

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 OPENSIR: OPEN-ENDED SELF-IMPROVING REAS- SONER

Anonymous authors

Paper under double-blind review

ABSTRACT

Recent advances in large language model (LLM) reasoning through reinforcement learning rely on annotated datasets for verifiable rewards, which may limit models' ability to surpass human-level performance. While self-play offers a promising alternative, existing approaches depend on external verifiers or cannot learn open-endedly. We present **Open**-Ended **S**elf-**I**mproving **R**easoner (OpenSIR), a self-play framework where an LLM learns to generate and solve novel problems by alternating teacher and student roles without external supervision. To generate novel problems, OpenSIR optimises for both difficulty and diversity, rewarding problems that challenge appropriately while exploring distinct concepts, enabling open-ended mathematical discovery. Starting from a single trivial seed problem, OpenSIR substantially improves instruction models: Llama-3.2-3B-Instruct advances from 73.9 to 78.3 on GSM8K, and from 28.8 to 34.4 on College Math, while Gemma-2-2B-Instruct rises from 38.5 to 58.7 on GSM8K. Our analyses reveal that OpenSIR achieves open-ended learning through co-evolving teacher-student roles that adaptively calibrate difficulty and drive diverse exploration, progressing autonomously from basic to advanced mathematics.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) drives recent advances in LLM reasoning. Recent works on DeepSeek-R1 (DeepSeek-AI et al., 2025) and OpenAI o1 (OpenAI, 2024) have shown that large-scale reinforcement learning improves reasoning capabilities. Yet, these methods require extensive human-annotated data for reward signals, which bottleneck scalability and potentially limit performance to human-level (Hughes et al., 2024b).

One promising direction to address these fundamental limitations is to generate synthetic training data through self-play, which demonstrated remarkable success in various games (Silver et al., 2016; 2017; Brown & Sandholm, 2019; FAIR et al., 2022), allowing systems to exceed human-level performance by learning from unambiguous reward signals (Silver et al., 2017; FAIR et al., 2022). Yet, mathematical reasoning poses a key challenge for self-play: unlike games that have clear rules and winners, generated mathematics problems lack the ground-truth answers to provide feedback signals. Recent works utilise external verifiers, such as compilers for coding tasks (Pourcel et al., 2024; Zhao et al., 2025) or game rules (Liu et al., 2025), while R-Zero (Huang et al., 2025) employs majority voting with basic repetition penalties. However, these approaches cannot achieve open-ended learning, the ability to continuously generate and pursue novel challenges without external supervision (Bauer et al., 2023; Hughes et al., 2024a), confining systems to known concepts instead of exploring diverse mathematical domains.

We present **Open**-ended **S**elf-**I**mproving **R**easoner (OpenSIR), a method for training a policy π_θ to generate and solve novel problems without external supervision. OpenSIR uses *self-play* — a single policy π_θ alternates between teacher and student roles: the teacher generates problems, while the student solves them, with problem-solution pairs selected for reinforcement learning updates. We reward teachers for generating appropriately challenging problems for the students, using consistency and solution length across multiple solution attempts. OpenSIR achieves open-ended learning through embedding-based diversity rewards that drive continuous exploration of novel mathematical concepts.

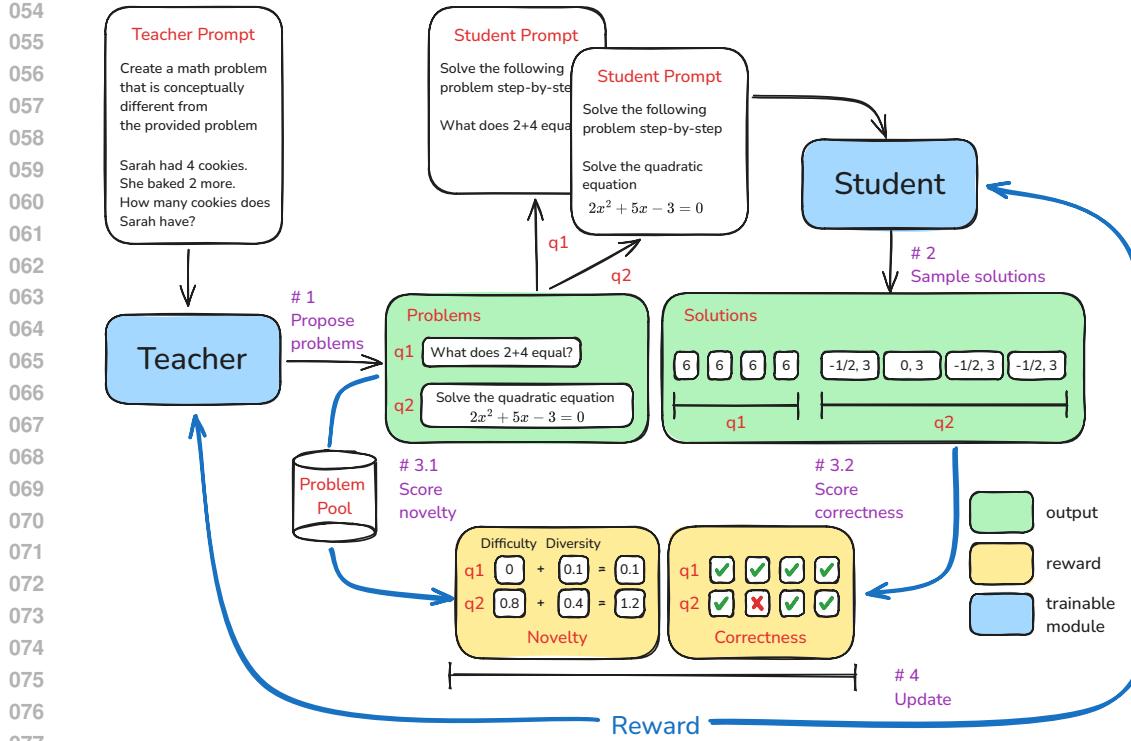


Figure 1: Overview of the OpenSIR framework. A single policy π_θ alternates between generating and solving novel problems without external supervision. Each training iteration consists of **problem generation, solution sampling, scoring, and model update**. Novelty is captured through both *difficulty* and *diversity*: problems must be challenging yet solvable, and they must explore new concepts. These dimensions together drive open-ended self-improvement in the LLM reasoning ability.

Our experiments show OpenSIR outperforms base instruction models and reinforcement learning baselines. Starting from a single trivial seed problem, OpenSIR improves base instruction models by up to 6.3 accuracy points, surpassing GRPO baselines trained on thousands of human-annotated examples. Specifically, Llama-3.2-3B-Instruct improves from $73.9 \rightarrow 78.3$ (+4.4) on GSM8K and $28.8 \rightarrow 34.4$ (+5.6) on College Math, while Gemma-2-2B-Instruct rises from $38.5 \rightarrow 58.7$ (+20.2) on GSM8K and $19.1 \rightarrow 23.4$ (+4.3) on College Math.

Our qualitative analysis reveals OpenSIR succeeds through adaptive difficulty calibration and diversity-driven exploration. Problem difficulty is automatically calibrated throughout training, while the range of topics expands from basic to advanced mathematics (§4.1). Generating harder problems risks invalidity, requiring a balance between challenge and correctness (§4.2). Diversity rewards incentives generate problems spanning varied mathematical concepts (§4.3). Teacher-student co-evolution proves essential: without teacher training, models cannot generate appropriate challenges or explore new topics (§4.4).

2 OPEN-ENDED SELF-IMPROVING REASONER

Figure 1 illustrates the Open-Ended Self-Improving Reasoner (OpenSIR), a self-play framework in which a policy π_θ learns to both generate and solve novel mathematical problems without external supervision. We use reinforcement learning to optimise two roles within one policy: the *teacher*, which creates new problems, and the *student*, which solves them. This open-ended approach enables the policy to bootstrap its learning and discover new and diverse challenges without annotated data. Each training iteration involves four phases:

1. **Problem generation** (§2.1): The teacher proposes new problems by conditioning on reference problems from an accumulated pool of previously generated problems;

108 2. **Solution sampling** (§2.2): The student attempts multiple solutions per problem, with majority
 109 voting determining the reference answer and solve rate measuring reliability;
 110
 111 3. **Scoring** (§2.3): We compute novelty scores for the teacher’s generated problems and correctness
 112 scores for the student’s solutions; and
 113
 114 4. **Model update** (§2.4): We update the policy’s parameters with role-specific rewards using the
 115 problem-solution pairs selected by the novelty scores.

116 Algorithm 1 summarizes the complete training procedure.

117 In OpenSIR, we define novelty along two dimensions that together drive continuous open-ended
 118 learning. First, problems must have an appropriate level of difficulty. It should be challenging
 119 enough to promote learning but solvable enough to provide reliable training signals. Second, prob-
 120 lems must explore diverse concepts, preventing the model from repeating learning on familiar con-
 121 cepts. This two-dimensional view of novelty ensures the model continuously expands both the depth
 122 and breadth of its mathematical reasoning abilities.

123 2.1 PROBLEM GENERATION

125 At each iteration t , the policy π_θ generates k groups of G problems each, denoted as $q_{1:G}$ within
 126 each group, for a total of $M = k \times G$ problems. To generate these problems, we sample k reference
 127 problems from a pool \mathcal{P}_{t-1} of accumulated problems from previous iterations, where each reference
 128 problem serves as a seed for generating G new problems. Each generated problem must explicitly
 129 include the mathematical concepts required for its solution. Problems with invalid formats are fil-
 130 tered out, and valid problems proceed to the solution-sampling phase. We initialise the problem pool
 131 \mathcal{P}_0 with a single trivial problem (“What is 1+1?”).

132 2.2 SOLUTION SAMPLING

135 Let a_j denote the parsed answer from solution attempt o_j . We select the most common answer
 136 across attempts as the reference answer a^* . We then compute the *solve rate* for each problem to
 137 determine the reliability of the answers. For brevity, we denote $s_{q_i} = \text{SolveRate}(q_i)$ when referring
 138 to the solve rate of problem q_i .

$$139 \text{SolveRate}(q_i) = \frac{\text{count}(a^*)}{G} \quad \text{where} \quad a^* = \arg \max_{a \in a_{1:G}} \text{count}(a), \quad (1)$$

142 In Eq. (1), $\text{count}(a)$ denotes the number of times answer a appears. The solve rate quantifies answer
 143 reliability. High solve rates indicate reliable reference answers due to solution convergence, while
 144 low solve rates suggest inconsistent solutions that may indicate flawed problem formulations.

145 2.3 SCORING

148 We evaluate the quality of generated problems and solutions with different scoring functions. The
 149 teacher’s problems are scored based on *difficulty* and *diversity*, while the student’s receive scores for
 150 *correctness*. Additionally, both roles incorporate format scores to ensure parseable outputs.

151 2.3.1 TEACHER SCORING

153 We capture novelty through two fundamental dimensions: difficulty and diversity. We measure dif-
 154 ficulty using *solvability* to ensure problems remain appropriately challenging and *solution length* to
 155 encourage multi-step reasoning, as these provide complementary signals about problem difficulty.
 156 Diversity is promoted through embedding distance, which encourages exploration of varied mathe-
 157 matical concepts. These components form a unified novelty score that guides problem generation.

158
 159 **Solvability (score_{sol}).** The solvability score identifies problems with appropriate challenge. We
 160 use solve rate as a proxy for solvability—problems with $s_{q_i} > s_{\max}$ are likely too easy, while those
 161 with $s_{q_i} < s_{\min}$ are either too difficult or malformed. We employ a triangular scoring function that
 peaks at the optimal solve rate and decreases linearly as problems become too easy or too hard.

162 We define the solve rate range as $[s_{\min}, s_{\max}]$. Easy problems ($s_{q_i} > s_{\max}$) fail to challenge the
 163 model, while problems that are too hard or malformed ($s_{q_i} < s_{\min}$) offer minimal training value.
 164

165 Formally, for $s_{q_i} \in [0, 1]$, let $s_{\text{mid}} = (s_{\min} + s_{\max})/2$ be the midpoint:

$$166 \quad \text{score}_{\text{sol}}(q_i) = \begin{cases} 1 - \alpha |s_{q_i} - s_{\text{mid}}| & \text{if } s_{q_i} \in [s_{\min}, s_{\max}], \\ 167 \quad 0 & \text{otherwise} \end{cases} \quad (2)$$

169 where $\alpha = (1 - 1/G) / (s_{\text{mid}} - s_{\min})$ is the slope coefficient, with G being the number of solution
 170 attempts. The score peaks at the midpoint s_{mid} and decreases to $1/n$ at the boundaries.

171 This creates a symmetric triangular score centred at the midpoint of the solve rate range, giving a
 172 maximum score for problems with moderate difficulty and progressively less score as the solve rate
 173 approaches either boundary.

175 **Solution Length (score_{len}).** Solution length complements solvability by measuring problem com-
 176 plexity. Problems requiring multi-step reasoning typically elicit longer solutions. We score problems
 177 using the average length of student solutions:

$$178 \quad \text{score}_{\text{len}}(q_i) = \min \left(\frac{\bar{l}(q_i)}{l_{\text{base}}}, \frac{l_{\text{cap}}}{l_{\text{base}}} \right) \quad (3)$$

181 where $\bar{l}(q_i)$ denotes average solution length for problem q_i , l_{base} is a normalisation factor (defaults to
 182 1000 tokens), and l_{cap} prevents outliers from dominating the scoring signal. This score complements
 183 the solvability score (see Appendix C.1).

184 **Diversity (score_{div}).** We compute the semantic distance between each new problem and the exist-
 185 ing problem pool:

$$186 \quad \text{score}_{\text{div}}(q_i) = \min_{q' \in \mathcal{P}_{t-1}} d(e_{q_i}, e_{q'}) \quad (4)$$

188 where e_{q_i} and $e_{q'}$ represent problem embeddings obtained from a pre-trained encoder, and $d(\cdot, \cdot)$
 189 denotes cosine distance. This score maximises when a problem is semantically distant from all
 190 existing problems in the pool.

192 **Format (score_{fom}^T).** The format score ensures proper problem structure. Generated problems must
 193 be enclosed in <question> tags with concepts listed in <concepts> tags (maximum three concepts).
 194 We assign $\text{score}_{\text{fom}}^T(q_i) = 1$ for correct formatting and $\text{score}_{\text{fom}}^T(q_i) = 0$ otherwise.

196 **Novelty Score.** We combine these components into a novelty score capturing both difficulty and
 197 diversity:

$$198 \quad \text{score}_{\text{novel}}(q_i) = \alpha \text{score}_{\text{sol}}(q_i) + \lambda \text{score}_{\text{len}}(q_i) + \gamma \text{score}_{\text{div}}(q_i) + \delta \text{score}_{\text{fom}}^T(q_i) \quad (5)$$

200 where $\alpha, \lambda, \gamma, \delta$ are hyperparameters that control the relative importance of each component. This
 201 novelty score is used to select high-quality problem-solution pairs for training.

202 2.3.2 STUDENT SCORING

204 The student’s score is based on solution correctness. For each solution attempt, we evaluate correct-
 205 ness by comparing the parsed answer against the reference answer from majority voting.

207 **Format (score_{fom}^S).** The format score ensures proper answer presentation. Solutions must present
 208 final answers in \boxed{} notation. We assign $\text{score}_{\text{fom}}^S(o_j) = 1$ for correct formatting and 0 other-
 209 wise.

211 **Correctness Score.** The student’s correctness score combines accuracy with the format score:

$$212 \quad \text{score}_{\text{correct}}(o_j, a_j) = \mathbf{1}[a_j = a^*] + \delta \text{score}_{\text{fom}}^S(o_j) \quad (6)$$

214 where $\mathbf{1}[a_j = a^*]$ is an indicator function that equals 1 when parsed answer a_j from outcome o_j
 215 matches the reference answer a^* , and 0 otherwise. This correctness score evaluates both solution
 accuracy and proper formatting.

216 **Algorithm 1** OpenSIR

217

218 **Require:** Problem pool \mathcal{P}_0 , policy $\pi_\theta^{(0)}$, embedding encoder ε , batch size B , generation group size G , solve
219 rate range $[s_{\min}, s_{\max}]$, teacher prompt p_T , student prompt p_S

220 1: **for** $t = 1$ to T **do** ▷ Problem Generation

221 2: Sample $k = B/G$ reference problems $\{p_1, \dots, p_k\}$ from \mathcal{P}_{t-1}

222 3: **for** $i = 1$ to k **do**

223 4: Sample $q_{i,1:G} \sim \pi_\theta^{(t)}(\cdot | p_i, p_T)$

224 5: **end for**

225 6: $\mathcal{Q}_{\text{valid}} \leftarrow \{q_{i,j} \mid q_{i,j} \text{ has valid format}\}$ ▷ Solution Sampling

226 7: **for** each $q_i \in \mathcal{Q}_{\text{valid}}$ **do**

227 8: Sample solutions $o_{i,1:G} \sim \pi_\theta^{(t)}(\cdot | q_i, p_S)$

228 9: Parse answers $a_{i,1:G}$ from solutions $o_{i,1:G}$

229 10: Compute reference answer $a_i^* = \arg \max_{a \in a_{i,1:G}} \text{count}(a)$ via majority voting

230 11: Compute solve rate $s_{q_i} = \text{count}(a_i^*)/G$

231 12: Compute embedding $e_{q_i} \leftarrow \varepsilon(q_i)$

232 13: **end for** ▷ Scoring

233 14: Compute $\text{score}_{\text{novel}}(q_i)$ for all $q_i \in \mathcal{Q}_{\text{valid}}$ via Eq. 5

234 15: $\mathcal{I}_T \leftarrow \text{top}_{B/(2G)}(i : \text{Var}(\text{score}_{\text{novel}}(q_{i,1:G})), i \in \{1, \dots, k\})$ ▷ Teacher sample selection

235 16: $\mathcal{Q}_S \leftarrow \text{top}_{B/(2G)}(q : \text{score}_{\text{novel}}(q), q \in \mathcal{Q}_{\text{valid}})$ ▷ Student sample selection

236 17: Compute $\text{score}_{\text{correct}}(o_{i,j}, a_{i,j})$ for solutions where $q_i \in \mathcal{Q}_S$ via Eq. 6

237 18: $\mathcal{D}_T \leftarrow \{(p_T, q_{i,j}, R_{i,j}^T) : i \in \mathcal{I}_T, 1 \leq j \leq G\}$ where $R_{i,j}^T = \text{score}_{\text{novel}}(q_{i,j})$ ▷ Model Update

238 19: $\mathcal{D}_S \leftarrow \{(p_S, o_{i,j}, R_{i,j}^S) : q_i \in \mathcal{Q}_S, 1 \leq j \leq G\}$ where $R_{i,j}^S = \text{score}_{\text{correct}}(o_{i,j}, a_{i,j})$

239 20: Update $\pi_\theta^{(t+1)} \leftarrow \text{GRPO}(\pi_\theta^{(t)}, \mathcal{D}_T \cup \mathcal{D}_S)$

240 21: $\mathcal{P}_t \leftarrow \mathcal{P}_{t-1} \cup \mathcal{Q}_{\text{valid}}$

241 22: **end for**

242 23: **return** $\pi_\theta^{(T)}$

241 2.4 MODEL UPDATE

243 After computing novelty scores, we select B high-quality samples from valid problems for reinforcement learning, allocating half to problem generation and half to solution solving. For teacher training, we choose problem groups with highest $\text{score}_{\text{novel}}$ variance to ensure diverse training signals. For student training, we select problems with the highest novelty scores to provide maximal training value.

244 We optimise the policy using π_θ with an objective similar to Group Relative Policy Optimization
245 (GRPO) (Shao et al., 2024), adapted for on-policy training to ensure stability (Chen et al., 2025):

246

$$\mathcal{J}(\theta) = \mathbb{E}_{\substack{q_{1:G} \sim \pi_\theta(\cdot | p_T) \\ o_{1:G} \sim \pi_\theta(\cdot | q_i, p_S)}} \left[\sum_{r \in \{T, S\}} \frac{1}{G} \sum_{i=1}^G A_i^r \right] - \beta \mathbb{D}_{KL}(\pi_\theta \| \pi_{\text{ref}}) \quad (7)$$

247 where p_T and p_S are the teacher and student prompts respectively, $r \in \{T, S\}$ refers to teacher
248 and student, \mathbb{D}_{KL} denotes the KL divergence, π_{ref} refers to the initial model before training. The
249 advantage for each role $r \in \{T, S\}$ is computed as:

250

$$A_i^r = \frac{R_i^r - \text{mean}(R_{1:G}^r)}{\text{std}(R_{1:G}^r)}. \quad (8)$$

251 We define role-specific rewards R_i^T and R_j^S using the scoring functions from Section 2.3:

252

$$R_i^T = \text{score}_{\text{novel}}(q_i), \quad R_j^S = \text{score}_{\text{correct}}(o_j, a_j) \quad (9)$$

253 All valid problems are then added to the problem pool \mathcal{P}_t for future iterations.

254 3 EXPERIMENTS

255 3.1 TRAINING SETUP

256 We experiment with four instruction-tuned models: Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct
257 (Dubey et al., 2024), Gemma-2-2B-Instruct (Team et al., 2024), and Qwen-2.5-3B-Instruct (Team,

270 2024) with GRPO (Shao et al., 2024). We use a learning rate of 3×10^{-7} and 10 warm-up steps.
 271 The KL divergence coefficient is set to 10^{-4} and the batch size is 256. To compare models trained
 272 on the same number of problem-solution pairs, we train the GRPO baselines with 100 steps, and
 273 OpenSIR for 200 steps since OpenSIR allocates half of its training budget to problem generation.
 274 Clipping is not applied since we strictly use on-policy samples. Each experiment is run with three
 275 random seeds. We provide full training details in Appendix D.1.

277 3.2 DATASET AND EVALUATION SETUP

279 We evaluate method on five mathematical benchmarks: GSM8K (Cobbe et al., 2021), MATH-500
 280 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and
 281 College Math (Tang et al., 2024).

282 We use sampling temperature 0.6 and top-p 0.95. The maximum response length is set to 4,096
 283 tokens. We report the average performance over 16 generations (avg@16). Answer extraction and
 284 comparison are performed using the `math_verify` library.

286 3.3 BASELINES

288 (1) **Base** We evaluate the instruction-tuned models using zero-shot prompting, where models generate
 289 step-by-step reasoning and provide final answers without additional training.

290 (2) **GRPO** We train the instruction models with GRPO (Shao et al., 2024) on established mathematical
 291 datasets. We train two variants: **GRPO_{math}** on the MATH dataset (7,500 training examples)
 292 (Hendrycks et al., 2021) and **GRPO_{gsm8k}** on the GSM8K dataset (7,473 training examples) (Cobbe
 293 et al., 2021).

294 (3) **Absolute Zero** (Zhao et al., 2025) A self-play framework for code generation that uses Python
 295 as external verifier, rewarding problems with minimal solve rates.

296 (4) **R-Zero** (Huang et al., 2025) A verifier-free self-play framework that trains separate challenger
 297 and solver models using rewards based on repetition penalties and solve rates near 0.5. OpenSIR
 298 differs by explicitly optimising for diversity and incorporating solution length to capture multiple
 299 dimensions of difficulty within a single model.

301 3.4 MAIN RESULTS

303 Table 1 demonstrates that OpenSIR achieves substantial gains over the base instruction models
 304 across different model scales and families. OpenSIR improves Llama-3.2-3B-Instruct by 3.6 points,
 305 Llama-3.1-8B-Instruct by 3.1, and Gemma-2-2B-Instruct by 6.3 points on average accuracy. One
 306 exception is Qwen-2.5-3B-Instruct (+0.6), where all methods show limited gains. The limited
 307 improvement aligns with observations of potential benchmark contamination (Wu et al., 2025).

308 OpenSIR outperforms all GRPO baselines without using human-annotated training data. GRPO
 309 baselines require over 7,000 labeled examples, yet OpenSIR generates its own training problems
 310 through self-play, starting from a single trivial seed problem. OpenSIR also substantially out-
 311 performs other self-play methods by 1.75 to 3.38 points on Llama-3.2-3B-Instruct, Gemma-2-2B-
 312 Instruct, and Llama-3.1-8B-Instruct. Although Absolute Zero and R-Zero demonstrate significant
 313 improvements over non-instruction-tuned models in their original work (Zhao et al., 2025; Huang
 314 et al., 2025), both show limited gains on instruction-tuned models. This challenging scenario is pre-
 315 cisely where self-play methods are intended to excel - when models have already consumed available
 316 human-annotated data. As we reveal later in our analysis, the success of OpenSIR can be attributed
 317 to its ability to explore diverse mathematical concepts, and calibrating difficulty adaptively to main-
 318 tain optimal challenge levels (§4.1). These capabilities enable OpenSIR to self-improve and expand
 319 its skills without external training data, achieving open-ended learning.

320 4 ABLATIONS AND ANALYSES

322 We perform a series of ablation studies and qualitative analyses on Llama-3.2-3B-Instruct to dissect
 323 the contribution of each key component in the OpenSIR framework. Our analysis investigates:

Model	GSM8K	MATH-500	Minerva	College Math	OlympiadBench	Avg.
Llama-3.2-3B-Instruct						
Base	73.94	42.86	15.21	28.78	13.09	34.78
GRPO _{gsm8k}	79.72	45.30	16.27	33.33	14.56	37.83 ^{+3.05}
GRPO _{math}	76.48	45.26	16.09	32.95	14.13	36.98 ^{+2.20}
Absolute Zero	74.37	44.71	14.78	31.93	14.42	36.04 ^{+1.26}
R-Zero	76.34	44.27	15.84	32.72	14.19	36.67 ^{+1.89}
OpenSIR	78.28	46.22	17.46	34.42	15.72	38.42 ^{+3.64}
Gemma-2-2B-Instruct						
Base	38.50	16.51	10.09	19.11	3.00	17.44
GRPO _{gsm8k}	58.75	19.15	7.75	20.45	3.21	21.86 ^{+4.42}
GRPO _{math}	56.03	22.76	7.96	16.31	3.24	21.26 ^{+3.82}
Absolute Zero	57.13	15.92	8.29	17.36	3.18	20.38 ^{+2.94}
R-Zero	56.37	17.31	8.49	19.86	3.12	21.03 ^{+3.59}
OpenSIR	58.03	24.75	9.51	23.36	3.15	23.76 ^{+6.32}
Qwen-2.5-3B-Instruct						
Base	84.43	65.36	25.23	48.22	27.94	50.24
GRPO _{gsm8k}	84.94	65.77	25.31	48.46	28.31	50.56 ^{+0.32}
GRPO _{math}	84.31	65.89	24.98	48.34	28.26	50.36 ^{+0.12}
Absolute Zero	84.62	65.33	25.21	48.31	28.12	50.32 ^{+0.08}
R-Zero	84.22	64.93	24.81	48.45	27.82	50.05 ^{-0.19}
OpenSIR	85.38	65.87	25.96	48.74	28.33	50.85 ^{+0.61}
Llama-3.1-8B-Instruct						
Base	84.50	47.89	22.75	34.10	16.26	41.10
GRPO _{gsm8k}	88.70	50.37	24.83	35.03	16.43	43.05 ^{+1.95}
GRPO _{math}	86.23	50.82	23.98	34.93	16.54	42.50 ^{+1.40}
Absolute Zero	86.89	51.38	23.21	34.39	15.96	42.37 ^{+1.27}
R-Zero	86.19	50.93	24.11	32.93	15.66	41.96 ^{+0.86}
OpenSIR	87.30	52.38	27.29	36.29	17.81	44.21 ^{+3.11}

Table 1: The avg@16 performance on five mathematical benchmarks. OpenSIR outperforms GRPO baselines trained on >7,000 human-annotated examples and other self-play methods (Absolute Zero, R-Zero) across model families, starting from a single trivial seed problem.

(1) the evolution of problem difficulty and diversity over training (§4.1), (2) the effect of solve rate thresholds on the difficulty-validity trade-off (§4.2), (3) the impact of diversity rewards on promoting exploration of novel problem types (§4.3), and (4) the necessity of dual-role training (§4.4).

4.1 EVOLUTION OF PROBLEM DIFFICULTY AND DIVERSITY

We track how difficulty and diversity evolve during training through human evaluation. We sample 20 problems from three OpenSIR training checkpoints (steps 0, 100, 200) and 20 each from GSM8K and MATH. Annotators evaluate mixed sets of five problems (one per source), identifying topics, assessing validity, and ranking difficulty. Figure 2 shows average difficulty rankings (1=easiest, 5=highest); see Appendix B for full annotation instructions.

Figure 2 (left) reveals a V-shaped difficulty trend across training stages. Problems start at 3.4 difficulty, drop to 3.0 at midpoint, then rise to 3.8. This pattern reflects OpenSIR’s self-calibration: the model first generates overly difficult problems, then learns appropriate difficulty, and finally increases challenge as its solving capabilities improve. The model also generates increasingly valid problems during training — validity improves from below 50% initially to 95% (19 of 20 problems) by the end.

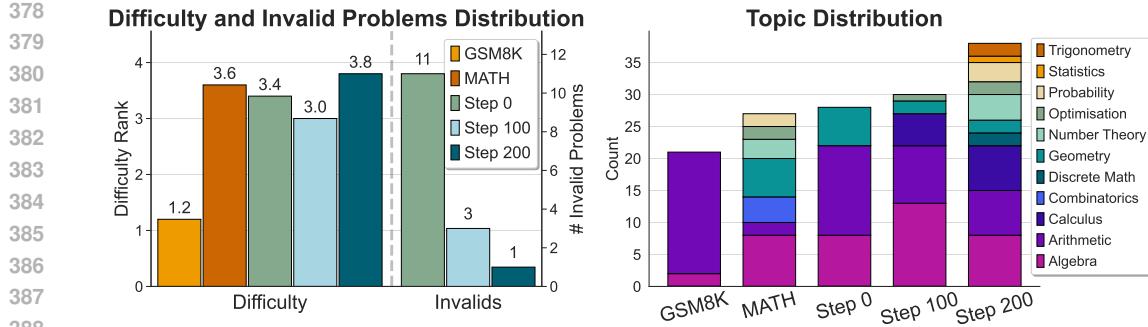


Figure 2: Evolution of problem difficulty, validity, and topic diversity during OpenSIR training. **(Left)** Human evaluation results showing difficulty rankings (1-5 scale where 1=easiest, 5=highest) and number of invalid problems for GSM8K, MATH, and problems generated at steps 0, 100, and 200 of training. Invalid problems are those with logical flaws, missing information, or ambiguities. **(Right)** Distribution of mathematical topics across training stages, demonstrating the increasing diversity of generated problems from step 0 to step 200.

Figure 2 (right) shows topic diversity expansion across training. OpenSIR progresses from basic topics (algebra, arithmetic, geometry) to advanced domains including calculus and optimisation, eventually incorporating trigonometry, statistics, and other mathematical areas. This progression demonstrates OpenSIR’s capacity for autonomous exploration of diverse mathematical concepts. Appendix A.2 provides detailed case studies that illustrate this evolution.

4.2 DIFFICULTY-VALIDITY TRADE-OFF

Model	Acc	Validity	Solve Rate
OpenSIR _{0.5}	38.42	70.82	89.82
OpenSIR _{0.3}	36.81	52.32	81.38
OpenSIR _{0.1}	35.97	42.31	78.31

Table 2: Performance, problem validity, and solve rate across different lower solve-rate thresholds, with the upper threshold fixed at 0.9 for all variants. Validity and solve rate are estimated using GPT-5. Lower thresholds produce harder problems but significantly more invalid ones, ultimately reducing overall performance.

We investigate the difficulty-validity trade-off by training OpenSIR variants with lower solve-rate thresholds of 0.1, 0.3, and 0.5, keeping the upper threshold at 0.9. From each variant, we sample 300 problems and assess quality with GPT-5 (OpenAI, 2025a) using 8 responses per problem. We measure validity by comparing GPT-5’s majority answer to our reference answer and difficulty by GPT-5’s solve rate.

Table 2 reveals a clear trade-off between validity and difficulty. While lowering the threshold from 0.5 to 0.1 produces moderately harder problems (GPT-5 solve rate decreases from 89.82% to 78.31%), validity plummets from 70.82% to 42.31%. This suggests that problems with very low solve rates frequently contain errors rather than representing genuine mathematical challenges. Performance consistently drops with lower thresholds, supporting our selection of 0.5 as the lower threshold for the solvability reward.

Besides solve-rate thresholds, we find that rewarding longer solutions provides another mechanism for promoting problem complexity that encourage sophisticated multi-step problems (Appendix C.1).

4.3 IMPACT OF DIVERSITY REWARDS

We analyse the impact of the diversity reward on problem diversity through problem embeddings, n-gram similarity, and concept overlap. Figure 3 visualises the problem embeddings with t-SNE, where red points represent problems without diversity reward, cyan points show problems with diversity reward, gold indicates MATH dataset problems, and purple marks GSM8K dataset problems. Without diversity rewards, problems cluster in narrow regions, generating similar types repeatedly and failing to achieve open-ended exploration. With diversity rewards, problems spread across the embedding space, reaching areas beyond MATH and GSM8K training sets. Further analysis of n-gram similarity and concept overlap support these findings, demonstrating consistent patterns of greater dispersion and novelty (Appendix A.3).

Table 3 empirically confirms the importance of diversity rewards, showing that removing diversity rewards reduces average performance by 1.97 (from 38.42 to 36.45). It also shows that the number of unique concepts has dropped significantly (from 5914 to 3328). This demonstrates that without diversity rewards, the model generates repetitive problems with limited learning value, constraining the teacher’s ability to present varied mathematical challenges to the student. Incorporating diversity rewards thus enables exploration of novel problems beyond existing datasets, supporting open-ended learning where the model continuously discovers new challenges rather than repeating known concepts. Notably, this improvement is robust to the choice of diversity metric (Appendix C.2), with different measurement approaches yielding comparable results.

4.4 IMPORTANCE OF DUAL-ROLE TRAINING

Trained Roles	Acc	Avg. Solve Rate
Both	38.42	72.20 (± 4.49)
Student	35.89	64.56 (± 17.37)

Table 4: Accuracy and average solve rate with standard deviation (\pm) for OpenSIR with teacher training (Both) versus without teacher training (Student only). Joint training achieves higher accuracy and remarkably stable problem difficulty (much lower solve rate variance), demonstrating that teacher training enables calibrated problem generation at optimal difficulty levels for effective learning.

We evaluate the contribution of the joint teacher-student training by testing a variant where only the student is updated while the teacher remains fixed at its initial state. Table 4 shows that accuracy drops significantly from 38.42 to 35.89 when only the student is trained. This demonstrates that effective self-play requires both components to co-evolve.

Without teacher training, generated problems become harder (solve rate drops from 72.20 to 64.56) and drift from the optimal 70% target solve rate established in Section 4.2. More critically, solve rate variance increases tremendously (from ± 4.49 to ± 17.37), indicating highly inconsistent diffi-

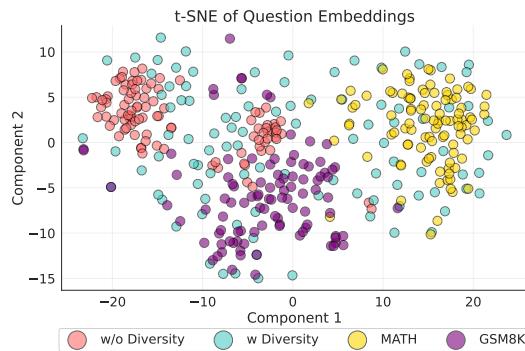


Figure 3: t-SNE visualization of problem embeddings showing the effect of diversity reward on problem distribution. With diversity reward, problems explore broader regions of the embedding space compared to the clustered distribution without diversity reward.

Model	Acc	# Concepts
w diversity	38.42	5914
w/o diversity	36.45	3328

Table 3: OpenSIR performance with and without diversity reward. Exploring diverse mathematical concepts through the diversity reward improves both accuracy and concept coverage, showing that variety in problem types is crucial for self-improvement.

486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809
 810
 811
 812
 813
 814
 815
 816
 817
 818
 819
 820
 821
 822
 823
 824
 825
 826
 827
 828
 829
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 840
 841
 842
 843
 844
 845
 846
 847
 848
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 880
 881
 882
 883
 884
 885
 886
 887
 888
 889
 890
 891
 892
 893
 894
 895
 896
 897
 898
 899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 910
 911
 912
 913
 914
 915
 916
 917
 918
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1870
 1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878
 1879
 1880
 1881
 1882
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1890
 1891
 1892
 1893
 1894
 1895
 1896
 1897
 1898
 1899
 1900
 1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 1909
 1910
 1911
 1912
 1913
 1914
 1915
 1916
 1917
 1918
 1919
 1920
 1921
 1922
 1923
 1924
 1925
 1926
 1927
 1928
 1929
 1930
 1931
 1932
 1933
 1934
 1935
 1936
 1937
 1938
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1950
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1960
 1961
 1962
 1963
 1964
 196

540 Nemotron-CrossThink: Scaling Self-Learning beyond Math Reasoning, April 2025.
 541

542 Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor
 543 Mordatch. Emergent Tool Use From Multi-Agent Autocurricula. In *International Conference*
 544 *on Learning Representations*, September 2019. URL <https://openreview.net/forum?id=SkxpxJBKws>.
 545

546 Jakob Bauer, Kate Baumli, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg,
 547 Michael Chang, Natalie Clay, Adrian Collister, Vibhavari Dasagi, Lucy Gonzalez, Karol Gre-
 548 gor, Edward Hughes, Sheleem Kashem, Maria Loks-Thompson, Hannah Openshaw, Jack Parker-
 549 Holder, Shreya Pathak, Nicolas Perez-Nieves, Nemanja Rakicevic, Tim Rocktäschel, Yannick
 550 Schroecker, Satinder Singh, Jakub Sygnowski, Karl Tuyls, Sarah York, Alexander Zacherl, and
 551 Lei M. Zhang. Human-Timescale Adaptation in an Open-Ended Task Space. In *Proceedings of*
 552 *the 40th International Conference on Machine Learning*, pp. 1887–1935. PMLR, July 2023. URL
 553 <https://proceedings.mlr.press/v202/bauer23a.html>.
 554

555 Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):
 556 885–890, August 2019. doi: 10.1126/science.aay2400.
 557

558 Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling Synthetic Data Creation
 559 with 1,000,000,000 Personas, June 2024. URL <http://arxiv.org/abs/2406.20094>.
 560

561 Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catan-
 562 zaro, and Wei Ping. AceReason-Nemotron: Advancing Math and Code Reasoning through Rein-
 563 forcement Learning, May 2025.

564 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
 565 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
 566 Schulman. Training Verifiers to Solve Math Word Problems, November 2021.
 567

568 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,
 569 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,
 570 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao
 571 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,
 572 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,
 573 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,
 574 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang
 575 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai
 576 Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang,
 577 Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang,
 578 Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang,
 579 Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang,
 580 R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng
 581 Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing
 582 Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen
 583 Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong
 584 Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu,
 585 Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xi-
 586 aosh Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia
 587 Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng
 588 Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong
 589 Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,
 590 Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou,
 591 Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying
 592 Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda
 593 Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu,
 Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu
 Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Rein-
 594 forcement Learning, January 2025.

594 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
 595 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
 596 *arXiv e-prints*, pp. arXiv–2407, 2024.

597

598 FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried,
 599 Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath,
 600 Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala,
 601 Stephen Roller, Dirk Rowe, Weiyuan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang,
 602 and Markus Zijlstra. Human-level play in the game of Diplomacy by combining language models
 603 with strategic reasoning. *Science*, 0(0):eade9097, November 2022. doi: 10.1126/science.ade9097.

604

605 Alex Havrilla, Andrew Dai, Laura O’Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, Fab-
 606 rizio Milo, Sharath Chandra Raparthy, Kanishk Gandhi, Baber Abbasi, et al. Surveying the effects
 607 of quality, diversity, and complexity in synthetic data from large language models. *arXiv preprint*
 608 *arXiv:2412.02980*, 2024.

609

610 Alex Havrilla, Edward Hughes, Mikayel Samvelyan, and Jacob Abernethy. Synthetic Problem Gen-
 611 eration for Reasoning via Quality-Diversity Algorithms, June 2025.

612

613 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han,
 614 Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench:
 615 A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Sci-
 616 entific Problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the*
 617 *62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,
 618 pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
 619 doi: 10.18653/v1/2024.acl-long.211.

620

621 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric
 622 Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solv-
 623 ing With the MATH Dataset. *Proceedings of the Neural Information Processing*
 624 *Systems Track on Datasets and Benchmarks*, 1, December 2021. URL <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html>.

625

626 Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin
 627 Huang, Haitao Mi, and Dong Yu. R-Zero: Self-Evolving Reasoning LLM from Zero Data, August
 628 2025.

629

630 Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge
 631 Shi, Tom Schaul, and Tim Rocktaschel. Open-Endedness is Essential for Artificial Superhuman
 632 Intelligence, June 2024a.

633

634 Edward Hughes, Michael D. Dennis, Jack Parker-Holder, Feryal M. P. Behbahani, Aditi Mavalankar,
 635 Yuge Shi, Tom Schaul, and Tim Rocktaschel. Position: Open-endedness is essential for artificial
 636 superhuman intelligence. In *ICML*. OpenReview.net, 2024b.

637

638 Jaehun Jung, Seungju Han, Ximing Lu, Skyler Hallinan, David Acuna, Shrimai Prabhumoye,
 639 Mostafa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. Prismatic synthe-
 640 sis: Gradient-based data diversification boosts generalization in Ilm reasoning. *arXiv preprint*
 641 *arXiv:2505.20161*, 2025.

642

643 Woosung Koh, Wonbeen Oh, Jaein Jang, MinHyung Lee, Hyeongjin Kim, Ah Yeon Kim, Joonkee
 644 Kim, Junghyun Lee, Taehyeon Kim, and Se-Young Yun. AdaSTAR: Adaptive data sampling for
 645 training self-taught reasoners. In *The Thirty-ninth Annual Conference on Neural Information*
 646 *Processing Systems*, 2025.

647

648 Joshua Ong Jun Leang, Giwon Hong, Wenda Li, and Shay B. Cohen. Theorem Prover as a Judge
 649 for Synthetic Data Generation, February 2025.

650

651 Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ra-
 652 masesh, Ambrose Sloane, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam
 653 Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving Quantitative Reasoning Problems with

648 Language Models. *Advances in Neural Information Processing Systems*, 35:3843–3857, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

649

650

651 Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

652

653

654

655

656 Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weian Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. SPIRAL: Self-Play on Zero-Sum Games Incentivizes Reasoning via Multi-Agent Multi-Turn Reinforcement Learning, June 2025.

657

658

659 Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, September 2018. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.

660

661

662 Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. #InsTag: Instruction Tagging for Analyzing Supervised Fine-tuning of Large Language Models. In *The Twelfth International Conference on Learning Representations*, October 2023. URL <https://openreview.net/forum?id=pszewhybU9>.

663

664

665

666

667 Zimu Lu, Aojun Zhou, Houxing Ren, Ke Wang, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. MathGenie: Generating Synthetic Data with Question Back-translation for Enhancing Mathematical Reasoning of LLMs, February 2024.

668

669

670 OpenAI. Learning to reason with LLMs, 2024. URL <https://openai.com/index/learning-to-reason-with-llms/>.

671

672 OpenAI. GPT-5 System Card, August 2025a. URL <https://openai.com/index/gpt-5-system-card/>.

673

674

675 OpenAI. Introducing OpenAI o3 and o4-mini, 2025b. URL <https://openai.com/index/introducing-o3-and-o4-mini/>.

676

677 OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with Large Scale Deep Reinforcement Learning, December 2019.

678

679

680

681

682

683 Julien Pourcel, Cédric Colas, Gaia Molinaro, Pierre-Yves Oudeyer, and Laetitia Teodorescu. ACES: Generating a Diversity of Challenging Programming Puzzles with Autotelic Generative Models. *Advances in Neural Information Processing Systems*, 37:67627–67662, December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7d0c6ff18f16797b92e77d7cc95b3c53-Abstract-Conference.html.

684

685

686

687

688 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, April 2024.

689

690

691

692 David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature16961.

693

694

695

696

697

698 David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017. ISSN 1476-4687. doi: 10.1038/nature24270.

702 Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan
 703 Zhang. Improving Data Efficiency for LLM Reinforcement Fine-tuning Through Difficulty-
 704 targeted Online Data Selection and Rollout Replay, June 2025.

705 Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. MathScale: Scaling Instruction
 706 Tuning for Mathematical Reasoning. In *Proceedings of the 41st International Conference on
 707 Machine Learning*, pp. 47885–47900. PMLR, July 2024. URL <https://proceedings.mlr.press/v235/tang24k.html>.

708 710 Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma
 711 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

712 713 Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

714 715 Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor
 716 Gitman. OpenMathInstruct-2: Accelerating AI for Math with Massive Open-Source Instruction
 717 Data, October 2024.

718 719 Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan
 720 Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou,
 721 Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David
 722 Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff,
 723 Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom
 724 Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver.
 725 Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):
 726 350–354, November 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1724-z.

727 728 Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan
 729 Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement
 730 learning. <https://github.com/huggingface/trl>, 2020.

731 732 Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao
 733 Zhou, Huijie Lv, Ming Zhang, Yanwei Fu, Qin Liu, Songyang Zhang, and Qi Zhang. Reasoning
 734 or Memorization? Unreliable Results of Reinforcement Learning Due to Data Contamination,
 735 August 2025.

736 737 Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo
 738 Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap Your Own Mathematical Questions for
 739 Large Language Models. In *The Twelfth International Conference on Learning Representations*,
 740 October 2023. URL <https://openreview.net/forum?id=N8N0hgNDrt>.

741 742 Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao
 743 Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond
 744 the Base Model?, April 2025.

745 746 Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping Reasoning With
 747 Reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, December
 748 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.

749 750 Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. SimpleRL-
 751 Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild,
 752 March 2025.

753 754 Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang,
 755 Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute Zero: Reinforced Self-play Reasoning
 756 with Zero Data, May 2025.

757 758 Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and
 759 Beidi Chen. Act Only When It Pays: Efficient Reinforcement Learning for LLM Reasoning via
 760 Selective Rollouts, June 2025.

756 A EXTENDED RESULTS AND ANALYSIS
757758 A.1 FULL RESULTS
759760 We provide the full results of all seeds in Table 5 and 6.
761762 A.2 CASE STUDY
763764 This section provides further analysis of question-solution pairs during training.
765766 As discussed in Section 4.1, the model generates predominantly invalid problems early in training.
767 Majority of these problems, primarily involve simple mathematical concepts like arithmetic, fail due
768 to missing information (Figures 4 and 5). When attempting complex topics like optimisation, which
769 are rare in the beginning, the model produces problems with missing information and fundamental
770 formulation errors (Figure 6). This reveal the model has limited understanding of underlying math-
771 ematical concepts. Invalid problems tend to exhibit low solve rates (≤ 0.25) and correspondingly
772 receive lower rewards, helping the model learn to generate valid problems. Consequently, invalid
773 problems decrease rapidly across training (§4.1).774 However, not all problems with low solve rates are invalid (§4.2). We find that some problems
775 involving certain topics that are challenging for the model, such as geometric series, persistently
776 exhibit low solve rates (Figures 8. The model struggles with exponentiation calculations, resulting
777 in poor performance on geometric series problems. This reveals a fundamental trade-off in OpenSIR:
778 while higher solve rate thresholds effectively filter out invalid problems, they inevitably discourage
779 exploration of genuinely difficult topics. Since these problems have low solvability scores, they are
780 likely to not receive sufficient encouragement to further explore these topics.781 In later training stages, we observe OpenSIR gradually expanding into advanced mathematical do-
782 mains. After 100 training steps, the model starts to generate problems involving concepts like optimi-
783 sation (Figure 9), calculus (Figure 10), trigonometry-based physics (Figure 11), probability (Fig-
784 ures 12), among others. While these advanced problems yield lower solve rates, which indicate the
785 model has a limited understanding of these domains, they achieve high novelty scores with large se-
786 mantic distances and longer solutions. This progression validates how novelty rewards in OpenSIR
787 drive exploration of diverse mathematical concepts, enabling open-ended learning.
788789 A concert venue charges an admission price per seat and also offers a VIP ticket package that includes
790 food, drinks, and other perks. If the food and other perks are included in the VIP ticket price and are
791 worth \$10 per person, and a group of friends want to buy the same number of VIP tickets as they would
792 with regular tickets, what is the minimum admission price per regular ticket to make the total cost of
793 the VIP tickets equal to or less than the total cost of the regular tickets?
794795 Figure 4: An invalid arithmetic question generated in step 0 with solve rate of 0.25. This question
796 is invalid since the VIP tick price is not provided, and therefore, it's impossible to calculate the
797 minimum regular ticket price.
798801 Find the percentage difference in the cumulative growth of two continuously compounded interest
802 functions after 5 years: $A = P * e^{(rt)}$, where A is the amount of money accumulated after n years,
803 including interest, P is the principal amount, r is the annual interest rate, and t is the time the money is
804 invested for.
805806 Figure 5: An invalid arithmetic question generated in step 0 with solve rate of 0.125. This question
807 is invalid since the two interest rates and principal amounts are not provided. Hence, it's impossible
808 to calculate the percentage difference with just the general formula provided.
809

810
811
812
813
814
815

Consider two positive integers m and n ($m \geq n$). Given a 2×2 matrix of numbers where each element is a non-negative integer, find the maximum value of the following expression: $ax^2 + by^2 + cxy$, where a , b , and c are constants, subject to the constraint that the sum of any two elements in each row and column of the matrix are equal. What is the maximum possible value of $ax^2 + by^2 + cxy$?

816
817
818
819
820
821
822

Figure 6: An invalid optimisation question generated in step 0 with solve rate of 0.125. This question is invalid because there are missing information about the constants a , b , and c . There are also ambiguities in the question, such as the role of m and n in the problem. It also did not explain what the elements of the matrix represent. Lastly, it contains problem formulation errors, specifically failing to specify constraints that ensure bounded solutions, demonstrating insufficient understanding of optimization problem structure.

823
824
825
826
827
828

Consider two positive integers m and n ($m \geq n$). Given a 2×2 matrix of numbers where each element is a non-negative integer, find the maximum value of the following expression: $ax^2 + by^2 + cxy$, where a , b , and c are constants, subject to the constraint that the sum of any two elements in each row and column of the matrix are equal. What is the maximum possible value of $ax^2 + by^2 + cxy$?

829
830
831
832
833
834
835

Figure 7: An invalid optimisation question generated in step 0 with solve rate of 0.125. This question is invalid because there are missing information about the constants a , b , and c . There are also ambiguities in the question, such as the role of m and n in the problem. It also did not explain what the elements of the matrix represent. Lastly, it contains problem formulation errors, specifically failing to specify constraints that ensure bounded solutions, demonstrating insufficient understanding of optimization problem structure.

836
837

A.3 FURTHER ANALYSIS ON QUESTIONS DIVERSITY

838
839
840
841
842
843
844

Figure 13 presents n-gram similarity and concept analysis. We compute ROUGE-L scores between problem texts and extract mathematical concepts using GPT-5 from problems at steps 0, 100, and 200, as well as from the MATH and GSM8K training sets. With diversity rewards (top row), problems maintain low ROUGE-L scores and minimal concept overlap both across training stages and with MATH/GSM8K. Without diversity rewards (bottom row), both textual similarity and concept overlap increase, confirming limited exploration of new problem types.

845
846

A.4 FURTHER ANALYSIS ON QUESTION DIFFICULTY PROGRESSION

847
848
849
850
851
852
853
854
855
856

Figure 14 compares solve rates between the evolving OpenSIR policy and the fixed instruction model (Base) on problems generated during training. While OpenSIR’s solve rate remains stable around 0.7 due to solvability-based problem selection (Section 2.3.1), this constant rate does not imply constant problem difficulty. As OpenSIR improves during training, maintaining the same solve rate requires generating progressively harder problems. To verify this difficulty progression objectively, we measure how the initial instruction model (Base) performs on the same problems. The base model’s solve rate first rises (0.48 → 0.58 at step 80) then declines (0.58 → 0.32 at step 200), confirming the V-shaped difficulty pattern trend from Section 4.1: problems initially become easier as OpenSIR learns appropriate calibration, then progressively harder as it increases challenge. Crucially, this pattern shows that OpenSIR’s reasoning ability improved over training.

857
858

A.5 SENSITIVITY TO THE INITIAL SEED PROBLEM

859
860
861
862
863

To address whether OpenSIR can robustly escape the limited starting point of a trivial arithmetic seed (“What is $1+1$?”), we experiment with two substantially different initial seeds using Llama-3.2-3B-Instruct: a geometry problem from the MATH dataset, representing a different mathematical domain, and a competition-level problem from AIME 2024, which is significantly more challenging than the trivial seed.

864

865

866

867

868

869

870

871

872

873

Question

A company is adding new employees to its workforce. Initially, the company has 500 employees and adds 10 new employees every quarter. If this trend continues for the next 5 years, and each quarter the company adds a fixed percentage increase in the number of new employees (starting with 10%), how many new employees can the company be expected to add during the first 2 years?

871

872

873

Solution (Incorrect)

874

875

To find the total number of new employees added in the first 2 years, we need to calculate the number of new employees added each quarter for the first 2 years and then sum them up.

876

877

878

First, let's calculate the number of quarters in 2 years: $2 \text{ years} * 4 \text{ quarters/year} = 8 \text{ quarters}$.

879

880

The company starts with 10 new employees and adds 10% more each quarter. To find the total number of new employees added each quarter, we can use the formula for compound interest:

881

882

$$A = P * (1 + r)^n$$

883

where:

884

A = total number of new employees after n quarters

885

P = initial number of new employees (10)

886

r = percentage increase per quarter (10% or 0.1)

887

n = number of quarters

888

889

However, since we want to find the total number of new employees added each quarter, we need to find the difference between A and the initial number of new employees (10). We can rewrite the formula as:

890

891

$$A - P = P * (1 + r)^n$$

892

$$A - 10 = 10 * (1 + 0.1)^n$$

893

$$A = 10 * (1.1)^n$$

894

Now, we can calculate A for each quarter:

895

$$\text{Quarter 1 (n = 1): } A = 10 * (1.1)^1 = 11$$

896

$$\text{Quarter 2 (n = 2): } A = 10 * (1.1)^2 = 12.1$$

897

$$\text{Quarter 3 (n = 3): } A = 10 * (1.1)^3 = 13.21$$

898

$$\text{Quarter 4 (n = 4): } A = 10 * (1.1)^4 = 14.41$$

899

$$\text{Quarter 5 (n = 5): } A = 10 * (1.1)^5 = 15.71$$

900

$$\text{Quarter 6 (n = 6): } A = 10 * (1.1)^6 = 17.31$$

901

$$\text{Quarter 7 (n = 7): } A = 10 * (1.1)^7 = 19.33$$

902

$$\text{Quarter 8 (n = 8): } A = 10 * (1.1)^8 = 21.59$$

903

904

Now, we can calculate the total number of new employees added in the first 2 years by summing up the number of new employees added each quarter:

905

906

$$\text{Total} = 11 + 12.1 + 13.21 + 14.41 + 15.71 + 17.31 + 19.33 + 21.59$$

907

$$\text{Total} = 119.66$$

908

909

Since we can't have a fraction of an employee, we can round down to the nearest whole number.

910

911

Therefore, the final answer is: 119

912

913

914

915

916

917

Figure 8: A valid arithmetic problem involving geometric series with a solve rate of 0.125. The solution requires either summing quarterly employee additions or applying the geometric series formula: $a * \frac{r^n - 1}{r - 1}$. Llama-3.2-3B-Instruct struggles with exponentiation calculations, resulting in poor performance on geometric series problems.

918
919
920921 A farmer has 10 different types of vegetables to store in a warehouse with 1,000 square meters of floor
922 space. There are four container sizes available:

- 923 - Small (5 sq m): Maximum 50 available
- 924 - Medium (10 sq m): Maximum 40 available
- 925 - Large (15 sq m): Maximum 30 available
- 926 - Extra-large (20 sq m): Maximum 25 available

927 The vegetables have different storage requirements:

- 928 - 3 bulky vegetables (pumpkins, watermelons, cabbages) require containers of at least 15 sq m
- 929 - 4 medium vegetables (tomatoes, peppers, eggplants, zucchini) require containers of at least 10 sq m
- 930 - 3 small vegetables (carrots, onions, potatoes) can fit in any container size

931 Each vegetable type must be stored in at least one container. What is the maximum number of containers that can be used while satisfying all constraints and not exceeding 1,000 sq m total space?

932

933 Figure 9: A valid optimisation problem with a solve rate of 0.375 generated at step 124.

934

935

936

937

938

939

940

941 Find the equation of the curve $y = f(x)$ where the derivative is given by $f'(x) = (3x^2 - x - 2)/2x$ and
942 the curve passes through the point (2, 3).

943

944

945 Figure 10: A valid calculus problem with a solve rate of 0.375 generated at step 156.

946

947

948

949

950

951

952 A golfer hits a ball from the top of a 50-meter high cliff with an initial velocity of 30 m/s at an angle
953 of 45 degrees above the horizontal. What is the horizontal distance traveled by the ball when it hits the
954 ground?

955

956

957

Figure 11: A valid physics problem that involves trigonometry with a solve rate of 0.5 generated at
step 172.

958

959

960

961

962

963

964 Consider a randomly ordered sequence of $n = 3q$ distinct integers $\{a_1, a_2, \dots, a_{3q}\}$ where q is a
965 positive integer. Define f as the number of adjacent pairs (a_i, a_{i+1}) in the sequence where both
966 integers have the same remainder when divided by 3 (i.e., $a_i \bmod 3 = a_{i+1} \bmod 3$). If the integers 1
967 through $3q$ are randomly permuted to form this sequence, what is the expected value of f ?

968

969

970

971

Figure 12: A valid probability problem with a solve rate of 0.25 generated at step 188.

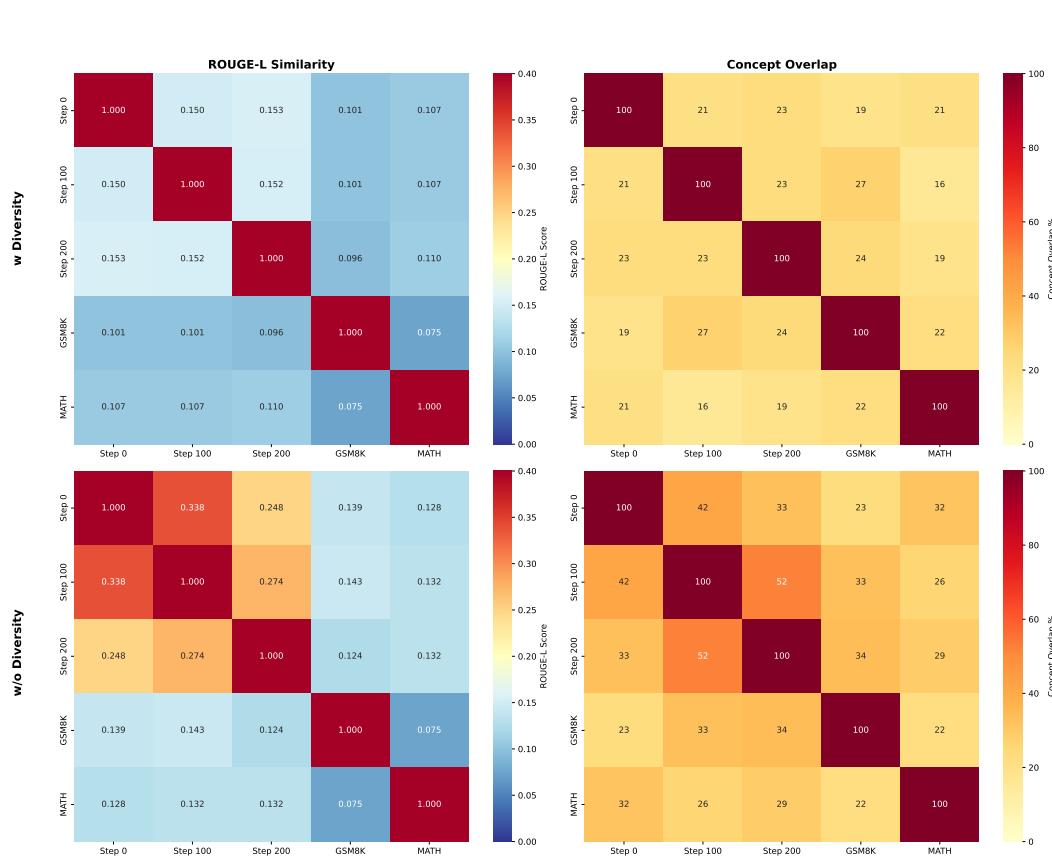


Figure 13: Heatmap visualisation of n-gram similarity (ROUGE-L scores) and concept overlap between generated problems at training steps 0, 100, 200 and reference datasets (MATH, GSM8K). Top row: with diversity reward; Bottom row: without diversity reward. With diversity reward incorporated, the generated problems exhibit low textual similarity and minimal concept overlap, demonstrating effective exploration of diverse problem types.

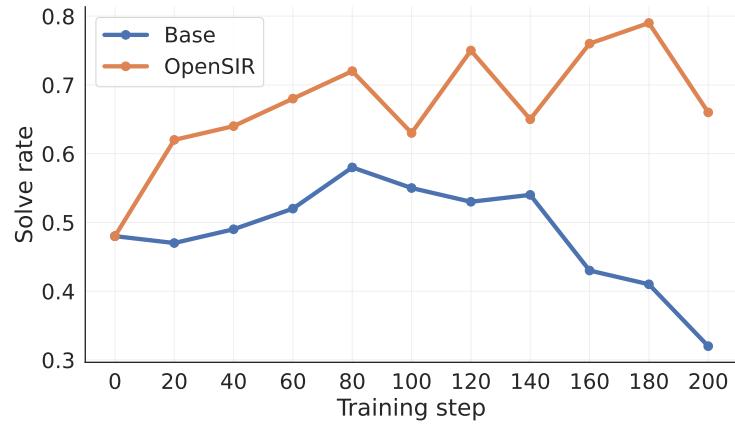


Figure 14: Progression of solve rates of OpenSIR and the initial instruction model as training goes.

1026

1027 B and C trisect \overline{AD} and M is the midpoint of \overline{AD} . $MC = 8$. How many units are in the length of
1028 \overline{AD} ?

1029

1030 Figure 15: A geometry problem from the MATH dataset, representing a different mathematical
1031 domain from the trivial arithmetic seed.

1032

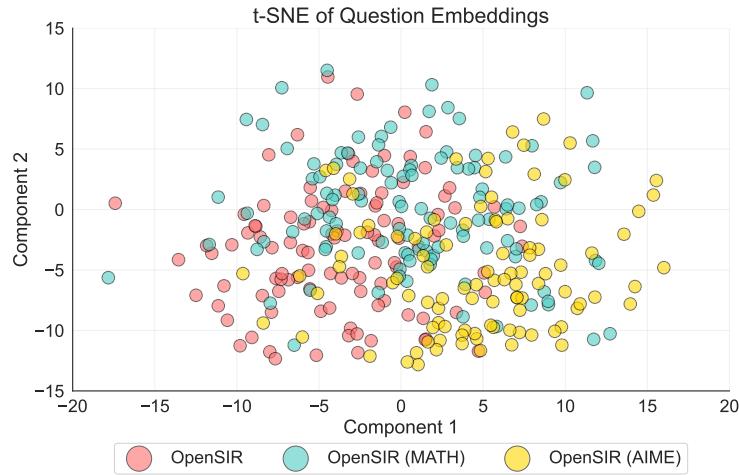
1033 Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she
1034 walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes
1035 spent in the coffee shop. When she walks $s + 2$ kilometers per hour, the walk takes her 2 hours and
1036 24 minutes, including t minutes spent in the coffee shop. Suppose Aya walks at $s + \frac{1}{2}$ kilometers per
1037 hour. Find the number of minutes the walk takes her, including the t minutes spent in the coffee shop.

1038

1039 Figure 16: A competition-level problem from AIME 2024, significantly more challenging than the
1040 trivial seed.

1041 Table 7 shows that all three variants achieve nearly identical performance (38.42, 38.67, and 38.81),
1042 with differences of less than 0.5 percentage points. This demonstrates that OpenSIR is robust to
1043 the initial seed problem, successfully escaping the limited starting point regardless of whether it
1044 begins with trivial arithmetic, a different mathematical domain (geometry), or a significantly more
1045 challenging competition-level problem.

1046



1062 Figure 17: t-SNE visualisation of problem embeddings generated by OpenSIR from three different
1063 initial seeds. The substantial overlap demonstrates that the method converges to similar problem
1064 distributions regardless of the starting point.

1066 Figure 17 visualises the diversity of problems generated at the final training step across the three
1067 different initial seeds. The t-SNE embeddings reveal that all three variants produce diverse prob-
1068 lems spanning similar regions of the semantic space, with substantial overlap in their distributions
1069 regardless of the initial seed. This confirms that OpenSIR successfully escapes its starting point by
1070 exploring a wide range of mathematical concepts, driven by the diversity and solvability rewards
1071 that encourage continuous exploration beyond the initial problem domain and difficulty level.

1080								
1081								
1082								
1083								
1084								
1085	Llama-3.2-3B-Instruct							
1086	Base	-	73.94	42.86	15.21	28.78	13.09	34.78
1087	GRPO _{gsm8k}	42	79.60	45.41	16.34	33.31	14.71	37.87
1088		43	79.62	44.56	16.64	33.35	14.52	37.74
1089		44	79.93	45.91	15.83	33.32	14.46	37.89
1090		Avg.	79.72 ± 0.19	45.30 ± 0.68	16.27 ± 0.41	33.33 ± 0.02	14.56 ± 0.13	37.83 ± 0.37
1091	GRPO _{math}	42	76.99	45.02	16.38	33.02	14.31	37.14
1092		43	76.51	45.23	15.95	32.87	13.85	36.88
1093		44	75.93	45.52	15.95	32.95	14.23	36.92
1094		Avg.	76.48 ± 0.53	45.26 ± 0.25	16.09 ± 0.25	32.95 ± 0.07	14.13 ± 0.24	36.98 ± 0.31
1095	OpenSIR	42	77.82	46.38	17.72	34.24	15.46	38.32
1096		43	78.58	45.91	17.23	34.58	15.86	38.43
1097		44	78.43	46.38	17.44	34.45	15.84	38.51
1098		Avg.	78.28 ± 0.40	46.22 ± 0.27	17.46 ± 0.24	34.42 ± 0.17	15.72 ± 0.23	38.42 ± 0.27
1099	Gemma-2-2B-Instruct							
1100	Base	-	38.50	16.51	10.09	19.11	3.00	17.44
1101	GRPO _{gsm8k}	42	58.32	18.86	7.53	20.17	3.18	21.61
1102		43	58.86	19.21	7.96	20.77	3.08	21.98
1103		44	59.06	19.36	7.76	20.42	3.37	21.99
1104		Avg.	58.75 ± 0.38	19.14 ± 0.26	7.75 ± 0.22	20.45 ± 0.30	3.21 ± 0.15	21.86 ± 0.27
1105	GRPO _{math}	42	55.14	22.31	7.95	15.71	3.03	20.83
1106		43	53.94	22.53	7.90	15.08	3.11	20.51
1107		44	59.01	23.44	8.02	18.15	3.57	22.44
1108		Avg.	56.03 ± 2.65	22.76 ± 0.60	7.96 ± 0.06	16.31 ± 1.62	3.24 ± 0.29	21.26 ± 1.42
1109	OpenSIR	42	58.68	24.09	8.89	22.29	2.99	23.39
1110		43	58.36	25.69	10.73	26.14	3.23	24.83
1111		44	57.03	24.49	8.89	21.66	3.24	23.06
1112		Avg.	58.03 ± 0.87	24.75 ± 0.83	9.51 ± 1.06	23.36 ± 2.43	3.15 ± 0.14	23.76 ± 1.30
1113	Qwen-2.5-3B-Instruct							
1114	Base	-	84.43	65.36	25.23	48.22	27.94	50.24
1115	GRPO _{gsm8k}	42	84.71	65.40	26.33	48.51	28.21	50.63
1116		43	85.16	65.80	24.84	48.46	28.50	50.55
1117		44	84.96	66.10	24.75	48.40	28.23	50.49
1118		Avg.	84.94 ± 0.23	65.77 ± 0.35	25.31 ± 0.89	48.46 ± 0.06	28.31 ± 0.16	50.56 ± 0.45
1119	GRPO _{math}	42	84.24	65.74	25.23	48.53	28.23	50.39
1120		43	84.19	65.64	25.14	48.20	27.98	50.23
1121		44	84.49	66.30	24.59	48.29	28.57	50.45
1122		Avg.	84.31 ± 0.16	65.89 ± 0.36	24.98 ± 0.35	48.34 ± 0.17	28.26 ± 0.30	50.36 ± 0.28
1123	OpenSIR	42	85.43	66.17	26.49	48.88	28.86	51.17
1124		43	85.26	65.64	25.30	48.62	28.30	50.62
1125		44	85.44	65.79	26.08	48.72	27.83	50.77
1126		Avg.	85.38 ± 0.10	65.87 ± 0.28	25.96 ± 0.61	48.74 ± 0.13	28.33 ± 0.52	50.85 ± 0.38
1127								
1128								
1129	Table 5: Math reasoning evaluation results for 2B/3B models with individual seed reporting. We report avg@16 per problem for each seed (42, 43, 44) and their average with standard deviation as superscript.							
1130								
1131								
1132								
1133								

1134

1135

1136

1137

1138

1139

1140

Models	Seed	GSM8K	MATH-500	Minerva	College Math	Olympiad-Bench	Avg.
Llama-3.1-8B-Instruct							
Base	-	84.50	47.89	22.75	34.10	16.26	41.10
GRPO _{gsm8k}	42	89.73	50.89	25.61	36.16	15.65	43.55
	43	88.30	49.93	24.45	34.33	16.29	42.66
	44	88.07	50.29	24.43	34.60	17.35	42.95
	Avg.	88.70 _{±0.73}	50.37 _{±0.39}	24.83 _{±0.55}	35.03 _{±0.81}	16.43 _{±0.70}	43.05 _{±0.62}
GRPO _{math}	42	86.93	51.02	24.43	35.74	17.13	43.05
	43	85.98	50.53	23.88	34.48	16.56	42.29
	44	85.77	50.91	23.63	34.57	15.93	42.16
	Avg.	86.23 _{±0.50}	50.82 _{±0.21}	23.98 _{±0.33}	34.93 _{±0.57}	16.54 _{±0.49}	42.50 _{±0.39}
OpenSIR	42	88.05	52.62	27.79	37.14	18.45	44.81
	43	87.03	52.07	27.15	35.82	17.81	43.98
	44	86.82	52.45	26.94	35.91	17.18	43.86
	Avg.	87.30 _{±0.54}	52.38 _{±0.23}	27.29 _{±0.36}	36.29 _{±0.60}	17.81 _{±0.51}	44.21 _{±0.42}

Table 6: Math reasoning evaluation results for 8B models with individual seed reporting. We report avg@16 per problem for each seed (42, 43, 44) and their average with standard deviation as superscript.

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

Model	Acc
OpenSIR	38.42
OpenSIR _{MATH}	38.67
OpenSIR _{AIME}	38.81

Table 7: Performance of OpenSIR with different initial seed problem.

1181

1182

1183

1184

1185

1186

1187

1188
1189

A.6 OPENSIR INCENTIVISES REASONING CAPACITY

1190
1191
1192

To verify whether OpenSIR elicits genuine reasoning improvements rather than memorisation, we evaluate pass@k performance on five challenging mathematical benchmarks following (Yue et al., 2025).

1193
1194
1195
1196

Figure 19 shows that OpenSIR consistently outperforms base instruction models across all k values (8–256) on all benchmarks. These results confirm that OpenSIR drives genuine advances in mathematical reasoning capacity.

1197

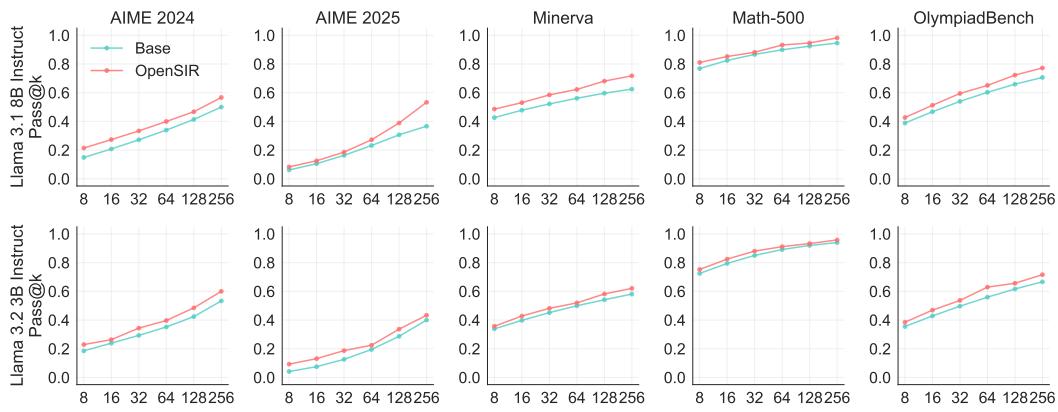
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214

Figure 18: Pass@k curves comparing base instruction models and OpenSIR across five mathematical benchmarks. OpenSIR consistently improves performance across all k values, with stable or increasing gaps at higher k, demonstrating genuine reasoning improvements rather than memorization.

1215

1216
1217

A.7 PROLONGED TRAINING ANALYSIS

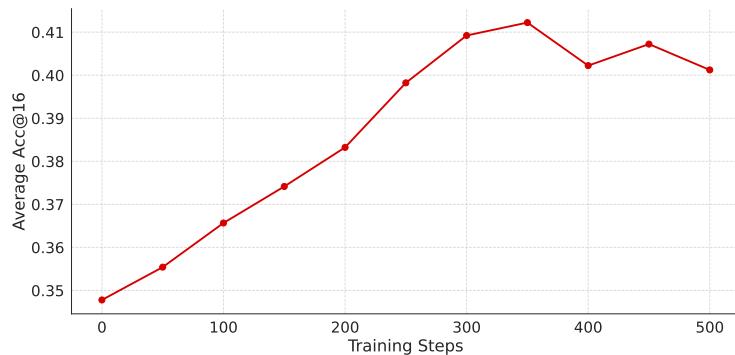
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232

Figure 19: Performance of OpenSIR extended training using Llama-3.2-3B-Instruct.

1233
1234
1235
1236
1237

To better understand the limitation of OpenSIR, we extend training of a single run of Llama-3.2-3B-Instruct to 500 steps, substantially beyond the 200 steps used in the main experiments. Figure 19 shows the evaluation performances over training. We observe consistent improvement from 34.8% to 41.3% at step 350, representing a gain of +6.5 points. Performance then plateaus after step 350 and remains stable around 40-41% through step 500, with no further statistically significant gains.

1238
1239
1240
1241

Preliminary examination of the generated problems suggests a likely cause for this saturation. The model appears to have explored most major mathematical topics by step 350, after which the generated problems become increasingly similar and repetitive. This indicates that the diversity reward mechanism may become less effective over extended training. Future work could investigate more sophisticated diversity mechanisms to foster open-ended exploration over long training horizons.

1242
1243
1244
1245 **A.8 SYNERGY WITH ANNOTATED DATA**
1246

Model	GSM8K	MATH-500	Minerva	College Math	OlympiadBench	Avg.
Base	73.94	42.86	15.21	28.78	13.09	34.78
GRPO _{gsm8k}	79.72	45.30	16.27	33.33	14.56	37.83 ^{+3.05}
OpenSIR	78.28	46.22	17.46	34.42	15.72	38.42 ^{+3.64}
GSM8K → OpenSIR	81.43	46.12	19.43	36.15	18.35	40.30 ^{+5.52}
GSM8K & OpenSIR	81.57	49.48	20.39	36.85	18.14	41.29^{+6.51}

1247
1248
1249
1250
1251
1252
1253 Table 8: The avg@16 performance on five mathematical benchmarks. OpenSIR obtains better re-
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295

While we showed that OpenSIR achieves significant improvements in math reasoning without using annotated data, we further investigate if OpenSIR can be combined with annotated data to achieve even greater performance gains. Having demonstrated that OpenSIR achieves significant improvements without annotated data, we investigate whether combining OpenSIR with annotated data can yield further gains. We focus on Llama-3.2-3B-Instruct and use Gsm8K as the training data, as Table 1 shows that fine-tuning on GSM8K consistently outperforms using MATH.

We explore two training strategies: (1) **GSM8K → OpenSIR**: The model is first trained on GSM8K for half the training iterations, then trained with OpenSIR for the remaining half. (2) **GSM8K & OpenSIR**: Each training iteration uses half GSM8k samples and half OpenSIR samples.

Table 8 shows that both setups achieve better performance than using OpenSIR alone or only on GSM8K. One possible explanation for the sequential approach’s effectiveness (GSM8K → OpenSIR) is that training on GSM8K first may improve the model’s foundational reasoning abilities, which could provide a stronger starting point for OpenSIR’s self-generated questions. The concurrent approach (GSM8K & OpenSIR) achieves a slight additional edge, which might be attributed to the model receiving better feedback signals for question calibration from the beginning, as it can leverage both supervised and self-generated data simultaneously throughout training. The precise underlying mechanisms for these improvements require further investigation.

A.9 COMPUTATIONAL COST ANALYSIS

In standard GRPO training, each iteration processes a batch of B prompts, generating G responses for each prompt, resulting in $B \times G$ total forward passes per iteration. In method, each training iteration involves generating B problems and G solution attempts for each problem, yielding B forward passes for problem generation and $B \times G$ forward passes for solution generation, for a total of $B + B \times G = B(1+G)$ forward passes. Compared to the $B \times G$ forward passes in standard GRPO training, method requires an additional B forward passes for problem generation. This represents a relative computational overhead of $\frac{B}{B \times G} = \frac{1}{G}$, or 12.5% with $G = 8$ solution attempts per problem.

Problem embeddings for diversity scoring are computed asynchronously during solution generation, incurring no additional wall-clock time. Cosine distance calculations between problem embeddings require $\mathcal{O}(B \times |\mathcal{P}_t|)$ operations where $|\mathcal{P}_t|$ is the problem pool size, but execute in under 3 seconds per iteration in our experiments—negligible compared to LLM forward passes. Note that while method generates B problems per iteration, only the top-scoring $B/2$ problems are selected for teacher training and another top-scoring $B/2$ for student training, as described in Section 2.4. Overall, method achieves improved performance without human-annotated training data at this modest computational overhead.

B ANNOTATION DETAILS

One of the authors prepare the samples for annotation, and the rest of the authors annotated the samples with the instructions provide in Figure 20.

```

1296
1297 You will be presented with multiple sets of 5 math problems to evaluate. For each set, please complete
1298 the following three-step annotation process.
1299 # Step 1: Identify Topics
1300 For each problem, identify ALL relevant mathematical topics from the following list:
1301 - Algebra
1302 - Geometry
1303 - Calculus
1304 - Probability
1305 - Statistics
1306 - Number Theory
1307 - Combinatorics
1308 - Optimization
1309 - Arithmetic
1310 - Discrete Math
1311 - Trigonometry
1312 # Step 2: Assess Validity
1313 For each problem, determine if it is valid or invalid:
1314 - Valid: The problem is logically sound, clearly stated, and can be answered with the given information
1315 - Invalid: The problem contains logical flaws, contradictions, insufficient information, or ambiguities
1316 that prevent a proper solution
1317 # Step 3: Rank Difficulty
1318 Rank all 5 problems from easiest to hardest. Provide your ranking as a sequence of problem numbers.
1319 Example: [3, 1, 5, 2, 4] means problem 3 is the easiest and 4 is the hardest.
1320 Consider these factors when assessing difficulty:
1321 - Number of steps required
1322 - Complexity of concepts involved
1323 - Level of mathematical knowledge needed
1324 - Computational complexity
1325 # Response Format
1326 Provide your annotations as a JSON list where each element represents one problem set. Here are
1327 some examples:
1328
1329 [
1330   {
1331     "set_id": "SET_1",
1332     "problems": {
1333       "1": {"topics": ["Algebra", "Calculus"], "valid": true},
1334       "2": {"topics": ["Geometry"], "valid": false},
1335       "3": {"topics": ["Probability"], "valid": true},
1336       "4": {"topics": ["Number Theory"], "valid": true},
1337       "5": {"topics": ["Arithmetic"], "valid": true}
1338     },
1339     "difficulty_ranking": [5, 3, 1, 2, 4]
1340   },
1341   {
1342     "set_id": "SET_2",
1343     "problems": {
1344       "1": {"topics": ["Statistics"], "valid": true},
1345       "2": {"topics": ["Discrete Math"], "valid": true},
1346       "3": {"topics": ["Optimization"], "valid": true},
1347       "4": {"topics": ["Algebra"], "valid": false},
1348       "5": {"topics": ["Geometry", "Algebra"], "valid": true}
1349     },
1350     "difficulty_ranking": [1, 2, 5, 3, 4]
1351   },
1352   ...
1353 ]

```

Figure 20: The instruction provided to the annotators to annotate problems.

1350 **C ADDITIONAL ABLATIONS**
13511352 **C.1 SOLUTION LENGTH REWARD INCREASES PROBLEM COMPLEXITY**
1353

1354 Model	1355 Question Length	1356 Solution Length	1357 Acc
w/ length	207	387	38.42
w/o length	150	238	37.86

1358 Table 9: Comparison of OpenSIR performance with and without solution length reward. Solution
1359 length reward improves OpenSIR accuracy and increases average question and solution lengths.
13601361 We investigate the impact of the solution length reward in OpenSIR. Table 9 shows this reward im-
1362 proves performance from 37.86% to 38.42%. It also increases the average question length (from 150
1363 to 207 tokens) and solution lengths (from 238 to 387 tokens). By examining the generated questions
1364 manually, we find that the policy tends to generate more sophisticated problems involving advanced
1365 concepts with this reward, such as linear programming and optimization, which naturally require
1366 longer multi-step solutions to solve. These results demonstrate that the solution length reward ef-
1367 fectively guides the policy toward generating more complex problems, which in turn leads to better
1368 performance.
13691370 **C.2 ROBUSTNESS TO DIVERSITY MEASUREMENTS**
1371

1372 Reward	1373 Acc	1374 # Concepts
Embedding	38.42	5914
Concepts	38.26	6213

1375 Table 10: Comparison of diversity measurement approaches in OpenSIR. Despite slight differences
1376 in concept coverage, both embedding-based and concept-based diversity rewards yield nearly iden-
1377 tical accuracy, demonstrating the framework’s robustness to the choice of diversity metric.
13781379 We have established the necessity of diversity rewards in Section 4.3. In this section, we further
1380 investigate OpenSIR’s robustness to different diversity measurement approaches. We implement
1381 concept-based diversity by measuring diversity through the mathematical concepts of the problems
1382 (Lu et al., 2023; Havrilla et al., 2025). Formally, we define the concept diversity reward as:
1383

1384
$$r_{\text{con}}(q) = \frac{|\mathcal{C}_q| - |\mathcal{C}_q \cap \mathcal{C}_{\mathcal{P}_{t-1}}|}{3} \quad (10)$$

1385 where \mathcal{C}_q are the concepts in problem q and $\mathcal{C}_{\mathcal{P}_{t-1}} = \bigcup_{q' \in \mathcal{P}_{t-1}} \mathcal{C}_{q'}$ represents the union of concepts
1386 from all problems in the existing pool. Since each problem contains at most three concepts, this
1387 reward calculates the fraction of new concepts introduced.
13881389 Table 10 shows that both embedding-based and concept-based diversity rewards achieve similar ac-
1390 curacy (38.42 vs 38.26), demonstrating the framework’s robustness to the choice of diversity metric.
1391 Beyond accuracy, we examine concept coverage, which refers to the number of unique mathemati-
1392 cal concepts discovered during training, as a direct measure of exploratory diversity. As expected,
1393 concept-based diversity achieves slightly higher coverage (6,213 concepts) since it explicitly op-
1394 timises for novel concept discovery. Surprisingly, embedding-based diversity attains comparable
1395 coverage (5,914 concepts), 95% of the concept-based approach, despite not tracking concepts ex-
1396 plicitly. This suggests that maximising representational spread in embedding space effectively pro-
1397 motes novelty discovery, achieving open-ended learning.
13981399 **D IMPLEMENTATION DETAILS**
14001401 **D.1 TRAINING DETAILS**1402 We implement OpenSIR based on the TRL framework (von Werra et al., 2020). Table 11 provides a
1403 summary of the training hyperparameters used in our experiments.
14041405 **D.2 PROMPTS**1406 We detailed the prompt for generating problems in Figure 21 and solving problems in Figure 22.
1407

Category	Hyperparameter	Value
Trainer	Learning rate	3×10^{-7}
	Optimiser	AdamW (Loshchilov & Hutter, 2018)
	Warmup steps	20
	Training steps	100/200
	KL loss coefficient	1×10^{-4}
	Gradient norm clipping	0.5
	Seeds	42/43/44
	GPUs	3 H100
Rollout	Batch size [†]	256
	Max prompt length	1024
	Max solution length	2048
	Number of rollouts per prompt	8
	Temperature	1.0
Teacher Rewards	Solvability weight (α)	1.0
	Solution length weight (λ)	1.0
	Diversity weight (γ)	1.0
	Format weight (δ)	0.1
	Embedding model	Linq-Embed-Mistral (7B)
Student Rewards	Accuracy weight	1.0
	Format weight (δ)	0.1

[†] The number of rollouts seen for one gradient update.

Table 11: The training configurations for the experiments.

You are given a math problem: {Problem}
 Your task is to create a math problem that is conceptually different from the provided problem. The new problem must be answerable with a numerical value or mathematical expression.
 First, explain how your new problem differs conceptually from the original problem inside the `<think>...</think>` tags. Then, present your new problem inside the `<problem>...</problem>` tags. Finally, identify at most three math concepts required to solve your problem. Provide these concepts in a comma separated list inside the `<concepts>...</concepts>` tags.

Figure 21: Prompt for generating math problems. {Problem} is a placeholder for the reference problem sampled from the problem pool.

You are a helpful AI Assistant, designed to provide well-reasoned and detailed responses. You FIRST think about the reasoning process step by step and then provide the user with the answer. The last line of your response should be 'Therefore, the final answer is: `\boxed{ANSWER}`' (without quotes) where ANSWER is just the final number or expression that solves the problem.
 {Problem}

Figure 22: Prompt for generating solutions to math problems. {Problem} is a placeholder for the actual problem.