ARES: AUTOMATED RUBRIC SYNTHESIS FOR SCALABLE LLM REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Rubric-based rewards offer a promising way to extend reinforcement learning (RL) for large language models beyond tasks with automatically verifiable answers. However, scaling rubric-based RL remains challenging: existing approaches often rely on expert-written rubrics and manually constructed question sets, while fixed task-level rubrics may fail to capture the evaluation requirements of individual questions. We propose **ARES** (**A**utomated **R**ubric synth**E**sis for **S**calable RL), a framework for automatically constructing rubric-based RL data at scale. Starting from raw pretraining documents, ARES converts source knowledge into self-contained question-answer pairs and co-generates question-specific weighted rubrics, enabling instance-level reward supervision for open-ended responses. To improve diversity and quality, ARES conditions generation on domain labels and persona information, and applies validation filters for question self-containment, answer faithfulness, and rubric validity. Using ARES, we construct **100K** rubric-annotated instances across ten domains. Experiments on seven benchmarks show that rubric-based RL trained with ARES, outperforms continual pretraining, supervised fine-tuning, and binary-reward RL, with the largest gains on multi-dimensional open-ended tasks such as healthcare and instruction following.

1 INTRODUCTION

Large Language Models (LLMs) are typically developed through large-scale pre-training on web-scale text corpora, which equips them with broad linguistic patterns and knowledge (Yang et al., 2025; Weber et al., 2024), followed by post-training to better align them with human instructions and enhance their problem-solving abilities (Ross et al., 2011; Bachmann & Nagarajan, 2024; Shao et al., 2024). Among recent post-training methods, Reinforcement Learning with Verifiable Rewards (RLVR) has become a particularly influential recipe for improving reasoning capabilities (Guo et al., 2025; Jaech et al., 2024). By optimizing models against automatically checkable outcomes, RLVR has achieved strong results in domains such as mathematics, coding, and short-answer reasoning. However, its success also exposes a fundamental limitation: *scalable RL training is currently concentrated in tasks where correctness can be cheaply and reliably verified.*

This dependence on verifiable outcomes limits RLVR in two important ways. Firstly, such verifiable rewards are mainly available for tasks with automatically checkable final answers, such as math, coding, and short-answer tasks (Luo et al., 2025; Face, 2025; Guha et al., 2025; Cen et al., 2026), limiting the diversity of domains and task types that can be reliably scaled. Secondly, the reward is sparse and often binary-like, making RL optimization difficult (Lightman et al., 2023). Therefore, RLVR is powerful but narrow: it scales well only where verification is cheap, and provides limited supervision over how a response should satisfy complex, multi-dimensional requirements. Rubric-based rewards (Gunjal et al., 2026) offer a promising alternative by evaluating responses against a set of weighted criteria, thereby decomposing response quality into multiple dimensions and providing finer-grained learning signals.

Despite their promise, scaling rubric-based RL remains challenging for two reasons. First, constructing rubric-annotated data usually requires experts to manually write questions, reference answers, and evaluation criteria, making it difficult to scale. Second, many rubrics are defined at the task or dataset level, so they provide general evaluation dimensions but cannot specify what should be emphasized for each individual question. To address these issues, we aim to automatically synthesize large-scale

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

rubric-based RL datasets with question-specific evaluation criteria and importance weights, enabling the reward to emphasize the aspects most critical to answer quality for each question.

In this work, we propose **ARES** (**A**utomated **R**ubric synth**E**sis for **S**calable RL), a framework for automatically constructing large-scale rubric-based RL training data from raw pretraining documents. Given a document containing knowledge to be transferred into the RL stage, ARES generates a self-contained question-answer pair that can be answered without access to the source document, together with a rubric consisting of question-specific evaluation criteria and their importance weights. To further improve data diversity, ARES conditions the generation process on domain and persona signals, producing questions and response styles that cover diverse scenarios. Using this pipeline, we synthesize **100K** rubric-annotated instances across ten domains, enabling RL training over a much broader range of open-ended capabilities than conventional verifiable-reward settings can support. Our main contributions are as follows:

- We propose ARES, a scalable document-to-RL pipeline that automatically transforms raw pretraining documents into RL instances with self-contained questions, reference answers, and question-specific weighted rubrics.
- We construct a large-scale rubric-annotated dataset of **100K** examples spanning ten domains, where each instance provides fine-grained reward supervision.
- We evaluate ARES across seven diverse benchmarks, showing that it outperforms continual pretraining, supervised fine-tuning, and binary-reward RL, with especially large gains on open-ended tasks such as healthcare (+6.4) and instruction following (+15.5).

2 Related Work

Data Synthesis for RL. Early efforts focused on curating domain-specific datasets for mathematics and code (Luo et al., 2025; Face, 2025). Subsequent work expanded coverage to broader web-scale sources (Yuan et al., 2026; Guha et al., 2025; Singhal et al., 2025), increasing diversity while remaining focused on tasks with verifiable ground truth. Our work most closely relates to WebScale-RL (Cen et al., 2026), which grounds both questions and answers in pretraining documents to enable large-scale RL data synthesis. However, all prior approaches generate short verifiable answers and rely on binary reward signals, restricting RL to tasks with unambiguous ground truth. We extend this paradigm to open-ended domains by synthesizing multi-dimensional rubric alongside QA pairs.

Reward Design for RL Training. The dominant approach uses binary rewards based on exact string matching (Cen et al., 2026; Luo et al., 2025; Face, 2025), which is robust and stable but limited to verifiable tasks. An alternative is using LLMs as judges (Zheng et al., 2023), where a model evaluates response quality holistically; this is flexible but can be inconsistent and expensive. Process reward models (Lightman et al., 2023; Wang et al., 2024a) provide step-level feedback for multi-step reasoning, improving training signal density over outcome-level binary rewards, but require substantial annotation to train. Rubric-based evaluation, common in educational assessment and long-form writing evaluation (Wu et al., 2026), decomposes quality into multiple weighted dimensions, enabling structured, transparent, and reproducible assessment. Gunjal et al. (2026) recently demonstrate the promise of rubric-based rewards for RL, showing gains over binary and Likert-scale baselines on medical and science domains. However, their approach relies on curated question sets from specific domains and requires separate rubric generation passes. We extend this direction by co-generating rubrics with QA pairs from raw pretraining documents in a single inference pass, enabling rubric-based RL at an order of magnitude greater scale and domain diversity.

Pretraining-Stage and Large-Scale RL. Recent work explores applying RL at earlier training stages or at larger scale. Cen et al. (2026) and Li et al. (2025) demonstrate that RL on pretraining-scale data yields significant gains. Liu et al. (2026; 2025) investigate prolonged RL training and its scaling properties. Our work complements these efforts by focusing on the reward signal design rather than the scale of training, showing that richer rewards enable RL to benefit a broader set of capabilities including open-ended domains.

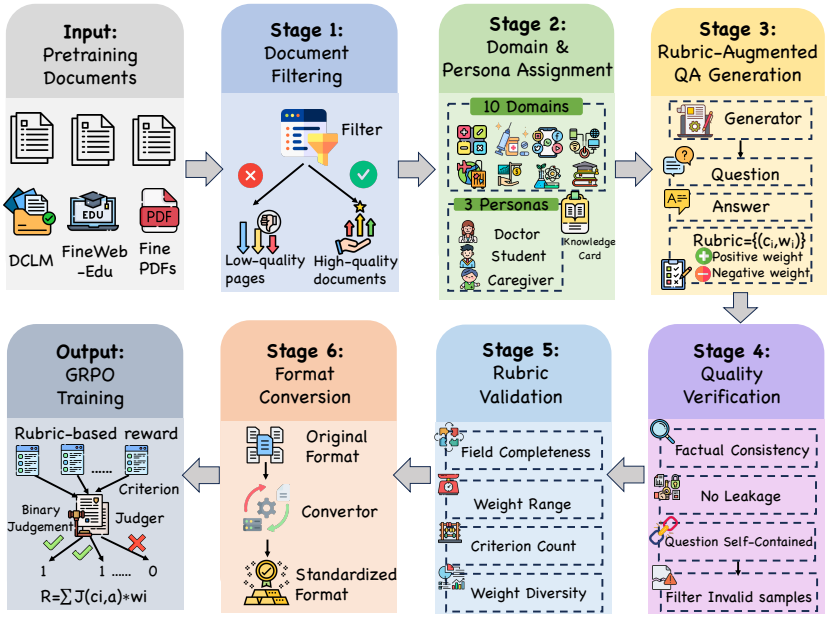


Figure 1: Overview of the six-stage ARES pipeline. Starting from raw pretraining documents, ARES performs document filtering, domain and persona conditioning, rubric-augmented QA generation, quality verification, rubric validation, and format conversion to produce training instances.

3 METHODOLOGY

3.1 PRELIMINARIES

RLVR. RLVR is a post-training paradigm that improves LLMs by optimizing them against automatically checkable outcomes. Given a question q, a policy samples multiple candidate responses, each of which receives a reward from an external verifier according to answer correctness or task success. RLVR has been particularly effective for domains such as mathematics and coding, where final answers or program outputs can be reliably verified.

In this work, we use Group Relative Policy Optimization (**GRPO**) (Shao et al., 2024) as the RL optimizer. For each question q, the behavior policy $\pi_{\theta_{\text{old}}}$ samples a group of G responses $\{y_i\}_{i=1}^{G}$, and each response y_i is assigned a reward R_i. GRPO optimizes the policy by comparing responses within the same group:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{y_i\}_{i=1}^{G}} \left[\frac{1}{G} \sum_{i=1}^{G} \min \left(\rho_i \hat{A}_i, \text{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon) \hat{A}_i \right) - \beta \, \text{KL}(\pi_\theta \| \pi_{\text{ref}}) \right], \quad (1)$$

where

$$\rho_i = \frac{\pi_\theta(y_i \mid q)}{\pi_{\theta_{\text{old}}}(y_i \mid q)}, \qquad \hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G}) + \epsilon_A}. \quad (2)$$

Here, ρ_i is the importance ratio, $\hat{A}_i$ is the group-normalized advantage, ε is the clipping threshold, β controls KL regularization, π_{ref} is the reference policy, and ϵ_A is a small constant for numerical stability. In standard RLVR, R_i is usually provided by a verifier that checks outcome-level correctness. While effective, such rewards are naturally restricted to tasks with reliable automatic verification and often provide sparse supervision.

RL with Rubric-Based Rewards. Rubric-based rewards extend RL training to more open-ended tasks by replacing outcome-only verification with multi-dimensional response evaluation (Gunjal et al., 2026). Instead of judging a response solely by final-answer correctness, a rubric specifies a set of criteria that describe different aspects of response quality, such as factual correctness, completeness, reasoning consistency, instruction following, or avoidance of unsupported claims.

Formally, for a question q, a rubric is defined as

$$\mathcal{R}(q) = \{(c_k, w_k)\}_{k=1}^{N}, \tag{3}$$

where c_k denotes the k-th evaluation criterion and w_k denotes its importance weight. Given a candidate response y, an LLM judge J_ϕ scores whether y satisfies each criterion. The rubric-based reward is computed as

$$R_{\text{rubric}}(q, y; \mathcal{R}) = \sum_{k=1}^{N} w_k \cdot J_\phi(q, y, c_k), \tag{4}$$

where $J_\phi(q, y, c_k) \in [0, 1]$ measures the degree to which response y satisfies criterion c_k. "1" encourages desirable properties, while negative weights can penalize common failure modes, such as hallucination, irrelevant information, or unsafe recommendations. During RL training, the rubric reward is used as the response reward in Eq. (1), i.e.,

$$R_i = R_{\text{rubric}}(q, y_i; \mathcal{R}). \tag{5}$$

Compared with final-answer verification, rubric-based rewards provide denser and more diagnostic supervision because they identify which aspects of a response are satisfactory or deficient. However, existing rubric-based RL methods typically rely on manually designed criteria, expert-written rubrics, or fixed task-level evaluation dimensions (Gallego, 2025; Akyürek et al., 2025). This limits both scalability and granularity: manually written rubrics are costly to construct at scale, while fixed rubrics may fail to capture the specific evaluation requirements of individual questions. ARES addresses this bottleneck by automatically synthesizing question-specific weighted rubrics together with self-contained question-answer pairs from raw pretraining documents.

3.2 Data Pipeline

To tackle these limitations, ARES constructs scalable rubric-based RL data by converting raw pretraining documents into instances with question-specific weighted rubrics. Given a document d, the pipeline produces a tuple

$$d \xrightarrow{\text{ARES}} (q, a^*, \mathcal{R}_q), \tag{6}$$

where q is a self-contained question, a^* is the reference answer, and $\mathcal{R}_q = \{(c_k^{(q)}, w_k^{(q)})\}_{k=1}^{N_q}$ is a rubric tailored to q. During RL training, $\mathcal{R}_q$ is used to compute the reward for each sampled response:

$$R_{\text{rubric}}(q, y; \mathcal{R}_q) = \sum_{k=1}^{N_q} w_k^{(q)} \cdot J(q, y, c_k^{(q)}). \tag{7}$$

The pipeline consists of three major phases: source preparation, rubric-augmented QA co-generation, and quality control.

Source preparation. ARES first filters raw pretraining documents using an LLM-based classifier, removing documents that are low-quality, boilerplate-heavy, or lack sufficient context for question generation. To enhance the diversity of both questions and rubrics, we further annotate each retained document with a semantic domain and three target personas. For instance, the same medical article may be used to generate questions tailored to medical professional, psychology student, or family caregiver, resulting in different levels of background knowledge, answer styles, and reasoning depth.

Rubric-augmented QA co-generation. Given a filtered document together with its domain and persona conditioning, ARES uses an LLM generator M_{gen}[1] to jointly produce the question, reference answer, and question-specific rubric:

$$(q, a^*, \mathcal{R}_q) = M_{\text{gen}}(d, \delta, p). \tag{8}$$

The generated question is required to be self-contained, so that it can be answered without access to the source document. The reference answer is grounded in the document, while the rubric specifies

[1]Document filtering and rubric-augmented QA generation use Gemini-3.1-Pro-Preview; domain/persona conditioning, question and rubric quality control use Claude-Sonnet-4.5.

Table 1: Comparison of existing datasets and RL training pipelines across four key desiderata. ✓ indicates the property is satisfied; × indicates it is not; / indicates the property is not applicable. **Multi-Domain**: covers diverse domains beyond math and code. **Rubric Rewards**: provides multi-dimensional rubric-based reward signals rather than binary verification. **Doc-Grounded**: questions and answers are grounded in source pretraining documents. **Scalable**: the pipeline can be readily scaled to larger data without manual curation. ARES is the only approach satisfying all four criteria.

Dataset	Multi-Domain	Rubric Rewards	Doc-Grounded	Scalable
DeepScaleR (Luo et al., 2025)	×	×	×	×
OpenR1-Math (Face, 2025)	×	×	×	×
OpenThoughts3 (Guha et al., 2025)	×	×	×	×
Nemotron (Singhal et al., 2025)	×	×	×	×
RaR (Gunjal et al., 2026)	×	✓	×	×
NaturalReasoning (Yuan et al., 2026)	✓	×	✓	✓
Webscale-RL (Cen et al., 2026)	✓	×	✓	✓
ARES (ours)	✓	✓	✓	✓

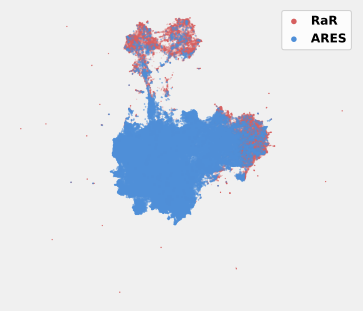


Figure 2: UMAP visualization of questions for RaR and ARES by Qwen3-Embeddings-0.6B.

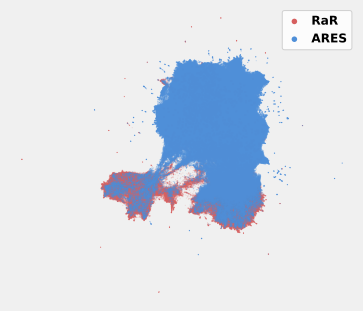


Figure 3: Distribution of rubric criteria in RaR and ARES by Qwen3-Embeddings-0.6B.

how candidate responses should be evaluated for this particular question. By co-generating q, a^*, and $\mathcal{R}_q$ in a single pass, ARES aligns the evaluation criteria with both the source knowledge and the intended answer, rather than relying on generic task-level rubrics. The rubric contains weighted positive criteria for desired response properties and negative criteria for common failure modes, with the number of criteria adapted to question complexity.

Quality control. ARES applies validation before using an instance for training. QA validation checks whether the question is self-contained, whether the reference answer is faithful to the source document, and whether the answer cannot be trivially inferred from surface cues in the question alone. Rubric validation checks whether each criterion is well-formed, whether the rubric covers the key requirements of the question, and whether the weights are valid and sufficiently diverse. Instances that fail validation are discarded. The remaining tuples are serialized with metadata such as domain, persona, and instance identifiers to form GRPO-compatible training data. Finally, we perform data deduplication and remove overlaps in the evaluation sets.

3.3 Comparison with Prior Work

Table 1 positions ARES relative to existing datasets and pipelines along four key dimensions. Math- and code-focused RL datasets (Luo et al., 2025; Face, 2025; Guha et al., 2025; Singhal et al., 2025) provide verifiable binary rewards but collect queries from limited curated sources, restricting both domain coverage and scalability. Webscale-RL (Cen et al., 2026) and NaturalReasoning (Yuan et al., 2026) extend data synthesis to diverse, document-grounded sources, but remain limited to binary reward signals, excluding open-ended domains where exact-match verification is infeasible. RaR (Gunjal et al., 2026) introduces rubric-based rewards but relies on curated question sets from science and medicine domains, limiting both scale and coverage. As visualized in Figure 2 and Figure 3, RaR (red) is confined to two narrow clusters, whereas ARES (blue) not only fully covers those regions but extends to substantially broader areas. ARES is the first pipeline to simultaneously

satisfy all four criteria: it draws from large-scale pretraining corpora, covers ten diverse domains, grounds all outputs in source documents, and produces multi-dimensional rubric rewards—enabling RL training at a breadth and scale previously unattainable.

3.4 Dataset Statistics

Table 2: Domain distribution of the ARES dataset spanning ten domains.

Domain	# Examples
Social Science	18,878
Technology & Engineering	18,321
Medicine & Health	13,974
Travel & Lifestyle	12,767
Commerce & Economics	12,439
Natural Science	10,154
Education	8,822
Other	3,931
Coding	1,631
Math	930
Total	**101,847**

Table 3: Rubric statistics of the ARES dataset.

Metric	Value
#Criteria	1,108,163
Avg. Criteria	10.88
Median Criteria	11
Positive Criteria	817,047
Negative Criteria	291,116

We run ARES on a small subset of pretraining corpora from DCLM (Li et al., 2024), FineWeb-Edu (Penedo et al., 2024), and FinePDFs (Kydlíček et al., 2025). Specifically, we first sample and filter approximately 0.1B tokens from these corpora as the source pool for rubric-augmented QA synthesis. After document filtering, co-generation, and validation, ARES produces 101,847 rubric-annotated QA instances, corresponding to an overall retention rate of 71.6%.

As summarized in Table 2, the resulting dataset spans ten domains, with Social Science and Technology & Engineering constituting the two largest categories. Coding and Math account for smaller proportions in the generated dataset, reflecting the domain composition after document filtering and classification. The remaining domains cover medicine, lifestyle, commerce, natural science, education, and other topics.

Table 3 reports the resulting rubric statistics. The dataset contains 1,108,163 criteria in total, with an average of 10.88 criteria per instance and a median of 11, consistent with our complexity-aware generation design. Among them, 817,047 are positive criteria that reward desired response properties, while 291,116 are negative criteria that penalize common failure modes. This mixture is intended to provide both positive incentives and discriminative penalties during rubric-based RL training.

4 Experiments

4.1 Experimental Setup

Base Model. All experiments use Qwen3-4B-Base (Yang et al., 2025) as the foundation model, which we train from scratch in each setting without any instruction tuning warmup unless specified.

Baselines. We compare against four baselines trained from the same foundation model. (1) **CPT** continues pretraining on the same source documents used by ARES. (2) **Natural Reasoning** fine-tunes on the NaturalReasoning dataset (Yuan et al., 2026), serving as a strong off-the-shelf SFT baseline. (3) **Webscale** applies GRPO with binary rewards on reformatted QA pairs, following WebScale-RL (Cen et al., 2026), to isolate the effect of rubric-based rewards. (4) **ARES-SFT** fine-tunes on ARES-generated QA pairs using reference answers as targets, isolating the effect of rubric-based RL from the ARES data construction.

ARES-RL. We apply GRPO with rubric-based rewards (Equation 7) following Zhou et al. (2025). We train for 3 epochs with batch size 128, using VeRL (Sheng et al., 2025) as the training backend and following the hyper-parameters from RuscaRL Zhou et al. (2025). For the rubric judge J in Equation 7, we use Qwen3-32B (Yang et al., 2025) following Zhou et al. (2025).

Benchmarks. We evaluate on seven benchmarks covering knowledge reasoning, math, code generation, healthcare QA, writing quality, and instruction following: MMLU-Pro (Wang et al.,

Table 4: Main results on seven diverse benchmarks. **Bold** indicates the best result per column.

Method	MMLU-Pro	GSM8K	HumanEval+	MBPP+	HealthBench	WritingBench	IFEval	Avg
CPT	46.02	82.34	32.32	59.40	35.04	36.98	39.39	47.36
NaturalReasoning	47.96	81.50	32.93	61.15	32.94	36.77	28.15	45.91
Webscale	49.50	84.91	33.54	61.40	36.08	37.09	35.61	48.30
ARES-SFT	**50.56**	85.67	31.10	61.40	35.78	37.05	46.41	49.71
ARES-RL (ours)	49.36	**86.96**	**34.76**	**63.16**	**41.45**	**38.24**	**54.88**	**52.69**

2024b), GSM8K (Cobbe et al., 2021), HumanEval+ (Chen et al., 2021), MBPP+ (Austin et al., 2021), HealthBench (Arora et al., 2025), WritingBench (Wu et al., 2026), and IFEval (Zhou et al., 2023).

4.2 Main Results

Table 4 presents the main experimental results. We highlight several key findings:

ARES-RL Achieves the Best Average Performance. ARES-RL achieves the strongest average performance among the compared training methods, surpassing CPT by 5.33 points, SFT-baseline (NaturalReasoning) by 6.78 points, and binary-reward RL (Webscale) by 4.39 points. Notably, ARES-RL also outperforms ARES-SFT by 2.98 points while using the same ARES-generated data. This comparison isolates the benefit of the training objective: SFT mainly learns to imitate reference answers, whereas rubric-based RL optimizes candidate responses against multi-dimensional evaluation criteria. As a result, the policy can improve aspects of response quality that are not fully captured by a single reference answer. These gains suggest that rubric-based rewards provide richer learning signals than imitation-based training or binary outcome verification.

Performance on Multi-Dimensional Evaluation Benchmarks. The largest improvements appear on benchmarks where response quality depends on multiple evaluation dimensions. Compared with binary-reward RL, ARES-RL improves HealthBench from 36.08 to 41.45 (+5.37), IFEval from 35.61 to 54.88 (+19.27), and WritingBench from 37.09 to 38.24 (+1.15). These tasks are difficult to capture with a single binary outcome reward: healthcare responses require factuality, safety, and reasoning quality; instruction following depends on satisfying multiple constraints; and writing quality involves coherence, relevance, and style. Rubric-based rewards, as a family of reward designs, are therefore well suited to these settings because they decompose response quality into explicit criteria.

Performance on Verifiable Reasoning and Coding Tasks. Beyond multi-dimensional evaluation benchmarks, ARES-RL also performs strongly on automatically scored reasoning and coding tasks. It achieves the best GSM8K score (86.96, +2.05 over Webscale) and leads on both code generation benchmarks: HumanEval+ (34.76, +1.22 over Webscale) and MBPP+ (63.16, +1.76 over Webscale). This is notable because Math and Coding are the two smallest domains in ARES, containing only 930 and 1,631 examples, respectively (Table 2). Despite limited domain-specific data, ARES-RL achieves the best results on all three corresponding benchmarks, suggesting that the benefits of rubric-based RL over diverse domains can transfer to structured reasoning and coding tasks. The main exception is MMLU-Pro, where ARES-RL (49.36) is slightly below ARES-SFT (50.56). This may be because MMLU-Pro is a multiple-choice benchmark, for which supervised learning or exact-answer-oriented training provides a more direct signal than open-ended rubric-based optimization.

4.3 Analysis

Why Do Rubric Rewards Help Open-Ended Domains? The gains on HealthBench, IFEval, and WritingBench can be attributed to two factors. First, these benchmarks evaluate response quality along multiple dimensions that are difficult to capture with a single binary outcome reward. For example, a health response may be partially correct, context-dependent, or appropriately cautious; a rubric can separately evaluate medical accuracy, completeness of risk disclosure, treatment reasoning, and clarity of explanation. Similarly, instruction-following tasks often contain multiple constraints, such as format, length, keywords, language, and structural requirements, which can be naturally decomposed into individual rubric criteria. Second, rubric rewards provide a denser and more diagnostic reward signal for GRPO. A response that correctly identifies a diagnosis but inadequately explains treatment options can still receive partial credit for its correct components while being penalized for missing dimensions, guiding the policy toward more complete and reliable responses.

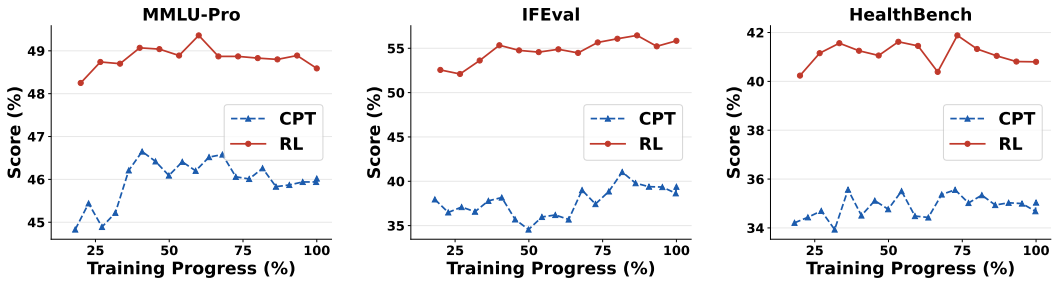


Figure 4: Per-benchmark comparison between CPT and ARES-RL.

Table 5: Ablation study on reward design. All variants use the same training dataset and GRPO setup. **Bold** indicates the best result per column.

Reward Strategy	MMLU-Pro	GSM8K	HumanEval+	MBPP+	HealthBench	WritingBench	IFEval	Avg
Blind Judge	47.56	85.97	33.54	62.16	44.43	38.84	34.20	49.53
General Rubric	49.00	86.05	**34.76**	61.90	45.97	**39.19**	45.66	51.79
Reference Answer	49.01	85.67	29.88	61.40	**50.13**	37.12	10.54	46.25
ARES-RL	**49.36**	**86.96**	**34.76**	**63.16**	41.45	38.24	**54.88**	**52.69**

From Passive Token Prediction to Targeted Reward Optimization. Figure 4 compares CPT and ARES-RL using the same source-document pool. CPT learns from raw documents through next-token prediction, whereas ARES-RL converts them into QA-rubric instances and optimizes the policy with rubric-based rewards. The gap is modest on knowledge-intensive or automatically scored benchmarks, but becomes much larger on open-ended tasks, where ARES-RL improves over CPT by +6.41 on HealthBench and +15.49 on IFEval. This suggests that targeted rubric supervision is more effective than token-level prediction for capabilities requiring nuanced judgment, constraint satisfaction, and multi-dimensional response quality. By assigning credit and penalties across multiple criteria, ARES-RL turns raw documents into more focused post-training supervision.

4.4 Ablation Study

To understand the contribution of question-specific rubric design, we compare ARES-RL against three alternative reward strategies, all using the same training dataset and GRPO setup:(1) **Blind Judge**: an LLM judge evaluates response quality holistically without any rubric or reference answer, producing a direct scalar reward. (2) **General Rubric**: a fixed set of generic rubric criteria (e.g., *accuracy*, *completeness*, *clarity*) is applied uniformly to all questions, rather than question-specific criteria. (3) **Reference Answer**: the judge evaluates by comparing the response against the reference answer, assigning a similarity-based score.

Table 5 reports the ablation results. We emphasize that no reward strategy dominates every individual benchmark. Different reward designs encode different inductive biases: reference-based rewards can be strong when benchmark evaluation aligns with semantic similarity or content coverage relative to a canonical answer, while general rubrics can be competitive when broad quality dimensions are sufficient. Therefore, we interpret the ablation primarily in terms of cross-benchmark robustness rather than per-benchmark dominance. Under this view, ARES-RL achieves the best average score and is top-performing or tied on five out of seven benchmarks, while avoiding the severe instability observed in reference-based evaluation.

Structured Rubrics Outperform Holistic Judgments on Average. Both rubric-based approaches, ARES-RL (52.69) and General Rubric (51.79), outperform the Blind Judge baseline (49.53) on average. This indicates that decomposing evaluation into explicit criteria provides a more effective reward signal than asking an LLM judge to produce a single holistic quality score. Rather than relying on an undifferentiated judgment, rubric evaluation makes the reward decision more transparent and provides more actionable feedback for policy optimization. The improvement of General Rubric over Blind Judge also suggests that even simple structured criteria can help stabilize reward estimation.

Question-Specific Rubrics Improve Cross-Task Robustness. ARES-RL outperforms the General Rubric baseline by 0.90 points on average, with the largest gain on IFEval (+9.22). However, question-

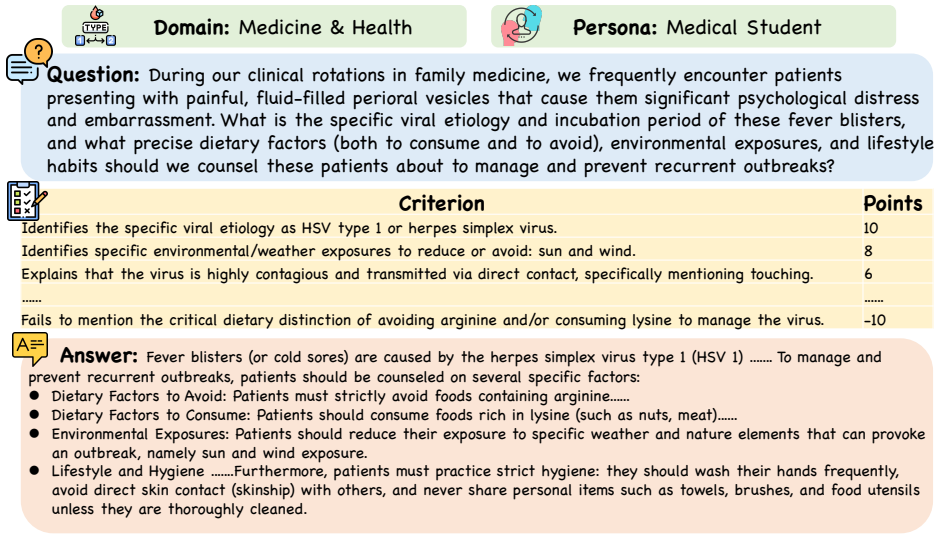


Figure 5: A representative ARES-generated instance from the Medicine & Health domain.

specific rubrics do not uniformly improve every benchmark: General Rubric performs better on HealthBench and WritingBench, suggesting that generic quality dimensions can already be effective for some broad evaluation settings. The advantage of ARES-RL is most evident on tasks whose evaluation requirements vary substantially across examples. For instance, instruction-following tasks may require different combinations of formatting, length, keyword, language, and structural constraints, which are difficult to express with a fixed set of generic criteria such as accuracy or clarity. Question-specific rubrics address this limitation by aligning the reward criteria with each generated question and reference answer, leading to better average performance and stronger robustness.

Reference-Based Evaluation Is Strong but Unstable. The Reference Answer reward achieves the highest HealthBench score (50.13), suggesting that reference-based comparison can be effective when the benchmark favors content coverage close to a canonical answer. This is plausible for health-related QA, where high-quality responses often require specific factual content, risk disclosure, and cautious recommendations that may be well captured by a strong reference answer. However, this reward design is much less stable across task types: it drops sharply on IFEval (10.54), where success depends less on matching a single target answer and more on satisfying explicit constraints such as formatting, length, keywords, and structural requirements. Consequently, Reference Answer obtains the lowest average score despite its strong HealthBench performance. This highlights the limitation of using a single reference answer as the reward target for heterogeneous open-ended RL training.

4.5 Case Study

Figure 5 shows a representative ARES instance from the medicine domain. The rubric decomposes an open-ended clinical question into weighted criteria: positive weights reward key knowledge dimensions (e.g., HSV-1 identification, +10; environmental triggers, +8), while negative weights penalize critical omissions (e.g., missing the arginine/lysine distinction, -10). Unlike a binary reward that would assign zero to any incomplete response, this structure provides partial credit proportional to the importance of satisfied criteria, guiding the policy toward more comprehensive outputs.

5 Conclusion

We presented ARES, a scalable pipeline that automatically converts raw pretraining documents into RL-ready instances with question-specific weighted rubrics. By co-generating questions, reference answers, and multi-dimensional rubrics in a single pass, ARES produces 100K rubric-annotated instances across ten domains at negligible additional cost. Experiments on seven benchmarks show that ARES-RL consistently outperforms CPT, SFT, and binary-reward RL, with the largest gains on open-ended tasks where multi-dimensional rubric rewards provide structured supervision that binary verification cannot offer.

AI USE STATEMENT

This work uses large language models in two capacities. First, as described in the main paper, LLMs serve as core components of the ARES pipeline: they perform document filtering, domain and persona identification, rubric-augmented question–answer generation, quality validation, and rubric-based reward judging during RL training. These uses are detailed in the method and experiment sections. Second, we used LLM-based assistants to polish the writing of this manuscript, including grammar correction, sentence rephrasing, and stylistic refinement. All LLM-generated or LLM-edited text has been manually reviewed and revised by the authors. We take full responsibility for the accuracy and integrity of every claim, result, and statement in this paper.

ETHICS STATEMENT

Automated questions, answers, and rubrics can inherit factual errors and biases from source documents and generator models. LLM judges may introduce further model-dependent biases. As discussed in Appendix B, human oversight and validation are particularly important for sensitive applications, including healthcare. Benchmark performance alone does not establish suitability for clinical use. The external assets and their reported licenses are listed in Appendix H; source provenance, privacy, and applicable terms should be considered when using or releasing derived data.

REPRODUCIBILITY STATEMENT

The methodology describes the ARES pipeline and reward formulation. Appendices C and D provide implementation and rubric-validation details; Appendix E describes training configurations and compute resources; Appendix F provides evaluation details; and Appendix G contains representative prompt templates.

REFERENCES

Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Maria Mazzone, David Randolph, Mohammad Mahmoudi Meymand, et al. Prbench: Large-scale expert rubrics for evaluating high-stakes professional reasoning. *arXiv preprint arXiv:2511.11562*, 2025.

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=76zq8Wkl6Z.

Zhepeng Cen, Haolin Chen, Shiyu Wang, Zuxin Liu, Zhiwei Liu, Ding Zhao, Caiming Xiong, Huan Wang, and Weiran Yao. Webscale-RL: Automated data pipeline for scaling RL data to pretraining levels. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hOJS9RB1NU.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, january 2025. *URL https://github.com/huggingface/open-r1*, 7, 2025.

Victor Gallego. Configurable preference tuning with rubric-guided synthetic data. In *2nd Workshop on Models of Human Feedback for AI Alignment*, 2025. URL https://openreview.net/forum?id=seA8en4ujl.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M. Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c1bTcrDmt4.

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

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

Hynek Kydlíček, Guilherme Penedo, and Leandro von Werra. Finepdfs. https://huggingface.co/datasets/HuggingFaceFW/finepdfs, 2025.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Mingjie Liu, Shizhe Diao, Jian Hu, Ximing Lu, Xin Dong, Hao Zhang, Alexander Bukharin, Shaokun Zhang, Jiaqi Zeng, Makesh Narsimhan Sreedhar, et al. Scaling up rl: Unlocking diverse reasoning in llms via prolonged training. *arXiv preprint arXiv:2507.12507*, 2025.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=YPsJha5HXQ.

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

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Soumye Singhal, Jiaqi Zeng, Alexander Bukharin, Yian Zhang, Gerald Shen, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Yoshi Suhara, Akhiad Bercovich, Itay Levy, Izik Golan, Mohammed Dabbah, Ran El-Yaniv, Somshubra Majumdar, Igor Gitman, Evelina Bakhturina, Jimmy J. Zhang, Bor-Yiing Su, Guyue Huang, Izzy Putterman, Mostofa Patwary, Oluwatobi Olabiyi, Olivier Delalleau, Bryan Catanzaro, Boris Ginsburg, Oleksii Kuchaiev, and Tugrul Konuk. Llama-nemotron: Efficient reasoning models. In *The Exploration in AI Today Workshop at ICML 2025*, 2025. URL https://openreview.net/forum?id=ev1xpo9mbI.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024b.

Maurice Weber, Daniel Y Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37: 116462–116492, 2024.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Wang Zijia, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. Writingbench: A comprehensive benchmark for generative writing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=Pkskg9drDQ.

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

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E Weston, and Xian Li. Naturalreasoning: Reasoning in the wild with 2.8m challenging questions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=CF2OfwkYdM.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, et al. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general llm reasoning. *arXiv preprint arXiv:2508.16949*, 2025.

A LIMITATIONS

Due to computational resource constraints, our experiments are conducted exclusively on Qwen3-4B-Base. This limits the scope of our findings. We are unable to validate whether ARES scales to larger models (e.g., 70B parameters), where rubric-based rewards may yield different dynamics due to increased model capacity and stronger baseline reasoning. Second, the rubric judge (Qwen3-32B) is constrained by our available GPU memory; using a larger or more capable judge model could improve reward signal quality but is infeasible within our current setup. We believe these are engineering rather than fundamental limitations, and we expect the core findings—that question-specific rubric rewards outperform binary and holistic alternatives—to hold at larger scales.

B BROADER IMPACT

ARES enables RL training on open-ended tasks that were previously inaccessible to verifiable-reward methods, broadening LLM capabilities in domains such as healthcare, education, and professional writing. On the positive side, by automating rubric synthesis from pretraining documents, ARES lowers the barrier to constructing diverse RL training data, making rubric-based post-training accessible to research groups with limited annotation budgets. However, this broader applicability also carries risks. Automated rubric generation may encode biases present in the source documents or the LLMs used for synthesis, potentially reinforcing stereotypes or inaccuracies. The use of LLM judges for reward assignment introduces a further layer of model-dependent bias. We encourage practitioners to carefully validate ARES-generated data and rubrics, and to combine automated rubric evaluation with human oversight when deploying models trained with this pipeline.

C ADDITIONAL IMPLEMENTATION DETAILS

This section provides additional details on the ARES data construction and training setup. The main paper is self-contained; the material below is intended to clarify implementation choices and reproducibility details.

C.1 PIPELINE INPUTS AND OUTPUTS

ARES takes raw pretraining documents as input and produces GRPO-ready training instances. Each accepted instance contains a user-facing prompt, an instance-specific rubric, a reference answer, and metadata describing the source domain and persona. In the training format, the prompt is represented as a single-turn conversation, the rubric is stored as a list of natural-language criteria with non-zero integer weights, and the reference answer is retained as an ideal completion for analysis and supervised baselines. The metadata is not used as a reward during RL training. It is retained for auditing, filtering, and dataset analysis.

C.2 DOMAIN AND PERSONA ASSIGNMENT

The domain classifier assigns each document to one of ten categories: Math, Coding, Medicine & Health, Technology & Engineering, Social Science, Natural Science, Travel & Lifestyle, Commerce & Economics, Education, and Other. If a document plausibly spans multiple categories, the classifier selects the most relevant primary domain. Persona conditioning is used to diversify the resulting questions: each document can be paired with up to three audience profiles, and the generator produces questions from the selected persona's perspective.

This design separates topical coverage from user intent. For example, two documents in the same medical domain can yield different questions when framed for a medical professional, a student, or a caregiver. Conversely, documents from different domains can share a similar persona while still requiring different rubric criteria.

D Rubric Generation and Validation

D.1 Rubric Format

Each rubric consists of a list of criteria $\mathcal{R} = \{(c_i, w_i)\}_{i=1}^{N}$, where c_i is a natural-language criterion and w_i is a non-zero integer weight. Positive weights reward desirable properties of an answer, while negative weights penalize important failure modes. Rubrics are generated jointly with the question and reference answer so that all three objects are grounded in the same source document.

The generator is instructed to write semantic criteria rather than string-match rules. A criterion should describe the concept, fact, reasoning step, constraint, or failure mode to be checked; it should not require reproducing a particular surface form unless the task itself is about formatting or exact wording.

D.2 Complexity-Adaptive Criteria

The number of criteria is adapted to question complexity. Easy questions use a small number of criteria focused on the central fact or concept. Medium questions use more criteria to cover multiple related aspects. Hard questions use the most criteria, including separate checks for sub-conditions, exceptions, and common failure modes. Across all complexity levels, the pipeline requires both positive and negative criteria so that the reward can distinguish partially correct answers from answers that omit or distort critical information.

D.3 Validation Checks

After generation, each rubric is validated before being serialized for training. The validator checks that every criterion has the required fields, that weights are non-zero and within the allowed range, that the rubric contains enough criteria for the corresponding complexity level, and that the rubric includes negative criteria. Criteria with malformed fields are rejected or repaired when the correction is unambiguous; examples that remain structurally invalid are not used for training.

The verifier also checks the associated Q&A pair before rubric validation. A generated instance must pass factual consistency, information-leakage, and self-containedness checks. This prevents the RL stage from receiving prompts whose answers are unsupported by the source document, trivially revealed by the question, or impossible to interpret without hidden context.

E Training Details

E.1 Compared Methods

All methods in the main comparison use Qwen3-4B-Base as the foundation model. Continual pretraining optimizes next-token likelihood on the same source documents used by ARES. Natural Reasoning fine-tunes on an off-the-shelf reasoning dataset. Webscale follows binary-reward RL on short-answer QA pairs. ARES-SFT fine-tunes on the generated ARES questions and reference answers. ARES-RL uses the same generated questions but optimizes rubric-based rewards.

This comparison is designed to isolate three factors: whether the model sees the source documents as language modeling data or as question-answer tasks, whether the training signal is imitation or reinforcement learning, and whether the reward is binary or multi-dimensional.

E.2 GRPO Configuration

ARES-RL uses GRPO with group-normalized advantages. For each prompt, the policy samples a group of candidate responses, scores each response with the associated rubric, and updates the policy using the clipped GRPO objective described in the main paper. The experiments use $G = 8$ rollouts per question, clipping threshold $\varepsilon = 0.2$, KL penalty $\beta = 0.01$, batch size 128, and 3 epochs with VeRL as the training backend.

Table 6: Training hyperparameters for NaturalReasoning-SFT/ARES-SFT and CPT baselines.

Setting	NaturalReasoning-SFT / ARES-SFT	CPT
Base model	Qwen3-4B-Base	Qwen3-4B-Base
Data mode	prompt + response	text
GPUs	8	8
Max length	2048	2048
Micro batch / GPU	2	4
Global batch size	128	256
Epochs	3	3
Learning rate	2e-5	5e-5
Warmup ratio	0.05	0.1
Scheduler	cosine	cosine
Betas	[0.9, 0.95]	[0.9, 0.95]
Weight decay	0.01	0.01
Grad clip	1.0	1.0
Strategy	FSDP2	FSDP2
Gradient ckpt	enabled	enabled
LoRA	disabled	disabled
Seed	42	42

Table 7: Main benchmark results with 95% confidence intervals (structured tasks).

Method	MMLU-Pro	GSM8K	HumanEval	MBPP
CPT	46.02 [45.11, 46.92]	82.34 [80.29, 84.39]	32.32 [25.03, 39.66]	59.40 [54.63, 64.15]
Natural Reasoning	47.96 [47.06, 48.86]	81.50 [79.46, 83.55]	32.93 [25.67, 40.30]	61.15 [56.36, 65.88]
Webscale	49.50 [48.62, 50.38]	84.91 [82.95, 86.81]	33.54 [26.23, 40.86]	61.40 [56.64, 66.16]
ARES-SFT	50.56 [49.67, 51.46]	85.67 [83.69, 87.55]	31.10 [23.82, 38.46]	61.40 [56.63, 66.16]
ARES-RL (ours)	49.36 [48.48, 50.25]	86.96 [85.13, 88.77]	34.76 [27.45, 42.08]	63.16 [58.37, 67.89]

The rubric score used for RL is the weighted sum of binary criterion judgments. The judge evaluates each criterion independently against the sampled answer, and the resulting weighted score is used as the scalar reward for GRPO.

E.3 SFT AND CPT HYPERPARAMETERS

Table 6 lists the training hyperparameters for the NaturalReasoning-SFT/ARES-SFT and CPT settings.

E.4 COMPUTE RESOURCES

All training experiments are conducted on a single node equipped with 8× NVIDIA A100 80GB GPUs, 224 CPU cores, and 2048 GB of system memory. The total compute budget across all training runs amounts to approximately 140 GPU-hours.

F EVALUATION DETAILS

The evaluation suite covers seven benchmarks: MMLU-Pro, GSM8K, HumanEval+, MBPP+, HealthBench, WritingBench, and IFEval. Structured tasks use their standard automatic evaluation protocols, including exact or normalized answer matching for math and test-case execution for code. Open-ended tasks are evaluated with their benchmark-specific rubric or judge protocols, which makes them especially relevant for testing whether rubric-based RL improves multi-dimensional response quality.

We report the unweighted average over the seven benchmark scores in the main tables. This average is intended as a compact summary of broad capability rather than a claim that all benchmarks have identical practical importance.

To quantify evaluation uncertainty without changing the reported point estimates, we compute 95% confidence intervals. The intervals are shown in Tables 7–8 and Tables 9–10.

Table 8: Main benchmark results with 95% confidence intervals (open-ended tasks).

Method	HealthBench	WritingBench	IFEval	Avg
CPT	35.04 [34.07, 36.04]	36.98 [36.08, 37.88]	39.39 [36.02, 42.77]	47.36 [46.01, 48.76]
Natural Reasoning	32.94 [32.01, 33.88]	36.77 [36.05, 37.50]	28.15 [25.06, 31.30]	45.91 [44.55, 47.28]
Webscale	36.08 [35.12, 37.03]	37.09 [36.20, 38.01]	35.61 [32.38, 38.87]	48.30 [46.94, 49.66]
ARES-SFT	35.78 [34.77, 36.81]	37.05 [36.12, 38.02]	46.41 [42.94, 49.93]	49.71 [48.34, 51.08]
ARES-RL (ours)	41.45 [40.34, 42.56]	38.24 [37.31, 39.21]	54.88 [51.34, 58.31]	52.69 [51.31, 54.07]

Table 9: Reward-design ablation results with 95% confidence intervals (structured tasks).

Reward Strategy	MMLU-Pro	GSM8K	HumanEval	MBPP
Blind Judge	47.56 [46.67, 48.45]	85.97 [84.08, 87.80]	33.54 [26.76, 40.79]	62.16 [57.35, 66.87]
General Rubric	49.00 [48.12, 49.88]	86.05 [84.16, 87.87]	34.76 [27.50, 42.13]	61.90 [57.16, 66.68]
Reference Answer	49.01 [48.11, 49.91]	85.67 [83.77, 87.56]	29.88 [23.17, 37.19]	61.40 [56.68, 65.95]
ARES-RL	49.36 [48.46, 50.23]	86.96 [85.15, 88.71]	34.76 [27.46, 42.09]	63.16 [58.37, 67.90]

G Prompt Templates

We include representative prompt excerpts used by the ARES data pipeline. The prompts below are lightly compressed for readability while preserving the operational instructions used in each stage.

G.1 Document Filtering Prompt

Stage 1: Document Filtering

```
You are a data quality analyst. You will be given material from
diverse sources. Our goal is to generate question-answer pairs WITH
evaluation rubrics from the material.
Identify whether the material is qualified for BOTH Q&A generation AND
rubric creation based on these criteria:
Content Quality:
• The material is informative and factually accurate.
• It contains specific, verifiable information.
• It is self-contained enough to formulate clear questions.
Suitable for Rubric Generation:
• It contains concrete facts or concepts that can be evaluated
  objectively.
• It allows distinguishing between correct and incorrect answers.
• It has sufficient detail to create meaningful evaluation criteria.
• It provides clear standards such as numbers, dates, definitions, or
  procedures.
Unsuitable Materials: Pure opinion pieces without factual basis;
overly vague or ambiguous content; lists without context; promotional
content without informative value; content with unclear or disputed
facts.
Output JSON: {"thought": "...", "qualified": "Y or N"}
```

G.2 Domain and Persona Identification Prompt

Stage 2: Domain & Persona Identification

```
You are a helpful data analyst. You will be given material from a
website which can come from very diverse sources and may not be well
structured. Our final goal is to generate a question-answer pair from
the material. In this stage, identify the domain and persona of the
material.
Instructions:
```

Table 10: Reward-design ablation results with 95% confidence intervals (open-ended tasks).

Reward Strategy	HealthBench	WritingBench	IFEval	Avg
Blind Judge	44.43 [43.25, 45.59]	38.84 [37.94, 39.77]	34.20 [30.81, 37.69]	49.53 [48.15, 50.91]
General Rubric	45.97 [44.84, 47.14]	39.19 [38.22, 40.16]	45.66 [42.13, 49.11]	51.79 [50.39, 53.21]
Reference Answer	50.13 [48.86, 51.42]	37.12 [36.33, 37.93]	10.54 [8.22, 12.90]	46.25 [44.97, 47.57]
ARES-RL	41.45 [40.34, 42.55]	38.24 [37.31, 39.19]	54.88 [51.49, 58.32]	52.69 [51.32, 54.04]

- Choose one domain from: Math, Technology & Engineering, Coding, Social Science, Natural Science, Travel & Lifestyle, Commerce & Economics, Medicine & Health, Education, Other.
- If multiple domains apply, choose the most relevant primary domain.
- The persona is the intended audience of the material.
- If multiple personas apply, list up to three personas.

Output JSON: {"thought": "...", "domain": "...", "persona": "..."}

G.3 Rubric-Augmented QA Generation Prompt

Stage 3: Rubric-Augmented QA Co-Generation

You will be given material from diverse sources. Your task is to: (1) Generate a question-answer pair from the material. (2) Generate evaluation rubrics based on question complexity.

Question & Answer Generation:

- Generate from the persona's perspective.
- Base the question and answer ONLY on the material content.
- The question must be clear and verifiable.
- The answer should be comprehensive based on the material depth.
- Add necessary background context because the material will not be provided during training.
- Never leak the answer in the question.

Evaluation Rubrics:

- Simple questions: 4-6 criteria for a single fact or concept.
- Medium questions: 6-10 criteria for multiple related points.
- Complex questions: 10-15 criteria for multifaceted answers with conditions, exceptions, or failure modes.

Scoring:

- Positive weights reward desirable answer properties.
- Negative weights penalize critical omissions, factual errors, or misleading statements.
- Each criterion should check one specific aspect.
- Scores should reflect the importance of that aspect in context.

Semantic Rubric Guidelines: Reference concrete concepts, facts, numbers, terms, or constraints from the material. Do not rely on exact string matching unless the task itself requires exact wording. Use negative criteria for common errors or important missing information.

Output JSON: {"thought": "...", "question": "...", "answer": "...", "rubrics": [{"criterion": "...", "points": 9}, {"criterion": "...", "points": 7}, {"criterion": "...", "points": -8}]}

G.4 Q&A and Rubric Suitability Checking Prompt

Stage 4–5: Quality & Rubric Validation

```
You are a data quality checker for Q&A pairs that will be used to
generate evaluation rubrics. Check the Q&A pair quality and assess
its suitability for rubric generation.
Basic Q&A Checks:
• Context sufficiency: the question should be self-contained and
  should not require seeing the original material.
• Answer correctness: the answer must be factually correct according
  to the material and must not add unsupported information.
• Information leakage: the question should not explicitly reveal the
  answer.
Rubric Suitability:
• Determine whether the answer contains concrete, verifiable facts.
• Determine whether multiple aspects can be objectively evaluated.
• Determine whether clear standards exist for good and bad responses.
• Reject examples that are too short, purely subjective, vague, yes/no
  only, or lack specific details.
Output JSON: {"thought": "...", "has_context": "Y/N",
"answer_correctness": "Y/N", "info_leakage": "Y/N",
"question_complexity": "simple/medium/complex", "answer_detail_level":
"low/medium/high", "rubric_suitable": "Y/N"}
```

H Licenses for Existing Assets

We list the licenses of all major external assets used in this work. All assets are used in compliance with their respective terms.

Table 11: Licenses for existing assets used in this work.

Asset	Type	License
Pretraining Corpora		
DCLM (Li et al., 2024)	Dataset	MIT
FineWeb-Edu (Penedo et al., 2024)	Dataset	ODC-By-1.0
FinePDFs (Kydlíček et al., 2025)	Dataset	ODC-By-1.0
Models		
Qwen3-4B-Base (Yang et al., 2025)	Model	Apache-2.0
Qwen3-32B (Yang et al., 2025)	Model	Apache-2.0
Training Framework		
VeRL (Sheng et al., 2025)	Code	Apache-2.0
Evaluation Benchmarks		
MMLU-Pro (Wang et al., 2024b)	Benchmark	MIT
GSM8K (Cobbe et al., 2021)	Benchmark	MIT
HumanEval+ (Chen et al., 2021)	Benchmark	Apache-2.0
MBPP+ (Austin et al., 2021)	Benchmark	Apache-2.0
HealthBench (Arora et al., 2025)	Benchmark	MIT
WritingBench (Wu et al., 2026)	Benchmark	Apache-2.0
IFEval (Zhou et al., 2023)	Benchmark	Apache-2.0
Baseline Datasets		
NaturalReasoning (Yuan et al., 2026)	Dataset	CC-BY-NC-4.0
Webscale-RL (Cen et al., 2026)	Dataset	CC-BY-NC-4.0

I SUPPLEMENTARY EXPERIMENTS: BASELINES AND TRAINING SETUP

This section presents additional baseline comparisons, human assessments of data quality, cross-judge evaluations, and rubric ablations. We also examine training variability, performance across policy checkpoints, and the cost of synthesizing the ARES dataset.

Baseline data and initialization. NaturalReasoning-100K and Webscale-style RL each use 101,847 examples. NaturalReasoning-100K is an external QA subset matched in size and token count, while Webscale-style RL uses short-answer QA from the ARES source pool. All methods initialize independently from Qwen3-4B-Base, and ARES-RL trains directly from this checkpoint (Table 12).

Table 12: Training data, dataset sizes, and epochs for the baseline and ARES methods.

Method	Training data	Size	Epochs
Direct Base	No additional data	0	0
CPT	Same source documents as ARES	$\sim$0.1B tokens	3
NaturalReasoning-100K	Size-matched external QA subset	101,847	3
Webscale-style RL	Short-answer QA from same sources	101,847	3
ARES-SFT	ARES questions and reference answers	101,847	3
ARES-RL	ARES questions and rubrics	101,847	3

Direct checkpoint evaluation. Table 13 compares seven-task means starting from the Direct Base score of 39.70. CPT and Webscale-style RL achieve 47.36 and 48.30, respectively, while ARES-SFT reaches 49.71 and ARES-RL reaches 52.69. ARES-RL has the highest mean among these methods, exceeding ARES-SFT by 2.98 points and Direct Base by 12.99 points.

Table 13: Seven-task average scores for the base checkpoint and post-training methods.

Method	Mean
Direct Base	39.70
CPT	47.36
Webscale-style RL	48.30
ARES-SFT	49.71
ARES-RL	52.69

Shared SFT settings. NaturalReasoning-100K and ARES-SFT use the same SFT recipe: maximum sequence length 2,048; global batch size 128; micro batch size 2 per GPU; three epochs; learning rate 2×10^{-5}; warmup ratio 0.05; cosine schedule; Adam betas $(0.9, 0.95)$; weight decay 0.01; gradient clipping 1.0; FSDP2; and gradient checkpointing. The main run uses seed 42, and additional three-seed results assess training variability. Appendix E.3 provides the corresponding training configuration.

Comparison design. CPT versus ARES-SFT compares raw-document training with task-oriented QA supervision. ARES-SFT versus ARES-RL uses the same ARES questions with different objectives and supervision. Webscale-style RL and ARES-RL compare short-answer QA with binary rewards against ARES questions with rubric-based rewards. The reward-form ablations further compare supervision strategies using identical ARES questions and the same judge.

J HUMAN AUDIT OF QUESTIONS, ANSWERS, AND RUBRICS

We audit 200 instances, stratified as 20 instances per domain. Each instance is independently assessed by two annotators, with disagreements adjudicated by a third. The annotation team consists of five PhD-level annotators with relevant backgrounds for the medical, mathematical, and coding examples. Eight candidate responses per question are also scored, yielding 1,600 responses for the agreement analysis.

Table 14: Question, reference-answer, and rubric quality in the 200-instance human audit.

Measure	Result
Question self-contained	97.0%
Reference answer fully supported	94.5%
Criterion factual validity	93.2%
Rubric key-aspect coverage	89.5%
Weight sign correctness	97.1%
Weight magnitude judged reasonable	88.4%
Ambiguous/redundant criterion rate	4.6%
Potentially unsafe criterion rate	0.4%

Table 15: Human–human and Qwen–human agreement on criterion assessments and total scores.

Comparison	Statistic
Human–human criterion agreement	$\kappa = 0.78$
Qwen–human criterion agreement	$\kappa = 0.72$
Qwen reward–human total score correlation	$\rho = 0.81$

Table 16: Audit error-rate summaries for the overall sample and the medical subset.

Error type	Overall	Medical subset
Unsupported reference answer	5.5%	7.5%
Incorrect criterion	6.8%	8.1%
Important aspect missing	10.5%	12.5%
Ambiguous/redundant criterion	4.6%	5.9%
Incorrect weight direction	2.9%	3.4%
Potentially unsafe instance	0.5%	2.5%

Audit results. Table 14 shows that 97.0% of questions are self-contained and 94.5% of reference answers are fully supported. Rubrics achieve 93.2% criterion factual validity and 89.5% key-aspect coverage, while weight signs and magnitudes are judged appropriate in 97.1% and 88.4% of assessments, respectively. Table 15 gives human–human criterion agreement of $\kappa = 0.78$, Qwen–human criterion agreement of $\kappa = 0.72$, and a reward–human score correlation of $\rho = 0.81$. These results indicate substantial agreement between the automated reward signal and human assessments. Table 16 summarizes complementary error categories, with higher rates in the medical subset across all listed categories. The pattern highlights factual support, coverage, and domain-specific review as useful priorities for data refinement.

K Cross-Judge Evaluation and Rubric-Weight Ablation

Independent rescoring and online judge replacement. We rescore the same 1,600 outputs using Qwen3-32B and Llama-3.1-70B-Instruct, obtaining criterion agreement of $\kappa = 0.69$ and reward correlation of $\rho = 0.79$. A separate RL run replaces the online reward judge with Llama while retaining the policy, GRPO setup, preprocessing, and evaluation. Its HealthBench score is 40.80, compared with 41.45 for the Qwen-judge run (Table 17).

Table 17: HealthBench scores in the supplementary cross-judge comparison.

Method / training reward judge	HealthBench
Webscale-style RL (supplementary)	37.09
ARES-RL / Qwen3-32B	41.45
ARES-RL / Llama-3.1-70B-Instruct	40.80

Judge comparison. Replacing the Qwen judge with Llama changes the HealthBench score by 0.65 points. Together with the agreement on the same candidate outputs, the similar downstream scores support the use of rubric-based rewards with either judge in this setting.

Weighted versus uniform criteria. The weight ablation keeps the data, policy, judge, and hyperparameters fixed while comparing uniform and weighted instance-specific rubrics (Table 18). Weighted rubrics increase the seven-task mean from 51.88 to 52.69 and IFEval from 52.40 to 54.88, gains of 0.81 and 2.48 points, respectively. This comparison supports assigning different importance to individual criteria when aggregating the reward.

Table 18: Uniform and weighted instance-specific rubric rewards under the same training setup.

Reward	Seven-task mean	IFEval
Uniform instance-specific rubric	51.88	52.40
Weighted instance-specific rubric	52.69	54.88

Task dependence of instance specificity. Blind Judge, General Rubric, and ARES-RL achieve seven-task means of 49.53, 51.79, and 52.69, respectively. ARES-RL improves over General Rubric by 0.90 points on average, with a gain of 9.22 on IFEval and differences of -4.52 on HealthBench and -0.95 on WritingBench. These task-level results show that the benefit of instance-specific criteria is strongest for explicit instruction constraints, while general rubrics remain competitive on medical and writing evaluations.

Separation of policy and evaluation roles. Rubrics and judge rationales remain within the reward-evaluation pipeline. The generator constructs the supervision, the verifier checks its quality, and the training judge scores policy outputs against the criteria. This separation keeps the policy focused on answering the task, while cross-judge evaluation examines the consistency of the resulting reward assessments.

L RaR Comparison and Additional Policy Checkpoint

RaR training comparison. The RaR baseline uses 18.3K Science and 17.9K Medicine examples, totaling 36.2K, with Qwen3-4B-Base and the same training framework and seven-benchmark suite. Table 19 compares the resulting systems using their respective datasets and includes an additional Qwen2.5-7B-Instruct experiment. ARES-RL exceeds RaR on six of seven benchmarks and achieves a higher seven-task mean, while RaR scores slightly higher on MMLU-Pro.

Table 19: RaR and ARES-RL results across seven benchmarks. The 4B runs use Qwen3-4B-Base, and the 7B run uses Qwen2.5-7B-Instruct.

Benchmark	4B + RaR	4B + ARES-RL	7B-Instruct + ARES-RL
MMLU-Pro	49.56	49.36	56.70
GSM8K	84.52	86.96	92.00
HumanEval+	32.56	34.76	74.50
MBPP+	60.25	63.16	68.20
HealthBench	36.44	41.45	36.41
WritingBench	37.77	38.24	37.16
IFEval	50.89	54.88	76.00
Mean	50.28	52.69	62.99

Policy-checkpoint comparison. The 7B experiment uses the same ARES data, reward pipeline, and evaluation suite with Qwen2.5-7B-Instruct as the policy initialization. This run scores higher than the 4B run on MMLU-Pro, GSM8K, HumanEval+, MBPP+, and IFEval, while the 4B run scores higher on HealthBench and WritingBench. The results demonstrate the application of ARES-RL to both a base checkpoint and an instruction-tuned checkpoint, with task-dependent performance across the two configurations.

Published checkpoint comparisons. Table 20 places published Qwen2.5-7B-Instruct checkpoint scores alongside the corresponding ARES-RL results. These cross-study reference values provide context for the 7B experiment. The ARES-RL scores are higher on all four listed benchmarks, with numerical differences of 0.40 on MMLU-Pro and GSM8K, 12.81 on HealthBench, and 4.80 on IFEval.

Table 20: Cross-study comparison of published Qwen2.5-7B-Instruct checkpoint scores and ARES-RL results.

Benchmark	Published 7B baseline	7B + ARES-RL	Difference
MMLU-Pro	56.30	56.70	+0.40
GSM8K	91.60	92.00	+0.40
HealthBench	23.60	36.41	+12.81
IFEval	71.20	76.00	+4.80

M Statistical Analysis and Task-Level Attribution

Paired evaluation and training seeds. Table 21 compares ARES-RL and Webscale-style RL using paired bootstrap evaluation on shared examples. ARES-RL improves the seven-task mean by 4.39 points, with the largest task-level gains on IFEval and HealthBench. The corresponding paired p-values summarize evaluation-sample uncertainty, while Table 22 examines variation across three training seeds.

Table 21: Paired-bootstrap comparisons between ARES-RL and Webscale-style RL. Differences are ARES-RL minus Webscale-style RL.

Benchmark	Difference	Paired p
MMLU-Pro	-0.14	0.81
GSM8K	$+2.05$	0.03
HumanEval+	$+1.22$	0.44
MBPP+	$+1.76$	0.18
HealthBench	$+5.37$	< 0.001
WritingBench	$+1.15$	0.08
IFEval	$+19.27$	< 0.001
Seven-task mean	$+4.39$	< 0.001

Table 22: Mean and standard deviation of the seven-task average across three training seeds.

Method	Mean $\pm$ standard deviation
Webscale-style RL	48.34 ± 0.25
ARES-SFT	49.73 ± 0.21
ARES-RL	52.64 ± 0.27

Across three seeds, ARES-RL achieves 52.64 ± 0.27, compared with 49.73 ± 0.21 for ARES-SFT and 48.34 ± 0.25 for Webscale-style RL, preserving the ordering of the main comparison. The paired results give $p < 0.001$ for the seven-task mean, HealthBench, and IFEval, and $p = 0.03$ for GSM8K. Together, the paired evaluation and repeated-training results characterize the performance gains at both the evaluation-sample and training-run levels.

Separating QA/SFT and RL effects. Table 23 compares CPT, ARES-SFT, and ARES-RL on math and code tasks. On GSM8K, ARES-SFT contributes 3.33 points of the 4.62-point CPT-to-ARES-RL difference, with a further 1.29-point gain from SFT to RL. On HumanEval+, ARES-SFT scores below CPT, while ARES-RL exceeds both. On MBPP+, performance increases from 59.40 with CPT to 61.40 with ARES-SFT and 63.16 with ARES-RL. The comparisons show how task-oriented supervision and rubric-based RL contribute differently across benchmarks.

Table 23: Math and code performance under CPT, ARES-SFT, and ARES-RL.

Benchmark	CPT	ARES-SFT	ARES-RL
GSM8K	82.34	85.67	86.96
HumanEval+	32.32	31.10	34.76
MBPP+	59.40	61.40	63.16

N SYNTHESIS COST AND OPEN-MODEL PIPELINE

The synthesis pipeline processes approximately 142K candidates and accepts 101,847 instances, yielding a 71.6% retention rate. Quality and rubric suitability checks are combined in one call, while structural checks, conversion, and deduplication are performed locally. Table 24 summarizes API calls, token usage, and estimated synthesis costs, including retries.

Table 24: Approximate API calls, token usage, and historical cost estimates for ARES data synthesis.

Stage or resource	Amount
Initial candidates	∼142K
Accepted instances	101,847
Retention	71.6%
Document filtering calls	∼142K
Domain/persona calls	∼125K
QA–rubric generation calls	∼125K
Joint validation calls	∼123K
Total API calls including retries	∼529K
Input tokens	∼0.67B
Output tokens	∼0.17B
Added calls from retries	∼2.7%
Standard API cost	∼US$4.3K
Batch API cost	∼US$2.1K
Standard cost per 1,000 accepted instances	∼US$42
Batch cost per 1,000 accepted instances	∼US$21

The four pre-retry stages total approximately 515K calls, increasing to approximately 529K with a 2.7% retry overhead. Estimated synthesis costs are approximately US$4.3K through the standard API and US$2.1K through the batch API, equivalent to about US$42 and US$21 per 1,000 accepted instances. Active quality filtering accounts for most candidate rejections, linking the final retention rate to the pipeline's validation criteria.

Open generator/verifier trial. A 10K-example trial uses Qwen3-235B-A22B-Instruct and Llama-3.1-70B as an open generator/verifier alternative. This configuration retains 65.8% of candidates, compared with 71.6% for the Gemini-and-Claude pipeline (Table 25). The trial demonstrates that the synthesis and validation workflow can operate with open models and produce instances that pass the pipeline checks.

Table 25: Validation retention for proprietary and open generator/verifier pipelines. The open-model trial uses 10K examples.

Generator / verifier pipeline	Retention
Gemini + Claude	71.6%
Qwen3-235B-A22B-Instruct + Llama-3.1-70B	65.8%

O Evaluation Protocols and Data Release Plan

IFEval metric. IFEval is evaluated using strict prompt-level accuracy. A response satisfies a prompt when it meets all associated instruction constraints under the strict evaluation rules. This metric summarizes the ability to follow the complete set of requirements in each prompt.

Contamination screening. Before training, we screen the generated data with the `lm-eval-harness` decontamination workflow. The procedure normalizes generated questions and answers, checks n-gram overlap against all seven evaluation sets, and removes potentially overlapping instances. It removes 113 candidates, approximately 0.11%, and the benchmark results use the filtered data. This benchmark-oriented screening complements source-document deduplication during data construction.

Planned dataset split. The planned release partitions source-document clusters into 96,847 training, 2,500 validation, and 2,500 test instances. Grouping by source cluster keeps near-duplicate persona variants derived from the same documents within a single partition.

Release contents. We plan to release all 101,847 instances, the six-stage synthesis pipeline, full uncompressed prompts, and training and evaluation configurations. The release plan also includes source corpus and document identifiers, the open-generator configuration, contamination-screening scripts, and implementation details for generation parameters, retry logic, and parsing.

Intended use. ARES constructs post-training supervision from raw domain documents by jointly generating tasks, reference answers, and instance-specific rubrics. The resulting collection supports QA-based SFT and rubric-based RL within the same data-construction framework. Joint document-grounded synthesis provides a practical route from domain corpora to both task supervision and structured reward criteria.