S^{3}D: SKILL-CONDITIONED SELF-DISTILLATION FOR LLM SOCIAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

While large language models (LLMs) have achieved impressive performance across many domains, their social intelligence, including the ability to reason about others' beliefs, intentions, and emotions, remains comparatively underexplored. Recent work has applied reinforcement learning (RL) to improve LLMs' social reasoning capabilities, but sparse verifier outcomes provide little guidance for the intermediate reasoning process. On-policy self-distillation (OPSD) alleviates this by using teacher-side privileged information to provide dense token-level supervision. However, conventional sample-specific privileged signals may fail to capture diverse, reusable reasoning strategies required for social reasoning. To address these limitations, we introduce **S**kill-Conditioned **S**elf-**D**istillation for **S**ocial Reasoning (**S^{3}D**), a framework that uses structured social reasoning skills as teacher-only privileged information. We extract compact natural-language planning skills from the model's own correct reasoning trajectories, where each skill describes a successful workflow for solving a class of social reasoning problems. During training, relevant skills are dynamically retrieved and provided only to the teacher, while the student always responds to the plain task prompt and learns to internalize the planning guidance through token-level distillation. We further combine this dense distillation signal with outcome rewards to jointly supervise both the reasoning process and the final answer. Experimental results on multiple social reasoning benchmarks demonstrate that S^{3}D substantially outperforms the standard RL baseline, improving both vanilla GRPO and vanilla OPSD.

1 INTRODUCTION

Large language models (LLMs) have demonstrated impressive capabilities across a range of domains, including mathematics, coding, and logical inference (Guo et al., 2025; Luo et al., 2025; Xie et al., 2025). However, reasoning in real-world social situations requires more than solving well-structured problems. It requires social intelligence—the ability to perceive social cues, infer latent mental states, and generate contextually appropriate responses. This has led to increased studies on social reasoning, particularly Theory of Mind (ToM), the ability to infer and reason about human beliefs, intentions, desires, and knowledge (Wu et al., 2023; Zhou et al., 2023; Kosinski, 2024; Strachan et al., 2024; Xiao et al., 2025; Xu et al., 2024; Gu et al., 2026).

Building on the success of reinforcement learning (RL) in structured reasoning domains Shao et al. (2024), recent work has begun to investigate whether rule-based RL can improve LLMs' social reasoning abilities (Lu et al., 2025; Wu et al., 2026). However, the gains from rule-based RL on ToM tasks remain limited, especially under distribution shift. Rule-based RL methods typically optimize sparse, outcome-level rewards based solely on final-answer correctness, leaving the intermediate reasoning process largely unsupervised. Such rewards provide limited guidance on how a correct answer should be derived. Models may therefore exploit superficial cues or shortcut strategies that achieve high rewards without developing robust social reasoning capabilities (Zhong et al., 2026). Moreover, recent work on RL for small-scale LLMs on ToM tasks has observed that RL can lead to overfitting to statistical patterns in the training data, resulting in substantial gains on in-distribution tasks but little improvement or even degraded performance on out-of-distribution ToM tasks (Sarangi & Salam, 2025). These limitations highlight the need for training methods that provide dense supervision throughout the reasoning process while encouraging reusable social reasoning strategies that generalize beyond the training distribution.

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.

On-policy distillation (OPD) (Agarwal et al., 2024) addresses this supervision gap by using a strong teacher to provide dense token-level supervision on the student's on-policy responses. On-policy self-distillation (OPSD) (Zhao et al., 2026) further eliminates the need for a separate teacher model: the same model acts as a teacher when conditioned on privileged information and as a student when prompted with the original task input. The effectiveness of OPSD, however, depends critically on the privileged information available to the teacher. Existing approaches typically use sample-specific signals, such as ground-truth answers or reference solutions (Shenfeld et al., 2026; Hübotter et al., 2026; Yang et al., 2026). While ground-truth answers identify the correct outcome, they provide little guidance about the underlying reasoning process. Reference solutions provide denser supervision, but they often encode a single instance-specific reasoning path, which can constrain exploration and obscure alternative strategies that are also valid. This motivates us to explore privileged information at a more transferable level, capturing reasoning procedures shared across similar social-reasoning problems rather than instance-specific outcomes or rationales.

In this work, we explore extending OPSD to social reasoning task. The key challenge is to design privileged information that conveys generalizable reasoning strategies rather than instance-specific solutions. To this end, we introduce **S**kill-Conditioned **S**elf-**D**istillation for **S**ocial Reasoning (**S^{3}D**). S^{3}D first extracts *social reasoning skills* from the model's own successful reasoning trajectories and then uses these skills as teacher-only privileged information in OPSD. Unlike ground-truth answers or complete reference solutions, each skill summarizes a generalizable reasoning procedure for a class of social reasoning problems without specifying the answer or a particular reasoning trace. During training, S^{3}D dynamically retrieves skills relevant to each task instance and provides them as teacher privileged information. Conditioned on these skills, the teacher provides token-level supervision for the student's on-policy responses. Through this process, the student internalizes reusable social reasoning strategies while preserving the standard inference-time setting, requiring neither skill retrieval nor additional skill input at inference time. We further combine skill-conditioned distillation with outcome-based rewards. The outcome reward promotes final-answer correctness, while the distillation signal provides process-level guidance grounded in reusable social reasoning workflows. By jointly optimizing these complementary signals, S^{3}D aims to reduce shortcut learning and improve the transferability of learned reasoning strategies across tasks and contexts.

We evaluate S^{3}D on multiple social reasoning benchmarks covering Theory-of-Mind reasoning and emotion understanding. Experimental results show that S^{3}D consistently improves over standard RL baselines and vanilla OPSD variants, indicating that dynamically retrieved social reasoning skills provide more effective privileged supervision than answer-only or reference-solution signals. The gains are particularly pronounced in settings requiring multi-step perspective tracking and latent mental-state inference.

Our contributions are summarized as follows:

- We propose S^{3}D, a skill-conditioned self-distillation framework that uses social reasoning skills as teacher-only privileged information to improve LLMs' social reasoning.
- We introduce a pipeline for extracting compact natural-language social reasoning skills from successful model-generated trajectories and dynamically retrieving them as privileged teacher guidance for new training instances.
- We combine skill-conditioned token-level distillation with outcome rewards and show that S^{3}D outperforms standard RL and vanilla OPSD baselines across multiple social reasoning benchmarks.

2 Related Work

2.1 Social Reasoning in LLMs

Social reasoning in LLMs has been primarily studied through Theory-of-Mind (ToM) tasks, which require models to infer the beliefs, intentions, knowledge, and emotions of other agents. While LLMs can perform well on simple ToM tasks, recent evaluations reveal substantial limitations as reasoning becomes more complex, including higher-order belief inference, dynamic mental-state tracking, and psychologically complex scenarios (Kosinski, 2024; Strachan et al., 2024; Wu et al., 2023; Xiao et al., 2025; Zhou et al., 2023; Xu et al., 2024; Gu et al., 2026). These findings suggest

that strong performance on simple ToM benchmarks does not necessarily translate into robust and transferable social reasoning, motivating training methods that provide supervision beyond final-answer optimization.

2.2 Reinforcement Learning for LLM Reasoning

Reinforcement Learning (RL) has become an important approach for improving LLM reasoning, with recent rule-based RL methods achieving strong results on tasks with automatically verifiable rewards (Ouyang et al., 2022; Jaech et al., 2024; Guo et al., 2025). Its application to social reasoning has also shown promising improvements on ToM tasks (Lu et al., 2025). However, recent studies suggest that outcome-based RL may encourage dataset-specific shortcuts and provide limited generalization beyond the training distribution (Sarangi & Salam, 2025; Zhong et al., 2026). Process-level supervision therefore offers a promising direction for encouraging more robust and reusable social reasoning strategies.

2.3 On-Policy Distillation and Self-Distillation

On-policy Distillation (OPD) provides dense supervision by evaluating the student's on-policy responses, while On-Policy Self-Distillation (OPSD) further removes the need for a separate teacher by using privileged information only in the teacher branch (Agarwal et al., 2024; Zhao et al., 2026; Ye et al., 2026). Recent work has extended this paradigm by incorporating reinforcement learning signals or natural-language skills as teacher-side guidance (Hübotter et al., 2026; Xu et al., 2025; Zhang et al., 2026; Xia et al., 2026; Wang et al., 2026b; Huang et al., 2026). However, existing skill-conditioned approaches focus primarily on agentic interaction or formal reasoning. Whether reusable skills can serve as privileged supervision for the context-dependent and multi-interpretive nature of social reasoning remains unexplored. Our work addresses this gap by applying skill-conditioned OPSD to LLM social reasoning.

3 Skill-Conditioned Self-Distillation for Social Reasoning

We introduce **S**kill-Conditioned **S**elf-**D**istillation for **S**ocial Reasoning (**S^{3}D**), a framework that improves LLM social reasoning by using reusable social reasoning skills as teacher-only privileged information. S^{3}D consists of three main components: constructing a skill bank from successful reasoning trajectories, dynamically retrieving relevant skills for each training instance, and jointly optimizing skill-conditioned self-distillation with outcome rewards.

3.1 Structured Skill Bank Construction

Unlike mathematical problem solving or code generation, social reasoning often requires inferring different types of mental or social states. Questions that target similar states can share underlying reasoning structures despite having different surface narratives. Motivated by this regularity, S^{3}D constructs a structured skill bank $\mathcal{B}$ by distilling reusable reasoning workflows from verified successful trajectories and organizing them by query target, as illustrated in Figure 1. Each skill captures a generalizable reasoning procedure rather than an instance-specific rationale, and is retrieved during distillation to provide the teacher with privileged information.

Verified Student Rollout. We first perform rollouts with the initial student policy on a task dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$. Each rollout consists of a reasoning trace ρ_i and a final response r_i. We retain only the rollouts whose final answers are verified as correct, yielding

$$\mathcal{T}^+ = \{(x_i, \rho_i, r_i) \mid r_i \text{ is verified as correct}\}.$$

These verified student rollouts provide the data from which we construct the skill bank.

Social Reasoning Annotation. For each problem x_i in $\mathcal{T}^+$, we annotate its social-reasoning demand with a schema-constrained LLM annotator. The annotator is given a social scenario and question to annotate, together with a controlled list of candidate labels for targets, mechanisms,

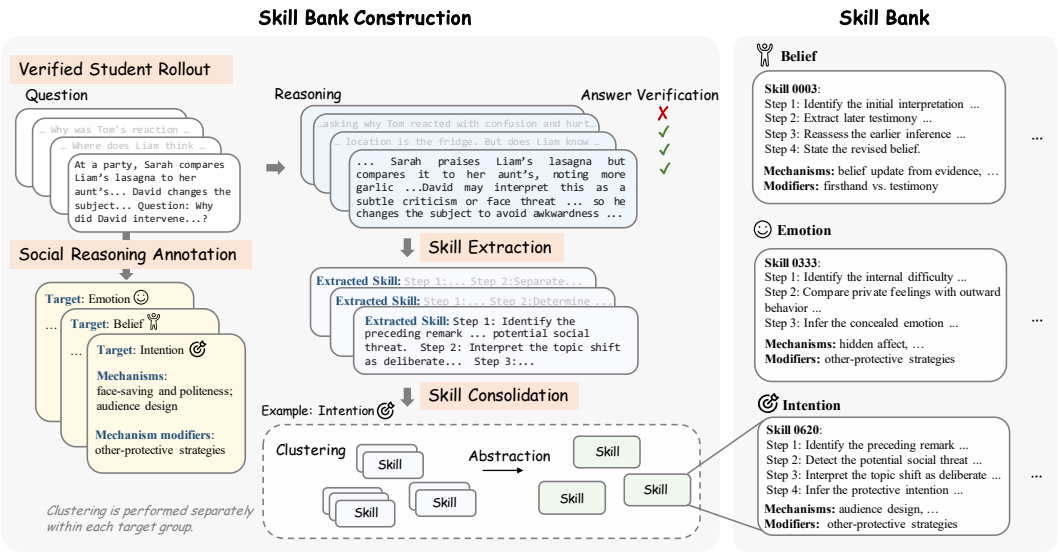


Figure 1: Skill bank construction pipeline. S^{3}D builds the skill bank from verified student rollouts by annotating each question with its query target, reasoning mechanisms, and mechanism modifiers. It then extracts an instance-level reasoning workflow from each verified trajectory. For each query-target group, semantically similar candidate workflows are clustered and abstracted into reusable skills; the intention group shown here illustrates this target-specific consolidation process. The final skill bank is organized by query target, with each skill storing reusable reasoning steps and associated mechanism metadata for downstream retrieval.

and mechanism modifiers. Each candidate label is accompanied by its definition and representative examples to guide the annotation. The annotation is represented as

$$a_i = (q_i, \mathcal{M}_i, \mathcal{U}_i), \qquad q_i \in \mathcal{Q}, \quad \mathcal{M}_i \subseteq \mathcal{M}, \quad \mathcal{U}_i \subseteq \mathcal{U}.$$

Here, q_i denotes the *query target*, i.e., the mental or social state to be inferred; $\mathcal{M}_i$ denotes the core *reasoning mechanisms* required to answer the question; and $\mathcal{U}_i$ contains optional *mechanism modifiers* that describe how these mechanisms are instantiated in the scenario. These annotations characterize the inference demand of each problem, providing a structured representation of the targets, reasoning mechanisms, and their contextual variations involved in social reasoning.

We use the query target as a routing key to partition the verified rollouts:

$$\mathcal{T}_q^+ = \{(x_i, \rho_i, r_i) \in \mathcal{T}^+ \mid q_i = q\}, \qquad q \in \mathcal{Q}.$$

This partition is imposed as a pre-clustering constraint: candidate skills are clustered independently within each query-target group. The mechanisms and modifiers are retained as auxiliary semantic metadata for downstream skill characterization, ranking, and retrieval, where they provide additional signals for assessing the semantic relevance of candidate skills. Details of label candidates and annotation prompts are provided in Appendix C.

Skill Extraction. For each verified trajectory $(x_i, \rho_i, r_i) \in \mathcal{T}^+$, we provide the extractor with the original problem x_i, the successful reasoning trace ρ_i, the final response r_i, and the structured annotation $a_i = (q_i, \mathcal{M}_i, \mathcal{U}_i)$. The annotation defines a compact reasoning scope: the query target identifies the question focus, the mechanisms specify the relevant reasoning operations, and the modifiers capture optional contextual factors. The extractor f then produces a candidate social reasoning skill:

$$p_i = f(x_i, \rho_i, r_i, a_i).$$

Specifically, f compresses the verified reasoning trace into an ordered natural-language workflow. It removes sample-specific names, objects, locations, answer options, and final answers while preserving the reusable reasoning structure. Each candidate p_i is stored as $(\mathrm{id}_i, z_i, a_i)$, where z_i is an ordered sequence of high-level steps for solving the social-reasoning problems, and a_i provides

metadata for subsequent clustering and retrieval. The resulting skills are therefore generalized abstractions of successful reasoning processes, rather than raw reasoning trajectories or fixed reference rationales.

Skill Consolidation. Skill extraction may produce functionally redundant candidates that differ mainly in wording or instance-specific details. We therefore consolidate semantically similar candidates into reusable reasoning templates. Given the candidate set $\mathcal{P} = \{p_i\}$, we discard candidates with invalid structured annotations and partition the remaining candidates by query target:

$$\mathcal{P}_q = \{p_i \in \mathcal{P} \mid q_i = q\}.$$

Within each target-specific partition, we embed the skill texts and apply complete-linkage clustering with cosine-similarity threshold τ, requiring every pair of skills in a cluster to satisfy the threshold. For each cluster, we select a small set of representative plans using a medoid-first diverse selection strategy and prompt an LLM to abstract their shared reasoning workflow into a compact, reusable plan. The abstraction is conditioned on the cluster's query target and mechanism coverage, while excluding instance-specific entities and answer details. If consolidation fails, the original candidates are retained rather than replaced with an unreliable abstraction. This process yields the final social reasoning skill bank $\mathcal{B}$ together with provenance metadata.

3.2 Skill-Conditioned Self-Distillation

During training, S^{3}D retrieves a relevant social-reasoning skill for each task instance and uses it exclusively as teacher-side privileged information. Following the annotation scheme introduced in Section 3.1, we annotate each training instance x_i with structured information $a_i = (q_i, \mathcal{M}_i, \mathcal{U}_i)$. For each training instance, we first restrict retrieval to the query-target-specific sub-bank $\mathcal{B}_{q_i}$. This hard routing ensures that skills associated with different mental-state targets are not considered as teacher guidance for the current instance.

The retrieval query is constructed from the task input, including the story and the question, rather than from the answer label. We embed both the query and all skill plans in $\mathcal{B}_{q_i}$, and first select a candidate set according to cosine similarity. Each candidate skill $s \in \mathcal{B}_{q_i}$ is then re-ranked by combining semantic similarity with annotation-derived signals:

$$\mathrm{score}(x_i, s) = \alpha\ \mathrm{sim}(x_i, s) + \beta\ \mathrm{align}(\mathcal{M}_i, s) + \gamma\ \mathrm{overlap}(\mathcal{U}_i, s) + \eta\ \mathrm{support}(s),$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes the embedding similarity between the task and the skill plan, $\mathrm{align}(\cdot,\cdot)$ quantifies the compatibility between the skill's supported mechanisms and the reasoning mechanisms required by the task, $\mathrm{overlap}(\cdot,\cdot)$ captures modifier overlap, and $\mathrm{support}$ favors skills consolidated from a larger number of source tasks. In our implementation, we set $(\alpha, \beta, \gamma, \eta) = (0.75, 0.17, 0.03, 0.05)$. We then select the highest-ranked skill, with a preference for general backbone skills when they have sufficient source-task support and comparable semantic similarity. We denote the selected skill as

$$s_i = R(x_i, a_i, \mathcal{B}).$$

Given the retrieved skill s_i, we instantiate OPSD by treating s_i as the privileged information available only to the teacher. The student samples an on-policy response from the original task input, $y_i = (y_{i,1}, \ldots, y_{i,T_i}) \sim \pi_\theta(\cdot \mid x_i)$, while the teacher evaluates the same response conditioned on both the task and the retrieved skill: $\pi_{\mathrm{T}}(y_{i,t} \mid x_i, s_i, y_{i,<t})$. The student distribution, by contrast, remains $\pi_\theta(y_{i,t} \mid x_i, y_{i,<t})$, so the skill affects training only through the teacher-side signal.

Following OPSD, we define the token-level skill-conditioned advantage as

$$\widetilde{A}_{i,t} = \log \pi_{\mathrm{T}}(y_{i,t} \mid x_i, s_i, y_{i,<t}) - \log \pi_\theta(y_{i,t} \mid x_i, y_{i,<t}),$$

where $\widetilde{A}_{i,t}$ is treated as a constant with respect to θ. The corresponding distillation loss is

$$\mathcal{L}_{\mathrm{S^3D}}(\theta) = -\mathbb{E}_{(x_i,a_i)\sim\mathcal{D}}\left[\mathbb{E}_{y_i\sim\pi_\theta(\cdot|x_i)}\frac{1}{T_i}\sum_{t=1}^{T_i}\widetilde{A}_{i,t}\log\pi_\theta(y_{i,t} \mid x_i, y_{i,<t})\right].$$

This objective increases the likelihood of tokens that receive higher teacher-side support under the retrieved skill, while leaving the student's input interface unchanged. Algorithm 1 in Appendix D summarizes the per-instance skill retrieval procedure used for skill-conditioned OPSD training.

At inference time, no skill retrieval is performed. The model receives only the original task prompt, and the retrieved skill serves solely as training-time privileged information. In this way, S^{3}D transfers reusable social-reasoning procedures into the student through distillation, without requiring explicit skill prompts during deployment.

3.3 Joint Optimization with Outcome Rewards

Skill-conditioned self-distillation provides dense token-level supervision, while the ultimate objective of social reasoning is to produce a correct final answer. We therefore jointly incorporate teacher-side skill guidance with outcome-based reinforcement learning. Given a prompt x, the student samples a group of responses $y^{(g)}$, and a task-specific verifier assigns each response an outcome reward $r^{(g)} = r(x, y^{(g)})$, according to whether the extracted final answer matches the ground truth or satisfies the task-specific correctness rule. The rewards are normalized within the group to obtain the GRPO advantage $\hat{A}^{(g)}$.

For each token $y_t^{(g)}$, we combine the S^{3}D token-level distillation signal with the outcome advantage by forming a mixed advantage:

$$\widetilde{A}_t^{(g)} = \hat{A}^{(g)} - \beta \left[\log \pi_\theta \left(y_t^{(g)} \mid x, y_{<t}^{(g)} \right) - \log \pi_{\mathrm{T}} \left(y_t^{(g)} \mid x, s(x), y_{<t}^{(g)} \right) \right],$$

where β controls the strength of skill-conditioned distillation. The first term preserves the outcome-level preference induced by GRPO, while the second term encourages the student to assign higher probability to tokens that are supported by the skill-conditioned teacher.

The final objective is the GRPO clipped policy objective with $\widetilde{A}_t^{(g)}$ replacing the original sequence-level advantage:

$$\mathcal{L}_{\mathrm{S^3D+GRPO}} = -\frac{1}{N} \sum_{g=1}^{G} \sum_{t=1}^{T_g} \min \left(\rho_t^{(g)} \widetilde{A}_t^{(g)},\ \mathrm{clip} \left(\rho_t^{(g)}, 1-\epsilon_l, 1+\epsilon_h \right) \widetilde{A}_t^{(g)} \right),$$

where

$$\rho_t^{(g)} = \frac{\pi_\theta \left(y_t^{(g)} \mid x, y_{<t}^{(g)} \right)}{\mathrm{sg} \left[\pi_{\theta_{\mathrm{old}}} \left(y_t^{(g)} \mid x, y_{<t}^{(g)} \right) \right]}$$

and $N = \sum_{g=1}^{G} T_g$. When $\beta = 0$, the objective reduces to vanilla GRPO. In contrast, positive β injects teacher-side skill guidance into the policy update without exposing the retrieved skill to the student at inference time. This joint objective preserves final-answer supervision while providing dense process-level guidance, reducing the tendency to exploit surface shortcuts in social reasoning problems.

4 Experiments

4.1 Experimental Setup

Training dataset. For training dataset, since most Theory-of-Mind benchmarks are designed primarily for evaluation, we construct an independent synthetic training dataset grounded in the Abilities in the Theory-of-Mind Space (ATOMS) framework (Osterhaus & Bosacki, 2022), covering six core dimensions of social intelligence: Belief, Desire, Emotion, Intention, Knowledge, and Non-literal Communication (NLC). we use GPT-5.5 within a structured generation protocol to construct social scenarios, Theory-of-Mind questions, answer options, and correct answers, replacing psychology expert construction with a standardized LLM-based generation procedure. After deduplication, leakage screening, quality verification, and difficulty calibration, we obtain 6,917 training examples; details are provided in Appendix E.

Evaluation datasets. We evaluate our model on five multiple-choice social-reasoning benchmarks, with ToMBench (Chen et al., 2024) serving as in-domain benchmark and four out-of-domain benchmarks. ToMBench follows the ATOMS framework and covers six broad Theory-of-Mind dimensions and 31 fine-grained abilities. ToMi (Le et al., 2019) focuses on unexpected-transfer and

Table 1: **Main comparison results on social reasoning benchmarks.** The best and second-best results are highlighted in **bold** and underlined, respectively. We report the pass@1 accuracy on ToMBench, ToMi, SimpleToM, Hi-ToM and EmoBench. S^3D and S^3D+GRPO improves over GRPO, and answer/reference-conditioned OPSD, especially on out-of-domain benchmarks.

Model	Method	In-domain	Out-of-domain				Avg.
		ToMBench	ToMi	SimpleToM	Hi-ToM	EmoBench	
Qwen3-4B	Base Model	70.42	78.81	60.10	65.83	58.67	66.77
	GRPO	69.20	61.95	52.72	57.58	51.50	58.59
	OPSD-answer	71.33	61.56	48.59	61.58	52.50	59.11
	OPSD-reference	**74.06**	76.39	54.11	66.25	53.50	64.86
	S^3D	71.75	**84.08**	**69.92**	**66.92**	**62.67**	**71.07**
	S^3D+GRPO	72.10	83.67	69.25	66.25	61.50	70.55
Qwen3-8B	Base Model	73.15	79.05	68.79	65.75	**64.33**	70.21
	GRPO	74.41	77.24	65.48	64.33	63.17	68.93
	OPSD-answer	74.16	70.35	52.98	64.33	59.67	64.30
	OPSD-reference	**76.12**	76.16	59.05	67.42	59.17	67.58
	S^3D	74.13	82.72	**79.51**	67.42	**64.33**	73.62
	S^3D+GRPO	73.78	**83.43**	79.10	**69.00**	**64.33**	**73.93**

false-belief reasoning. SimpleToM (Gu et al., 2026) tests mental-state inference, behavior prediction, and moral judgment. Hi-ToM (Wu et al., 2023) evaluates higher-order Theory-of-Mind reasoning with nested mental states. EmoBench (Sabour et al., 2024) evaluates emotion understanding in socially grounded scenarios.

Models and metrics. We experiment with the Qwen3 (Yang et al., 2025) model family at two scales: Qwen3-4B and Qwen3-8B, using the instruct-tuned versions. We report **accuracy** (pass@1) on five social-reasoning benchmarks.

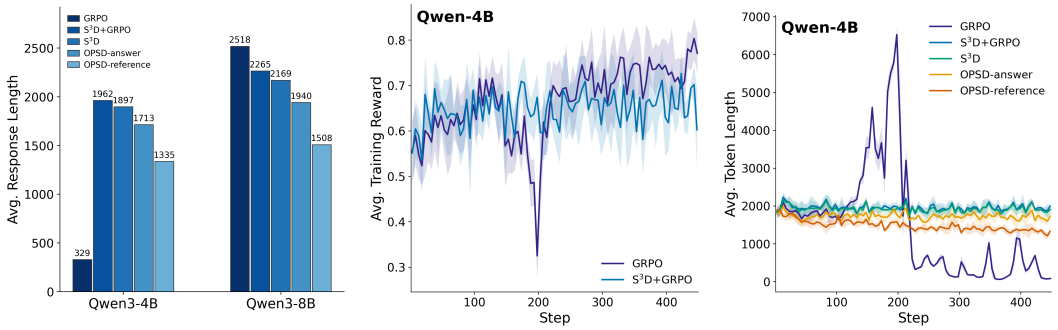
Baselines. We compare S^3D against base model and several representative training methods. (1) **Base Model**, the Qwen3-4B and Qwen3-8B model without further post-training. (2) **GRPO** (Shao et al., 2024), group relative policy optimization with binary outcome rewards verified against ground-truth answers. (3) **OPSD** (Zhao et al., 2026), a representative SD baseline that uses ground-truth answer or reference solution as teacher-side PI for dense token-level self-supervision. We denote these two variants as **OPSD-answer** and **OPSD-reference**, respectively.

S^3D is our skill-conditioned self-distillation method, which retrieves query-target-aligned social-reasoning skills as teacher-side PI. **S^3D+GRPO** further combines dense token-level process supervision with the outcome-based reward optimization.

Implementation details. For skill-bank construction, we extract candidate skills from 4,050 verified correct reasoning trajectories generated on the training dataset by Qwen3-8B. After redundancy-aware consolidation, the final skill bank contains 1,213 reusable skills. The training process was implemented with the slime framework (Zhu et al., 2025). More experimental details are provided in Appendix F.

4.2 Main Results

S^3D improves social reasoning and exhibits strong out-of-domain generalization. Table 1 summarizes the results on five social-reasoning benchmarks. Across both model scales, S^3D consistently improves performance over the base model and existing OPSD variants. For Qwen3-8B, S^3D increases the average accuracy from 70.21% to 73.62%, outperforming GRPO by 4.69 percentage points and OPSD-reference by 6.04 percentage points. The improvement is particularly pronounced on the out-of-domain benchmarks: S^3D achieves 82.72% on ToMi, 79.51% on SimpleToM, and 67.42% on HiToM, substantially exceeding the base model on ToMi and SimpleToM. Similar trends are observed for Qwen3-4B, where S^3D attains the best performance on four of the five benchmarks. These results indicate that dynamically organized skill supervision transfers effec-



(a) Final Response Length by Scale (b) Training Reward Dynamic (c) Response Length Dynamic

Figure 2: **Model response length and training dynamics.** (a) Qwen3-4B trained with GRPO produces substantially shorter final responses than the other methods and the Qwen3-8B counterpart. (b) GRPO exhibits a pronounced instability in training reward, whereas S^{3}D+GRPO maintains a more stable trajectory on Qwen3-4B. (c) GRPO causes a sharp and sustained decrease in Qwen3-4B's response length, while OPSD-based methods preserve longer responses throughout training.

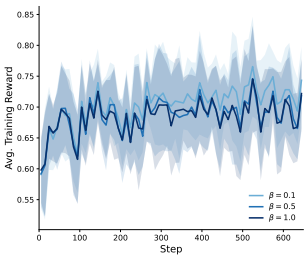
tively across datasets, and is more effective than using fixed target answers or fixed reference CoTs for improving social reasoning.

Dense token-level supervision provides complementary benefits to outcome optimization. Combining S^{3}D with GRPO further improves the Qwen3-8B average accuracy from 73.62% to 73.93%. In particular, S^{3}D+GRPO obtains the best results on ToMi (83.43%) and HiToM (69.00%), while remaining competitive on the other benchmarks. This result suggests that the dense, token-level learning signal introduced by S^{3}D is the primary contributor to improved social reasoning, whereas outcome-level optimization can provide additional but comparatively modest gains after skill-based supervision. The complementary behavior of the two objectives supports the value of combining process-level supervision with outcome-based rewards.

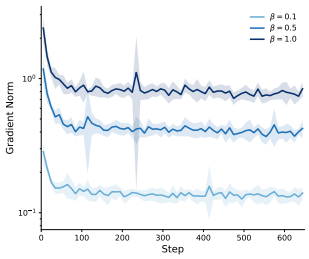
S^{3}D mitigates GRPO-induced degradation, particularly for smaller models. As illustrated in Figure 2, applying GRPO directly to Qwen3-4B leads to substantially shorter generated responses during training (Lu et al., 2025). This length collapse coincides with marked degradation in social-reasoning performance: compared with the base model, GRPO reduces accuracy from 78.81% to 61.95% on ToMi, from 65.83% to 57.58% on HiToM, and from 58.67% to 51.50% on EmoBench. A similar, although less severe, trend is observed for Qwen3-8B, where GRPO decreases the average score from 70.21% to 68.93%. In contrast, S^{3}D avoids this degradation and consistently improves over both the base model and GRPO across out-of-domain benchmarks. For Qwen3-4B, S^{3}D improves upon GRPO by 22.13, 17.20, and 11.17 percentage points on ToMi, HiToM, and EmoBench, respectively. These findings suggest that dense token-level supervision discourages overly short, reward-seeking generations and provides a more stable optimization signal for preserving and improving social reasoning capabilities, especially in smaller models.

4.3 Further Discussions

Sensitivity to the GRPO–OPSD balance. Figures 3 analyze the effect of the OPSD coefficient β in S^{3}D+GRPO. Although $\beta = 0.1$ achieves the highest final training reward, reaching approximately 0.72, it does not perform as well as $\beta = 1.0$ on the in-domain ToMBench benchmark, where the stronger OPSD signal yields 75.21% accuracy compared with 74.44%. The intermediate setting $\beta = 0.5$ achieves the best average performance across the five benchmarks, reaching 73.93%, while $\beta = 0.1$ and $\beta = 1.0$ obtain 72.74% and 73.79%, respectively. Its advantage is mainly reflected in the out-of-domain benchmarks, where $\beta = 0.5$ achieves the best results on HiToM and ToMi. Nevertheless, the differences among the three settings remain relatively small, suggesting that the method is not highly sensitive to the precise value of β. The training dynamics further show that increasing β substantially enlarges the gradient norm, indicating stronger OPSD-driven parameter updates, whereas the training reward does not increase accordingly. These results suggest that the intermediate value $\beta = 0.5$ provides a favorable balance between outcome optimization and dense



(a) Training reward.



(b) Gradient norm.

β	ToMBench	HiToM	SimpleToM	ToMi	Avg.
0 (GRPO)	74.41	64.33	65.48	77.24	70.37
0.1	74.44	68.67	75.85	81.75	72.74
0.5	73.78	69.00	79.10	83.43	73.93
1.0	75.21	68.75	79.16	82.35	73.79

(c) Evaluation results.

Figure 3: Sensitivity to the GRPO–OPSD balance. (a) The corresponding training-reward dynamics. (b) The gradient-norm dynamics, showing the increasing optimization intensity as β grows. (c) Final benchmark performance under different values of β.

skill-conditioned supervision, achieving strong generalization without requiring excessively aggressive updates.

5 Conclusion

We present S^{3}D, a skill-conditioned self-distillation framework for improving LLM social reasoning. Instead of relying only on sparse outcome rewards or sample-specific privileged signals, S^{3}D constructs a structured bank of reusable social-reasoning skills from verified successful trajectories and retrieves relevant skills as teacher-only privileged information during training. Through OPSD, the skill-conditioned teacher provides token-level supervision that encourages the student to internalize reusable reasoning patterns. Experiments on multiple social-reasoning benchmarks show that S^{3}D improves over the base model, standard GRPO, and conventional OPSD baselines, demonstrating the value of reusable intermediate reasoning strategies for social intelligence. Our results suggest that skill-conditioned privileged supervision is a promising direction for aligning LLMs with complex social reasoning demands.

6 Limitations

Our work has several limitations. First, skill extraction and consolidation rely on LLM-based annotation and merging, which may introduce noise or overlook useful minority reasoning patterns. Second, our current skill retrieval strategy in training provides only a small number of skills based primarily on query-target routing and semantic similarity, which may be insufficient for examples requiring multiple interacting social mechanisms.

7 AI use statement

We used LLMs for two purposes. First, LLMs were used as writing assistants to identify grammatical errors, improve phrasing and readability, and polish the academic presentation of the manuscript. Second, LLMs were used to construct synthetic training resources for social reasoning. Specifically, they were used to annotate each instance with its query target and mechanisms; extract generalized skills from student-rollout reasoning traces; and consolidate semantically similar skills after clustering.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, et al. Tombench: Benchmarking theory of mind in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15959–15983, 2024.

Yuling Gu, Oyvind Tafjord, Hyunwoo Kim, Jared Moore, Ronan Le Bras, Peter Clark, and Yejin Choi. Simpletom: Exposing the gap between explicit tom inference and implicit tom application in llms. In *International Conference on Learning Representations*, volume 2026, pp. 50252–50297, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jiazhen Huang, Xiao Chen, Xiao Luo, Yong Dai, Senkang Hu, and Yuzhi Zhao. Skill-conditioned gated self-distillation for llm reasoning. *arXiv preprint arXiv:2605.28791*, 2026.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Michal Kosinski. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45):e2405460121, 2024.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of EMNLP-IJCNLP*, pp. 5872–5877. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1598.

Yi-Long Lu, Chunhui Zhang, Jiajun Song, Lifeng Fan, and Wei Wang. Tom-rl: Reinforcement learning unlocks theory of mind in small llms. *arXiv e-prints*, pp. arXiv–2504, 2025.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 3(5), 2025.

Christopher Osterhaus and Sandra L Bosacki. Looking for the lighthouse: A systematic review of advanced theory-of-mind tests beyond preschool. *Developmental Review*, 64:101021, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Sahand Sabour, Siyang Liu, Zheyuan Zhang, June Liu, Jinfeng Zhou, Alvionna Sunaryo, Tatia Lee, Rada Mihalcea, and Minlie Huang. EmoBench: Evaluating the emotional intelligence of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5986–6004. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.326.

Sneheel Sarangi and Hanan Salam. Small llms do not learn a generalizable theory of mind via reinforcement learning. *arXiv preprint arXiv:2507.15788*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. 2024.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

James WA Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, et al. Testing theory of mind in large language models and humans. *Nature human behaviour*, 8(7):1285–1295, 2024.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, and Shumin Deng. Skillx: Automatically constructing skill knowledge bases for agents. 2026a.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, et al. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*, 2026b.

Jincenzi Wu, Yuxuan Lei, Jianxun Lian, Yitian Huang, Lexin Zhou, Haotian Li, Xing Xie, and Helen Meng. Social-r1: Towards human-like social reasoning in llms. *arXiv preprint arXiv:2603.09249*, 2026.

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-tom: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10691–10706, 2023.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Yang Xiao, Jiashuo Wang, Qiancheng Xu, Changhe Song, Chunpu Xu, Yi Cheng, Wenjie Li, and Pengfei Liu. Towards dynamic theory of mind: Evaluating llm adaptation to temporal evolution of human states. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24036–24057, 2025.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. Opentom: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8593–8623, 2024.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. Kdrl: Post-training reasoning llms via unified knowledge distillation and reinforcement learning. *arXiv preprint arXiv:2506.02208*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for llm reasoning. *arXiv preprint arXiv:2602.22495*, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Jike Zhong, Yuxiang Lai, Ming Li, Yuheng Li, Wuao Liu, Behzad Dariush, Konstantinos Psounis, and Shao-Yuan Lo. From shortcuts to reasoning: Robust post-training of theory of mind with reinforcement learning. *arXiv preprint arXiv:2606.09092*, 2026.

Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R McKee, Ari Holtzman, Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, et al. How far are large language models from agents with theory-of-mind? *arXiv preprint arXiv:2310.03051*, 2023.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. `https://github.com/THUDM/slime`, 2025. GitHub repository. Corresponding author: Xin Lv.

A Related Work

A.1 Social Reasoning in LLMs

Social reasoning in LLMs has been studied primarily through Theory-of-Mind tasks, which require models to infer the beliefs, intentions, knowledge, and emotions of other agents. Recent evaluations suggest that LLMs can perform well on simple ToM tasks involving shallow mental-state inference. Kosinski (2024) found that GPT-4 achieved child-level performance on false-belief tasks, solving 75% of the cases, while Strachan et al. (2024) showed that GPT-4 and LLaMA2 exhibited performance patterns broadly consistent with human mental-state inference across a range of ToM tasks. However, broader evaluations reveal substantial limitations as the reasoning demands of ToM tasks increase. Hi-ToM reports sharp performance degradation as the order of nested beliefs increases, particularly beyond first- and second-order reasoning (Wu et al., 2023). Dynamic evaluation further shows that LLMs struggle to track mental states as they evolve over time (Xiao et al., 2025). Comprehensive benchmarks also reveal inconsistent performance across task settings and particular weaknesses in psychologically complex scenarios (Zhou et al., 2023; Xu et al., 2024; Gu et al., 2026). Taken together, these findings suggest that strong performance on simple ToM benchmarks does not necessarily translate into robust and transferable social reasoning. This motivates training methods that go beyond final-answer optimization by providing dense supervision for intermediate reasoning and encouraging strategies that generalize across tasks and contexts.

A.2 Reinforcement Learning for LLM Reasoning

Reinforcement learning has become an important approach for improving the reasoning capabilities of LLMs. Early alignment methods such as RLHF use preference-based feedback to optimize model behavior (Ouyang et al., 2022), while more recent rule-based RL approaches use automatically verifiable rewards to strengthen structured reasoning in mathematical and coding domains (Jaech et al., 2024; Guo et al., 2025).

Recent work has begun to apply rule-based RL to social reasoning and Theory-of-Mind tasks. ToM-RL demonstrated that outcome-based RL can substantially improve ToM reasoning in small LLMs, including on higher-order and out-of-distribution tasks (Lu et al., 2025). However, Sarangi & Salam (2025) found that these gains often fail to generalize beyond the training distribution, while extended RL training may amplify reliance on dataset-specific shortcuts. Similarly, Zhong et al. (2026) showed that ToM benchmarks contain exploitable shortcuts and that explicit reasoning supervision is important for improving robustness and transferability. These findings suggest that outcome-based RL without sufficient process supervision may reward shortcut strategies rather than reusable social reasoning abilities. Social-R1 addresses this issue by augmenting rule-based RL with trajectory-level rewards, but its reliance on multiple manually designed reward components results in a relatively complex reward framework (Wu et al., 2026). This motivates simpler and more reusable forms of process supervision, such as using natural-language social reasoning skills as teacher-only privileged information in OPSD.

A.3 On-Policy Distillation and Self-Distillation

On-policy distillation provides dense token-level supervision by using a teacher to evaluate the student's on-policy responses (Agarwal et al., 2024). On-policy self-distillation further eliminates the need for a separate teacher by conditioning the same model on privileged information in the teacher branch, while the student receives only the original task prompt (Zhao et al., 2026; Ye et al., 2026). Recent work has integrated distillation with reinforcement learning by converting environment feedback into a dense self-distillation target (Hübotter et al., 2026), unifying Knowledge Distillation (KD) and RL in a single objective (Xu et al., 2025), or mitigating their interference through selective imitation (Zhang et al., 2026).

More recent studies have explored using skills as teacher-side privileged information. Skills are compact natural-language abstractions of successful trajectories that capture reusable procedures, principles, or tactics rather than instance-specific solutions. By distilling procedural guidance from complete trajectories, skills can be retrieved and applied across related tasks (Xia et al., 2026; Wang et al., 2026a). Skill-SD extracts reusable skills from agent trajectories and uses them to guide self-distillation for multi-turn LLM agents (Wang et al., 2026b), while Skill-Conditioned Gated Self-

Distillation applies skill-conditioned supervision to mathematical reasoning and filters unreliable guidance through a gated objective (Huang et al., 2026). However, these methods focus on agentic interaction or formal mathematical reasoning. Whether reusable skills can provide effective privileged supervision for the context-dependent and multi-interpretive nature of social reasoning remains unexplored. Our work addresses this gap by applying skill-conditioned OPSD to LLM social reasoning.

B GRPO AND OPSD BACKBONE

We consider a social-reasoning task instance x from dataset $\mathcal{D}$, consisting of a narrative context and a question. Given x, the student policy π_θ generates a response $y_i = (y_i^1, \ldots, y_i^{|y_i|})$, where $|y_i|$ is the response length. An outcome verifier evaluates the response and assigns a scalar reward $r_i = r(x, y_i)$.

For multiple-choice social-reasoning tasks, this reward can be defined by whether the extracted answer matches the ground-truth answer. More generally, $r(x, y)$ may be any task-level verifier that measures the correctness of the final response.

For each input x, GRPO samples a group of G responses $\{y_1, y_2, \ldots, y_G\}$ from the rollout policy $\pi_{\theta_{\text{old}}}$. The rewards within the group are normalized to obtain a relative advantage:

$$A_i = \frac{r_i - \mu_x}{\sigma_x + \epsilon}, \qquad \mu_x = \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad \sigma_x = \text{std}(r_1, \ldots, r_G). \tag{1}$$

The same sequence-level advantage is assigned to all response tokens in the corresponding trajectory. GRPO then optimizes the clipped policy objective

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|y_i|}\sum_{n=1}^{|y_i|} \min\left(\rho_i^n A_i,\ \text{clip}\left(\rho_i^n, 1-\epsilon, 1+\epsilon\right) A_i\right) \right], \tag{2}$$

where $\rho_i^n = \frac{\pi_\theta(y_i^n | x, y_i^{<n})}{\pi_{\theta_{\text{old}}}(y_i^n | x, y_i^{<n})}$ is the token-level importance ratio and ϵ control the clipping range. Thus, GRPO provides outcome-level supervision by encouraging responses with higher relative verifier rewards, but the resulting signal remains tied to the final outcome and is broadcast across the response tokens.

On-policy self-distillation (OPSD) complements this sparse outcome signal with token-level supervision from a privileged teacher. Let z denote privileged information available only to the teacher during training. The student π_θ generates the response $\hat{y}$ using the ordinary task input x, whereas the teacher π_{T} evaluates the same response conditioned on (x, z). For each position n in a sampled sequence $\hat{y}$, OPSD define the student–teacher log-probability discrepancy as the token-level advantage term

$$\widetilde{A}_n(x, \hat{y}) = \log \pi_{\text{T}}(\hat{y}_n \mid x, z, \hat{y}_{<n}) - \log \pi_\theta(\hat{y}_n \mid x, \hat{y}_{<n}), \tag{3}$$

and optimize the policy-gradient-style objective

$$\mathcal{L}_{\text{OPSD}}(\theta) = -\mathbb{E}_{(x,z)\sim\mathcal{D}} \left[\mathbb{E}_{\hat{y}\sim\pi_\theta(\cdot|x)} \frac{1}{|\hat{y}|}\sum_{n=1}^{|\hat{y}|} \widetilde{A}_n(x, \hat{y}) \times \log \pi_\theta(\hat{y}_n \mid x, \hat{y}_{<n}) \right]. \tag{4}$$

$\widetilde{A}_n(x, \hat{y})$ is treated as a constant with respect to θ (i.e., gradients do not flow through the advantage). Importantly, the privileged information z is used only to condition the teacher and is not exposed to the student during response generation. In our framework, z is instantiated by a retrieved social-reasoning skill, whose construction and teacher-side integration are described in the following sections.

C PROMPTS FOR SKILL BANK CONSTRUCTION

We provide the prompts used at each stage of skill bank construction, including social reasoning annotation, skill extraction from successful rollouts, and consolidation of extracted skills into reusable skill entries.

Table 2: Prompt for Social Reasoning Annotation.

Social Reasoning Annotation Prompt

You are annotating ToMBench Theory-of-Mind data with the V2 multi-axis schema. Use only the story and question. Do not infer unstated facts, and do not use external knowledge.
The query target is WHAT the question asks us to identify or predict. A reasoning mechanism is WHY that answer follows from the story. Do not treat surface question wording as a mechanism.
Return JSON only, using exactly this schema:

```
{
  "query_target": "one allowed query target",
  "reasoning_mechanisms": [
    "one to three allowed mechanisms"
  ],
  "mechanism_modifiers": [
    "zero to two allowed modifiers"
  ],
  "legacy_primary":
    "one legacy label in the form Top Level: Sub Level",
  "secondary_abilities": [
    "zero to two other legacy labels"
  ],
  "evidence": [
    "one to three concise, source-grounded facts"
  ],
  "annotation_confidence": "high, medium, or low",
  "brief_reason": "one concise sentence"
}
```

Hard constraints:
- Select exactly one query target, one to three distinct core mechanisms, and zero to two distinct modifiers.
- Select exactly one legacy_primary. secondary_abilities may contain zero to two distinct legacy labels and must not repeat legacy_primary.
- Evidence must be one to three short facts from the story or question, not a chain-of-thought and not generic label definitions.
- Use only the allowed strings shown in the user message. Do not invent labels.

Disambiguation:
- competing_goals means incompatible goals of ONE agent. divergent_agent_preferences means different agents have conflicting preferences.
- strategic_deception requires intent to make another agent believe something false. A false statement made by mistake uses unintentional_false_statement.
- belief_driven_action and belief_driven_emotion describe an action or emotion explained by an agent's belief. Use false_content, false_location, false_identity, or false_sequence as an additional mechanism when that belief is specifically false in that way.
- second_order_belief is required when reasoning is about one agent's belief, knowledge, or perception of another agent's mental state.
- perception_to_knowledge, attention_to_knowledge, and information_access_to_knowledge distinguish seeing, attending, and receiving information/testimony.
- nonliteral_meaning is for irony, sarcasm, humor, or indirect intended meaning. face_saving_or_politeness is for socially protective wording; they may co-occur.

Table 3: Prompt for Extracting Skills from Student Rollouts.

Skill Extraction Prompt

You are a Theory-of-Mind planning expert extracting one reusable plan from a successful final response.
The input contains one user task, the model's successful final response, and a compact task reasoning scope:
- `question_focus` identifies the question type the plan must solve.

- `reasoning_scope` constrains the plausible reasoning path.
- `conditional_factors` describe optional conditions.

Use the task and final response as evidence for the central successful reasoning path. Do not add a step merely to mention a listed item in the reasoning scope. Write a plan that transfers to similar tasks with the same question focus, without names, objects, locations, numerical details, candidate options, or the final answer from this example.

Output exactly:

```
<plan>
# step 1:  ...
# step 2:  ...
</plan>
```

Requirements:

1. Use 3-6 concise, executable reasoning steps.
2. Center the plan on the supplied question focus.
3. Use the scope as a boundary, not as a checklist of mandatory steps.
4. Make the final step state the requested belief, intention, emotion, knowledge state, behavior, or interpersonal effect in the task's requested form.
5. Never phrase the final step as selecting a candidate, option, answer, or answer choice.
6. Do not explain the metadata, produce a final answer, or add text outside the plan tags.

Table 4: Prompt for Consolidating Skills within Semantic Clusters.

Skill Consolidation Prompt

You are merging reusable Theory-of-Mind planning templates from one semantic cluster.
All input plans share the supplied `question_focus`. `reasoning_coverage` reports how often each reasoning operation appears among source tasks; it is descriptive evidence, not a checklist and not a requirement that every operation appear in the merged plan.

Output exactly:

```
<plan>
# step 1:  ...
# step 2:  ...
</plan>
```

Requirements:

1. Keep the same question focus.
2. Preserve only operations that are genuinely shared across the representative plans.
3. Express supported variants as short conditionals only when needed.
4. Make the final step state the requested conclusion directly; never phrase it as selecting a candidate, option, answer, or answer choice.
5. Do not mention sample-specific entities, labels, candidate options, or final answers.
6. Use 3-6 concise steps and no text outside the plan tags.

Candidate label lists for social-reasoning demand annotation. The following label sets define the candidate query targets, reasoning mechanisms, and mechanism modifiers used by the social reasoning annotation.

Candidate label lists for social-reasoning demand annotation

Query targets

- `belief_or_expectation`
- `knowledge_or_information_state`
- `desire_or_preference`
- `emotion_or_affect`
- `intention_or_motive`
- `next_action_or_behavior`
- `utterance_meaning_or_truthfulness`
- `social_appropriateness_or_interpersonal_effect`

Core mechanisms

- `false_content`

- false_location
- false_identity
- false_sequence
- second_order_belief
- belief_persistence
- belief_update_from_evidence
- belief_driven_action
- belief_driven_emotion
- perception_to_knowledge
- attention_to_knowledge
- information_access_to_knowledge
- pretense_or_representation
- goal_or_preference
- competing_goals
- divergent_agent_preferences
- desire_action_conflict
- intention_action_link
- intention_outcome_mismatch
- interrupted_or_failed_plan
- outcome_appraisal
- expectation_violation
- agent_specific_appraisal
- mixed_affect
- hidden_affect
- emotion_regulation
- moral_self_evaluation
- strategic_deception
- unintentional_false_statement
- nonliteral_meaning
- social_norm_violation
- face_saving_or_politeness
- audience_design

Mechanism modifiers

- self_protective
- other_protective
- accidental_or_unintentional
- private_vs_public
- direct_vs_indirect
- literal_vs_nonliteral
- firsthand_vs_testimony
- immediate_vs_delayed

D SKILL-CONDITIONED OPSD TRAINING PROCEDURE

Algorithm 1 illustrates the two-stage training procedure used in S^3D: per-instance skill retrieval to construct a skill-augmented training set, followed by OPSD training where the retrieved skill is used only as teacher-side privileged information.

E SYNTHETIC TRAINING CORPUS CONSTRUCTION

E.1 ATOMS-GUIDED SCENARIO GENERATION

We construct the training corpus according to the six core dimensions and 31 fine-grained abilities defined in the ATOMS framework (Osterhaus & Bosacki, 2022). GPT-5.5 is used within a structured generation procedure to produce realistic, self-contained social scenarios. The story-generation prompt requires concrete character interactions, latent mental-state structures, and sufficient contextual evidence for subsequent Theory-of-Mind reasoning, while prohibiting explicit ability names, psychological interpretations, questions, and answers. The complete prompt is provided in Table 5. Each generated story is then passed to a second GPT-5.5 prompt, which produces three multiple-choice Theory-of-Mind items. Each item contains a question, four answer options, and one unique correct answer. The prompt provides the complete 31-ability inventory and requires the generated questions to test distinct, story-supported aspects of social reasoning rather than surface-level re-

Algorithm 1 Skill-Conditioned OPSD Training

Require: Training data $\mathcal{D}$, skill bank $\mathcal{B}$, student policy π_θ, teacher policy π_{T}
Ensure: Skill-augmented training set $\mathcal{D}_{\mathrm{skill}}$ and updated student policy π_θ

Stage 1: Instance-level skill retrieval

Initialize $\mathcal{D}_{\mathrm{skill}} \leftarrow \emptyset$
for each training instance $x_i \in \mathcal{D}$ **do**
  Annotate x_i as $a_i = (q_i, \mathcal{M}_i, \mathcal{U}_i)$
  Route retrieval to the target-specific sub-bank $\mathcal{B}_{q_i}$
  Construct the retrieval query from the story and question in x_i
  $\mathcal{C}_i \leftarrow \mathrm{TopK}_{s \in \mathcal{B}_{q_i}} \mathrm{sim}(x_i, s)$
  Re-rank each $s \in \mathcal{C}_i$ using $\mathrm{score}(x_i, s)$
  $s_i \leftarrow \arg\max_{s \in \mathcal{C}_i} \mathrm{score}(x_i, s)$
  Add (x_i, s_i) to $\mathcal{D}_{\mathrm{skill}}$
end for

Stage 2: Skill-conditioned OPSD

for each training iteration **do**
  for each $(x_i, s_i) \in \mathcal{D}_{\mathrm{skill}}$ **do**
    Sample an on-policy response $y_i = (y_{i,1}, \ldots, y_{i,T_i}) \sim \pi_\theta(\cdot \mid x_i)$ ▷ student sees only x_i
    Compute token-level student log-probabilities $\boldsymbol{\ell}_i^{\mathrm{stu}} \leftarrow \log \pi_\theta(y_i \mid x_i)$ ▷ plain prompt
    Compute token-level teacher log-probabilities $\boldsymbol{\ell}_i^{\mathrm{tea}} \leftarrow \log \pi_{\mathrm{T}}(y_i \mid x_i, s_i)$ ▷ skill-augmented prompt
    $\widetilde{\mathbf{A}}_i \leftarrow \mathrm{sg}(\boldsymbol{\ell}_i^{\mathrm{tea}} - \boldsymbol{\ell}_i^{\mathrm{stu}})$
  end for
  Update θ by minimizing $\mathcal{L}_{\mathrm{S^3D}}(\theta)$
end for

call. Distractors are designed to reflect plausible perspective-taking errors, such as confusing an agent's belief with reality or interpreting an utterance literally instead of pragmatically. The complete question-generation prompt is provided in Table 6. ToMBench test instances are not used as generation inputs.

Table 5: Prompt for training dataset story generation.

Story Generation Prompt

You are helping construct Theory-of-Mind benchmark data in the style of ToMBench.
Your job is to generate ONLY one story that can support one or more high-quality Theory-of-Mind multiple-choice questions later.

Optional control:
"focus_type": {`focus_type`},
If "focus_type" is provided, the story should naturally center on that type of social reasoning.
If "focus_type" is not provided, generate any story that satisfies the requirements below.

Requirements:
1. Write one natural, realistic, everyday social story in English.
2. The story should involve 2–4 characters.
3. The story should be concrete, with specific actions, dialogue, and context.
4. The story should be self-contained and understandable without outside knowledge.
5. The story should include at least one socially meaningful mental-state structure that can later support Theory-of-Mind reasoning.
6. If "focus_type" is provided, make it the primary latent structure of the story, but do NOT explicitly name it.
7. Do NOT explicitly mention psychological terms, task names, ability names, or the correct interpretation.
8. Do NOT explain anyone's mental state directly unless such wording would naturally appear in a real story.
9. Do NOT include a question, options, answer, analysis, labels, or metadata.
10. Keep the story concise and focused: around 40–90 words.

11. Make the story rich enough that one or more Theory-of-Mind questions could later be asked about it.
12. Prefer concise, event-driven narration rather than elaborate scene-setting.
13. Avoid unnecessary descriptive details, emotional embellishment, or literary phrasing.
14. Do not over-explain motivations or reactions.
15. Avoid stories that are purely factual, descriptive, or action-only with no social reasoning needed.
16. Avoid overly dramatic, fantastical, or highly specialized scenarios.

Style guidance:

1. Use natural names and everyday settings such as school, family, workplace, friendship, neighborhood, shopping, dining, travel, volunteering, or hobbies.
2. Prefer realistic social interactions over dramatic plots.
3. Include enough hidden social cues, perspective differences, or contextual gaps to support later question generation.
4. Avoid repetitive rigid templates unless they arise naturally.
5. Make the story feel like a believable slice of daily life.

Return valid JSON only, following the provided schema.

Table 6: Prompt for training dataset question-options-answer generation.

Story Generation Prompt

You are helping construct Theory-of-Mind benchmark data in the style of ToMBench.
Your job is to generate THREE multiple-choice QA items based on the given story.

[STORY]
{story}

Reference ability pool (31 abilities):

1. Typical emotional reactions
2. Atypical emotional reactions
3. Discrepant emotions
4. Mixed emotions
5. Hidden emotions
6. Moral emotions
7. Emotion regulation
8. Multiple desires
9. Desires influence on action/emotion
10. Desire-action contradiction
11. Discrepant desires
12. Discrepant intentions
13. Prediction of actions
14. Intentions explanations
15. Completion of failed actions
16. Knowledge-pretend play links
17. Percepts-knowledge links
18. Information-knowledge links
19. Knowledge-attention links
20. Content false beliefs
21. Location false beliefs
22. Identity false beliefs
23. Second-order beliefs
24. Belief-based action/emotion
25. Sequence false beliefs
26. Irony/Sarcasm
27. Egocentric lies
28. White lies
29. Involuntary lies
30. Humor
31. Faux pas

Requirements:

1. Generate exactly 3 QA items.
2. Each QA item must contain:
   - one question sentence
   - four answer options: OPTION-A, OPTION-B, OPTION-C, OPTION-D
   - one correct answer: A, B, C, or D
3. Choose 3 different abilities from the list above.
4. Do NOT choose abilities arbitrarily if the story does not support them. Select abilities that are genuinely supported by the story.
5. Each question must test social reasoning, not surface facts.
6. Questions should resemble ToMBench style:
   - concise
   - natural
   - directly answerable from the story
   - focused on a character's emotion, belief, knowledge, desire, intention, implied meaning, or social appropriateness

7. The three questions should test different aspects of the story and should not be paraphrases of each other.
8. Each question must be a single question sentence.
9. Each QA item must have exactly one correct answer.
10. Wrong options must be plausible, subtle, and close enough to the story to be misleading.
11. Avoid absurd, irrelevant, overly generic, or obviously wrong distractors.
12. Prefer distractors that reflect:
    - confusion between reality and a character's belief
    - confusion between different characters' perspectives
    - literal vs implied meaning confusion
    - a nearby but incorrect emotion, intention, or motivation
13. Do not include explanations, reasoning steps, or any text outside the JSON output.
14. Keep wording concise and annotation-like rather than literary.
15. If the story supports fewer than 3 high-quality abilities, still produce 3 items, but prioritize the most defensible and distinct ones.
16. The three QA items should be coherent with the same story context, but each should probe a different inferential angle.

Output must strictly follow the provided JSON schema.

E.2 Post-Generation Curation

We apply a multi-stage curation pipeline to the generated candidates. We first discard malformed records and remove duplicate or near-duplicate story–question pairs within the synthetic corpus. We then compare the generated examples with the ToMBench test set and remove instances that directly overlap with or closely reproduce evaluation examples.

For quality verification, GPT-5.5 receives the complete story, question, options, and proposed answer. It checks story coherence, answer correctness, question–story consistency, distractor plausibility, and whether exactly one option is defensible. Items that are ambiguous, logically inconsistent, unsupported by the story, or associated with multiple valid answers are discarded.

Finally, we calibrate the difficulty of the remaining items using repeated answering trials. An item is removed if GPT-5.5 answers it incorrectly in all three trials, since it is likely flawed or excessively difficult. It is also removed if Qwen3-8B answers it correctly in all three trials, since it is insufficiently challenging for post-training. The remaining examples form the final training corpus of 6,917 multiple-choice instances.

E.3 Corpus Statistics

Table 7 reports the distribution of the 6,917 retained instances across the 31 ATOMS abilities. Counts are computed using the primary ability label associated with each item. The distribution is not forced to be uniform: some abilities naturally receive more examples because they are easier to instantiate with sufficient evidence and a unique answer, while overly ambiguous or low-quality candidates are removed during curation.

F Implementation Details

We set the global training batch size to 256 and the number of rollout generations to 8. A consistent learning rate of 1e-6 and a temperature parameter of 1 were used. The Qwen3-8B model was trained for 3 epochs and the Qwen3-4B model was trained for 2 epochs, resulting in a total of 650 or 450 optimization steps. All experiments were performed on 8 H100 (80GB) GPUs. Detailed training hyperparameters for the experiments are listed in Table 8

Table 7: Distribution of the 6,917 retained training instances across the 31 ATOMS abilities.

Dimension	Ability	Count
Belief (2990)	Second-order beliefs	1675
	Location false beliefs	585
	Content false beliefs	489
	Beliefs based action/emotions	149
	Identity false beliefs	79
	Sequence false beliefs	13
Emotion (997)	Typical emotional reactions	379
	Hidden emotions	373
	Mixed emotions	86
	Atypical emotional reactions	82
	Moral emotions	38
	Discrepant emotions	21
	Emotion regulation	18
Non-Literal Communication (992)	Faux pas	351
	White lies	280
	Irony/Sarcasm	213
	Egocentric lies	104
	Involuntary lies	37
	Humor	7
Intention (955)	Intentions explanations	768
	Prediction of actions	174
	Completion of failed actions	9
	Discrepant intentions	4
Knowledge (653)	Information-knowledge links	307
	Percepts-knowledge links	269
	Knowledge-attention links	40
	Knowledge-pretend play links	37
Desire (330)	Desire-action contradiction	139
	Desires influence on actions/emotions	126
	Discrepant desires	34
	Multiple desires	31
Total		**6917**

Table 8: Training hyperparameters for the experiments.

Hyperparameter	**Value**
Group size G (rollouts per task)	8
Rollout batch size	32
Global batch size	256
Optimization steps per rollout	1
Max prompt length	8192
Max response length	8192
Training temperature	1.0
Eval temperature	0.6
Entropy coefficient	0.0
clip ϵ_l	0.2
clip ϵ_h	0.28
KL loss coefficient	0.0
Learning rate	1×10^{-6}
LR decay style	constant
Weight decay	0.1
Adam β_1	0.9
Adam β_2	0.98
Optimizer	Adam
GPU	8×NVIDIA H100