Skywork R1V: Bridging Vision and Language for Advanced Multimodal Reasoning

Yi Peng,[*] Chris,[*] Xiaokun Wang, Yichen Wei, Jiangbo Pei, Weijie Qiu,
Ai Jian, Yunzhuo Hao, Jiachun Pan, Tianyidan Xie, Li Ge,
Rongxian Zhuang, Xuchen Song,[†] Yang Liu,[†] Yahui Zhou

Abstract

We introduce Skywork R1V, a multimodal reasoning model extending the an R1-series Large language models (LLM) to visual modalities via an efficient multimodal transfer method. Leveraging a lightweight visual projector, Skywork R1V facilitates seamless multimodal adaptation without necessitating retraining of either the foundational language model or the vision encoder. To strengthen visual-text alignment, we propose a hybrid optimization strategy that combines Iterative Supervised Fine-Tuning (SFT) with Group Relative Policy Optimization (GRPO), significantly enhancing cross-modal integration efficiency. Additionally, we introduce an adaptive-length Chain-of-Thought distillation approach for reasoning data generation. This approach dynamically optimizes reasoning chain lengths, thereby enhancing inference efficiency and preventing excessive reasoning overthinking. Empirical evaluations demonstrate that Skywork R1V, with only 38B parameters, delivers competitive performance, achieving a score of 69.0 on the MMMU benchmark and 67.5 on MathVista. Meanwhile, it maintains robust textual reasoning performance, evidenced by impressive scores of 72.0 on AIME and 94.0 on MATH500. The Skywork R1V model weights have been publicly released to promote openness and reproducibility[‡].

1 Introduction

In recent years, significant advances have been made in artificial intelligence, especially in natural language processing. Large language models (LLMs), represented by OpenAI GPT-4o [11], Claude 3.5 [1] and Deepseek-R1 [5], have achieved groundbreaking progress in complex reasoning tasks, reaching human-expert levels in logical reasoning and mathematical problem-solving within textual contexts. These models demonstrate proficiency in accurately interpreting complex problems, performing detailed step-by-step analyses, and ultimately arriving at correct solutions in intricate mathematical and logical reasoning tasks.

However, extending these advancements into multimodal contexts presents substantial challenges. Although vision-language models (VLMs) [19, 20] excel at descriptive tasks—such as generating coherent and contextually relevant textual descriptions for images, their performance in deeply logical multimodal tasks (e.g., geometric proofs and scientific problem-solving) remains inferior to that of single-modal systems [22]. For example, For instance, geometric reasoning tasks demand that models accurately interpret intricate geometric relationships from visual inputs to carry out logical deductions effectively. Existing VLMs often struggle to accurately understand complex geometric relationships in images and conduct effective reasoning and proof.

---

[*]Equal contribution.

[†]Corresponding author.

[‡]https://huggingface.co/Skywork/Skywork-R1V-38B.

Tech report. Preprint.

Integrating reasoning-capable language models into VLMs to augment their reasoning capabilities presents a promising solution. Nevertheless, practical implementation of this integration faces significant obstacles. Specifically, the alignment between visual backbones and LLMs necessitates extensive datasets, while the unique nature of reasoning tasks demands specialized, reasoning-formatted training data. However, current VLM datasets predominantly consist of non-reasoning content, with only a limited subset containing traditional VLM chain-of-thought (CoT) [21] examples, which often lack the complexity needed for advanced reasoning tasks. Consequently, training VLMs on such datasets may inadvertently weaken rather than strengthen their multimodal reasoning capabilities.

To address these issues, we introduce Skywork R1V, a novel multimodal reasoning model. It transfers the reasoning capabilities of the R1 text model series to the visual domain via cross-modal transfer technology, achieving visual reasoning performance comparable to closed-source large models like Gemini2.0 [14] and K1.5 [15]. owing to three core technical innovations:

1. **Efficient Multimodal Transfer of Reasoning-Capable LLMs**: Utilizing a lightweight multilayer perceptron (MLP) as a visual projector, Skywork R1V seamlessly transfers the reasoning prowess of R1-series text models into multimodal scenarios without retraining either the base language model or the visual encoder.

2. **Hybrid Optimization Framework**: This framework strategically integrates Iterative Supervised Fine-Tuning (SFT) with Group Relative Policy Optimization (GRPO) reinforcement learning, progressively aligning visual-textual representations for efficient cross-modal reasoning.

3. **Adaptive-Length Chain-of-Thought Distillation**: By dynamically adjusting the length of reasoning chains, this technology mitigates excessive computational deliberation overthinking, significantly enhancing reasoning efficiency and inference effectiveness.

All components and weights of Skywork R1V have been fully open-sourced, aiming to foster broader research and innovation within the multimodal reasoning community.

2 Methodology

In this section, we elaborate on the technical details of Skywork R1V. Section 2.1 describes our approach for transferring a reasoning-capable LLM into the MLP-based VLM framework. Section 2.2 outlines our training framework, and Section 2.3 provides detailed insights into the training data generation process.

2.1 Efficient Multimodal Transfer of Reasoning-Capable LLMs

We propose an Efficient Multimodal Transfer method, efficiently aligning a reasoning-capable language model with a vision backbone through an MLP structure, substantially reducing the requirement for extensive multimodal reasoning data.

The insight behind our approach lies in decoupling the alignment of visual-language representations from the preservation of reasoning capabilities. Directly connecting the reasoning-capable language model (f_l) to a vision backbone (f_v) would necessitate extensive multimodal reasoning data in the R1-style format to simultaneously achieve both objectives. However, acquiring such data is prohibitively expensive and impractical for most applications.

Instead, we adopt a staged strategy. First, we train an MLP adapter to align f_v with a substitutive language model (f_l^s) sharing the same architecture as f_l but without reasoning capabilities. This initial step allows the MLP to learn a generalized mapping from the visual space of f_v to the language space of f_l^s, utilizing existing multimodal datasets. Subsequently, leveraging the latent similarity between f_l^s and f_l, transferring this pretrained MLP to align f_v with the original reasoning-capable model f_l becomes significantly more efficient, requiring substantially fewer data. Our method is detailed further in the following three steps.

MLP Initialization Given a vision encoder f_v (we employ the `Vision Transformer (ViT)` [6]), a reasoning-capable language model f_l (`DeepSeek-R1-distill-Qwen2.5-32B` [5]), and a
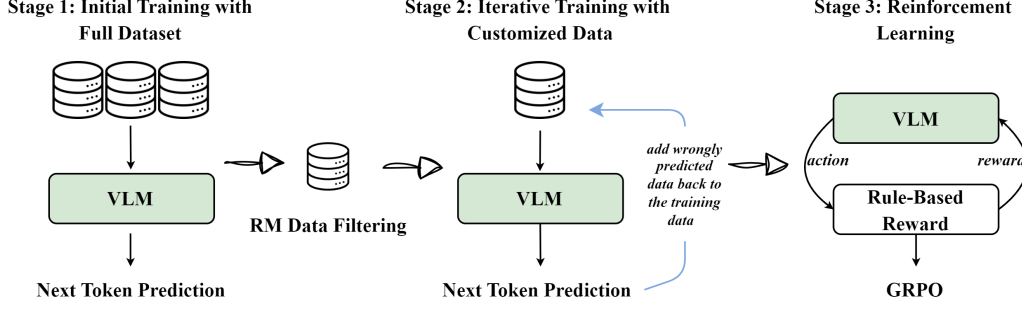
Figure 1: Hybrid Optimization Framework.

substitutive language model f_l^s (`Qwen2.5-32B-Instruct` [16]), we first initialize the MLP adapter by aligning f_v with f_l^s. Specifically, the MLP adapter θ connects f_v and f_l^s, forming a preliminary vision-language model $M' = f_v \circ \theta \circ f_l^s$. Keeping both f_v and f_l^s frozen, we optimize the MLP parameters through the following SFT process: 1) initial fine-tuning on the full dataset (2M samples); 2) refinement on a curated subset consisting of 200K high-quality samples selected via GPT-4 evaluation; and 3) a final fine-tuning step on 40K high-quality Chain-of-Thought (CoT) samples. The learning rate is set to 2×10^{-4} for the initial fine-tuning stage, and reduced to 4×10^{-5} for the subsequent refinement stages (2 and 3). Additional hyperparameters include a context length of 16384 tokens, a weight decay of 0.05, a warmup ratio of 0.03, a batch size of 512, and a training epoch of 1 for each stage[§].

Model Re-Assembly   Using the pretrained MLP adapter θ obtained from Step 1, we transfer it to bridge the vision encoder f_v and the original reasoning-capable language model f_l, thus constructing the complete R1V model $M = f_v \circ \theta \circ f_l$. The tokenizer for the combined model follows that of f_l. **Notably**, despite the changes in both the language model parameters and the tokenizer, the assembled model surprisingly retains a significant portion of its original performance. For further details on this phenomenon, please refer to Section 3.2.

Modality Alignment   Finally, we perform modality alignment between the visual and textual representations within the model M. During this phase, only the MLP adapter parameters θ are fine-tuned, while both the vision encoder f_v and the reasoning-capable language model f_l remain fixed. This approach ensures the model retains robust reasoning abilities inherited from the R1-series LLM, while effectively aligning visual and language modalities. This alignment is carried out via the Hybrid Optimization Framework detailed in Section 2.2, utilizing training data in the form of reasoning chains generated through our Adaptive-Length Chain-of-Thought method as described in Section 2.3.

2.2 Hybrid Optimization Framework

We propose a Hybrid Optimization Framework (Figure 1) that strategically integrates iterative SFT and GRPO. Specifically, during the iterative SFT phase, we sequentially train a series of models $M_0, M_1, \ldots, M_T$. Each subsequent model M_{t+1} is trained using high-quality data identified by our reward model, along with challenging samples misclassified by the previous iteration. In the RL phase, we apply Group Relative Policy Optimization (GRPO) to further enhance model generalization

Stage 1: Initial Training with Full Dataset   We commence by performing SFT of the base model using our full dataset $\mathcal{D}$. The training configuration follows the aforementioned **MLP Initialization**. The data generation process is introduced in Section 2.3. This stage produces the initialized model M_0.

Stage 2: Iterative Training with Customized Data   During the iterative supervised fine-tuning (SFT) phase, we sequentially train a series of models $\{M_1, M_2, \ldots, M_T\}$ (with $T = 4$). Each model M_t is iteratively fine-tuned based on its predecessor M_{t-1}, leveraging high-quality samples identified

---

[§]The developed model (i.e., M') is publicly available at SkyworkVL-38B.

3

by our Reward Model (RM), as well as challenging cases incorrectly processed by M_{t-1}. Specifically, the Reward Model assigns a quality score to each sample as follows:

$$\text{RM} : \mathcal{X} \rightarrow \{0, 1, 2, 3, 4, 5\}.$$

Utilizing these scores, we construct a refined dataset $\mathcal{D}_{\text{rm}}$ by selecting high-quality samples from the original dataset $\mathcal{D}$ according to a dynamically increasing threshold:

$$\mathcal{D}_{\text{rm}} = \{(x, y) \in \mathcal{D} \mid \text{RM}(x) \geq \tau\},$$

where the threshold τ is progressively set to $2, 3, 4$, and 5 for iterations $t = 1, 2, 3, 4$, respectively.

Additionally, for each iteration t, we construct an error-focused dataset $\mathcal{E}_{t-1}$, explicitly targeting challenging samples that were misclassified in the previous iteration:

$$\mathcal{E}_{t-1} = \{(x, y) \in \mathcal{D} \mid \Phi(M_{t-1}(x)) \neq y\},$$

where $M_{t-1}(x)$ denotes the response from the model at iteration $t - 1$, y represents the ground-truth label, and Φ is a function used to extract answers from the model's outputs.

The combined, customized training dataset for each iteration t is thus formulated as:

$$\mathcal{D}_t = \mathcal{D}_{\text{rm}} \cup \mathcal{E}_{t-1}.$$

Finally, we fine-tune the previous iteration's model M_{t-1} using the customized dataset $\mathcal{D}_t$, thus enhancing the model's robustness and generalization capabilities, resulting in the improved model M_t. For each iteration, we train the model for 1 epoch, employing a context length of 16,384 tokens, weight decay of 0.05, warmup ratio of 0.03, and batch size of 512. The learning rate is set to 1×10^{-4} for the first iteration and subsequently reduced to 2×10^{-5} for the following iterations.

Stage 3: Reinforcement Learning   Following the approach proposed in DeepSeek R1, we utilize the GRPO with rule-based reward system (Accuracy reward & Format reward) to further boost the generalizability of our model. The reward-model-filtered subset $\mathcal{D}_{\text{rm}}$ ($\tau = 5$) is utilized as the training dataset. The model is trained with the following hyperparameters: learning rate of 1×10^{-6}, temperature of 1.0, generation number of 8, and a maximum completion length of 8k tokens. After RL training, we select the model that achieves the optimal balance between performance and reasoning rationality, designating it as the final model.

2.3   Adaptive-Length Chain-of-Thought Distillation

We propose an Adaptive-Length Chain-of-Thought Distillation (AL-CoTD) framework (Figure 2), specifically designed to dynamically optimize the reasoning chain length when generating high-quality reasoning-oriented data. The generated data effectively mitigates the common issue of excessive reasoning or overthinking during inference.

Quality and Difficulty Assessment Module (QDAM)   The QDAM leverages GPT-4o to systematically evaluate image-text query pairs across two primary dimensions: the *vision score* (S_v) and the *text score* (S_t). Specifically, the *vision score* assesses visual characteristics via two criteria—*image clarity* and *image necessity*. *Image clarity* quantifies perceptual quality using blur detection and resolution analysis, whereas *image necessity* evaluates the dependency of the text on visual context through context ablation tests and relevance classification. The *text score* examines linguistic properties through three distinct aspects: *question quality*, which assesses clarity using grammatical validation and semantic coherence checks; *difficulty level*, which measures conceptual complexity based on domain-specific knowledge requirements; and *reasoning demand*, which quantifies the complexity of inference steps via multi-hop reasoning analysis. Together, these measures offer a comprehensive framework for capturing both perceptual and cognitive complexities inherent in multimodal query understanding. All these properties are obtained by using GPT-4o, except for image clarity.

Vision-Text Integration Analyzer (VTIA)   VTIA quantifies the required depth of cross-modal integration by performing syntactic and semantic analyses, generating an *integration score* (S_I) through pattern recognition within image-text queries using GPT-4o. Queries with high integration patterns, resulting in increased S_I, are typically found in tasks demanding scientific explanations
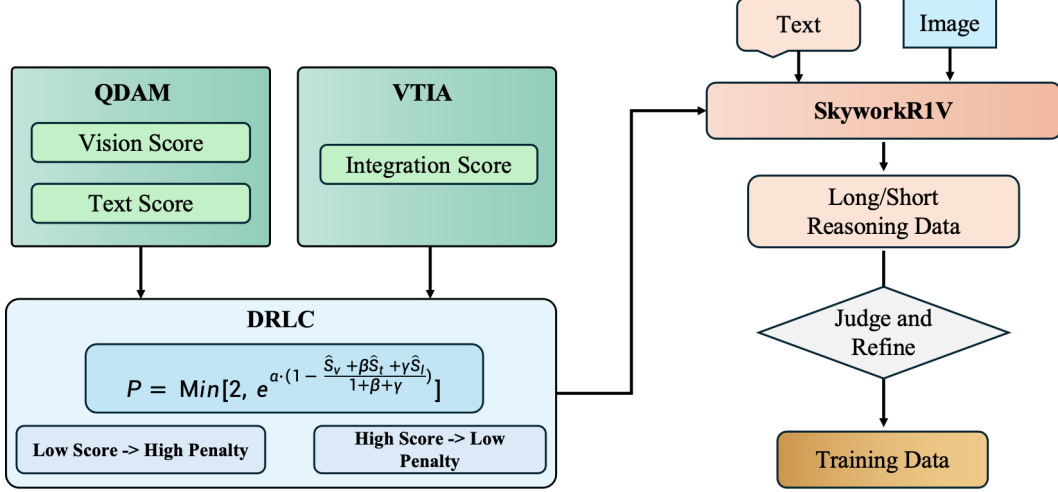
4

Figure 2: Adaptive-Length Chain-of-Thought Distillation.

or detailed reasoning. Such patterns include the presence of causal connectives (*"why"/"how"*) accompanied by presupposition triggers, multiple-object visual references necessitating spatial relationship comprehension, and co-occurrence of domain-specific terminologies. Conversely, queries exhibiting low integration patterns lead to reduced S_I. These are commonly seen in simpler tasks like object recognition, characterized by straightforward interrogatives (*"what"/"where"*) accompanied by definite articles, queries targeting direct object identification, and minimal dependence between the textual content and visual input. This pattern-driven analytical framework facilitates adaptive cross-modal fusion precisely tailored to the complexity of each query.

Dynamic Reasoning Length Controller (DRLC) The DRLC module operates on normalized scores $\hat{S}_v$, $\hat{S}_t$, and $\hat{S}_I$, derived from the original scores S_v, S_t, and S_I via min-max scaling to the range $[0,1]$. The controller dynamically adjusts the reasoning chain length by modulating the *repetition penalty* based on query complexity. Specifically, queries characterized by high visual-textual quality $(\hat{S}_v, \hat{S}_t)$, substantial cognitive difficulty, and complex visual scenarios demanding deeper reasoning (reflected by higher values of $\hat{S}_v$, $\hat{S}_t$, and $\hat{S}_I$) receive a lower repetition penalty, allowing for longer reasoning chains. Conversely, queries of lower difficulty, simpler visual identification tasks, and minimal cross-modal integration requirements (indicated by lower $\hat{S}_v$, $\hat{S}_t$, and $\hat{S}_I$ scores) are assigned higher repetition penalties to prevent unnecessary reasoning. The repetition penalty (P) is calculated as:

$$P = \min \left[2,\ e^{\alpha \cdot \left(1 - \frac{\hat{S}_v + \beta \hat{S}_t + \gamma \hat{S}_I}{1+\beta+\gamma}\right)} \right], \tag{1}$$

where α, β, and γ are hyperparameters controlling the relative influence of these components.

Multi-Stage Self-Distillation Pipeline Building upon our DRLC module, we further propose a progressive self-distillation strategy. In this pipeline, the model initially generates reasoning-oriented data explicitly annotated with <think> tokens, where the repetition penalty P, computed by the DRLC module, dynamically regulates the inference length. Subsequently, GPT-4o evaluates the correctness of the generated answers. If an answer is assessed as correct, the original reasoning chain (<think> annotations) is preserved; otherwise, GPT-4o revises the reasoning process to realign it with the correct answer. This procedure is conducted prior to Stage 1 and repeated before each iteration of Stage 2 to refine the reasoning chain within our Hybrid Optimization Framework.

3 Experiments

We conducted a comprehensive evaluation of our model across multiple benchmarks designed to assess performance in different modalities. The benchmarks primarily fall into two categories:

	Benchmark	LLM	VLM				
		QwQ-32B -Preview	QwenVL 2-72B	InternVL- 2.5-38B	VILA 1.5-40B	InternVL2 -40B	Skywork R1V (38B)
Reasoning	MATH-500	90.6	-	-	-	-	**94.0**
	AIME 2024	50.0	-	-	-	-	**72.0**
	GPQA	54.5	-	-	-	-	**61.6**
Vision	MathVista(mini)	-	70.5	**71.9**	49.5	63.7	67.5
	MMMU(Val)	-	64.5	63.9	55.1	55.2	**69.0**

Table 1: Evaluation results of state-of-the-art LLMs and VLMs.

- **Reasoning Benchmarks**:
  1. **MATH-500** [7]: This dataset comprises 500 undergraduate-level mathematical problems spanning algebra, calculus, probability, and various other topics. It evaluates both computational proficiency and advanced mathematical reasoning, with higher scores reflecting superior problem-solving abilities.
  2. **AIME 2024**: This benchmark includes competition problems from the 2024 American Invitational Mathematics Examination (AIME), a prestigious and highly selective contest for elite high school students. It assesses advanced mathematical competencies, requiring deep conceptual understanding and rigorous logical reasoning skills.
  3. **GPQA** [13]: GPQA evaluates the general-purpose question-answering capabilities of language models. It comprises carefully designed questions spanning diverse domains, providing a robust measure of a model's ability to comprehend, analyze, and accurately respond to complex queries across multiple knowledge areas.
- **VLM Benchmarks**:
  1. **MathVista** [10]: MathVista presents challenges integrating mathematical reasoning and visual understanding. It combines diverse tasks requiring precise visual interpretation and structured analytical reasoning, thus evaluating a model's capability to handle intricate multimodal problems.
  2. **MMMU** [23]: MMMU consists of approximately 11,500 questions sourced from college-level exams, quizzes, and textbooks, covering six academic disciplines: Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Tech & Engineering. It assesses the model's proficiency in comprehending and responding effectively to complex multimodal inputs.

Evaluation Setup In our evaluations, the maximum generation length is set to 64K tokens. For textual reasoning benchmarks, the test prompts strictly adhere to the implementation guidelines provided by DeepseekR1. For visual-language model (VLM) benchmarks, including MMMU, and MathVista, we utilize a unified test prompt. The reported performance metric is the Pass@1 score, averaged across 5 independent runs.

➤ **Prompt for Multi-Choice QA Problems**: Answer the multiple choice preceding question. The last line of your response should follow this format: 'Answer: \boxed{$LETTER}' (without quotes), where LETTER is one of the options. If you are uncertain or the problem is too complex, make a reasoned guess based on the information provided. Avoid repeating steps indefinitely—provide your best guess even if unsure. Think step by step logically, considering all relevant information before answering.

➤ **Prompt for Other Problems**: Answer the preceding question. The last line of your response should follow this format: 'Answer: \boxed{$FINAL_ANSWER}' (without quotes), where FINAL_ANSWER is is your conclusion. If you are uncertain or the problem is too complex, make a reasoned guess based on the information provided. Avoid repeating steps indefinitely—provide your best guess even if unsure. Think step by step logically, considering all relevant information before answering.

Baselines We conduct comprehensive evaluations against several strong closed-source models, including Claude-3.5-Sonnet (20241022) [1], GPT-4o-0513 [12], OpenAI-o1-mini [8], and Kimi k1.5 [15]. Additionally, we compare our method with advanced open-source models, such as InternVL2-40B [3], InternVL2.5-38B [3], InternVL2.5-78B [3], VILA-1.5-40B [9], QwQ-32B-Preview [17] , Deepseek V3 [4], Deepseek R1 [5], Qwen2-VL-72B-Instruct [18] and Qwen2.5-VL-72B-Instruct [2].
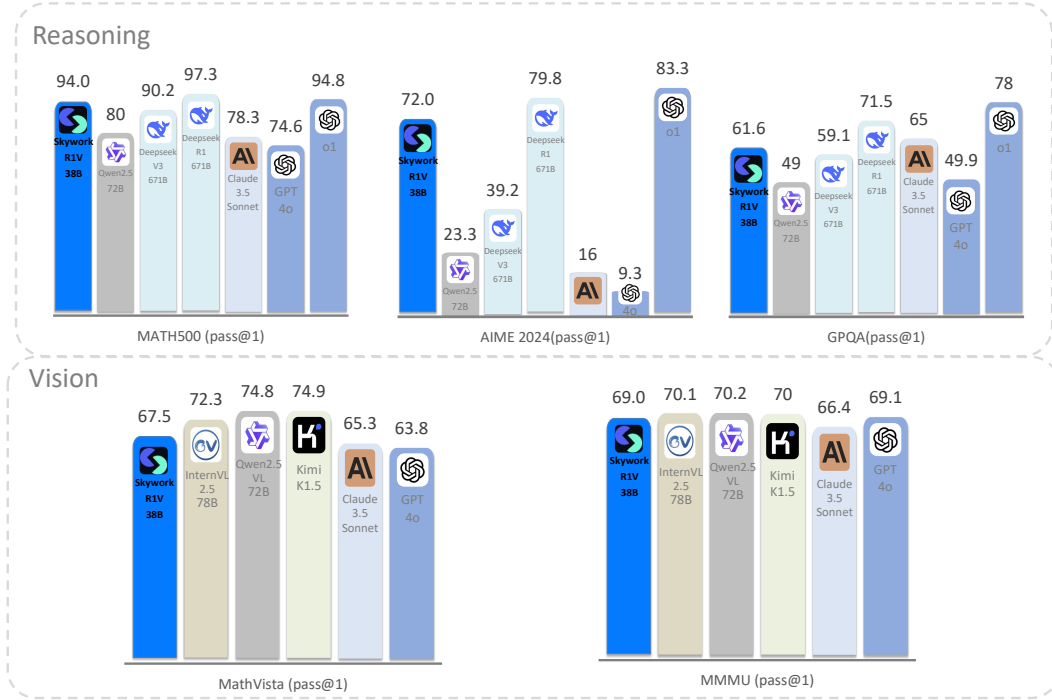
Figure 3: Comparison with Larger-Scale Open-Source and Closed-Source Models.

3.1 Main Results

Comparison with Models of Similar Scale We comprehensively compare the performance of our Skywork R1V model with other state-of-the-art models of similar scale across various benchmarks. As shown in the updated evaluation results (Table 1), Skywork R1V exhibits outstanding performance in both reasoning and visual tasks.
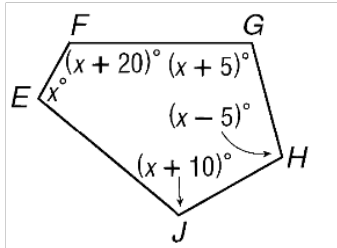
In text-based reasoning tasks, Skywork R1V achieves exceptional results, notably scoring 94.0 on the MATH-500 benchmark, surpassing similar-scale models such as QwQ-32B-Preview (90.6), and demonstrating significant advantages on the AIME 2024 benchmark with a remarkable score of 72.0.

In visual multimodal tasks, Skywork R1V also demonstrates robust performance. Specifically, on the MathVista (mini) benchmark, it achieves a score of 67.5, maintaining a strong performance relative to InternVL2-40B (63.7), though trailing InternVL2.5-38B (71.9). Moreover, on the MMMU (Val) benchmark, Skywork R1V attains a Pass@1 score of 69.0, significantly surpassing VILA-40B (55.1), InternVL2-40B (55.2), InternVL2.5-38B (63.9), and QwenVL2-72B (64.5). These results collectively underscore the superior capability of Skywork R1V across a wide spectrum of tasks, affirming its effectiveness in both advanced reasoning and complex visual understanding scenarios.

Comparison with Larger-Scale Open-Source and Closed-Source Models We further compare our Skywork R1V model with larger-scale open-source and closed-source models (Figure 3). Despite being only a 38B-scale model, Skywork R1V achieves competitive performance compared to significantly larger counterparts. Specifically, on the MathVista benchmark, our model achieves a score of 67.5, surpassing the performance of some closed-source models such as Claude 3.5 Sonnet (65.3). More notably, on the MMMU benchmark, Skywork R1V attains a score of 69.0, outperforming Claude 3.5 Sonnet (66.4) and matching GPT4-o (69.1). These results indicate that our approach effectively transfers advanced reasoning capabilities from textual modalities to vision, enabling our smaller-scale model to perform on par with larger-scale models.

3.2 Analysis

Analysis of Reasoning Capability As shown in Figure 4, Skywork R1V addresses the pentagon angle problem with structured mathematical reasoning. It first applies the geometric principle that a

Figure 4: Reasoning Capability of Skywork R1V on Mathematical Problems.

pentagon's interior angles sum to $540°$, constructs a linear equation from symbolic angle expressions, and solves for $x = 102$ through algebraic simplification. The model then validates its solution by substituting x back into all angles to confirm the total equals $540°$, and calculates angle $H = 97°$ to align with contextual objectives. This dual-phase approach—systematic problem-solving coupled with self-verification—demonstrates rigorous integration of geometric and algebraic reasoning, critical for complex mathematical tasks.

Figure 5 tests the model's reasoning capabilities through a chart analysis task. The model begins by accurately interpreting the graph structure, identifying axes, gender-specific trends, and temporal patterns in U.S. life expectancy data (2009–2019). To pinpoint the year with the largest gender gap, it systematically calculates yearly differences between female and male values, validates results through recalculation to eliminate arithmetic errors, and cross-checks numerical findings against the visual trend of a widening gap. This dual-phase verification ensures robustness, culminating in the identification of 2019 as the peak disparity year. The model's ability to integrate structured data processing, self-correction, and contextual alignment of quantitative and visual evidence underscores its proficiency in multimodal reasoning tasks.

Chart & OCR
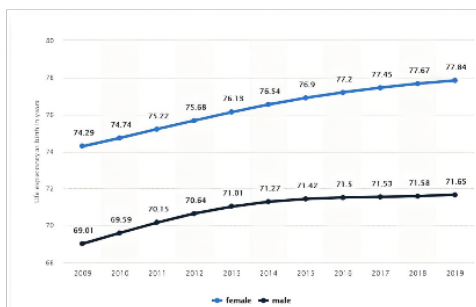
Question:
Which year showed the largest difference in the data points between the two lines?

Skywork–R1V:

So I've got this graph here that shows life expectancy at birth for males and females in the United States from 2009 to 2019. The x–axis is the year, and the y–axis is life expectancy in years. There are two lines: one for females and one for males. The female line is in blue, and the male line is in black. First, I need to understand what the graph is showing. It seems like both lines are increasing over time, which makes sense because life expectancy generally tends to increase due to better healthcare, nutrition, and living conditions. However, the question is asking for the year that showed the largest difference in the data points between the two lines. To find this, I need to look at the difference between the female and male life expectancies for each year and identify which year has the biggest gap. Let me list out the data points for each year: 2009: Female: 74.29, Male: 69.01, Difference: 74.29 – 69.01 = 5.28. 2010: Female: 74.74, Male: 69.59, Difference: 74.74 – 69.59 = 5.15. 2011: Female: 75.22, Male: 70.15, Difference: 75.22 – 70.15 = 5.07. 2012: Female: 75.68, Male: 70.64, Difference: 75.68 – 70.64 = 5.04. 2013: Female: 76.13, Male: 71.01, Difference: 76.13 – 71.01 = 5.12. 2014: Female: 76.54, Male: 71.27, Difference: 76.54 – 71.27 = 5.27. 2015: Female: 76.9, Male: 71.42, Difference: 76.9 – 71.42 = 5.48. 2016: Female: 77.2, Male: 71.55, Difference: 77.2 – 71.55 = 5.65. 2017: Female: 77.45, Male: 71.53, Difference: 77.45 – 71.53 = 5.92. 2018: Female: 77.67, Male: 71.58, Difference: 77.67 – 71.58 = 6.09. 2019: Female: 77.84, Male: 71.65, Difference: 77.84 – 71.65 = 6.19. Now, I need to compare these differences to find the year with the largest gap. Looking at the differences: 2009: 5.28, 2010: 5.15, 2011: 5.07, 2012: 5.04, 2013: 5.12, 2014: 5.27, 2015: 5.48, 2016: 5.65, 2017: 5.92, 2018: 6.09, 2019: 6.19. From this list, it's clear that the difference increases over time, with the largest difference in 2019 at 6.19 years. Wait a minute, but the question is asking for the year that showed the largest difference in the data points between the two lines. So, according to this, 2019 has the largest difference. However, I should double–check to make sure I haven't made any calculation errors. Let me recalculate the differences: 2009: 74.29 – 69.01 = 5.28. 2010: 74.74 – 69.59 = 5.15. 2011: 75.22 – 70.15 = 5.07. 2012: 75.68 – 70.64 = 5.04. 2013: 76.13 – 71.01 = 5.12. 2014: 76.54 – 71.27 = 5.27. 2015: 76.9 – 71.42 = 5.48. 2016: 77.2 – 71.55 = 5.65. 2017: 77.45 – 71.53 = 5.92. 2018: 77.67 – 71.58 = 6.09. 2019: 77.84 – 71.65 = 6.19. All the calculations seem correct. So, the largest difference is indeed in 2019. But just to be thorough, I'll look at the graph again to see if there's any visual indication that might contradict this. Looking at the graph, the female line is consistently above the male line, and the gap between them seems to be widening over time. The differences I calculated align with this observation. Therefore, based on the data provided and the calculations, the year that showed the largest difference in life expectancy between females and males is 2019. **Final Answer** \boxed{2019}

Ground Truth: 2019

Figure 5: Reasoning Capability of Skywork R1V on Chart Problems.

Model	Initial	Stage 1	Stage 2				Stage 3 (RL)
			t=1	t=2	t=3	t=4	
Performance	60.2	62.5	63.9	64.7	65.2	65.6	69.0

Table 2: Model Performance of Skywork R1V at Different Stages on the MMMU Dataset.

Performance of the Preliminary Model   We first evaluate the performance of the preliminary VLM (combined by ViT, MLP and a Qwen2.5-32B-Instruct) obtained after the initial MLP initialization step. This preliminary model achieves a competitive score of 64.0 on MMMU benchmark[¶].

Performance of the Newly Assembled Skywork R1V   Upon transferring and applying the pre-trained MLP adapter to the DeepSeek-R1-distill-Qwen-32B model (i.e., after performed Model Re-Assembly), the newly assembled multimodal model achieves an impressive score of 60.2 (Table 2). Remarkably, this performance not only exceeds many smaller-scale models explicitly trained for multimodal alignment but also rivals larger models such as InternVL2-40B (55.2). Moreover, the reassembled model's performance closely approaches state-of-the-art models at a similar scale, notably InternVL2.5-38B-MPO (64.1). These results illustrate a surprising effectiveness of the pretrained MLP in aligning the ViT vision encoder with another reasoning-capable LLM from the same series (DeepSeek-R1-distill-Qwen-32B), despite employing a different tokenizer and without additional fine-tuning.

Effectiveness of Iterative SFT   The iterative Supervised Fine-Tuning (SFT) strategy yields consistent performance improvements across successive training stages, as demonstrated in Table 2. Beginning from an initial score of 60.2, the model exhibits steady incremental gains at each iteration. By the conclusion of the fifth stage, performance reaches 65.6, clearly evidencing the effectiveness and stability of iterative fine-tuning in progressively refining model capabilities.

Effectiveness of RL Training   The introduction of Group Relative Policy Optimization (GRPO) utilizing a ReLU-based reward function significantly boosts the model's performance, attaining an impressive increase to 69.0. This notable improvement underscores the efficacy of RL techniques in further enhancing multimodal reasoning capabilities. Additionally, an intriguing observation emerged during the RL training phase: employing GRPO results in an increase in the length and detail of the model's outputs. This phenomenon aligns with previous observations from DeepSeekR1, where models undergoing RL training exhibit "aha moments", spontaneously generating more comprehensive and elaborative responses.

References

[1] Anthropic. Claude-3.5, 2024. 1, 6

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6

[3] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 6

[4] DeepSeek-AI. Deepseek-v3 technical report, 2024. 6

[5] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 1, 2, 6

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al.

[¶]Please refer to https://huggingface.co/Skywork/SkyworkVL-38B.

An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[7] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021. 6

[8] 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. 6

[9] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models, 2023. 6

[10] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 6

[11] OpenAI. Gpt-4 technical report, 2023. 1

[12] OpenAI. Gpt-4o system card, 2024. 6

[13] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. 6

[14] Gemini Team. Gemini: A family of highly capable multimodal models, 2024. 2

[15] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. 2, 6

[16] Qwen Team. Qwen2.5: A party of foundation models, September 2024. 3

[17] Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. 6

[18] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6

[19] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models, 2023. 1

[20] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. VisionLLM: Large language model is also an open-ended decoder for vision-centric tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2

[22] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. Gpt4tools: Teaching large language model to use tools via self-instruction. *arXiv preprint arXiv:2305.18752*, 2023. 1

[23] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 6