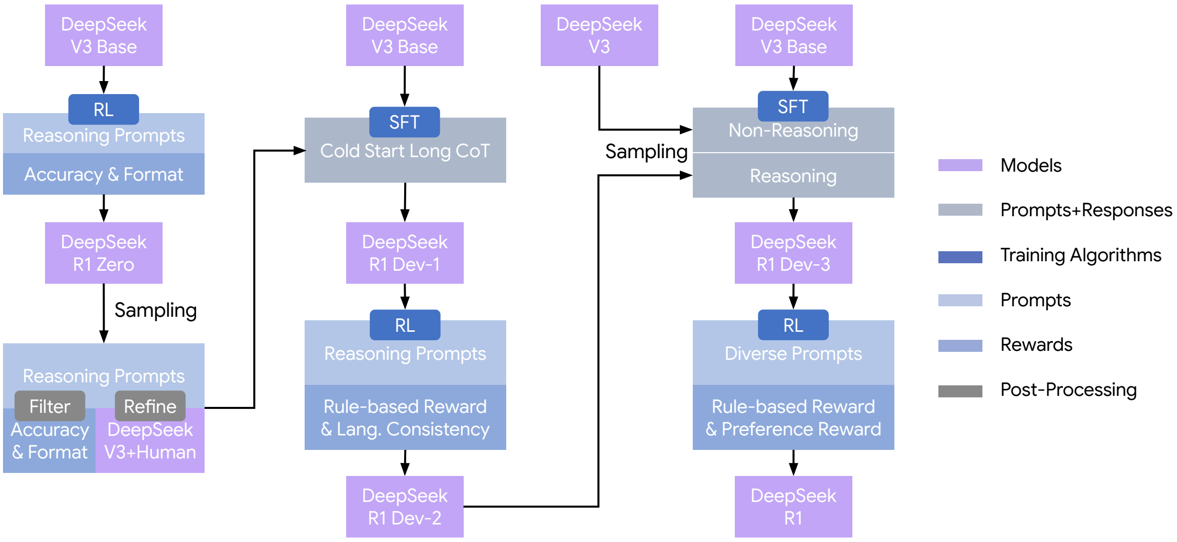

Figure 2 | The multi-stage pipeline of DeepSeek-R1. A detailed background on DeepSeek-V3 Base and DeepSeek-V3 is provided in Supplementary A.1. The models DeepSeek-R1 Dev1, Dev2, and Dev3 represent intermediate checkpoints within this pipeline.

3. DeepSeek-R1

Although DeepSeek-R1-Zero exhibits strong reasoning capabilities, it faces several issues. DeepSeek-R1-Zero struggles with challenges like poor readability, and language mixing, as DeepSeek-V3-Base is trained on multiple languages, especially English and Chinese. To address these issues, we develop DeepSeek-R1, whose pipeline is illustrated in Figure 2.

In the initial stage, we collect thousands of cold-start data that exhibits a conversational, human-aligned thinking process. RL training is then applied to improve the model performance with the conversational thinking process and language consistency. Subsequently, we apply rejection sampling and SFT once more. This stage incorporates both reasoning and non-reasoning datasets into the SFT process, enabling the model to not only excel in reasoning tasks but also demonstrate advanced writing capabilities. To further align the model with human preferences, we implement a secondary RL stage designed to enhance the model's helpfulness and harmlessness while simultaneously refining its reasoning capabilities.

The remainder of this section details the key components of this pipeline: Section 3.1 introduces the Reward Model utilized in our RL stages, and Section 3.2 elaborates on the specific training methodologies and implementation details. Data we used in this stage is detailed in Supplementary B.3.

3.1. Model-based Rewards

For general data, we resort to reward models to capture human preferences in complex and nuanced scenarios. We build upon the DeepSeek-V3 pipeline and adopt a similar distribution of preference pairs and training prompts. For helpfulness, we focus exclusively on the final summary, ensuring that the assessment emphasizes the utility and relevance of the response to the user while minimizing interference with the underlying reasoning process. For harmlessness, we evaluate the entire response of the model, including both the reasoning process and the summary, to identify and mitigate any potential risks, biases, or harmful content that may arise during the generation process.

Helpful Reward Model Regarding helpful reward model training, we first generate preference pairs by prompting DeepSeek-V3 using the arena-hard prompt format, listed in Supplementary B.2, where each pair consists of a user query along with two candidate responses. For each preference pair, we query DeepSeek-V3 four times, randomly assigning the responses as either Response A or Response B to mitigate positional bias. The final preference score is determined by averaging the four independent judgments, retaining only those pairs where the score difference (Δ) exceeds 1 to ensure meaningful distinctions. Additionally, to minimize length-related biases, we ensure that the chosen and rejected responses of the whole dataset have comparable lengths. In total, we curated 66,000 data pairs for training the reward model. The prompts used in this dataset are all non-reasoning questions and are sourced either from publicly available open-source datasets or from users who have explicitly consented to share their data for the purpose of model improvement. The architecture of our reward model is consistent with that of DeepSeek-R1, with the addition of a reward head designed to predict scalar preference scores.

$$Reward_{helpful} = RM_{helpful}(Response_A, Response_B) \tag{5}$$

The helpful reward models were trained with a batch size of 256, a learning rate of 6e-6, and for a single epoch over the training dataset. The maximum sequence length during training is set to 8192 tokens, whereas no explicit limit is imposed during reward model inference.

Safety Reward Model To assess and improve model safety, we curated a dataset of 106,000 prompts with model-generated responses annotated as “safe" or “unsafe" according to predefined safety guidelines. Unlike the pairwise loss employed in the helpfulness reward model, the safety reward model was trained using a point-wise methodology to distinguish between safe and unsafe responses. The training hyperparameters are the same as the helpful reward model.

$$Reward_{safety} = RM_{safety}(Response) \tag{6}$$

For general queries, each instance is categorized as belonging to either the safety dataset or the helpfulness dataset. The general reward, $Reward_{General}$, assigned to each query corresponds to the respective reward defined within the associated dataset.

3.2. Training Details

3.2.1. Training Details of the First RL Stage

In the first stage of RL, we set the learning rate to 3e-6, the KL coefficient to 0.001, the GRPO clip ratio ε to 10, and the sampling temperature to 1 for rollout. For each question, we sample 16 outputs with a maximum length of 32,768. Each training step consists of 32 unique questions, resulting in a training batch size of 512 per step. Every 400 steps, we replace the reference model with the latest policy model. To accelerate training, each rollout generates 8,192 outputs, which are randomly split into 16 minibatches and trained for only a single inner epoch. However, to mitigate the issue of language mixing, we introduce a language consistency reward during RL training, which is calculated as the proportion of target language words in the CoT.

$$Reward_{language} = \frac{Num(Words_{target})}{Num(Words)} \tag{7}$$

Although ablation experiments in Supplementary B.6 show that such alignment results in a slight degradation in the model's performance, this reward aligns with human preferences, making it more readable. We apply the language consistency reward to both reasoning and non-reasoning data by directly adding it to the final reward.

Note that the clip ratio plays a crucial role in training. A lower value can lead to the truncation of gradients for a significant number of tokens, thereby degrading the model's performance, while a higher value may cause instability during training.

3.2.2. Training Details of the Second RL Stage

Specifically, we train the model using a combination of reward signals and diverse prompt distributions. For reasoning data, we follow the methodology outlined in DeepSeek-R1-Zero, which employs rule-based rewards to guide learning in mathematical, coding, and logical reasoning domains. During the training process, we observe that CoT often exhibits language mixing, particularly when RL prompts involve multiple languages. For general data, we utilize reward models to guide training. Ultimately, the integration of reward signals with diverse data distributions enables us to develop a model that not only excels in reasoning but also prioritizes helpfulness and harmlessness. Given a batch of data, the reward can be formulated as

$$Reward = Reward_{\text{reasoning}} + Reward_{\text{general}} + Reward_{\text{language}} \tag{8}$$

$$\text{where, } Reward_{\text{reasoning}} = Reward_{\text{rule}} \tag{9}$$

$$Reward_{\text{general}} = Reward_{\text{reward\_model}} + Reward_{\text{format}} \tag{10}$$

The second stage of RL retains most of the parameters from the first stage, with the key difference being a reduced temperature of 0.7, as we find that higher temperatures in this stage lead to incoherent generation. The stage comprises a total of 1,700 training steps, during which general instruction data and preference-based rewards are incorporated exclusively in the final 400 steps. We find that more training steps with the model based preference reward signal may lead to reward hacking, which is documented in Supplementary B.5. The total training cost is listed in Supplementary B.4.4.

4. Experiment

We evaluate our models on MMLU (Hendrycks et al., 2021), MMLU-Redux (Gema et al., 2025), MMLU-Pro (Wang et al., 2024), C-Eval (Huang et al., 2023), and CMMLU (Li et al., 2024), IFEval (Zhou et al., 2023b), FRAMES (Krishna et al., 2024), GPQA Diamond (Rein et al., 2023), SimpleQA (OpenAI, 2024a), C-SimpleQA (He et al., 2024), SWE-Bench Verified (OpenAI, 2024b), Aider (Gauthier, 2025), LiveCodeBench (Jain et al., 2024) (2024-08 – 2025-01), Codeforces (Mirzayanov, 2025), Chinese National High School Mathematics Olympiad (CNMO 2024) (CMS, 2024), and American Invitational Mathematics Examination 2024 (AIME 2024) (MAA, 2024). The details of these benchmarks are listed in Supplementary D.

Table 3 summarizes the performance of DeepSeek-R1 across multiple developmental stages, as outlined in Figure 2. A comparison between DeepSeek-R1-Zero and DeepSeek-R1 Dev1 reveals substantial improvements in instruction-following, as evidenced by higher scores on the IF-Eval and ArenaHard benchmarks. However, due to the limited size of the cold-start dataset, Dev1 exhibits a partial degradation in reasoning performance compared to DeepSeek-R1-Zero, most notably on the AIME benchmark. In contrast, DeepSeek-R1 Dev2 demonstrates

Table 3 | Experimental results at each stage of DeepSeek-R1. Numbers in bold denote the performance is statistically significant (t−test with $p < 0.01$).

	Benchmark (Metric)	**R1-Zero**	**R1-Dev1**	**R1-Dev2**	**R1-Dev3**	**R1**
English	MMLU (EM)	88.8	89.1	**91.2**	91.0	90.8
	MMLU-Redux (EM)	85.6	90.0	93.0	93.1	92.9
	MMLU-Pro (EM)	68.9	74.1	83.8	83.1	**84.0**
	DROP (3-shot F1)	89.1	89.8	91.1	88.7	**92.2**
	IF-Eval (Prompt Strict)	46.6	71.7	72.0	78.1	**83.3**
	GPQA Diamond (Pass@1)	**75.8**	66.1	70.7	71.2	71.5
	SimpleQA (Correct)	30.3	17.8	28.2	24.9	30.1
	FRAMES (Acc.)	82.3	78.5	81.8	81.9	**82.5**
	AlpacaEval2.0 (LC-winrate)	24.7	50.1	55.8	62.1	**87.6**
	ArenaHard (GPT-4-1106)	53.6	77.0	73.2	75.6	**92.3**
Code	LiveCodeBench (Pass@1-COT)	50.0	57.5	63.5	64.6	**65.9**
	Codeforces (Percentile)	80.4	84.5	90.5	92.1	**96.3**
	Codeforces (Rating)	1444	1534	1687	1746	**2029**
	SWE Verified (Resolved)	43.2	39.6	44.6	45.6	**49.2**
	Aider-Polyglot (Acc.)	12.2	6.7	25.6	44.8	**53.3**
Math	AIME 2024 (Pass@1)	77.9	59.0	74.0	78.1	**79.8**
	MATH-500 (Pass@1)	95.9	94.2	95.9	95.4	**97.3**
	CNMO 2024 (Pass@1)	**88.1**	58.0	73.9	77.3	78.8
Chinese	CLUEWSC (EM)	93.1	92.8	92.6	91.6	92.8
	C-Eval (EM)	**92.8**	85.7	91.9	86.4	91.8
	C-SimpleQA (Correct)	66.4	58.8	64.2	66.9	63.7

marked performance enhancements on benchmarks that require advanced reasoning skills, including those focused on code generation, mathematical problem solving, and STEM-related tasks. Benchmarks targeting general-purpose tasks, such as AlpacaEval 2.0, show marginal improvement. These results suggest that reasoning-oriented RL considerably enhances reasoning capabilities while exerting limited influence on user preference-oriented benchmarks.

DeepSeek-R1 Dev3 integrates both reasoning and non-reasoning datasets into the SFT pipeline, thereby enhancing the model's proficiency in both reasoning and general language generation tasks. Compared to Dev2, DeepSeek-R1 Dev3 achieves notable performance improvements on AlpacaEval 2.0 and Aider-Polyglot, attributable to the inclusion of large-scale non-reasoning corpora and code engineering datasets. Finally, comprehensive RL training on DeepSeek-R1 Dev3 using mixed reasoning-focused and general-purpose data produced the final DeepSeek-R1. Marginal improvements occurred in code and mathematics benchmarks, as substantial reasoning-specific RL was done in prior stages. The primary advancements in the final DeepSeek-R1 were in general instruction-following and user-preference benchmarks, with AlpacaEval 2.0 improving by 25% and ArenaHard by 17%.

In addition, we compare DeepSeek-R1 with other models in Supplementary D.2. Model safety evaluations are provided in Supplementary D.3. A comprehensive analysis is provided in Supplementary E, including a comparison with DeepSeek-V3, performance evaluations on both fresh test sets, a breakdown of mathematical capabilities by category, and an investigation of test-time scaling behavior. Supplementary F shows that the strong reasoning capability can be transferred to smaller models.

5. Ethics and Safety Statement

With the advancement in the reasoning capabilities of DeepSeek-R1, we deeply recognize the potential ethical risks. For example, R1 can be subject to jailbreak attacks, leading to the generation of dangerous content such as explosive manufacturing plans, while the enhanced reasoning capabilities enable the model to provide plans with better operational feasibility and executability. Besides, a public model is also vulnerable to further fine-tuning that could compromise inherent safety protections.

In Supplementary D.3, we present a comprehensive safety report from multiple perspectives, including performance on open-source and in-house safety evaluation benchmarks, and safety levels across multiple languages and against jailbreak attacks. These comprehensive safety analyses conclude that the inherent safety level of the DeepSeek-R1 model, compared to other state-of-the-art models, is generally at a moderate level (comparable to GPT-4o (2024-05-13)). Besides, when coupled with the risk control system, the model's safety level is elevated to a superior standard.

6. Conclusion, Limitation, and Future Work

We present DeepSeek-R1-Zero and DeepSeek-R1, which rely on large-scale RL to incentivize model reasoning behaviors. Our results demonstrate that pre-trained checkpoints inherently possess substantial potential for complex reasoning tasks. We believe that the key to unlocking this potential lies not in large-scale human annotation but in the provision of hard reasoning questions, a reliable verifier, and sufficient computational resources for reinforcement learning. Sophisticated reasoning behaviors, such as self-verification and reflection, appeared to emerge organically during the reinforcement learning process.

Even if DeepSeek-R1 achieves frontier results on reasoning benchmarks, it still faces several capability limitations, as outlined below:

Structure Output and Tool Use: Currently, the structural output capabilities of DeepSeek-R1 remain suboptimal compared to existing models. Moreover, DeepSeek-R1 cannot leverage tools, such as search engines and calculators, to improve the performance of output. However, as it is not hard to build an RL environment for structure output and tool use, we believe the issue will be addressed in the next version.

Token efficiency: Unlike conventional test-time computation scaling approaches, such as majority voting or Monte Carlo Tree Search (MCTS), DeepSeek-R1 dynamically allocates computational resources during inference according to the complexity of the problem at hand. Specifically, it uses fewer tokens to solve simple tasks, while generating more tokens for complex tasks. Nevertheless, there remains room for further optimization in terms of token efficiency, as instances of excessive reasoning—manifested as overthinking—are still observed in response to simpler questions.

Language Mixing: DeepSeek-R1 is currently optimized for Chinese and English, which may result in language mixing issues when handling queries in other languages. For instance, DeepSeek-R1 might use English for reasoning and responses, even if the query is in a language other than English or Chinese. We aim to address this limitation in future updates. The limitation may be related to the base checkpoint, DeepSeek-V3-Base, mainly utilizes Chinese and English, so that it can achieve better results with the two languages in reasoning.

Prompting Engineering: When evaluating DeepSeek-R1, we observe that it is sensitive to