CasePlay: Self-Play Reinforcement Learning from Case Reports for Medical Reasoning

Hao Wang[*,1] **Zihan Wang**[*,2] **Kai Wu**[*,3] **Jian Yao**[†,4]
Yiwen Ye[5] **Yongcan Luo**[1] **Dapeng Wu**[1] **Jiale Chen**[5] **Haihua Yang**[†,5]

[1]City University of Hong Kong [2]Northeastern University
[3]Tongji University [4]The Hong Kong Polytechnic University [5]ByteDance

Abstract

Reinforcement learning (RL) is a promising way to improve the clinical reasoning abilities of large language models (LLMs), but current pipelines often rely on expert-labeled questions or costly synthetic-data generation. Meanwhile, the medical literature offers a wealth of high-quality unlabeled documents, including clinical case reports that trace the full clinical course from initial presentation to final treatment decisions. We study whether document-grounded self-play can convert these reports into effective RL training signals. We identify two challenges that limit existing self-play methods in this setting: *knowledge-point fixation*, where the proposer repeatedly asks about a narrow set of salient clues in a source report, and *difficulty-quality mismatch*, where appropriately difficult questions may still be poorly formed or shallow, limiting their value for reasoner training. To address these challenges, we propose CasePlay, a self-play framework with two key components: (a) a *Knowledge-Conditioned Proposer*, which anchors question generation to case-specific knowledge points so that questions draw on a broader range of evidence from the source report; and (b) a *Rubric-Judge Reasoner*, which augments answer-accuracy feedback with rubric-based quality feedback on clue integration, reasoning depth, and option quality, guiding the proposer toward reasoning-intensive questions that better support reasoner training. Extensive experiments across seven medically related benchmarks demonstrate that CasePlay outperforms existing self-play baselines and improves the SFT warm-start model by 3.1 points on average, highlighted by a 6.3-point gain on LiveClin. This approach offers an effective path toward continually improving medical reasoning ability through self-play RL.

1 Introduction

Large language models (LLMs) have shown growing promise in medical reasoning, including differential diagnosis [1–5], clinical decision support [6–9], and health question answering [10–12]. Many recent advances in medical LLMs have followed the common post-training paradigm of supervised fine-tuning (SFT) followed by reinforcement learning (RL), which has proven effective for improving reasoning ability. However, scaling this recipe remains costly as it depends on expert annotations and synthetic data pipelines, a bottleneck that becomes increasingly severe as models improve. In contrast, medicine already offers a continually growing collection of unlabeled documents (e.g., case reports) that contain rich real-world medical knowledge. This raises a fundamental question: *can we continually improve medical reasoning by learning directly from unlabeled clinical corpora?*

[*]Equal contribution.
[†]Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

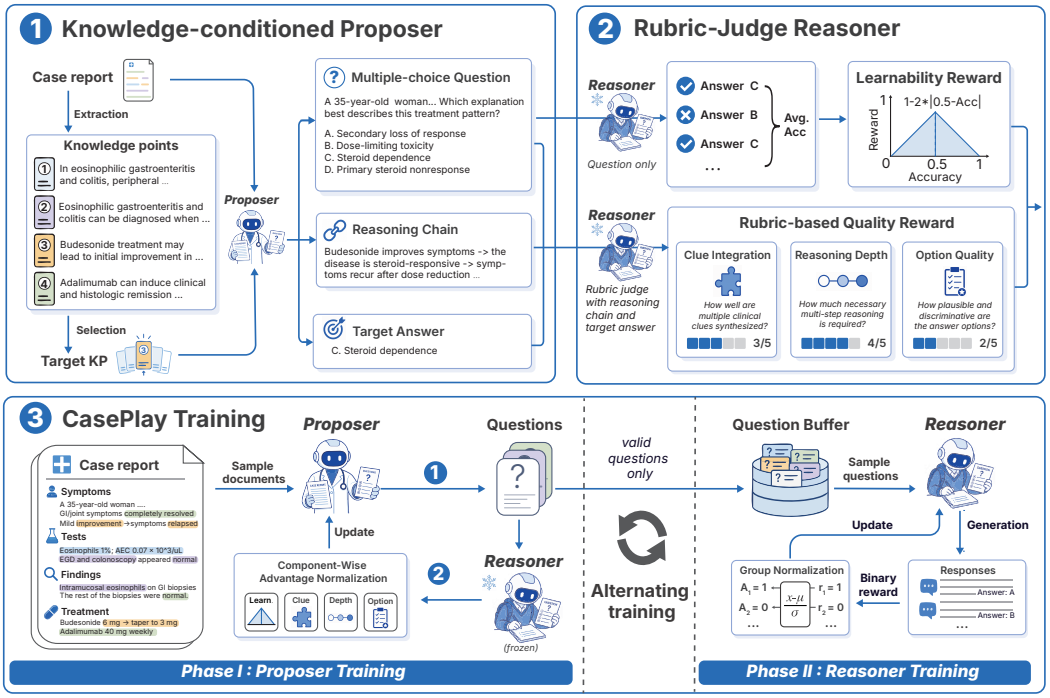


Figure 1: Overview of CasePlay.

A promising direction is self-play RL over text corpora [13–21]. In this paradigm, a *Proposer* generates questions from either its internal knowledge or external documents, and a *Reasoner* learns to solve them. The reasoner's performance provides feedback for updating the proposer, while the generated questions provide training data for improving the reasoner, forming an automated loop for deriving learning signals from unlabeled corpora. Although this paradigm has shown promising results in domains such as mathematics [22, 23] and code [24], directly applying it to the medical domain raises two challenges: (a) *knowledge-point fixation*: medical case reports are information-dense and often contain multiple interdependent clinical clues, yet a naive proposer tends to anchor on the most salient evidence and repeatedly generate questions from the same narrow perspective; and (b) *difficulty-quality mismatch*: existing self-play methods often use the reasoner's answer accuracy as the main feedback signal, treating intermediate-difficulty questions as useful training data. However, such questions may still be poorly constructed or require only shallow reasoning, making them poor candidates for improving medical reasoning through RL.

To address these challenges, we propose **CasePlay**, a self-play learning framework consisting of two key components: (a) a **Knowledge-Conditioned Proposer**, which conditions question generation on case-specific knowledge points to broaden coverage of each source report and reduce redundant questions around the same salient clue; and (b) a **Rubric-Judge Reasoner**, which supplements answer-accuracy feedback with rubric-based quality feedback along three dimensions: clue integration, reasoning depth, and option quality. This quality-aware feedback guides the proposer toward higher-quality questions that provide more useful training examples for the reasoner. Experiments on seven medically related benchmarks show that CasePlay improves the warm-start model by **3.1 points** on average and outperforms existing self-play baselines, with the largest gain on LiveClin [25] (**+6.3 points**). Together, these results position CasePlay as a new way to leverage unlabeled case reports for improving medical reasoning through self-play RL. In summary, our contributions are threefold:

- We study self-play RL over clinical case reports for improving medical reasoning, and empirically identify two characteristic failure modes of existing methods: *knowledge-point fixation* and *difficulty-quality mismatch*.
- We propose CasePlay, a medical self-play framework where a Knowledge-Conditioned Proposer uses case-specific knowledge points to encourage broader document coverage, while a Rubric-Judge Reasoner leverages rubric-based feedback as an additional quality

signal for proposer optimization, helping generate less redundant and higher-quality training questions for improving medical reasoning.

- We validate CasePlay through extensive experiments on 7 medically related benchmarks, where it improves the warm-start model by *3.1 points* on average, outperforms medical reasoning models and self-play baselines, and remains effective across ablations, hyperparameter settings, and model scales.

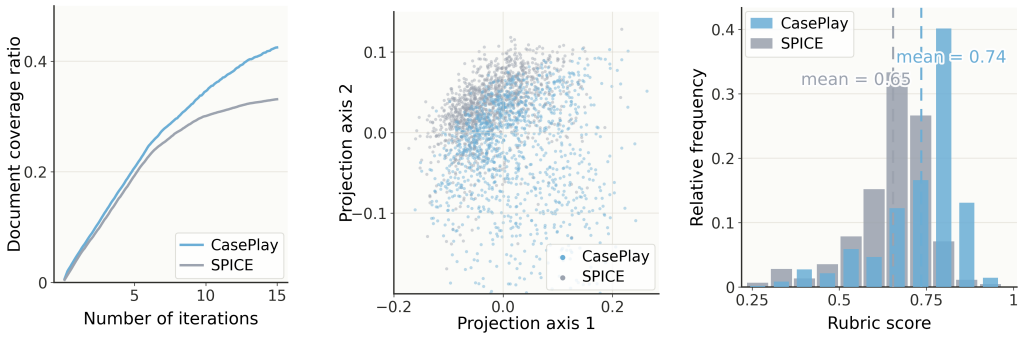
2 Pitfalls of Self-Play RL in Medical Reasoning

Preliminaries. We formulate self-play training as an alternating game between a *Proposer* π_θ^P and a *Reasoner* π_ϕ^R over an unlabeled document corpus $\mathcal{D}$:

$$\mathcal{G} = \left\langle \mathcal{M}_1^P(\phi_0), \mathcal{M}_1^R(\theta_1), \mathcal{M}_2^P(\phi_1), \mathcal{M}_2^R(\theta_2), \ldots, \mathcal{M}_T^P(\phi_{T-1}), \mathcal{M}_T^R(\theta_T) \right\rangle, \tag{1}$$

where T is the number of self-play iterations. At iteration t, the proposer is first updated against the frozen reasoner $\pi_{\phi_{t-1}}^R$, and the reasoner is then updated using valid questions accumulated during the proposer-update phase. In the proposer phase $\mathcal{M}_t^P = (\mathcal{O}^P, \mathcal{A}^P, R^P)$, the observation $o_t^P \in \mathcal{O}^P$ is a batch of clinical documents $\{d_{t,i}\}_{i=1}^B \subset \mathcal{D}$, the action $a_t^P \in \mathcal{A}^P$ is to generate grounded multiple-choice questions with target answers, and the reward r_t^P is computed from the current reasoner's performance on these questions. In the reasoner phase $\mathcal{M}_t^R = (\mathcal{O}^R, \mathcal{A}^R, R^R)$, the observation $o_t^R \in \mathcal{O}^R$ consists only of proposer-generated training questions, without access to the source documents; the action $a_t^R \in \mathcal{A}^R$ is to answer these questions; and the reward r_t^R is a binary correctness reward.

However, existing self-play RL algorithms often yield limited gains when directly applied to medical corpora. To understand the limitations of current self-play methods in this setting, we analyze SPICE, a representative document-grounded baseline, and identify two recurring pitfalls in medical self-play: *knowledge-point fixation* and *difficulty-quality mismatch*. Details of the visualization are provided in Appendix B.



(a) Source-document coverage. (b) Question distribution. (c) Quality at matched difficulty.

Figure 2: (a) Cumulative source-document coverage of generated questions over training. (b) MDS [26] projection of generated-question embeddings. (c) Rubric-based quality-score distribution among intermediate-difficulty questions with reasoner accuracy of 0.5.

Knowledge-Point Fixation. Clinical case reports are information-dense, often containing multiple clinical clues about symptoms, treatments, outcomes, and disease mechanisms. Ideally, document-grounded self-play should turn these diverse clues into diverse training questions. However, when the proposer is conditioned only on the full document, it tends to anchor on the most salient evidence and repeatedly generate questions from the same narrow subset of the report. Figure 2(a) shows that SPICE's source-document coverage quickly saturates at 0.33, compared with 0.42 for CasePlay, indicating that later rollouts mostly reuse already-covered content rather than explore new case evidence. The resulting questions are also less diverse: in Figure 2(b), SPICE forms a tighter cluster and occupies a smaller region than CasePlay in the projection space. We refer to this failure mode as *knowledge-point fixation*, where an information-rich case report is reduced to redundant training

questions from a narrow subset of clinical evidence. This observation indicates that the proposer needs an explicit mechanism to control which part of the case report each rollout should explore.

Difficulty-Quality Mismatch. Existing self-play methods often update the proposer using reasoner-estimated difficulty, favoring questions that are neither too easy nor too hard [13, 14, 17]. However, intermediate difficulty does not guarantee training value in medical reasoning: such questions may still be poorly constructed, or require only shallow reasoning, making them poor candidates for improving medical reasoning through RL. To examine this issue, we select questions with empirical reasoner accuracy of 0.5 and evaluate their quality using the rubrics introduced in Section 3.2. As shown in Figure 2(c), even within this intermediate-difficulty set, SPICE produces many low-quality questions, with 31.3% receiving rubric scores no higher than 0.6. We then construct two matched training sets from these questions: a random subset and a rubric-filtered subset that keeps questions with average rubric score above 0.6. As shown in Table 1, RL training on the rubric-filtered set improves average benchmark accuracy from 63.1 to 64.3, whereas random selection reaches only 63.5 and reduces the average response length. This highlights the need for quality-aware feedback that goes beyond difficulty and favors well-formed, reasoning-intensive questions.

Table 1: Impact of question quality.

Method	Avg. Acc.	Resp. Len.
Warm-start Model	63.1	3794.5
+ RL (Random)	63.5	3025.4
+ RL (Filtered)	**64.3**	**4932.8**

3 Proposed Solution: CasePlay

In this paper, we propose **CasePlay**, a self-play framework for improving medical reasoning from case reports. The central idea of CasePlay is to turn two implicit choices in document-grounded self-play into explicit controls: *which clinical focus* in a case report the proposer should explore, and *how* to evaluate the quality of generated questions for training. CasePlay implements these controls with a *Knowledge-Conditioned Proposer*, which guides question generation with case-specific knowledge points, and a *Rubric-Judge Reasoner*, which provides quality-aware feedback for proposer learning.

3.1 Knowledge-Conditioned Proposer

Prior document-grounded self-play methods typically condition the proposer on the entire document [17, 18], leaving the clinical focus implicit during the generation process. Conceptually, generating o from a case report d can be viewed as marginalizing over a latent focus z:

$$\pi_\theta(o \mid d) = \sum\nolimits_z \pi_\theta(z \mid d)\pi_\theta(o \mid d, z), \tag{2}$$

where o denotes the proposer output. When z is left implicit, the proposer must choose the focus while writing the question, often biasing generation toward salient clues such as the final diagnosis. Chunking offers a crude focus proxy, but can remove the temporal or treatment context needed for faithful medical questions. We therefore make z explicit as a document-grounded knowledge point: a discrete clinical focus in the report that guides what the proposer should ask about, while the full report remains available as context.

Knowledge-Point Extraction. Before self-play, for each document $d \in \mathcal{D}$, we use the proposer to extract candidate knowledge points from the source report, remove redundant or overlapping candidates, and retain at most M of them, denoted as $K(d) = \{k_1, \ldots, k_M\}$. Each knowledge point k_m summarizes a clinically meaningful aspect of the case, such as a symptom progression, diagnostic clue, or outcome pattern, and is supported by explicit evidence in the source document, as illustrated in Appendix Figure 6. We keep $K(d)$ fixed throughout self-play to provide stable clinical focuses and avoid introducing additional non-stationarity into proposer optimization. Details of the extraction procedure are provided in Appendix C.3.

Knowledge-Conditioned Question Generation. CasePlay implements the latent-focus view in Eq. (2) by factorizing each proposer rollout into knowledge-point selection and knowledge-conditioned generation. Specifically, for the j-th rollout of document d:

$$\tilde{\pi}_\theta^P(k, o \mid d, K(d), j) = \pi^{\text{sel}}(k \mid K(d), j)\, \pi_\theta^P(o \mid d, k), \tag{3}$$

where k is the selected knowledge point and $o = (q, c, y)$ consists of the generated multiple-choice question q, proposer-provided reasoning chain c, and target answer y. In our implementation, π^{sel} is a fixed round-robin selector, which avoids introducing an additional learned selector while remaining computationally efficient and empirically effective. Specifically, for $K(d) = \{k_1, \ldots, k_M\}$:

$$\pi^{\text{sel}}(k = k_m \mid K(d), j) = \mathbb{I}[m = 1 + ((j-1) \bmod M)]\,. \tag{4}$$

Thus, repeated rollouts of the same case report are anchored to different knowledge points, encouraging the proposer to cover multiple clinical focuses rather than repeatedly asking about the same salient one. We compare alternative knowledge-point selection strategies in Appendix E. After generation, each output o is evaluated by the current reasoner to produce feedback rewards for the proposer update, as described next.

3.2 Rubric-Based Feedback from the Reasoner

As discussed in Section 2, answer accuracy mainly reflects a question's difficulty for the current reasoner, but does not necessarily capture whether the question is useful for RL training. Therefore, CasePlay evaluates the generated question with two reward components: *learnability*, estimated from the reasoner's answer accuracy without access to the source report, and *quality*, judged by rubric-based criteria using the target answer and proposer-provided reasoning chain.

Learnability Reward. Given a proposer-generated multiple-choice question q with target answer y, the reasoner samples N_{acc} answers to q without access to the source case report, yielding an estimated accuracy $\text{acc}(q)$. Following [14], the learnability reward is defined as:

$$r^{\text{learn}} = 1 - 2\,|\text{acc}(q) - 0.5|\,, \tag{5}$$

which is maximized when $\text{acc}(q) = 0.5$, encouraging the proposer to generate questions that are neither too easy nor too hard for the current reasoner.

Rubric-Based Quality Reward. Medical questions with similar difficulty can differ substantially in the reasoning they elicit. To assess question quality, we ask the reasoner to judge the generated question together with the target answer y and proposer-provided reasoning chain c. With the proposer-provided answer and reasoning chain, the reasoner can more directly assess whether the intended reasoning path is coherent and whether the options are discriminative, without inferring the rationale from the question alone. Formally, the quality reward is calculated as:

$$\mathbf{r}^{\text{qual}} = \left(r^{\text{clue}}, r^{\text{depth}}, r^{\text{option}}\right) = \pi_\phi^J(o), \tag{6}$$

where the three dimensions capture complementary aspects of question utility for RL training. *Clue integration* measures whether answering the question requires synthesizing multiple clinical clues; *reasoning depth* measures whether the question requires multi-hop medical reasoning rather than surface-level fact matching; and *option quality* measures whether the answer options include plausible and discriminative distractors. Detailed rubric prompts are provided in Appendix C.5.

3.3 Alternating Self-Play Optimization

CasePlay is trained over multiple self-play iterations. Each iteration contains two phases: a *proposer-update phase* and a *reasoner-update phase*. In each phase, we update one model for multiple steps while keeping the other frozen. Both models are optimized with the GRPO [27] objective:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_i \left[\min\left(\rho_i \hat{A}_i,\ \text{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon)\hat{A}_i\right)\right], \tag{7}$$

where $\rho_i = \frac{\pi(o_i|x_i)}{\pi_{\text{old}}(o_i|x_i)}$ is the importance-sampling ratio for the corresponding phase, with x_i denoting the model input and o_i denoting the sampled output. The advantage $\hat{A}_i$ is normalized within each sampled group, with phase-specific reward definitions described below.

Proposer Update with Component-Wise Advantage Normalization. In the proposer-update phase, we sample a batch of documents, and for each document the proposer generates a group of N_p candidate outputs according to the knowledge-conditioned rollout in Eq. (3). The frozen reasoner

evaluates each valid candidate with the learnability reward in Eq. (5) and the rubric-based quality rewards in Eq. (6). If a candidate violates the required format, we assign the format penalty r^{format} to all reward components and skip reasoner evaluation. Since these reward components capture different aspects of question utility, we normalize each component separately within the candidate group before aggregation. For a generated question i, the final proposer advantage $\hat{A}_i^P$ is computed as:

$$\tilde{r}_i^s = \frac{r_i^s - \text{mean}_j(r_j^s)}{\text{std}_j(r_j^s) + \epsilon}, \qquad \hat{A}_i^P = \tilde{r}_i^{\text{learn}} + \lambda \left(\tilde{r}_i^{\text{clue}} + \tilde{r}_i^{\text{depth}} + \tilde{r}_i^{\text{option}} \right), \tag{8}$$

where $s \in \{\text{learn}, \text{clue}, \text{depth}, \text{option}\}$ indexes the reward component and λ controls the overall contribution of rubric-based feedback.

Reasoner Update. During the proposer-update phase, generated valid questions are stored in a training buffer $\mathcal{B}$. After the proposer phase finishes, we sample a subset of questions from this buffer to update the reasoner. For each sampled question, the reasoner generates N_r responses without access to the source report, and each response receives a binary reward: 1 if the final answer matches the target answer and 0 otherwise. These rewards are normalized within the response group to obtain $\hat{A}_i^R$, and the reasoner is updated by Eq. (7). The pseudocode of CasePlay is provided in Appendix C.6.

4 Experiments

4.1 Setup

Models and Baselines. We evaluate CasePlay on two model families and multiple scales, including Qwen3-4B-Base [28], Qwen3-8B-Base [28], and OctoThinker-3B-Hybrid-Base [29]. We compare against two categories of baselines. The first category consists of recent medical reasoning models, including HuatuoGPT-o1-8B [2], AlphaMed-8B [30], and Fleming-R1-7B [31]. The second category consists of self-play methods, including R-Zero [14], SPICE [17], and DARC [18]. Details of baseline implementations are provided in Appendix D.1.

Training Data. For training data collection, we collect open-access case reports from PubMed Central (PMC [32]) published between 2020 and 2025. We apply a strict filtering pipeline to remove incomplete reports, image-dominant articles, and documents with potential benchmark leakage, yielding a final corpus of 4,281 case reports. For each filtered document, we extract up to $M = 4$ knowledge points to support knowledge-conditioned question generation. To ensure stable instruction following and output formatting, we construct a small SFT warm-start dataset from the training case reports. Both CasePlay and all self-play baselines use this dataset for warm-start initialization to ensure a fair comparison. Additional details of training data collection are provided in Appendix C.2.

Evaluation. We evaluate CasePlay on seven benchmarks across three categories. First, we test whether our method improves performance on expert-level medical reasoning benchmarks, including MedXpert-Text [33], MedBullets-OP4/OP5 [34], and MedQA-USMLE [35]. Second, we evaluate clinical case reasoning with LiveClin [25], which focuses on patient narratives and requires integrating case-specific clues from clinical context. Third, we evaluate transfer to scientific QA using MMLU-Pro [36] (Health and Biology subsets) and GPQA [37] (Genetics and Molecular Biology subset). For each benchmark question, we sample 8 responses and report the average accuracy as the final metric. Detailed benchmark descriptions are provided in Appendix D.2.

Implementation Details. We train CasePlay for 15 iterations, with each iteration consisting of 10 proposer-update steps and 10 reasoner-update steps. The rubric weight λ is set to 0.2. For both the proposer and the reasoner, we use a training batch size of 128 and a learning rate of 1×10^{-6}. During training rollouts, we use temperature 1.0 and top-p 0.95, with a maximum input length of 4,096 tokens and a maximum output length of 8,192 tokens. During evaluation, we set the sampling temperature to 0.6 and top-p to 0.95, while keeping the other generation settings unchanged. Full experimental hyperparameters are listed in Appendix D.3.

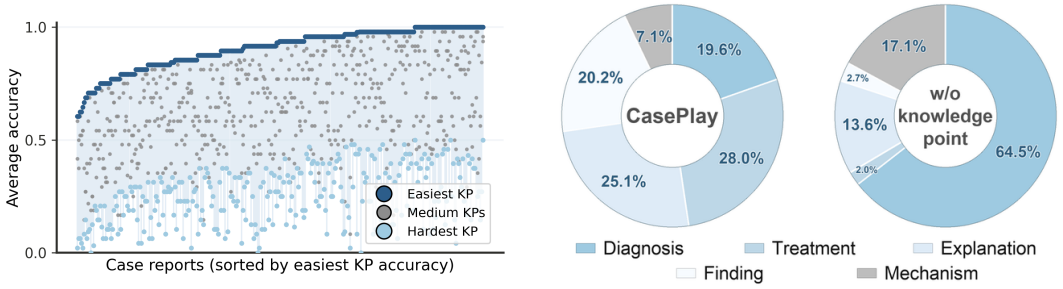
Table 2: Main results on seven medically related benchmarks.

Method	Expert Medical Reasoning				Clin. Case	Science QA		
	MedXpert	MB-OP4	MB-OP5	MedQA	LiveClin	MMLU.	GPQA	Avg.
Base Models								
Qwen3-8B-Base [28]	12.3	41.2	35.3	50.8	48.6	56.4	42.2	41.0
+ SFT (Warm-Start)	22.7	72.9	68.1	83.6	66.0	75.3	53.4	63.1
Medical Reasoning Models								
HuatuoGPT-o1-8B [2]	15.4	52.6	45.2	72.6	55.4	63.1	50.4	50.7
AlphaMed-8B [30]	21.4	62.4	58.7	76.2	62.5	66.7	52.1	57.1
Fleming-R1-7B [31]	19.1	57.8	54.1	75.1	60.2	69.3	48.7	54.9
Self-Play RL Methods								
R-Zero [14]	19.7	72.9	68.6	83.5	66.0	73.3	52.9	62.4
SPICE [17]	21.8	74.2	70.0	83.2	67.1	75.6	54.2	63.7
DARC [18]	23.5	75.7	69.8	84.8	70.5	75.3	55.2	65.0
Proposed Solution								
CasePlay	**25.1**	**76.9**	**70.1**	**85.4**	**72.3**	**75.9**	**57.6**	**66.2**
Impr. vs SFT Model	**+2.4**	**+4.0**	**+2.0**	**+1.8**	**+6.3**	**+0.6**	**+4.2**	**+3.1**
Ablation Study								
w/o knowledge-point	23.3	75.8	69.4	84.7	69.8	75.2	56.3	64.9
w/o rubric judge	22.5	75.0	68.4	83.5	70.8	74.5	55.2	64.3
w/o proposer update	21.4	74.1	67.9	83.2	68.4	73.8	53.9	63.2

4.2 Performance Comparison

As shown in Table 2, CasePlay achieves the best performance on all seven benchmarks. Compared with the SFT warm-start model, CasePlay improves the average score from 63.1% to 66.2%, yielding a +3.1 point gain. The largest gain appears on LiveClin [25] (+6.3 points), because its patient-case narratives require integrating symptoms, diagnostic clues, and treatment context, which matches the reasoning ability improved by self-play over clinical case reports. Among self-play baselines, R-Zero does not use external documents during question generation and slightly underperforms the warm-start model on average. Compared with document-grounded baselines such as SPICE and DARC, CasePlay achieves stronger gains, indicating that medical self-play benefits from broader coverage of case-specific knowledge points and quality-aware proposer feedback.

4.3 Ablation Study



(a) Within-case difficulty variation.

(b) Question-type distribution.

Figure 3: (a) Different knowledge points from the same report yield substantially different difficulty. (b) Knowledge-conditioned generation produces a more balanced distribution of question types.

Knowledge-Conditioned Generation Diversifies Within-Case Questions. As shown in Figure 3(a), we randomly sample 10% of the training documents and visualize the reasoner's accuracy on questions generated from different knowledge points within the same case report. Questions grounded in different knowledge points often have substantially different difficulty, indicating that they probe distinct aspects of the case and provide complementary learning opportunities. We then remove knowledge-point conditioning during question generation and compare the resulting question-type distribution in Figure 3(b). Without knowledge-point conditioning, generation becomes highly skewed toward diagnosis-related questions, which account for 64.5% of all generated questions. In contrast, CasePlay produces a more balanced distribution across the five clinical question types. Consistent with this improved diversity, CasePlay outperforms the variant without knowledge points by 1.3 point on average, improving from 64.9 to 66.2 as shown in Table 2.

Necessity of Proposer Updates. To verify the benefit of proposer updates over static question generation, we conduct an additional ablation in which the proposer is frozen at the SFT warm-start checkpoint throughout training, while the reasoner is still updated on the generated questions. As shown in the last row of Table 2, this variant reaches only 63.2 average accuracy, 3.0 points lower than CasePlay, and also underperforms other document-grounded self-play methods. This result highlights the necessity of proposer adaptation, where the proposer adjusts its questions according to feedback from the current reasoner to keep producing useful training data.

Rubric Judge Improves Question Quality. We compare the average rubric scores of questions generated by CasePlay and by the variant without rubric-judge feedback in Figure 4. For the latter, we still score generated questions with the reasoner for analysis, but exclude rubric rewards from proposer updates. Without rubric-judge feedback, the rubric score remains lower and less stable, indicating that learnability-based feedback alone is insufficient to promote high-quality medical question generation. In contrast, CasePlay steadily increases the average rubric score, helping the proposer generate questions with better clue integration, greater reasoning depth, and more discriminative options. This improvement in question quality translates into better reasoner performance: as shown in Table 2, CasePlay improves the average score from 64.3 to 66.2 compared with the variant without rubric judge.

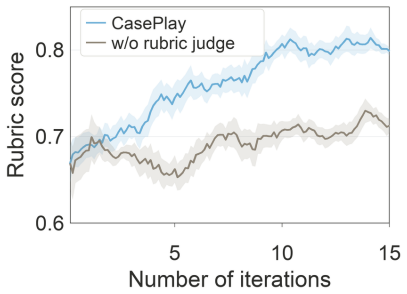


Figure 4: Training curve of rubric score.

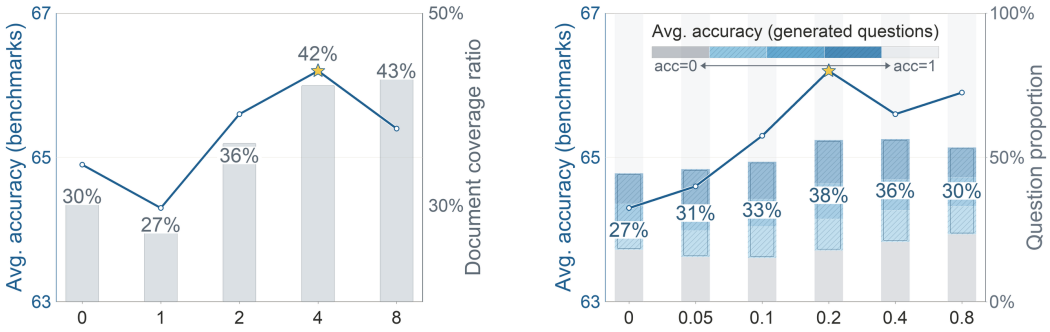
Reliability of Reasoner-Based Rubric Judging. Since the rubric scores are produced by the current reasoner rather than a stronger external judge, an important concern is whether these scores provide reliable quality feedback. To assess this, we compare the reasoner's rubric scores with scores assigned by DeepSeek-V3.2 [38] using the same rubric prompt. As shown in Table 3, the reasoner achieves strong agreement with DeepSeek-V3.2 on *clue integration* and *reasoning depth*, with Pearson correlations of 0.783 and 0.752, respectively. The correlation is lower on *option quality*, likely because judging distractor discriminativeness requires broader medical knowledge. The correlation for the average rubric score reaches 0.705, indicating that the reasoner provides a sufficiently reliable quality signal for proposer optimization.

Table 3: Agreement with DeepSeek-V3.2.

Rubric Dimension	Pearson Corr. ↑
Clue Integration	0.783
Reasoning Depth	0.752
Option Quality	0.627
Average Score	0.705

4.4 Hyperparameter Sensitivity

Impact of the Number of Knowledge Points. The number of extracted knowledge points controls the granularity of clinical focus during question generation. We vary the maximum number of knowledge points per document, including the no-knowledge-point setting, and report the results in Figure 5(a). With too few knowledge points, the proposer conditions generation on a limited set of clinical focuses, leading to lower document coverage and weaker performance. As M increases, document coverage ratio improves, rising from 27% at $M = 1$ to 43% at $M = 8$. However, when M becomes larger than 4, document coverage improves only marginally while downstream accuracy begins to decline, indicating that too many knowledge points can make the clinical focuses overly



(a) Number of knowledge points M

(b) Rubric weight λ

Figure 5: Sensitivity to (a) the number of knowledge points M and (b) the rubric weight λ.

fine-grained or weakly distinguishable, weakening the quality of generated questions. We therefore set $M = 4$ as the default number of knowledge points per case report.

Impact of Rubric Weight. The rubric weight controls the trade-off between learnability and question quality during proposer optimization. In Figure 5(b), we report the proportion of learnable questions in GRPO (i.e., $0 < \text{acc}(q) < 1$). When λ is small, the proposer is primarily guided by estimated difficulty, so it may generate questions that are non-trivial for the reasoner but still shallow or poorly constructed. Increasing λ to 0.2 raises the learnable-question proportion from 27% to 38% and yields the best accuracy, indicating that rubric feedback helps align difficulty with clinically meaningful question quality. However, larger λ over-emphasizes rubric scores relative to learnability, encouraging questions that are more complex but less matched to the current reasoner's capability. We therefore set $\lambda = 0.2$ to balance quality guidance with learnability for proposer training.

Table 4: Performance of CasePlay across different base models.

Method	**Expert Medical Reasoning**				**Clin. Case**	**Science QA**		
	MedXpert	MB-OP4	MB-OP5	MedQA	LiveClin	MMLU.	GPQA	Avg.
Qwen3-4B-Base								
SFT (Warm-Start)	16.8	65.6	58.4	72.4	57.3	66.5	43.5	54.4
R-Zero [14]	16.3	65.5	59.4	73.8	59.4	68.4	44.2	55.3
SPICE [17]	17.6	66.4	60.9	77.6	61.7	70.7	45.5	57.2
DARC [18]	18.4	67.1	60.8	78.4	62.6	71.4	**47.8**	58.1
CasePlay	**19.3**	**67.6**	**61.6**	**79.1**	**64.4**	**72.2**	47.4	**58.8**
OctoThinker-3B-Hybrid-Base								
SFT (Warm-Start)	12.0	45.8	38.2	57.0	48.9	45.5	28.2	39.4
R-Zero [14]	10.5	44.5	39.4	55.4	49.6	45.1	28.4	39.0
SPICE [17]	12.9	47.2	41.7	58.5	50.8	47.1	29.8	41.1
DARC [18]	12.6	47.9	40.1	58.0	51.3	46.2	28.9	40.7
CasePlay	**13.5**	**49.1**	**41.8**	**60.8**	**51.8**	**47.8**	**30.2**	**42.1**

4.5 Robustness Across Model Scales and Families

Table 4 examines whether the benefits of CasePlay extend beyond Qwen3-8B-Base. On Qwen3-4B-Base, CasePlay improves the average score from 54.4% to 58.8%, a gain of 4.4 points, and outperforms the strongest self-play baseline, DARC, by 0.7 points on average. On OctoThinker-3B-Hybrid-Base, CasePlay again achieves the best average performance, improving the warm-start

model from 39.4% to 42.1%, a gain of 2.7 points. These results indicate that CasePlay is not tied to a specific base model and remains effective across different model scales and families.

5 Conclusion

In this paper, we show that directly applying document-grounded self-play RL to medical corpora is limited by two failure modes: *knowledge-point fixation* and *difficulty-quality mismatch*. To address these issues, we introduce CasePlay, which makes the clinical focus of question generation explicit through knowledge-point conditioning and augments proposer optimization with rubric-based quality feedback from the reasoner. Across seven medically related benchmarks, CasePlay consistently improves the warm-start model and outperforms existing self-play baselines, with especially strong gains on clinical case reasoning. Further ablations and generalization experiments support the robustness of these gains. These results demonstrate that CasePlay provides a scalable way to turn clinical case reports into continual training signals for improving medical reasoning.

Acknowledgments

This paper is partially supported by the Hong Kong Research Grants Council grant R1018-25F.

References

[1] Xiaoke Huang, Juncheng Wu, Hui Liu, Xianfeng Tang, and Yuyin Zhou. m1: Unleash the potential of test-time scaling for medical reasoning with large language models. *arXiv preprint arXiv:2504.00869*, 2025.

[2] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*, 2024.

[3] Shuyang Jiang, Yusheng Liao, Zhe Chen, Ya Zhang, Yanfeng Wang, and Yu Wang. Meds^3: Towards medical slow thinking with self-evolved soft dual-sided process supervision. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor, editors, *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 31319–31327. AAAI Press, 2026.

[4] Keer Lu, Zheng Liang, Youquan Li, Jiejun Tan, Xili Wang, Da Pan, Shusen Zhang, Guosheng Dong, Bin Cui, Yunhuai Liu, et al. Med-r^3: Enhancing medical retrieval-augmented reasoning of llms via progressive reinforcement learning. *arXiv preprint arXiv:2507.23541*, 2025.

[5] Qiaoyu Zheng, Yuze Sun, Chaoyi Wu, Weike Zhao, Pengcheng Qiu, Yongguo Yu, Kun Sun, Jian Zhang, Yanfeng Wang, Ya Zhang, et al. End-to-end agentic rag system training for traceable diagnostic reasoning. *arXiv preprint arXiv:2508.15746*, 2025.

[6] Pengcheng Qiu, Chaoyi Wu, Junwei Liu, Qiaoyu Zheng, Yusheng Liao, Haowen Wang, Yun Yue, Qianrui Fan, Shuai Zhen, Jian Wang, et al. Evolving interactive diagnostic agents in a virtual clinical environment. *arXiv preprint arXiv:2510.24654*, 2025.

[7] Josip Vrdoljak, Zvonimir Boban, Marino Vilović, Marko Kumrić, and Joško Božić. A review of large language models in medical education, clinical decision support, and healthcare administration. *Healthcare*, 13:603, 2025.

[8] Jasmine Chiat Ling Ong, Liyuan Jin, Kabilan Elangovan, Gilbert Yong San Lim, Daniel Yan Zheng Lim, Gerald Gui Ren Sng, Yu He Ke, Joshua Yi Min Tung, Ryan Jian Zhong, Christopher Ming Yao Koh, et al. Large language model as clinical decision support system augments medication safety in 16 clinical specialties. *Cell Reports Medicine*, 6(10), 2025.

[9] Farieda Gaber, Maqsood Shaik, Fabio Allega, Agnes Julia Bilecz, Felix Busch, Kelsey Goon, Vedran Franke, and Altuna Akalin. Evaluating large language model workflows in clinical decision support for triage and referral and diagnosis. *npj Digital Medicine*, 8(1):263, 2025.

[10] Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

[11] Yusheng Liao, Chaoyi Wu, Junwei Liu, Shuyang Jiang, Pengcheng Qiu, Haowen Wang, Yun Yue, Shuai Zhen, Jian Wang, Qianrui Fan, et al. Ehr-r1: A reasoning-enhanced foundational language model for electronic health record analysis. *arXiv preprint arXiv:2510.25628*, 2025.

[12] Han Yang, Mingchen Li, Huixue Zhou, Yongkang Xiao, Qian Fang, and Rui Zhang. One llm is not enough: Harnessing the power of ensemble learning for medical question answering. *medRxiv*, 2023.

[13] Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[14] Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025.

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

[16] Ziyi Yang, Weizhou Shen, Chenliang Li, Ruijun Chen, Fanqi Wan, Ming Yan, Xiaojun Quan, and Fei Huang. Spell: Self-play reinforcement learning for evolving long-context language models. *arXiv preprint arXiv:2509.23863*, 2025.

[17] Bo Liu, Chuanyang Jin, Seungone Kim, Weizhe Yuan, Wenting Zhao, Ilia Kulikov, Xian Li, Sainbayar Sukhbaatar, Jack Lanchantin, and Jason Weston. Spice: Self-play in corpus environments improves reasoning. *arXiv preprint arXiv:2510.24684*, 2025.

[18] Shengda Fan, Xuyan Ye, and Yankai Lin. Darc: Decoupled asymmetric reasoning curriculum for llm evolution. *arXiv preprint arXiv:2601.13761*, 2026.

[19] Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. Teaching models to teach themselves: Reasoning at the edge of learnability. *arXiv preprint arXiv:2601.18778*, 2026.

[20] Gengsheng Li, Jinghan He, Shijie Wang, Dan Zhang, Ruiqi Liu, Renrui Zhang, Zijun Yao, Junfeng Fang, Haiyun Guo, and Jinqiao Wang. R-diverse: Mitigating diversity illusion in self-play llm training. *arXiv preprint arXiv:2602.13103*, 2026.

[21] Fangyuan Li, Pengfei Li, Shijie Wang, Junqi Gao, Jianxing Liu, Biqing Qi, and Yuqiang Li. Wist: Web-grounded iterative self-play tree for domain-targeted reasoning improvement. *arXiv preprint arXiv:2603.22352*, 2026.

[22] Yongxian Wei, Yilin Zhao, Li Shen, Xinrui Chen, Runxi Cheng, Sinan Du, Hao Yu, Xiaohan Wang, Gang Liu, Jiahong Yan, et al. Learning to pose problems: Reasoning-driven and solver-adaptive data synthesis for large reasoning models. *arXiv preprint arXiv:2511.09907*, 2025.

[23] Zhengxin Zhang, Chengyu Huang, Aochong Oliver Li, and Claire Cardie. Better llm reasoning via dual-play. *arXiv preprint arXiv:2511.11881*, 2025.

[24] Yuxiang Wei, Zhiqing Sun, Emily McMilin, Jonas Gehring, David Zhang, Gabriel Synnaeve, Daniel Fried, Lingming Zhang, and Sida Wang. Toward training superintelligent software agents through self-play swe-rl. *arXiv preprint arXiv:2512.18552*, 2025.

[25] Xidong Wang, Shuqi Guo, Yue Shen, Junying Chen, Jian Wang, Jinjie Gu, Ping Zhang, Lei Liu, and Benyou Wang. Liveclin: A live clinical benchmark without leakage. *arXiv preprint arXiv:2602.16747*, 2026.

[26] A. Mead. Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society Series D: The Statistician*, 41(1):27–39, 03 1992. ISSN 2515-7884. doi: 10.2307/2348634.

[27] 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. *arXiv preprint arXiv:2402.03300*, 2024.

[28] 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, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[29] Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025. Preprint.

[30] Che Liu, Haozhe Wang, Jiazhen Pan, Zhongwei Wan, Yong Dai, Fangzhen Lin, Wenjia Bai, Daniel Rueckert, and Rossella Arcucci. Beyond distillation: Pushing the limits of medical llm reasoning with minimalist rule-based rl. *arXiv preprint arXiv:2505.17952*, 2025.

[31] Chi Liu, Derek Li, Yan Shu, Robin Chen, Derek Duan, Teng Fang, and Bryan Dai. Fleming-r1: Toward expert-level medical reasoning via reinforcement learning. *arXiv preprint arXiv:2509.15279*, 2025.

[32] National Library of Medicine. Pubmed central (pmc). `https://pmc.ncbi.nlm.nih.gov/`.

[33] Yuxin Zuo, Shang Qu, Yifei Li, Zhang-Ren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[34] Hanjie Chen, Zhouxiang Fang, Yash Singla, and Mark Dredze. Benchmarking large language models on answering and explaining challenging medical questions. 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)*, pages 3563–3599, 2025.

[35] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[36] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

[37] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[38] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[39] Zhihong Shao, Yuxiang Luo, Chengda Lu, ZZ Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*, 2025.

[40] Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

[41] Jian Yao, Ran Cheng, Xingyu Wu, and Jibin Wu. Diversity-aware policy optimization for large language model reasoning. *Advances in Neural Information Processing Systems*, 38:94801–94826, 2026.

[42] Jian Yao, Xiongcai Luo, Ran Cheng, and Kay Chen Tan. Slat: Segment-level adaptive trimming for efficient cot reasoning. *arXiv preprint arXiv:2605.30832*, 2026.

[43] Hao Wang, Hao Gu, Hongming Piao, Kaixiong Gong, Yuxiao Ye, Xiangyu Yue, Sirui Han, Yike Guo, and Dapeng Wu. Learning while staying curious: Entropy-preserving supervised fine-tuning via adaptive self-distillation for large reasoning models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13567–13581, 2026.

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

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

[46] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[47] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

[48] Ali Hatamizadeh, Syeda Nahida Akter, Shrimai Prabhumoye, Jan Kautz, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. Rlp: Reinforcement as a pretraining objective. *arXiv preprint arXiv:2510.01265*, 2025.

[49] Ximing Lu, David Acuna, Jaehun Jung, Jian Hu, Di Zhang, Shizhe Diao, Yunheng Zou, Shaokun Zhang, Brandon Cui, Mingjie Liu, et al. Golden goose: A simple trick to synthesize unlimited rlvr tasks from unverifiable internet text. *arXiv preprint arXiv:2601.22975*, 2026.

[50] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.

[51] Bingxiang He, Yuxin Zuo, Zeyuan Liu, Shangziqi Zhao, Zixuan Fu, Junlin Yang, Cheng Qian, Kaiyan Zhang, Yuchen Fan, Ganqu Cui, et al. How far can unsupervised rlvr scale llm training? *arXiv preprint arXiv:2603.08660*, 2026.

[52] Siheng Li, Kejiao Li, Zenan Xu, Guanhua Huang, Evander Yang, Kun Li, Haoyuan Wu, Jiajia Wu, Zihao Zheng, Chenchen Zhang, et al. Reinforcement learning on pre-training data. *arXiv preprint arXiv:2509.19249*, 2025.

[53] Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, et al. Medreason: Eliciting factual medical reasoning steps in llms via knowledge graphs. *arXiv preprint arXiv:2504.00993*, 2025.

[54] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H. Chen, Tom J. Pollard, Joyce C. Ho, and Tristan Naumann, editors, *Conference on Health, Inference, and Learning, CHIL 2022, 7-8 April 2022, Virtual Event*, Proceedings of Machine Learning Research, pages 248–260. PMLR, 2022.

[55] Kevin Wu, Eric Wu, Rahul Thapa, Kevin Wei, Angela Zhang, Arvind Suresh, Jacqueline J Tao, Min Woo Sun, Alejandro Lozano, and James Zou. Medcasereasoning: Evaluating and learning diagnostic reasoning from clinical case reports. *arXiv preprint arXiv:2505.11733*, 2025.

[56] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[57] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[58] Hynek Kydlíček. Math-Verify: Math verification library, 2025. URL `https://github.com/huggingface/math-verify`. Version 0.6.1. Apache-2.0 license.

[59] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

[60] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[61] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018.

Appendix Contents

A Related Work **15**

B Details of Visualization **15**

C Implementation Details **16**

C.1 Overview 16

C.2 Data Preparation 16

C.3 Knowledge-Point Extraction 18

C.4 Knowledge-Conditioned Question Generation 18

C.5 Rubric-Based Judge with the Reasoner 19

C.6 Self-Play Training 19

D Experimental Setup **24**

D.1 Comparison Baselines 24

D.2 Benchmarks 24

D.3 Training Setup and Hyperparameters 25

E Additional Experimental Results **25**

F Limitations **27**

G Broader Impacts **28**

A Related Work

RL for LLM Post-Training. RL has become increasingly important in LLM post-training, especially for improving reasoning in domains where rewards can be automatically verified [39–43]. Recent representative works, such as OpenAI-o1 [44] and DeepSeek-R1 [45], rely on large amounts of synthesized or curated training data with verifiable rewards, enabling effective RL training but often requiring substantial effort for data construction, verification, and filtering. Another line of work explores more weakly supervised or unsupervised alternatives, deriving training signals either from the model's own prior, such as majority voting [46] and semantic similarity [47], or from unlabeled text [48–50]. The former can be brittle when model confidence is misaligned with correctness [51], while the latter seeks to extract supervision directly from raw text and thereby provides an alternative source of training signal. For example, Golden Goose [49] extends the fill-in-the-middle paradigm to construct supervision from document context, while Quiet-STaR [50] and RLPT [52] optimize future tokens as supervision targets. These developments motivate our setting: medicine contains abundant high-quality unlabeled clinical text, making it a natural domain for studying whether self-play RL can convert real clinical corpora into useful supervision for medical reasoning.

RL for Medical Reasoning Models. A growing body of work applies RL post-training to medical reasoning and reports promising gains on medical benchmarks [2, 11, 53]. HuatuoGPT-o1 [2] is among the earliest efforts in this direction, using exam-style multiple-choice questions such as MedQA-USMLE [35] and MedMCQA [54] as verifiable training data. Fleming-R1 [31] and MedReason [53] further synthesize QA-style training instances by extracting medical entities and relations to construct medical knowledge graphs. AlphaMed [30] builds a carefully curated medical RL dataset and highlights the importance of informative questions and diverse difficulty levels for effective training. MedCaseReasoning [55] identifies clinical case reports as a valuable source of medical reasoning data, but converts them into a static supervised dataset through extensive offline processing. Despite their encouraging progress, these methods predominantly rely on fixed offline supervision, including exam-style QA pairs, curated benchmarks, and synthesized reasoning datasets. In contrast, CasePlay treats clinical case reports not as a source of static annotations, but as grounding documents in an adaptive self-play loop that dynamically generates training questions for the current reasoner.

Self-Play RL. Self-play has long been a central paradigm in RL, with early successes in games such as AlphaZero [56], where models improve by repeatedly competing against themselves and refining their policies over iterations. Recent work extends this idea to LLM post-training. Absolute Zero [13] and R-Zero [14] are among the earliest attempts to cast post-training as a co-evolutionary process between a problem proposer drawing on internal knowledge and a solver, enabling continual improvement without manually curated datasets. Subsequent methods introduce additional components into this loop. SPICE [17] grounds problem generation in unlabeled text, while WIST [21] explores tree-structured self-play for acquiring domain-relevant knowledge from the open web. Despite this progress, existing self-play methods leave two limitations that are especially important in document-grounded medical reasoning. First, they typically do not explicitly control coverage over distinct knowledge points within a source document, which can cause the proposer to repeatedly anchor on a small subset of salient clues. Second, proposer optimization is still driven primarily by solver performance, which reflects difficulty but provides limited supervision over whether a question is clinically well-formed and useful for RL training. Motivated by these limitations, CasePlay addresses document-grounded medical self-play through knowledge-conditioned question generation and rubric-based quality feedback.

B Details of Visualization

Document Coverage Ratio. To measure how broadly generated questions cover the source case report, we first extract a set of fine-grained clues from each document d, denoted as $\mathcal{C}_d = \{c_1, \ldots, c_{|\mathcal{C}_d|}\}$. These clues are more fine-grained than the knowledge points used by CasePlay and correspond to self-contained clinical facts, findings, interventions, outcomes, or mechanistic statements. We extract these clues once for each document using Qwen3-32B [28], obtaining an average of 48.3 clues per document.

Given a generated question q for document d, we use Qwen3-32B to identify the subset of clues in $\mathcal{C}_d$ that are explicitly used by the question, denoted as $\mathcal{C}_{q,d}^{\text{used}} \subseteq \mathcal{C}_d$. At proposer-update step u, the document coverage ratio is computed by aggregating the unique clues covered by all valid questions generated from the same document up to that step, and then averaging the resulting document-level coverage ratios:

$$\text{DCR}(u) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{\left| \bigcup_{q \in \mathcal{Q}_d^{\leq u}} \mathcal{C}_{q,d}^{\text{used}} \right|}{|\mathcal{C}_d|}, \tag{9}$$

where $\mathcal{Q}_d^{\leq u}$ denotes the set of valid questions generated from document d up to proposer-update step u. Invalid questions are excluded before computing the ratio, and repeated use of the same clue within a document is counted only once.

Question Distribution Visualization. To examine the diversity of questions generated during self-play, we visualize the semantic distribution of questions produced by SPICE and CasePlay. Since the full set of generated questions is large, we randomly sample 100 source documents and analyze all questions generated from these documents. For each question, we concatenate the question lead-in and answer options, and encode the resulting text with Qwen3-Embedding-8B [57] to obtain semantic embeddings. We then compute pairwise cosine distances over the pooled question set from both methods and apply classical multidimensional scaling (MDS) [26] to project the questions into two dimensions. Note that questions from SPICE and CasePlay are embedded and projected jointly rather than separately, so their relative positions in the MDS plane reflect semantic dispersion and overlap under the same embedding model and distance metric. The resulting visualization is shown in Figure 2(b).

Quality Distribution of Intermediate-Difficulty Questions. To compare the quality of questions generated by SPICE and CasePlay, we collect all proposer-generated questions throughout self-play training. We then select intermediate-difficulty questions, defined as questions for which the reasoner achieves an empirical answer accuracy of 0.5. This yields 7,915 questions for SPICE and 9,152 questions for CasePlay. From each set, we randomly sample 5,000 questions for evaluation. We use DeepSeek-V3.2 [38] to score these questions with the same rubric prompt used in our framework. The rubric contains three dimensions: clue integration, reasoning depth, and option quality. We compute the overall quality score by averaging the three rubric dimensions and linearly normalizing the result to $[0, 1]$. The final quality-score distributions are shown in Figure 2(c).

C Implementation Details

C.1 Overview

This section provides implementation details for CasePlay. We begin by describing the construction of the unlabeled clinical case-report corpus from the internet, including document-level filtering and preprocessing (Section C.2). We then describe how we extract a fixed set of document-grounded knowledge points for each retained report (Section C.3). During self-play, the proposer conditions on both the full case report and a selected knowledge point to generate a multiple-choice question, a supporting rationale, and a target answer (Section C.4). The reasoner evaluates each valid question using answer-based learnability feedback and rubric-based quality feedback (Section C.5). Finally, we detail the alternating self-play training procedure for updating the proposer and the reasoner (Section C.6).

C.2 Data Preparation

We collect open-access clinical case reports from the internet published between 2020 and 2025. We apply a three-stage filtering pipeline to ensure that each retained document contains sufficient text-grounded clinical evidence for question generation. *First*, we parse the raw XML files and discard reports that cannot be parsed, lack full text, have incomplete formatting, or are too short (fewer than 500 tokens). *Second*, we perform corpus-level filtering and deduplication. We first remove duplicate entries by document ID, then remove near-duplicate reports using n-gram similarity. We also use regular expressions to count figures and tables, excluding reports dominated by visual content because

Clinical Case Report

A 35-year-old White woman with hypertension and primary hyperaldosteronism presented with intermittent abdominal pain and chronic diarrhea for the past 6 years. She had a few episodes of small rectal bleeding. Review of systems was notable for heartburn and severe pain in her entire spine. The patient denied vomiting, weight loss, or fevers. Family history was notable for ulcerative colitis in her mother. *<omitted parts>*

On physical examination, the patient had temperature 37.2°C, pulse rate 75, and blood pressure 140/100 with oxygen saturation 100% on room air. There was mild epigastric tenderness on deep palpation. Initial laboratory examination showed a leukocyte count of 11.1 $\times$ 10^3/μL (70% neutrophils, 1% eosinophils, absolute eosinophil count 0.07 $\times$ 10^3/μL), hemoglobin 13.2 $\times$ 10^3/μL, platelet count 350 $\times$ 10^3/μL, C-reactive protein 10 mg/dL, blood urea nitrogen 14 mg/dL, serum creatinine 0.76 mg/dL, and vitamin D 25 mg/dL.

An esophagogastroduodenoscopy and colonoscopy at an outside clinic 8 months ago appeared normal. Pathology was notable for intramucosal eosinophil at 53 eos/hpf on duodenal biopsy and 68 eos/hpf on gastric antrum biopsy. Helicobacter pylori was absent. On terminal ileum biopsy, there were 120 intramucosal eos/hpf. Other random colon biopsies showed increased eosinophils of 87 eos/hpf and occasional noncaseating granulomas. However, there were no crypt abscesses or inflammatory cells in the lamina propria. *<omitted parts>*

The patient was started on budesonide 6 mg daily with mild improvement, but her symptoms relapsed with taper to budesonide 3 mg daily. Oral prednisone was contraindicated due to primary hyperaldosteronism. Given steroid dependence and concomitant arthritis, she underwent induction with adalimumab and was bridged to adalimumab 40 mg every 14 days. *<omitted parts>*

Eventually, the patient had complete resolution of her GI and joint symptoms on an adalimumab 40 mg weekly dose after budesonide was tapered off. Five months after adalimumab initiation, her inflammatory markers were normalized *<omitted parts>* The biopsies showed nonspecific antrum erosion and nonspecific mild prominence of eosinophils in the cecum; the rest of the biopsies were normal.

Knowledge Point 1

In eosinophilic gastroenteritis and colitis, peripheral absolute eosinophil count may remain within normal limits despite dense tissue eosinophilic infiltration.

Knowledge Point 2

Eosinophilic gastroenteritis and colitis can be diagnosed based on increased intramucosal eosinophils on biopsy even when endoscopic visualization appears normal.

Knowledge Point 3

Budesonide treatment may lead to initial improvement in eosinophilic gastroenteritis and colitis, but symptoms can relapse upon dose reduction, indicating steroid dependence.

Knowledge Point 4

Adalimumab can induce clinical and histologic remission in adult patients with steroid-refractory eosinophilic gastroenteritis and colitis.

Figure 6: Case study of knowledge-point extraction from a clinical case report.

our experiments focus on text-grounded question generation. To reduce benchmark leakage, we apply both document-level and fine-grained benchmark deduplication. At the document level, we remove any training report whose document ID appears in LiveClin. At the fine-grained level, we split each report into chunks and compare each chunk against every benchmark question using n-gram similarity. If any chunk is detected as a near duplicate of a benchmark question, we discard the entire report from the training corpus. *Third*, for the remaining reports, we remove publication-related and medically irrelevant sections, including acknowledgements, funding statements, author contributions, conflicts of interest, references, and other metadata. When section information is available, we prioritize clinically relevant sections such as case presentation, diagnosis, treatment, outcome, and discussion. Finally, we retain document content that fits within the maximum input length of 4,096 tokens. After preprocessing, the final corpus contains $|\mathcal{D}| = 4{,}281$ case reports. These documents are used only as unlabeled source documents during self-play.

Before self-play, we construct a small warm-start dataset to stabilize instruction following, role-specific behavior, and structured output formatting. Specifically, we randomly sample 20% of the retained training reports and use DeepSeek-V3.2 [38] to generate one proposer question for each

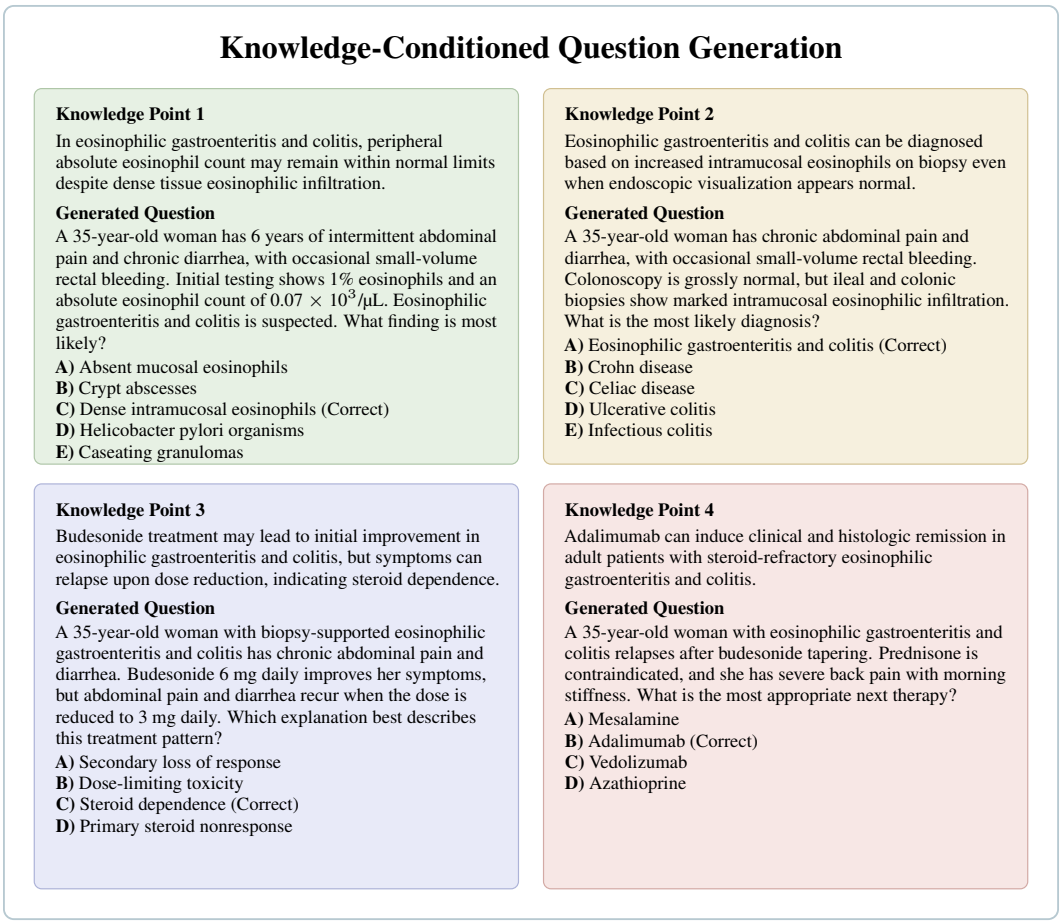


Figure 7: Examples of knowledge-conditioned question generation from the same clinical case report.

extracted knowledge point. For each generated question, we further sample two correct reasoner responses. This process yields 3,384 proposer examples and 6,768 reasoner examples, which are used for SFT warm-up before self-play training.

C.3 Knowledge-Point Extraction

Before self-play training, we extract a set of document-grounded knowledge points $K(d) = \{k_1, \ldots, k_M\}$ for each case report $d \in \mathcal{D}$. Each knowledge point is intended to capture a clinically meaningful and case-specific unit that is explicitly supported by the document, such as a distinctive clinical finding, a treatment-response pattern, or a report-supported disease association or mechanism. The extraction procedure is guided by the prompt in Table 5 and follows three principles: (a) every knowledge point must be grounded in evidence from the case report, without introducing unsupported facts from external medical knowledge; (b) knowledge points extracted from the same report should be mutually non-overlapping and cover different clinical aspects; and (c) we keep at most $M = 4$ knowledge points for each case report in our main experiments. The resulting knowledge points are fixed throughout RL training, which avoids introducing additional non-stationarity into CasePlay. Figure 6 presents an example case report together with its extracted knowledge points. As shown in the figure, non-overlapping knowledge points align with different parts of the original report, helping the proposer cover a broader range of source content during question generation.

C.4 Knowledge-Conditioned Question Generation

As shown in Table 6, we concatenate the case report with the selected target knowledge point and ask the proposer to generate one self-contained USMLE-style multiple-choice question centered on that target. The prompt emphasizes four requirements: (a) the question must remain grounded in

the source document; (b) it should focus on the selected knowledge point rather than test it through direct restatement; (c) it should require nontrivial medical reasoning, preferably integrating at least three distinct clues and involving a two-hop inference chain when possible; and (d) the answer options should be well-formed, with 4 to 8 clinically plausible options of the same semantic type and granularity.

We further require the proposer to return a structured output with five fields: `question_type`, `exam_question`, `options`, `correct_answer`, and `knowledge_and_reasoning_steps`. The `question_type` field is chosen from `diagnosis`, `mechanism`, `finding`, `explanation`, and `treatment`; `exam_question` contains the self-contained vignette and lead-in; `options` contains the answer candidates; `correct_answer` gives the label of the gold option; and `knowledge_and_reasoning_steps` summarizes the intended clue-to-inference chain. Outputs with missing fields, malformed options, non-contiguous option labels, ambiguous correct answers, or unparseable structure are marked invalid. Invalid outputs receive a fixed format penalty r^{format} during proposer optimization and are excluded from reasoner training. Figure 7 presents examples of questions generated from the same source document under different knowledge-point conditions.

C.5 Rubric-Based Judge with the Reasoner

In CasePlay, we use a rubric-based judge prompt to evaluate the quality of each proposer-generated question, providing richer feedback signals for proposer optimization. The full prompt is shown in Table 7. The judge is given the generated question, the target answer, and the proposer-provided reasoning chain, and is asked to assess the question along three dimensions:

- **Clue integration** measures whether answering the question requires synthesizing multiple non-redundant, case-derived clues, rather than relying on a single giveaway fact or a simple knowledge-matching shortcut. It discourages the proposer from generating overly direct questions that can be solved through surface-level clue recognition alone.
- **Reasoning depth** measures whether the question requires a necessary multi-step clinical inference process, rather than shallow fact retrieval or post-hoc explanation. Importantly, the rubric distinguishes between using multiple clues and requiring multiple reasoning hops: several clues pointing directly to the same answer do not by themselves imply deep reasoning.
- **Option quality** measures whether the option set is homogeneous, clinically plausible, and sufficiently discriminative, such that the correct answer is most reliably identified through case-based reasoning rather than wording cues or obviously weak distractors.

Each dimension is scored independently on a 1–5 scale according to the detailed criteria in Table 7. We then linearly normalize the resulting scores to $[0, 1]$ and use them as separate quality reward channels in proposer optimization.

C.6 Self-Play Training

Algorithm 1 summarizes the full training procedure of CasePlay. We first extract a fixed set of document-grounded knowledge points for the training corpus (line 1). In each self-play iteration, we initialize an empty question buffer $\mathcal{B}_t$ (line 3) and then alternate between a proposer-update phase and a reasoner-update phase.

During the proposer-update phase (lines 4–17), we sample a minibatch of documents and, for each document, select rollout-specific knowledge points according to the knowledge-point selection strategy (lines 7–9). The proposer then generates N_p candidate outputs conditioned on the document and the selected knowledge points. Invalid outputs receive the format penalty and are excluded from reasoner evaluation (line 10). For the remaining generated questions, the frozen reasoner samples answers without access to the source document and computes the learnability reward (lines 11–12). The same generated questions are also evaluated by the rubric judge to obtain quality rewards (line 13). We add the generated question-answer pairs to the question buffer and compute normalized proposer advantages from the reward components before updating the proposer with GRPO (lines 15–17).

During the reasoner-update phase (lines 19–28), we sample valid question-answer pairs from the question buffer (line 21). For each sampled question, the reasoner generates N_r responses without

Algorithm 1 Training CasePlay

Require: Proposer π_θ^P, reasoner π_ϕ^R, document corpus $\mathcal{D}$

1: Extract document-grounded knowledge points $\{K(d)\}_{d\in\mathcal{D}}$
2: **for** iteration $t = 1, \ldots, T$ **do**
3: Initialize the question buffer $\mathcal{B}_t \leftarrow \emptyset$
4: **// Phase I: Proposer update**
5: **for** proposer update step $u = 1, \ldots, U_P$ **do**
6: Sample a minibatch of documents $\mathcal{D}_u \subset \mathcal{D}$
7: **for** each document $d \in \mathcal{D}_u$ **do**
8: Select rollout knowledge points $\{k_{d,j}\}_{j=1}^{N_p}$ according to $\pi^{\mathrm{sel}}(\cdot \mid K(d), j)$
9: Sample candidate outputs $\{o_{d,j} = (q_{d,j}, c_{d,j}, y_{d,j})\}_{j=1}^{N_p}$ from $\pi_\theta^P(\cdot \mid d, k_{d,j})$
10: Assign the format penalty r^{format} to invalid outputs and skip their evaluation
11: Sample N_{acc} answers from $\pi_\phi^R(\cdot \mid q_{d,j})$ without access to d
12: Estimate answer accuracy against $y_{d,j}$ and compute $r_{d,j}^{\mathrm{learn}}$ by Eq. (5)
13: Evaluate generated questions with the rubric judge and obtain $(r_{d,j}^{\mathrm{clue}}, r_{d,j}^{\mathrm{depth}}, r_{d,j}^{\mathrm{option}})$
14: Add generated question-answer pairs $\{(q_{d,j}, y_{d,j})\}$ to question buffer $\mathcal{B}_t$
15: Compute normalized proposer advantages $\{\hat{A}_{d,j}^P\}_{j=1}^{N_p}$ by Eq. (8)
16: **end for**
17: Update π_θ^P on the current minibatch by Eq. (7)
18: **end for**
19: **// Phase II: Reasoner update**
20: **for** reasoner update step $u = 1, \ldots, U_R$ **do**
21: Sample a minibatch of valid question-answer pairs $\{(q_i, y_i)\}$ from $\mathcal{B}_t$
22: **for** each sampled pair (q_i, y_i) **do**
23: Sample N_r responses $\{a_{i,j}\}_{j=1}^{N_r} \sim \pi_\phi^R(\cdot \mid q_i)$
24: Assign rewards $\{r_{i,j}^R\}_{j=1}^{N_r}$ according to whether the final answer in $a_{i,j}$ matches y_i
25: Compute group-normalized reasoner advantage $\{\hat{A}_{i,j}^R\}_{j=1}^{N_r}$
26: **end for**
27: Update π_ϕ^R on the current minibatch by Eq. (7)
28: **end for**
29: **end for**

access to the source document (line 23). Each response receives a binary reward according to whether its final answer matches the target answer, and the resulting rewards are group-normalized to obtain reasoner advantages (lines 24–25). The reasoner is also updated using GRPO (line 27).

Table 5: Prompt for Knowledge Point Extraction

```
You are a medical knowledge-point extraction model working on a single case report. Your task is to
extract multiple high-value knowledge points directly from the case report. You must therefore identify
 clinically useful knowledge points from scratch, using only the case report content.

# Input

## Document
<Case Report>

# Goal
Before writing the final output, first read the report and identify clinically meaningful findings,
tests, management details, complications, and outcomes that are explicitly supported. Then propose
candidate knowledge points, remove overlapping candidates, and keep the 4 strongest remaining
candidates based on case-specificity and evidence strength that:
1. are directly supported by the report text
2. are clinically meaningful
3. are mutually non-overlapping
4. reflect distinct reasoning targets, findings, or management implications from the same case
Never invent unsupported facts just to increase the number of points.

# Hard Grounding Rules
You must obey all of the following:
1. Use the case report as the primary source of truth.
2. Every generated point must be supported by explicit content in the report text.
3. Do not use outside medical knowledge to add facts that are not stated in the report.
4. Do not turn weak hints into confident claims.
5. Do not produce generic textbook facts that could fit many unrelated cases.
6. Do not produce image-only points unless the imaging finding is explicitly described in text.
7. Do not produce publication trivia, ethics details, or administrative details.
8. Do not use "this is the first reported case" style novelty as a knowledge point unless it creates a
real clinical teaching point, which is usually not the case.

# Non-Overlap Rules
The generated points must be distinct from each other. Treat the following as duplicates and avoid:
- the same diagnosis phrased in two different ways
- the same clinical clue boundary rewritten in another form
- a broad point and a narrower restatement of the same point

# Preferred Knowledge Point Types
- a clinically important diagnosis stated or strongly established in the report
- a distinctive clinical, laboratory, imaging, pathology, or procedural finding
- a management or treatment response detail that is specific and clinically useful
- a complication, progression pattern, or follow-up finding explicitly shown in the report
- a disease association or mechanism only if the report itself supports it

# Writing Rules
- Write all output in English.
- Each knowledge point should be one concise sentence or clause.
- Keep it specific, case-grounded, and teaching-oriented.
- Avoid repeating the same disease across multiple points unless the clinical boundary is clearly
different.
- Prefer wording that would still be useful as retrieval-oriented supervision later.

# Output Format
Return exactly one JSON object with this schema:
{
  "generated_knowledge_points": [
    "knowledge point 1",
    "knowledge point 2",
    "..."
  ]
}
```

Table 6: Prompt for Proposer Question Generation

```
Your task is to write one self-contained USMLE-style multiple-choice question from the case document for
 student learning. You will be given a case document and one target knowledge point. Your job is to use
 the provided knowledge point to guide the question you write, so that solving the question helps
reinforce that target. When deciding what to ask:
- keep the focus on the provided target knowledge point
- adjust the question angle and distractors so the item clearly teaches and reinforces that target
- aim to teach, reinforce, and clarify the target knowledge point rather than only test it mechanically

# Input

## Document
<Case Report>

## Target Knowledge Point
<Target Knowledge Point>

# Requirements
## Grounding
- The correct answer must be supported by the document.
- Do not invent decisive diagnoses, mechanisms, findings, timelines, management steps, or outcomes.
- The question must be solvable without mentioning the source article.

## Reasoning Quality
- Prefer items that require integrating at least 3 distinct clues.
- Prefer a real 2-hop or 3-hop reasoning chain when possible.
- Avoid direct restatement, buzzword-only items, trivia, and image-related questions.

## Question Type
Choose exactly one `question_type` from `diagnosis`, `mechanism`, `finding`, `explanation`, and `
treatment`.

## Options
- Use 4 to 8 options when possible.
- Use contiguous labels starting from `A`.
- All options must be the same semantic type and level.
- Include plausible near-neighbor distractors.
- Do not use duplicate option text.

# Output Format
1. First write the reasoning content inside a <think>...</think> block.
2. Immediately after </think>, output a JSON object.
3. Do not output markdown, code fences, or any extra text outside the think block and the JSON object.
4. Strictly follow the JSON schema below:
{
  "question_type": "diagnosis | mechanism | finding | explanation | treatment",
  "exam_question": "self-contained vignette and lead-in only",
  "options": {
    "A": "option text",
    "B": "option text",
    "..."
  },
  "correct_answer": "label of the correct answer",
  "knowledge_and_reasoning_steps": [
    "clue -> inference",
    "..."
  ]
}
```

Table 7: Prompt for Rubric Judge

```
You are an expert USMLE item quality reviewer with deep knowledge of clinical reasoning assessment and
NBME item-writing standards.

# Evaluation Dimensions
## 1. Clue Integration
- Definition: Solving the item requires integrating multiple clinically meaningful clues rather than
matching a single giveaway fact.
- Scoring Criteria:
  - 1: Almost entirely single-clue; extra details are decorative and do not materially affect choices.
  - 2: More than one clue is present, but one dominant clue still does nearly all of the work and the
others merely confirm it.
  - 3: At least two meaningful clues must be combined, but the synthesis is still fairly shallow or
straightforward.
  - 4: Multiple distinct clues must be integrated to narrow the differential or choose among close
alternatives; no single clue is sufficient.
  - 5: The answer depends on rich, case-specific synthesis across several clues, including supporting
and discriminating details, and would become unstable if any major clue were removed.

## 2. Reasoning Depth
- Definition: Judge whether the correct answer requires a multi-hop reasoning chain, not merely multiple
clues pointing directly to one answer.
- Scoring Criteria:
  - 1: One-hop reasoning: long vignette, but the answer is still selected from one dominant clue.
  - 2: Appears multi-step on the surface, but one step is weak, optional, or post-hoc explanation rather
 than necessary reasoning.
  - 3: Clear and necessary 2-hop reasoning: clues must first support an intermediate clinical judgment,
and that judgment is then needed to choose the answer.
  - 4: Strong 2-hop or genuine 3-hop reasoning: multiple clues support a necessary intermediate
conclusion, which then supports option-level discrimination against strong distractors.
  - 5: Clear, necessary, and non-compressible 3-hop reasoning: each hop is required, each hop depends on
 the prior one, and at least one strong distractor survives early reasoning but is eliminated only by
the full chain.

## 3. Option Quality
- Definition: The option set supports fair one-best-answer discrimination through comparable, plausible,
and clinically coherent alternatives. The item may contain  4 to 8 options labeled from `A` onward.
Penalize this dimension only when the added options reduce homogeneity, plausibility, or one-best-answer
 fairness.
- Scoring Criteria:
  - 1: Most distractors are weak, implausible, or cueing; one-best-answer discrimination is largely
artificial.
  - 2: Some distractors are plausible, but the set still has noticeable heterogeneity, stylistic cues,
or obvious throwaway options.
  - 3: Options are mostly on the same semantic level and at least one distractor is reasonably strong,
but the correct answer still stands out too easily.
  - 4: Options are homogeneous and clinically plausible, with nearby distractors that require real case-
based reasoning to eliminate.
  - 5: The option set is exceptionally tight: multiple distractors remain genuinely attractive until the
 full case is reasoned through, and the correct answer wins without wording cues or weak-option
shortcuts.

# Counting Criteria
- Clue: A clue is an independent, non-redundant piece of case information that changes answer
discrimination. If two details provide the same clinical signal, count them as one clue, not two.
- Hop: A hop is a necessary intermediate inference step that produces a new clinical conclusion needed
for the next step or the final answer. Do not count post-hoc explanation or optional intermediate
wording as an extra hop.

# Evaluation Rules
- Score each dimension separately. Do not let strength in one compensate for weakness in another.
- Do not confuse "many clues" with "many hops." Multiple clues that all point directly to one diagnosis
 do not by themselves justify a high `reasoning_depth` score.
- Treat 4 to 8 options as acceptable; judge the quality of the set, not the count alone.

# Output Format
Return exactly one JSON object with this schema:
{
  "scores": {
    "clue_integration": { "score": 1, "reason": "Evidence-based rationale" },
    "reasoning_depth": { "score": 1, "reason": "Evidence-based rationale" },
    "option_set_quality": { "score": 1, "reason": "Evidence-based rationale" }
  }
}
# Input
## Questions
<Generated Question>
## Answers
<Generated Target Answer>
## Reasoning chain
<Generated Reasoning Chain>
```

D Experimental Setup

D.1 Comparison Baselines

We compare CasePlay against two groups of baselines: *existing medical reasoning models* and *state-of-the-art self-play methods*. For medical reasoning models, we select three representative models, including:

- **HuatuoGPT-o1 [2]** is among the first works to study how to improve medical reasoning through RL. It first uses SFT to elicit medical reasoning behavior and then applies RL with medical verifiable rewards to enhance reasoning performance.
- **AlphaMed [30]** systematically studies how minimalist rule-based RL can improve model performance on medical reasoning benchmarks. It further highlights the importance of training data diversity for enhancing medical reasoning ability, making it a strong representative of medical RL post-training.
- **Fleming-R1 [31]** follows an offline data construction pipeline that first builds medical knowledge graphs and then synthesizes QA questions from the constructed graphs. The synthesized QA data are then used for RL training, leading to strong performance among medical reasoning models.

For self-play methods, we include one document-free self-play baseline and two recent document-grounded self-play baselines:

- **R-Zero [14]**: R-Zero is a zero-data self-play framework in which a proposer generates questions based on its own internal knowledge, and a reasoner learns from the generated questions. We include this baseline to examine whether self-play that relies only on the model's internal knowledge is sufficient for medical reasoning. In our implementation, we remove both document input and knowledge-point conditioning from the proposer, and also remove rubric-judge feedback. All other training details follow the original R-Zero setup as closely as possible under our backbone model and training budget.
- **SPICE [17]**: SPICE is a document-grounded self-play method that uses unlabeled text to support challenger question generation. Unlike CasePlay, SPICE uses the same model as both proposer and reasoner, and updates them jointly within each batch. Following SPICE, we merge proposer-side and reasoner-side data for warm-start training, and then jointly sample and update the shared model during self-play. We keep the number of update examples matched to CasePlay for a fair comparison.
- **DARC [18]**: DARC is a two-stage document-grounded self-play method that trains the challenger and reasoner separately. Following its original design, we first train the proposer with explicit difficulty targets using grounding and difficulty rewards, and then use the trained proposer to generate questions for reasoner training. The total amount of training data is matched to CasePlay to ensure a fair comparison.

D.2 Benchmarks

We evaluate CasePlay on 7 medically related benchmarks covering expert-level medical reasoning, clinical case reasoning, and scientific QA. All benchmarks are evaluated in a multiple-choice format. During evaluation, we prompt the model to place its final answer in `\boxed{}`, and use `math-verify` [58] to extract the boxed answer and compare it with the ground-truth option. The details of each benchmark are as follows.

- **MedXpert-QA** [33] is an expert-level medical QA benchmark designed to evaluate advanced clinical knowledge and reasoning. In this paper, we evaluate on the MedXpert-Text subset, which contains 2,450 text-only multiple-choice questions.
- **MedBullets-OP4/OP5** [34] consists of exam-style medical questions derived from MedBullets. We evaluate both the four-option and five-option variants, denoted as OP4 and OP5, respectively. Each variant contains 308 questions.
- **MedQA-USMLE** [35] is a widely used medical exam benchmark collected from United States Medical Licensing Examination-style questions, evaluating a model's ability to apply

medical knowledge and clinical reasoning in realistic clinical scenarios. We use the standard test split, which contains 1,273 questions.

- **LiveClin** [25] is a clinical case reasoning benchmark built from patient-centered clinical scenarios. It emphasizes realistic case-based reasoning, including symptom interpretation, diagnosis, and treatment decision-making. Since the original benchmark contains multi-modal questions, we filter out questions that depend on figures or images, resulting in 1,450 text-only questions.
- **MMLU-Pro** [36] is a challenging extension of MMLU with harder questions and expanded answer choices. We evaluate the medically related Biology and Health subsets, which contain 717 and 818 questions, respectively, for a total of 1,535 questions.
- **GPQA** [37] consists of graduate-level scientific QA questions designed to test expert-level scientific reasoning. We evaluate the medically related Genetics and Molecular Biology subsets, which contain 20 and 83 questions, respectively, for a total of 103 questions.

D.3 Training Setup and Hyperparameters

Our implementation is built on Verl [59] and vLLM [60]. Since proposer and reasoner rollouts are executed alternately, we disable GPU colocation in Verl and use Ray [61] to repartition computational resources. Our final setup uses six 8-GPU nodes, with GPUs allocated to the proposer and reasoner in a 2:4 ratio, which yields the best runtime performance in our experiments.

Before self-play, we construct a small warm-start dataset as described in Section C.2. For fair comparison, all self-play methods use this same warm-start dataset for initialization. We warm-start the proposer and reasoner separately for 5 epochs with learning rate 1×10^{-5}. For self-play RL, we optimize the proposer and reasoner alternately with GRPO for $T = 15$ iterations. Each iteration contains $U_P = 10$ proposer-update steps and $U_R = 10$ reasoner-update steps. During proposer rollouts, each document produces $N_p = 4$ proposer candidates, and each candidate is answered by the reasoner $N_{\text{acc}} = 4$ times to estimate learnability. During reasoner updates, we sample $N_r = 8$ responses per selected question. Full hyperparameters are listed in Table 8.

Table 8: Key hyperparameters.

SFT Warm-Start					
Epochs	5	Batch size	128	Max response len.	8,192
Learning rate	1×10^{-5}	LR schedule	Cosine	LR warmup ratio	0.1
Self-Play RL					
Optimization		Rollout		Others	
Learning rate	1×10^{-6}	Proposer samples N_p	4	Invalid penalty r^{format}	-0.1
Train batch	128	Answer samples N_{acc}	4	Learnability weight	1.0
Proposer steps U_P	10	Reasoner samples N_r	8	Rubric weight λ	0.2
Reasoner steps U_R	10	Max input len.	4,096	Number of knowledge points M	4
Clip upper	0.28	Max response len.	8,192		

E Additional Experimental Results

Robustness of Performance Gains and Confidence Intervals. Table 9 reports mean accuracies and standard deviations over three training seeds. Each checkpoint is evaluated with eight responses per question (24 across the three checkpoints); 95% CIs are estimated by percentile bootstrapping over training runs and evaluation questions. CasePlay achieves the highest mean accuracy on all seven benchmarks, with an average accuracy of 66.2%, compared with 65.0% for DARC, 63.7% for SPICE, and 62.4% for R-Zero. The reported confidence intervals for CasePlay lie entirely above those for SPICE on MedQA and LiveClin, and above those for R-Zero on MedXpert, MB-OP4, MedQA, LiveClin, and MMLU-Pro. Although the intervals overlap in the remaining comparisons, the consistent advantage in mean accuracy across all seven benchmarks provides empirical support for the effectiveness of CasePlay in improving medical reasoning under the three-run evaluation.

Table 9: Full results for the main self-play comparisons.

Method	Expert Medical Reasoning				Clin. Case	Science QA		
	MedXpert	MB-OP4	MB-OP5	MedQA	LiveClin	MMLU.	GPQA	Avg.
R-Zero	19.7 ± 2.3	72.9 ± 0.5	68.6 ± 2.5	83.5 ± 0.4	66.0 ± 1.5	73.3 ± 0.4	52.9 ± 2.6	62.4
95% CI	[16.8, 21.6]	[71.8, 74.2]	[65.1, 71.4]	[82.7, 84.1]	[64.8, 69.1]	[72.6, 74.1]	[50.0, 56.5]	—
SPICE	21.8 ± 2.2	74.2 ± 1.0	70.0 ± 2.0	83.2 ± 0.4	67.1 ± 1.2	75.6 ± 0.2	54.2 ± 1.6	63.7
95% CI	[17.1, 24.3]	[72.8, 75.2]	[66.3, 73.1]	[82.4, 83.9]	[65.1, 69.9]	[74.9, 75.9]	[51.8, 57.1]	—
DARC	23.5 ± 1.0	75.7 ± 1.0	69.8 ± 1.4	84.8 ± 0.5	70.5 ± 0.7	75.3 ± 0.1	55.2 ± 1.3	65.0
95% CI	[22.0, 25.4]	[73.6, 77.5]	[66.2, 73.5]	[83.1, 86.7]	[68.7, 72.8]	[74.3, 76.4]	[53.6, 58.4]	—
CasePlay	$\mathbf{25.1 \pm 1.2}$	$\mathbf{76.9 \pm 1.4}$	$\mathbf{70.1 \pm 1.8}$	$\mathbf{85.4 \pm 0.5}$	$\mathbf{72.3 \pm 0.8}$	$\mathbf{75.9 \pm 0.3}$	$\mathbf{57.6 \pm 1.0}$	**66.2**
95% CI	[23.5, 27.9]	[74.8, 79.9]	[67.4, 73.4]	[84.6, 86.0]	[70.9, 74.0]	[75.0, 76.8]	[55.8, 59.8]	—

Table 10: Comparison with GRPO on a fixed synthetic question pool.

Method	Expert Medical Reasoning				Clin. Case	Science QA		
	MedXpert	MB-OP4	MB-OP5	MedQA	LiveClin	MMLU.	GPQA	Avg.
Static Synthetic Data + GRPO	22.6	75.0	69.5	84.1	66.9	75.6	55.3	64.1
CasePlay	**25.1**	**76.9**	**70.1**	**85.4**	**72.3**	**75.9**	**57.6**	**66.2**

Effectiveness of the Self-Play Training Paradigm. To assess adaptive self-play relative to a fixed-data training pipeline, we use DeepSeek-V3.2 [38] to generate approximately 19K question–answer pairs from the same 4,281 case reports. This synthetic pool remains fixed throughout GRPO training. We match the warm-start reasoner, the number of training pairs, and the reasoner GRPO update budget to CasePlay. As shown in Table 10, CasePlay outperforms the static baseline on all seven benchmarks, achieving an average accuracy of 66.2% versus 64.1%. This comparison supports the effectiveness of adaptive self-play under the matched reasoner update budget. A plausible explanation is that adapting question generation to the evolving reasoner helps maintain informative response groups: questions that elicit uniformly correct or incorrect responses provide no correctness-driven group-relative learning signal under binary rewards.

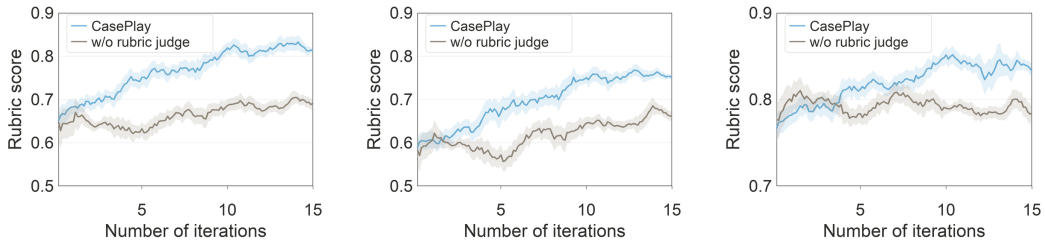
Table 11: Continuing medical RL post-training with CasePlay.

Method	Expert Medical Reasoning				Clin. Case	Science QA		
	MedXpert	MB-OP4	MB-OP5	MedQA	LiveClin	MMLU.	GPQA	Avg.
Fleming-R1-7B	19.1	57.8	54.1	75.1	60.2	69.3	48.7	54.9
+ CasePlay	**20.2**	**61.6**	**57.9**	**76.5**	**65.3**	**69.6**	**50.2**	**57.3**

Compatibility with Medical RL Post-Training. We next investigate whether CasePlay can further improve a reasoner that has already undergone medical RL post-training. We initialize the reasoner from Fleming-R1-7B [31] and continue training with CasePlay on the clinical case-report corpus, using the same Avg@8 evaluation protocol. As shown in Table 11, average accuracy increases from 54.9% to 57.3%, a gain of 2.4 percentage points. Performance improves on all seven benchmarks, with the largest gain on LiveClin, from 60.2% to 65.3% (+5.1 percentage points). This continuation experiment shows that CasePlay provides additional gains for the evaluated medical RL model, supporting its use as a subsequent training stage after medical RL post-training.

Quality Analysis of Generated Questions. We next examine how the rubric scores of generated questions evolve during training. Figure 8 compares the three rubric scores throughout training with and without rubric-based feedback. Incorporating this feedback improves all three dimensions, with the most pronounced gains in clue integration and reasoning depth. This pattern suggests that the rubric encourages the proposer to construct questions requiring the integration of multiple

clinical clues and multistep reasoning, extending the training signal beyond estimated difficulty. Improvements in option quality are more modest. One plausible explanation is that constructing discriminative distractors depends more heavily on the proposer's parametric medical knowledge, while clue integration and reasoning depth can be influenced more directly through the selection and composition of case evidence. The variant without rubric feedback also improves at some stages, but its scores remain lower and less stable. These results indicate that learnability-based feedback alone provides weaker control over the quality of generated questions.



(a) Clue integration (b) Reasoning depth (c) Option quality

Figure 8: Training curves of rubric scores.

Impact of Knowledge-point Selection Strategy. We compare three strategies for selecting knowledge points during question generation: *random*, *round-robin*, and *accuracy-priority*. For a document d with extracted knowledge points $K(d) = \{k_1, \ldots, k_M\}$, the *random* strategy uniformly samples one knowledge point from $K(d)$ at each rollout. The *round-robin* strategy selects knowledge points cyclically, as described in Eq. (4). The *accuracy-priority* strategy samples knowledge points according to the current reasoner's average answer accuracy on questions generated from each knowledge point. Specifically, for knowledge point k_m, its sampling probability is defined by applying a softmax over the learnability scores, $p(k_m \mid d) = \mathrm{softmax}_{k_m \in K(d)}(1 - 2|\bar{a}_m - 0.5|)$, where $\bar{a}_m$ is the current reasoner's average accuracy on questions associated with k_m.

Table 12: Impact of knowledge-point selection strategies.

Strategy	DCR ↑	Avg. Acc. ↑
Random	0.40	65.7
Acc.-priority	0.32	64.1
Round-robin	**0.42**	**66.2**

As shown in Table 12, *round-robin* achieves the best performance, with the highest document coverage ratio (0.42) and the highest average accuracy (66.2). The *random* strategy is competitive, but its stochastic selection can repeatedly sample the same knowledge points, leading to slightly lower coverage and downstream performance. In contrast, *accuracy-priority* substantially reduces document coverage (0.32) and achieves the lowest average accuracy. Since each document is sampled only a limited number of times during self-play, prioritizing knowledge points based on noisy intermediate accuracy estimates can over-concentrate generation on a small subset of knowledge points, reducing within-document diversity and weakening the effectiveness of self-play RL. We therefore adopt *round-robin* as the default strategy in CasePlay.

F Limitations

Despite the strong empirical gains of CasePlay, our current implementation extracts a fixed set of knowledge points before self-play training and keeps them unchanged throughout RL. This design provides a stable and controllable basis for knowledge-conditioned generation, which helps reduce redundant question generation and improve training efficiency. However, it also limits the framework's ability to adapt its knowledge-point coverage as the proposer and reasoner improve. Future work could explore online discovery, pruning, and refinement of knowledge points when learning from unlabeled medical corpora. Additionally, our experiments focus on text-grounded case reports and text-only benchmark settings, while many medical corpora contain images that provide important clinical evidence. Extending CasePlay to multimodal self-play is therefore an important direction for improving its practical utility in medical reasoning. Such an extension could allow the framework to generate and evaluate questions grounded in both textual and visual evidence, making self-play learning better aligned with real clinical case reasoning.

G Broader Impacts

This work aims to improve medical reasoning models by learning from unlabeled clinical case reports. Potential benefits include reducing reliance on costly expert annotations and supporting research on medical education and clinical question answering. However, the work also has risks. Models trained with this framework may produce incorrect or overconfident medical answers, and case reports may contain biases because they often describe unusual or publication-worthy cases rather than representative patient populations. There are also privacy concerns when using medical text, even if the data are publicly available, and stronger medical reasoning models could be misused to generate misleading health information. We view CasePlay as a research framework, not a deployable clinical decision-making system. Any downstream use should require de-identified and properly licensed data, careful factuality and bias evaluation, clear communication of limitations, and human clinical oversight.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details are provided in Appendix C and Appendix D

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is available at supplemental material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Details are provided in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For the main comparison, we run 3 seeds for each self-play method. And we sample eight responses per question for each evaluated checkpoint and report the average performance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computer resources are provided in Appendix D.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that the research conducted in this paper conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix G.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We construct training data from PMC Open Access Subset and apply safeguards for responsible release. We use only articles with appropriate access permissions, respect the original licenses, and do not release raw crawled content beyond what is permitted.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited and credited the existing assets used in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce new training data, code, and model checkpoints, and provide documentation for their usage, limitations, and licenses. Released assets are anonymized during submission where applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: This work trains LLMs for medical reasoning through self-play RL.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.