DREAM-R: MULTIMODAL SPECULATIVE REASONING WITH RL-BASED REFINED DRAFTING, PRECISE VERIFICATION, AND FULLY PARALLEL EXECUTION

Yunhai Hu[1] **Zining Liu**[4] **Xiangyang Yin**[1] **Tianhua Xia**[2] **Bo Bao**[3] **Eric Sather**[3]
Vithursan Thangarasa[3] **Sai Qian Zhang**[1,2]
[1]Courant Institute of Mathematical Sciences, New York University
[2]Tandon School of Engineering, New York University
[3]Cerebras Systems Inc.
[4] University of Pennsylvania
{yh5961, tx856, xy2053, sai.zhang}@nyu.edu
zliu0@seas.upenn.edu
{bo.bao, eric.sather, vithu}@cerebras.net

ABSTRACT

Speculative reasoning has recently been proposed as a means to accelerate reasoning-intensive generation in large multimodal models, but its effectiveness is often constrained by misalignment between speculative drafts and target-verified reasoning. In this work, we introduce *DREAM-R*, a framework that substantially improves the performance of speculative reasoning. At its core, DREAM-R employs *Speculative Alignment Policy Optimization* (SAPO), a reinforcement-learning objective that trains draft models to generate reasoning steps that are faithful to target trajectories and concise. We further propose a *Contrastive Probability Normalization* (CPN) that uses a ratio-based criterion to provide stable and interpretable acceptance of speculative steps only when positive evidence clearly dominates, thereby preventing error propagation. Building on these components, we develop a *Fully Parallel Speculative Reasoning* (FPSR) framework that parallelizes draft generation, target-side reasoning, and verification across multi-step reasoning, enabling early stopping and clean fallback. Experiments on reasoning-heavy benchmarks demonstrate up to $2.48\times$ speedup while preserving target-model accuracy, yielding substantial efficiency gains without compromising reasoning quality.

1 INTRODUCTION

Large models (LMs) Grattafiori et al. (2024); Brown et al. (2020); Ouyang et al. (2022); Achiam et al. (2023) have become the backbone of modern artificial intelligence, driving major advances across language understanding, vision, and multimodal perception by leveraging massive model capacity and data scale. While these models excel at representation learning and pattern recognition, many real-world tasks require more than direct prediction, demanding structured, multi-step inference and long-horizon decision-making. This need has given rise to large reasoning models (LRMs) Bai et al. (2025); Hong et al. (2025); Wang et al. (2025a), which extend large models with explicit reasoning capabilities that can decompose complex problems, integrate heterogeneous evidence, and maintain consistency across extended inference chains. LRMs are particularly important in settings such as multimodal understanding, scientific problem solving, and embodied intelligence, where accurate outcomes depend on coherent reasoning rather than surface-level correlations. However, the increased computational cost of reasoning-intensive generation presents significant efficiency challenges, motivating research into methods that can accelerate reasoning while preserving accuracy.

To address this challenge, recent advances in speculative decoding Li et al. (2024a;b; 2025b); Cai et al. (2024); Ankner et al. (2024); Xia et al. (2023); Zhang et al. (2023); Miao et al. (2023); Chen et al. (2024); Sun et al. (2024); Hu et al. (2025a;b) have shown promise in accelerating inference by parallelizing generation and verification using lightweight draft models. Building on this idea,

speculative reasoning Pan et al. (2025); Fu et al. (2025) extends speculation beyond token-level decoding to reasoning-step granularity, enabling more effective acceleration for LRMs by exploiting parallelism while preserving correctness. These developments point toward a promising direction for scaling reasoning performance without proportionally increasing inference cost.

Most existing speculative decoding and speculative reasoning methods are developed and evaluated primarily for large language models (LLMs) and LRMs, with limited investigation into multimodal LLMs (MLLMs), especially multimodal large reasoning models (MLRMs). In practice, we observe that directly applying these techniques to MLRMs often yields suboptimal performance, as the reasoning characteristics and error modes in multimodal settings differ substantially from those of text-only models. Through an empirical analysis of multimodal reasoning behavior, we observe that design choices effective for LLMs do not transfer reliably to MLRMs. Motivated by these observations, we introduce *DREAM-R*, a multimodal speculative reasoning framework that integrates RL-based refined drafting, precise threshold-based verification, and fully parallel execution to effectively accelerate multimodal reasoning while preserving accuracy. In particular, our contribution can be summarized as follows:

- DREAM-R incorporates a *Speculative Alignment Policy Optimization* (SAPO), a reinforcement-learning objective that trains draft models to produce reasoning steps that are both faithful to target-model trajectories and concise.
- Furthermore, we introduce a *Contrastive Probability Normalization* (CPN) that employs a contrastive probability normalization to ensure stable and interpretable acceptance of speculative steps only when positive evidence clearly outweighs negative evidence, effectively preventing error propagation.
- Finally, we present a *Fully Parallel Speculative Reasoning* (FPSR) framework that parallelizes draft generation, target-side reasoning, and verification across multi-step reasoning, enabling early stopping and clean fallback.
- Experiments on reasoning-heavy benchmarks demonstrate up to $2.48\times$ speedup while maintaining target-model accuracy, delivering substantial efficiency gains without compromising reasoning quality.

2 Related Works

Large Reasoning Models Large reasoning models (LRMs) are distinguished by their ability to generate extended chains of thought (CoT) Wei et al. (2022), enabling explicit intermediate reasoning steps prior to producing a final answer. In language-only settings, this inference paradigm has yielded substantial gains on complex tasks such as mathematical reasoning Cobbe et al. (2021) and code generation Chen (2021). For vision-language reasoning models (VLRMs), CoT reasoning naturally extends to multimodal scenarios that require integrating visual evidence with textual inference. Recent VLRMs Bai et al. (2025); Hong et al. (2025); Wang et al. (2025a) demonstrate strong performance on vision-dependent reasoning benchmarks, highlighting the effectiveness of multimodal CoT reasoning.

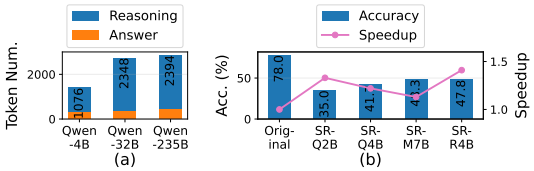

Figure 1: (a) Numbers of reasoning and answer tokens. Qwen-4B, Qwen-32B, and Qwen-235B refer to Qwen3-VL-4B, Qwen3-VL-32B, and Qwen3-VL-235B-A22B. (b) Accuracy and speedup of Qwen3-VL-32B under different decoding methods. Original denotes standard decoding. SR-Q2B, SR-Q4B, SR-M7B, and SR-R4B denote SpecReason Pan et al. (2025) using Qwen3-VL-2B, Qwen3-VL-4B, MiMo-VL-7B-RL, and Qwen3-VL-R1-VL-4B as draft models, respectively. Speedup is normalized to Original.

However, VLRMs typically incur high decoding latency, which is largely driven by the generation of intermediate reasoning steps. Analyzing Qwen3-VL-4B, Qwen3-VL-32B, and Qwen3-VL-235B on 50 samples from the MathVerse dataset, we find that the models generate an average of 2,330 tokens per sample, with 1,939 tokens corresponding to intermediate reasoning, accounting for approximately 83% of the total output. In contrast, answer tokens make up only 17% of the output,

as illustrated in Figure 1(a). These results highlight that the reasoning phase dominates decoding cost and constitutes the primary bottleneck for end-to-end inference latency in VLRMs.

Speculative Decoding Speculative decoding accelerates autoregressive inference by pairing a lightweight draft model with a larger target model (Stern et al., 2018). After the target model completes prefilling, the draft model proposes multiple speculative tokens, which are verified in parallel by the target model using causal attention. If all speculative tokens are accepted, decoding advances without more target computation; otherwise, decoding resumes from the first rejected token.

Prior work has explored speculative decoding along multiple axes (Xia et al., 2023; Hu et al., 2025a). Algorithmically, some methods replace the draft model with retrieval-based (Yang et al., 2023; He et al., 2023) or n-gram-based generators (Ou et al., 2024; Stewart et al., 2024). Architecturally, self-speculative approaches reuse shallow target layers or apply layer skipping to generate drafts (Zhang et al., 2023; Xia et al., 2024; Elhoushi et al., 2024; Liu et al., 2024b), while others introduce lightweight trainable draft heads that share embeddings or output layers with the target model (Li et al., 2024a; Gao et al., 2024b). Verification strategies range from sequential acceptance to tree-based parallel verification (Miao et al., 2023; Chen et al., 2024), and recent systems work explores asynchronous or pipelined execution to improve hardware utilization (Chen et al., 2023; McDanel et al., 2025).

Multimodal Acceleration Extending speculative decoding to multimodal models introduces additional challenges, as draft generation must account for both linguistic and visual evidence. Several recent works adapt speculative decoding to multimodal settings by modifying draft model architectures or training strategies. VADUSA (Li et al., 2025a) applies speculative decoding with tolerance mechanisms in text-to-speech systems. For vision-language models, Gagrani et al. (2024) pair a language-only draft model with a VLM target, while IbED (Lee et al., 2025) improves efficiency through batch-level ensembling without additional parameters. Other approaches focus on strengthening multimodal draft models: DREAM (Hu et al., 2025b) incorporates cross-attention and adaptive feature selection, ViSpec (Kang et al., 2025) compresses visual tokens via lightweight vision adapters, and MASSV (Ganesan et al., 2025) adapts small language models into multimodal drafters through architectural changes and self-distillation.

Speculative Reasoning Large reasoning models, including both language-only and multimodal variants, achieve strong performance on complex tasks but incur high inference latency due to extended CoT generation. To alleviate this overhead, recent work proposes speculative reasoning Pan et al. (2025), where a lightweight draft model generates candidate reasoning steps that are verified by a larger target model. The target model assigns a scalar score to each step to determine acceptance. The key insight is that many intermediate reasoning steps do not require the full capacity of the target model and can be accurately produced by a smaller model, while preserving semantic correctness even when token-level outputs differ.

Building on this line of work, Lookahead Reasoning Fu et al. (2025) proposes an asynchronous architecture that overlaps the drafting and verification stages, allowing the target model to verify earlier steps while new reasoning steps are generated in parallel. However, these approaches are developed and evaluated primarily for language-only models, and our experiments indicate that directly extending them to vision-language models provides limited benefits. To better understand this limitation, we conduct a simple diagnostic study using 1,000 samples from the MathVista dataset, focusing on the behavior of speculative reasoning when there is a large capability gap between the draft and target models. We evaluate the SpecReason Pan et al. (2025) method using Qwen3-VL-32B as the target model and Qwen3-VL-2B, Qwen3-VL-4B, MiMo-VL-7B-RL, and R-4B as draft models, respectively. We compare the accuracy and speedup of SpecReason against the standard decoding baseline. As shown in Figure 1 (b), SpecReason's accuracy drops from 78% to 43.2%. One contributing factor is perceptual error in vision-language reasoning, where failures stem from incorrect or incomplete visual grounding rather than flawed logic. For example, in the blue tape positioning case, the visual cue exists in the image but is mis-recognized by the draft model, leading to a coherent yet visually unsupported reasoning trace that complicates verification. This highlights that visual reasoning fundamentally requires grounding in images, which standard language-model drafts tend to ignore.

We find that when the target model is a vision–language model, its judgments of reasoning correctness follow different logic and distributions from those of standard language models. As a result, directly using the log-probabilities of its step-level scores as a reliability signal becomes inaccurate and unstable.

Reinforcement-Learning Based Large Model Training Reinforcement learning (RL) has emerged as a central component of the post-training pipeline for foundation models, including both LLMs and VLMs (Grattafiori et al., 2024; Bai et al., 2025; Liu et al., 2024a; Guo et al., 2025; Comanici et al., 2025). Through alignment, recent models such as DeepSeek-R1 (Liu et al., 2024a; Guo et al., 2025) have demonstrated substantial performance improvements across a broad range of downstream tasks, particularly those involving complex multi-step reasoning. Existing RL-based alignment approaches can be broadly categorized based on the source of their reward signals into two paradigms: reinforcement learning from human feedback (RLHF) (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023; Sun et al., 2023; Ethayarajh et al., 2024) and reinforcement learning with verifiable rewards (RLVR) (Gao et al., 2024a; Zeng et al., 2025; Guan et al., 2025; Wang et al., 2025b; Cui et al., 2025; Yu et al., 2025). RLHF leverages human preference annotations to guide alignment, whereas RLVR derives reward signals from automated or programmatic verification mechanisms.

Despite its effectiveness, RLHF depends on human preference annotations and primarily captures relative preferences rather than objective correctness. Extensions such as reinforcement learning from AI feedback (RLAIF) (Bai et al., 2022) help reduce annotation costs, but preference-based signals remain limited in their ability to reliably evaluate correctness for reasoning-intensive tasks, including mathematical problem solving and code generation. In contrast, RLVR relies on task-specific verifiers (Cheng et al., 2024; Kydlíček, 2025) to assess correctness during training, reducing dependence on human feedback and enabling more reliable correctness evaluation.

Following the RLVR paradigm, our work investigates verifier-guided reward design for speculative reasoning, with the goal of improving the quality of draft-generated reasoning steps and, in turn, enhancing both accuracy and latency in speculative reasoning pipelines, as described in Section 3.

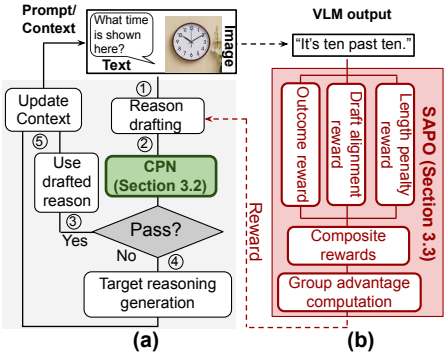

Figure 2: (a) The DREAM-R decoding step. All the step numbers are highlighted in circles. (b) A highlight of SAPO training.

3 METHOD

In this section, we first present the *Contrastive Probability Normalization* (CPN) for validating drafted reasoning, followed by *Speculative Alignment Policy Optimization* (SAPO), an RL-based alignment method that trains the draft model to produce reasoning steps consistent with multimodal evidence. Finally, we introduce *Fully Parallel Speculative Reasoning* (FPSR), which concurrently performs drafting, target generation, and verification to maximize hardware utilization and reduce wall-clock latency.

3.1 OVERVIEW OF DREAM-R

Figure 2 illustrates the overall architecture of DREAM-R. As shown in Figure 2(a), the draft model M_{draft} first processes the input sample and generates a CoT reasoning trace (Step 1). The target model M_{target} then performs CPN to assess the quality of this reasoning (Step 2). If the reasoning produced by M_{draft} passes the verification check (Step 3), it is fed back to M_{draft} to update the CoT context for the next execution round (Step 5). Otherwise, if the reasoning fails the CPN check (Step 4), the target model M_{target} directly generates the CoT reasoning from the input.

3.2 CONTRASTIVE PROBABILITY NORMALIZATION

As illustrated in Figure 2(a), CPN is used to assess the quality of the reasoning x_{reason} generated by M_{draft}. DREAM-R adopts a verification scheme in which x_{reason}, together with a predefined prompt template x_{prompt}, is sent to the target model M_{target}. The prompt x_{prompt} instructs M_{target} to evaluate the quality of x_{reason} by selecting one of two keywords: "positive" or "negative". As shown in Figure 3, let s_+ and s_- denote the predicted probabilities of the keywords "positive" and "negative", respectively. The verification decision is then made by computing the ratio $\rho = \frac{s_+}{s_+ + s_-}$.

This ratio ρ captures the relative dominance between the two signals. When the verifier assigns comparable probabilities to both keywords (e.g., a near 1:1 balance), ρ approaches 0.5, indicating insufficient confidence in a positive judgment. Conversely, larger values of ρ reflect stronger confidence in the positive assessment. Empirically, we adopt a conservative policy: a reasoning step x_{reason} proposed by M_{draft} is accepted only if ρ exceeds a predefined threshold $\alpha = 0.7$; otherwise, the reasoning is rejected.

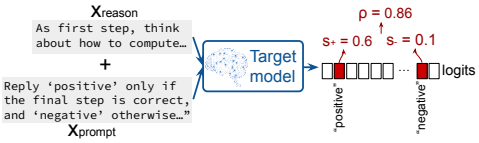

Figure 3: An example on CPN.

Compared with utility score–based methods such as those proposed in Pan et al. (2025), which require M_{target} to assign a discrete score (e.g., from 0 to 9) to the proposed reasoning and accept it if the score exceeds a predefined threshold, CPN provides a more principled and fine-grained assessment. Instead of relying on an arbitrary scalar rating, CPN directly captures the confidence of M_{target}'s judgment by examining the predicted probabilities of the verification keywords. This probabilistic formulation enables a smoother and more interpretable acceptance criterion, reduces sensitivity to prompt-specific scoring biases, and avoids ambiguity introduced by coarse-grained discrete scores. As a result, CPN yields more stable and reliable verification decisions across different reasoning steps and inputs, as demonstrated in Section 4.

3.3 SPECULATIVE ALIGNMENT POLICY OPTIMIZATION

To improve the quality of reasoning produced by M_{draft}, we introduce Speculative Alignment Policy Optimization (SAPO), a policy optimization framework designed for synchronous speculative reasoning. SAPO extends group-based policy optimization by incorporating alignment-aware reward signals that explicitly measure the consistency between the draft model's proposed reasoning steps and the target model's verified reasoning trajectory.

As described in Section 3.1, during LRM decoding, M_{draft} iteratively proposes candidate reasoning steps, which are verified by M_{target} via CPN. This process yields a complete reasoning trace and a final prediction for each input. SAPO assigns a composite verification reward to the resulting reasoning trajectory x_{reason}, consisting of an outcome reward R_{outcome} that reflects final-answer correctness, a draft alignment reward R_{draft} that measures the proportion of draft-generated steps accepted by the target model, and a length penalty reward term R_{length} that penalizes unnecessarily long reasoning traces.

$$R_{\text{outcome}} = \begin{cases} 1, & \text{if } x_{\text{reason}} \text{ yields a correct answer,} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

while R_{draft} is defined as $R_{\text{draft}} = \frac{N_{\text{accepted}}}{N_{\text{draft}}}$, where N_{accepted} and N_{draft} denote the number of rounds in which the draft reasoning is accepted or rejected by the target model. Finally, to discourage unnecessarily long CoT generation, SAPO employs a length-dependent penalty:

$$R_{\text{length}} = \begin{cases} 0, & L \le C, \\ \min\big(1,\ k \cdot (L - C)\big), & L > C. \end{cases} \tag{2}$$

where L denotes the token length of the reasoning and C is a predefined threshold. Equation 2 imposes a penalty proportional to the excess length beyond C. The overall reward is computed as

$$R = w_1 \cdot R_{\text{outcome}} + w_2 \cdot R_{\text{draft}} - w_3 \cdot R_{\text{length}}, \tag{3}$$

where w_1, w_2, and w_3 determine the relative contribution of each component. After computing the rewards, SAPO updates the draft model using a simplified Group Relative Policy Optimization

(GRPO) objective Shao et al. (2024), with the group baseline defined as $b_{\mathcal{G}} = \frac{1}{n}\sum_{i=1}^{N} R_i$, where N is the number of input samples. R_i is the reward corresponding to the i-th draft reason. The group-relative advantage for each trajectory is $A_i = R_i - b_{\mathcal{G}}$. To improve stability, advantages A_i are normalized within the group, leading to $\tilde{A}_i = \frac{A_i}{\sigma_{\mathcal{G}}+\delta}$, where $\sigma_{\mathcal{G}}$ denotes the standard deviation of $\{A_i\}$, and δ is a small constant introduced to improve stability. The policy is updated using a clipped ratio objective:

$$L_i(\theta) = \min\Big(u_i(\theta)\tilde{A}_i,\ \text{clip}(u_i(\theta),\ 1-\epsilon,\ 1+\epsilon)\,\tilde{A}_i\Big) \tag{4}$$

where θ denotes the parameter set. $u_i(\theta) = \frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_{\text{old}}}(a_i|x_i)}$ that follows the definition in Shao et al. (2024), the overall SAPO process is highlighted in Figure 2 (b).

3.4 Fully Parallel Speculative Reasoning

While Sections 3.2 and 3.3 focus on the algorithmic design of DREAM-R to improve the reasoning quality of the LRM, this section introduces an advanced Fully Parallel Speculative Reasoning (FPSR) approach that enables high parallelism across the drafting, verification, and decoding stages to further reduce latency.

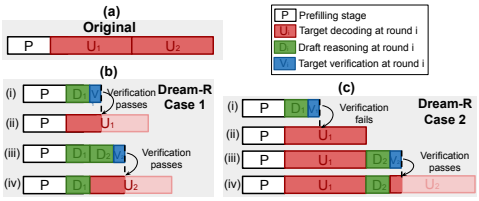

Figure 4: Timeline comparison of (a) Standard Autoregressive Reasoning, and (b) Fully Parallel Speculative Reasoning (FPSR). FPSR pipelines the drafting (Green), verification (Blue), and target generation (Red) stages, ensuring the GPU is never idle. Unlike lookahead methods, we allow 'rollback' (c) to recover from verification failures without recomputation.

The FPSR method is illustrated in Figure 4. For ease of exposition, we assume a total of two rounds of reasoning. Figure 4 (a) depicts the baseline approach, in which the target model M_{target} sequentially performs the prefilling stage P, generates intermediate reasoning steps U_1 and produces the final answer U_2, resulting in high end-to-end generation latency.

In contrast, Figure 4 (b) shows the FPSR workflow. At each reasoning round, the draft model M_{draft} continuously proposes candidate reasoning steps (e.g., D_1 in Figure 4 (b)(i)), while the target model M_{target} concurrently generates its own reasoning (U_1 in Figure 4 (b)(ii)). Once M_{draft} completes a reasoning step, it is immediately verified (V_1 in Figure 4 (b)(i)). If the draft reasoning is accepted, M_{target} early-terminates its ongoing reasoning generation (U_1 in Figure 4 (b)(ii)). This process repeats for subsequent reasoning rounds, assuming the draft reasoning continues to pass verification. As a result, FPSR significantly reduces overall latency compared to the baseline scheme shown in Figure 4 (a).

In contrast, when the draft reasoning generated by M_{draft} fails verification (Figure 4(c)), the subsequent round cannot begin until the current update stage completes (U_1 in Figure 4 (c)(ii)). Once U_1 finishes, a second reasoning round is initiated (D_2 in Figure 4 (c)(iii)) and then verified (V_2 in Figure 4 (c)(iii)). Assuming this verification succeeds, the final result is produced in the D_2 (Figure 4 (c)(iv)).

Compared to prior parallelization approaches that only overlap target generation Fu et al. (2025) or speculative reasoning schemes that perform step verification largely sequentially as shown in Figure 4 (a), FPSR maintains continuous parallelism across all three components: the draft model proposing candidate reasoning steps, the target model extending the trusted reasoning prefix, and the verifier determining acceptance or termination. By tightly interleaving proposal, verification, and fallback, FPSR significantly reduces end-to-end wall-clock latency for reasoning-intensive decoding, while fully preserving the target model's answer quality.

3.5 DREAM-R System Implementation

We implement DREAM-R as an asynchronous speculative reasoning engine with four coordinated components: a *Drafter*, *Verifier*, *Corrector*, and a centralized *Manager*. The Drafter continuously proposes speculative steps whenever `can_draft` is active, each of which is synchronously verified for acceptance. Upon rejection, the Manager cancels all in-flight speculative tasks within a bounded lookahead window, rolls the state back to the latest verified prefix, and invokes the Corrector to

generate a non-speculative replacement step, which is then committed before drafting resumes. By decoupling drafting from verification while enforcing ordered commitment and rollback, the system achieves high speculative throughput while strictly preserving stepwise correctness and logical consistency. Detailed prompts are provided in the Appendix C for completeness.

4 EXPERIMENTS

We evaluate our fully parallel speculative reasoning framework on four well-established multimodal reasoning benchmarks, MathVerse Zhang et al. (2024), MMBench Liu et al. (2024c), RealWorldQA xAI (2024), and MMMU Yue et al. (2024), using their full evaluation splits. For each benchmark, we report accuracy, acceptance rate, and speedup relative to target model autoregressive decoding. We experiment with multiple draft and target model combinations. The draft model is chosen from Qwen3-VL-2B (Q2B) Yang et al. (2025a), Qwen3-VL-4B (Q4B), R-4B (R4B) Yang et al. (2025b), and MiMo-VL-7B-RL (M7B-RL) Xiaomi et al. (2025). The target model is selected from either Qwen3-VL-32B (Q32B) Yang et al. (2025a) or Qwen3-VL-235B-A22B (Q235B). All draft and target models are used in their thinking versions.

The draft model is trained using SAPO on a mixture of multimodal reasoning datasets with step-level annotations, including Geo3K Lu et al. (2021), OCR-VQA Mishra et al. (2019), and the ScienceQA Lu et al. (2022) dev split. The target model is kept frozen throughout training, while the draft model is fully optimized.

To evaluate DREAM-R, we compare it against three baseline methods. The first, *Standard SD*, applies conventional speculative decoding Leviathan et al. (2023) to LRMs to accelerate generation. We further compare against two recent speculative reasoning approaches, SpecReason Pan et al. (2025) and LR Fu et al. (2025). Finally, to isolate the contribution of SAPO, we include a variant of DREAM-R without SAPO, termed *DREAM-R-NS*, which incorporates CPN and FPSR but omits SAPO. Specifically, we adopt three evaluation metrics: outcome accuracy (Acc.), the acceptance rate of draft reasoning steps (Acpt.), and execution latency speedup measured relative to standard autoregressive decoding.

Table 1: Model performance (%) across four benchmarks.

Model	MathVerse	MMBench	RealWorldQA	MMMU
Q32B	76.00	83.40	75.55	77.85
Q235B	85.40	83.20	75.17	71.59
Q2B	53.94	62.90	63.29	59.27
Q4B	65.40	71.00	68.98	59.60
M7B-RL	69.80	78.50	58.80	66.80
R4B	75.40	74.40	66.48	65.20

4.1 EVALUATION RESULTS

Table 1 summarizes the autoregressive accuracy of all candidate models and serves as the baseline for evaluating speculative reasoning. The target models, Qwen3-VL-32B and Qwen3-VL-235B, exhibit strong baseline performance, achieving 76.00% and 85.40% on MathVerse, 83.40% and 83.20% on MMBench, and over 75.00% on RealWorldQA. In contrast, draft models are substantially weaker, with Qwen3-VL-2B and Qwen3-VL-4B reaching only 53.94% and 65.40% on MathVerse, respectively. This performance gap highlights the challenge of effective speculative reasoning when draft models differ significantly from the target.

Table 2 demonstrates that our framework preserves accuracy close to the vanilla baseline while delivering substantial acceleration. Using Qwen3-VL-32B as the target and Qwen3-VL-2B as the draft, the supervised variant achieves 74.30% on MathVerse and 80.32% on MMBench, with speedups ranging from $1.8\times$ to $2.2\times$ across benchmarks. The RL-enhanced variant further improves acceptance, yielding speedups of up to $2.38\times$ on MathVerse and $2.48\times$ on MMBench and RealWorldQA, while maintaining high accuracy. In contrast, SpecReason underperforms in the VLM setting because its verification relies on discrete, score-based judgments that are brittle under multimodal uncertainty. Small draft models are particularly prone to hallucinations, and the resulting noisy scores can interfere with the target verifier. Moreover, discrete scoring often produces closely clustered or extreme values, making threshold-based accept/reject decisions highly unstable. These factors jointly lead to low acceptance rates and degraded final accuracy.

Table 2: Performance evaluations across different models and datasets.

Target	Draft	Method	MathVerse			MMBench			RealWorldQA			MMMU		
			Acc.%	Acpt.%	Speedup	Acc.%	Acpt.%	Speedup	Acc.%	Acpt.%	Speedup	Acc.%	Acpt.%	Speedup
Q32B	Q2B	Standard SD	75.99	–	1.09 ×	79.11	–	1.15 ×	69.72	–	1.12 ×	76.15	–	1.15 ×
		SpecReason Pan et al. (2025)	44.57	14.60	1.26 ×	62.49	24.26	1.43 ×	39.80	21.35	1.41 ×	46.88	16.04	1.21 ×
		LR Fu et al. (2025)	72.36	35.21	1.53 ×	74.52	71.75	1.73 ×	72.97	66.85	2.13 ×	75.57	41.63	1.72 ×
		DREAM-R-NS	74.30	35.70	1.86 ×	80.32	69.99	1.77 ×	75.55	60.06	2.19 ×	77.85	42.46	2.03 ×
		DREAM-R	75.98	52.61	2.38 ×	92.65	63.52	1.86 ×	76.10	67.98	2.48 ×	85.79	56.44	2.19 ×
	Q4B	Standard SD	71.31	–	1.04 ×	83.90	–	1.11 ×	73.48	–	1.08 ×	77.18	–	1.10 ×
		SpecReason	40.22	20.48	0.94 ×	73.58	32.13	1.28 ×	69.86	48.51	1.23 ×	53.73	32.86	1.24 ×
		LR	72.08	40.28	1.55 ×	76.41	48.92	1.58 ×	72.12	49.29	1.66 ×	58.96	34.35	1.66 ×
		DREAM-R-NS	74.80	37.33	1.64 ×	79.42	49.15	2.37 ×	81.38	47.43	1.71 ×	66.00	34.23	1.84 ×
		DREAM-R	81.60	31.13	1.84 ×	89.44	53.28	2.48 ×	81.39	52.36	1.80 ×	77.37	42.37	1.91 ×
	M7B-RL	Standard SD	73.20	–	1.02 ×	86.90	–	1.05 ×	71.10	–	1.03 ×	74.00	–	1.04 ×
		SpecReason	48.30	18.20	0.86 ×	45.20	15.10	0.80 ×	51.60	23.40	0.83 ×	50.10	27.90	0.84 ×
		LR	72.10	32.50	1.40 ×	84.30	35.60	1.32 ×	70.50	39.70	1.38 ×	69.20	37.30	1.36 ×
		DREAM-R-NS	76.50	34.10	1.55 ×	88.10	60.20	1.48 ×	73.00	41.50	1.52 ×	71.10	39.80	1.49 ×
		DREAM-R	79.80	36.70	1.70 ×	90.40	63.80	1.62 ×	75.20	44.00	1.66 ×	73.50	42.10	1.61 ×
	R4B	Standard SD	72.10	–	1.07 ×	82.10	–	1.09 ×	72.10	–	1.09 ×	68.15	–	1.08 ×
		SpecReason	46.35	21.80	1.27 ×	71.20	29.30	1.24 ×	62.10	38.70	1.26 ×	54.30	40.50	1.22 ×
		LR	71.25	43.90	1.67 ×	77.35	47.80	1.78 ×	70.95	53.80	1.82 ×	67.10	46.90	1.80 ×
		DREAM-R-NS	74.10	45.60	1.83 ×	80.15	52.60	2.18 ×	72.80	50.90	1.93 ×	69.05	49.20	1.88 ×
		DREAM-R	78.45	49.70	1.96 ×	83.60	57.10	2.31 ×	75.25	56.10	2.03 ×	71.40	53.60	1.98 ×
Q235B	Q2B	Standard SD	82.22	–	1.11 ×	83.20	–	1.06 ×	75.17	–	1.16 ×	71.59	–	1.07 ×
		SpecReason	55.10	41.55	1.47 ×	73.29	68.47	1.13 ×	67.87	68.75	1.30 ×	56.71	35.65	1.07 ×
		LR	66.47	48.53	1.71 ×	80.50	66.42	1.93 ×	74.51	52.12	1.61 ×	71.11	36.66	1.36 ×
		DREAM-R-NS	72.70	49.80	2.22 ×	81.73	66.80	2.28 ×	75.17	57.29	2.13 ×	71.59	37.20	1.52 ×
		DREAM-R	80.00	51.82	2.28 ×	88.45	71.45	2.31 ×	77.89	68.05	2.26 ×	78.33	39.21	1.60 ×
	Q4B	Standard SD	81.24	–	1.11 ×	81.31	–	1.09 ×	75.36	–	1.10 ×	69.83	–	1.11 ×
		SpecReason	47.35	29.91	1.23 ×	69.74	39.18	1.22 ×	55.43	36.88	1.23 ×	59.72	51.86	1.22 ×
		LR	70.70	57.65	1.68 ×	79.10	37.34	1.45 ×	69.88	48.22	1.48 ×	68.20	53.70	1.48 ×
		DREAM-R-NS	80.40	56.94	1.77 ×	81.32	39.01	1.65 ×	68.52	44.23	1.51 ×	71.69	53.29	1.55 ×
		DREAM-R	84.00	57.96	1.79 ×	82.20	37.96	1.73 ×	72.53	47.22	1.53 ×	72.53	57.82	1.58 ×
	M7B-RL	Standard SD	81.20	–	1.08 ×	82.70	–	1.04 ×	75.10	–	1.07 ×	71.50	–	1.07 ×
		SpecReason	64.40	36.52	0.80 ×	17.10	14.82	0.71 ×	11.30	24.79	1.09 ×	48.60	23.77	0.63 ×
		LR	72.10	37.91	1.46 ×	74.70	35.46	1.46 ×	73.30	25.24	1.40 ×	79.80	23.03	1.45 ×
		DREAM-R-NS	78.00	36.80	1.62 ×	79.60	35.45	1.62 ×	76.10	25.56	1.56 ×	81.00	23.69	1.65 ×
		DREAM-R	82.10	40.20	1.78 ×	82.30	47.00	1.78 ×	78.40	28.50	1.72 ×	83.10	25.00	1.75 ×
	R4B	Standard SD	80.10	–	1.10 ×	80.95	–	1.08 ×	74.20	–	1.09 ×	68.05	–	1.10 ×
		SpecReason	49.20	27.40	1.21 ×	67.10	37.50	1.20 ×	57.10	35.60	1.21 ×	60.10	49.80	1.21 ×
		LR	69.85	54.20	1.65 ×	78.25	35.80	1.43 ×	69.05	46.90	1.47 ×	67.30	51.90	1.46 ×
		DREAM-R-NS	79.10	53.70	1.74 ×	79.85	37.90	1.62 ×	67.95	42.70	1.49 ×	70.25	50.70	1.54 ×
		DREAM-R	82.35	55.90	1.78 ×	81.95	36.90	1.71 ×	71.80	45.90	1.52 ×	71.95	54.10	1.57 ×

4.2 Ablation Studies

4.2.1 Ablation on Reward Function Design

Based on Equation 3, we examine how different weighting schemes for draft alignment reward R_{draft}, the outcome reward R_{outcome}, and length penalty reward term R_{length} affect the behavior of the draft model. Unless stated otherwise, the baseline setting uses equal weights for all three components, and we vary the relative importance w_1, w_2, w_3 of each term to analyze its impact. According to Figure 5, the balanced weighting strategy with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$ delivers the best overall performance across benchmarks. Specifically, on MathVerse it achieves 75.98% accuracy with a 52.61% acceptance rate and a $2.38\times$ speedup, while on MMBench it attains 82.65% accuracy with a 73.52% acceptance rate and a $1.86\times$ speedup.

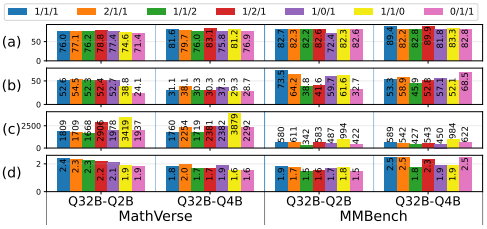

Figure 5: Impact of RL reward weighting on (a) output accuracy, (b) reasoning acceptance rate, (c) reasoning token length, and (d) speedup. Values separated by "/" indicate the weights w_1, w_2, w_3.

Increasing the weight w_2 on R_{draft} while keeping other terms fixed leads to only marginal accuracy changes on MathVerse (77.12%) and a modest increase in acceptance (54.54%), but with reduced speedup ($2.31\times$). On MMBench, acceptance instead drops to 64.15%, suggesting that emphasizing verification alignment alone does not consistently improve verifier agreement in multimodal reasoning.

Adjusting the balance between R_{outcome} and R_{length} introduces more pronounced trade-offs. Emphasizing R_{outcome} increases MathVerse accuracy to 78.80%, but substantially lengthens generation

(2906 tokens) and reduces speedup (2.16×). In contrast, strengthening R_{length} shortens outputs on MathVerse (1668 tokens) but lowers acceptance to 52.26% and causes a sharp acceptance drop on MMBench (38.78%). These results indicate that while outcome-level supervision is beneficial, overly aggressive or imbalanced regularization can negatively affect verifier acceptance.

Employing a stronger draft model (Qwen3-VL-4B) further improves both accuracy and acceptance, resulting in more stable speculative decoding. Across all benchmarks, our approach consistently outperforms existing baselines: LR achieves moderate acceleration but often at the cost of accuracy, while SpecReason suffers from low acceptance when the draft is substantially weaker. In contrast, DREAM-R-NS and DREAM-R simultaneously maintains high accuracy and high acceptance, achieving the most favorable balance between efficiency and correctness.

Scaling to the larger Qwen3-VL-235B target shows that DREAM-R-NS and DREAM-R generalize well across model sizes. When using Qwen3-VL-2B as the draft, DREAM-R-NS achieves speedups ranging from 1.6× to 2.3× while maintaining accuracy close to the vanilla Qwen3-VL-235B baseline, which spans 71.59% to 85.40%. The RL-enhanced variant further improves acceptance by approximately 15%–25%, leading to additional acceleration across all benchmarks. With Qwen3-VL-4B as the draft, accuracy closely matches the vanilla target performance on MMBench and RealWorldQA, while speculative efficiency remains robust, delivering speedups of 1.5× to 1.8×.

4.2.2 Ablation on FPSR

Overall, the results demonstrate that our speculative decoding framework consistently achieves the best trade-off among accuracy retention, acceptance rate, and decoding efficiency. Both the supervised and RL-enhanced variants preserve accuracy close to the vanilla baseline in Table 1 while delivering substantial and stable speedups across four multimodal reasoning benchmarks.

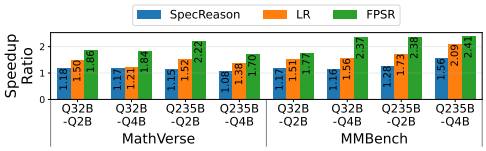

Figure 6: Impact of scheduling methods.

We analyze the impact of FPSR by comparing it with two baselines: SpecReason Pan et al. (2025), LR Fu et al. (2025), and FPSR. SpecReason does not incorporate explicit scheduling mechanisms and instead directly follows the baseline scheduling strategy illustrated in Figure 4(a). In contrast, LR allows the draft and target models to generate in parallel, overlapping their computation to improve hardware utilization and reduce end-to-end latency. However, verification is deferred until generation finishes, delaying feedback and preventing early rejection of incorrect draft steps. As shown in Figure 6, for the Qwen3-VL-2B to Qwen3-VL-32B setting, speedup increases from 1.18× and 1.17× with SpecReason Pan et al. (2025) on MathVerse and MMBench to 1.50× and 1.51× with FPSR, and further to 1.86× and 1.77× under full parallelism. With a stronger draft model (Qwen3-VL-4B), FPSR achieves speedups of 1.84× and 2.37× on the two benchmarks. When the target is Qwen3-VL-235B, the benefit becomes even more significant. For example, with a 2B draft, speedup increases from 1.15× and 1.28× to 1.52× and 1.73× under LR Fu et al. (2025), and to 2.22× and 2.38× under FPSR.

4.2.3 Ablation on Acceptance Threshold for CPN

We analyze how the threshold α in CPN (Section 3.2) affects the performance of DREAM-R. By comparing ϕ with α, this threshold controls the acceptance ratio of drafted reasoning steps. A lower α makes the verifier more permissive, leading to higher acceptance rates and the largest speedups across model pairs, but at the cost of reduced accuracy due to accepting lower-quality reasoning. In contrast, a higher α makes the verifier more conservative, improving accuracy while increasing latency. As shown in Figure 7, setting α to 0.7 achieves the best overall trade-off between accuracy and speedup

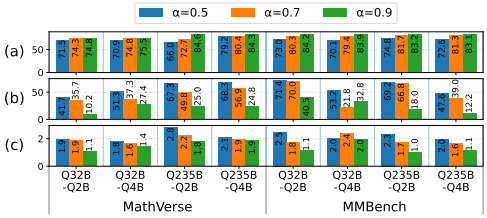

Figure 7: Impact of α on (a) accuracy (b) reasoning acceptance rate and (c) speedup.

across different models and datasets. This indicates
that the acceptance threshold serves as an effective control knob for tuning system behavior: lower thresholds prioritize efficiency, higher thresholds prioritize accuracy, and intermediate thresholds strike a strong balance.

5 CONCLUSION

We propose *DREAM-R*, a multimodal speculative reasoning framework that accelerates reasoning-heavy decoding while preserving target-model accuracy. Experiments on four multimodal reasoning benchmarks show up to $2.48\times$ speedup with accuracy comparable to vanilla target decoding, outperforming prior speculative reasoning baselines.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Zachary Ankner, Rishab Parthasarathy, Aniruddha Nrusimha, Christopher Rinard, Jonathan Ragan-Kelley, and William Brandon. Hydra: Sequentially-dependent draft heads for medusa decoding. *arXiv preprint arXiv:2402.05109*, 2024.

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.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

Mark Chen. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. Sequoia: Scalable, robust, and hardware-aware speculative decoding. *arXiv preprint arXiv:2402.12374*, 2024.

Ziyi Chen, Xiaocong Yang, Jiacheng Lin, Chenkai Sun, Kevin Chen-Chuan Chang, and Jie Huang. Cascade speculative drafting for even faster llm inference. *arXiv preprint arXiv:2312.11462*, 2023.

Yao Cheng, Jianfeng Chen, Jie Chen, Li Chen, Liyu Chen, Wentao Chen, Zhengyu Chen, Shijie Geng, Aoyan Li, Bo Li, et al. Fullstack bench: Evaluating llms as full stack coders. *arXiv preprint arXiv:2412.00535*, 2024.

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.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *ArXiv*, abs/2502.01456, 2025. URL https://api.semanticscholar.org/CorpusID:276107672.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layer skip: Enabling early exit inference and self-speculative decoding. *arXiv preprint arXiv:2404.16710*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Yichao Fu, Rui Ge, Zelei Shao, Zhijie Deng, and Hao Zhang. Scaling speculative decoding with lookahead reasoning. *arXiv preprint arXiv:2506.19830*, 2025.

Mukul Gagrani, Raghavv Goel, Wonseok Jeon, Junyoung Park, Mingu Lee, and Christopher Lott. On speculative decoding for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8285–8289, 2024.

Mugilan Ganesan, Shane Segal, Ankur Aggarwal, Nish Sinnadurai, Sean Lie, and Vithursan Thangarasa. Massv: Multimodal adaptation and self-data distillation for speculative decoding of vision-language models. *arXiv preprint arXiv:2505.10526*, 2025.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weiling Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. On designing effective rl reward at training time for llm reasoning. *ArXiv*, abs/2410.15115, 2024a. URL https://api.semanticscholar.org/CorpusID:273502123.

Xiangxiang Gao, Weisheng Xie, Yiwei Xiang, and Feng Ji. Falcon: Faster and parallel inference of large language models through enhanced semi-autoregressive drafting and custom-designed decoding tree. *arXiv preprint arXiv:2412.12639*, 2024b.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *ArXiv*, abs/2501.04519, 2025. URL https://api.semanticscholar.org/CorpusID:275357888.

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

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D Lee, and Di He. Rest: Retrieval-based speculative decoding. *arXiv preprint arXiv:2311.08252*, 2023.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Yunhai Hu, Zining Liu, Zhenyuan Dong, Tianfan Peng, Bradley McDanel, and Sai Qian Zhang. Speculative decoding and beyond: An in-depth survey of techniques. *arXiv preprint arXiv:2502.19732*, 2025a.

Yunhai Hu, Tianhua Xia, Zining Liu, Rahul Raman, Xingyu Liu, Bo Bao, Eric Sather, Vithursan Thangarasa, and Sai Qian Zhang. Dream: Drafting with refined target features and entropy-adaptive cross-attention fusion for multimodal speculative decoding. *arXiv preprint arXiv:2505.19201*, 2025b.

Jialiang Kang, Han Shu, Wenshuo Li, Yingjie Zhai, and Xinghao Chen. Vispec: Accelerating vision-language models with vision-aware speculative decoding. *arXiv preprint arXiv:2509.15235*, 2025.

Hynek Kydlíček. Math-Verify: Math Verification Library, 2025. URL https://github.com/huggingface/math-verify.

Minjae Lee, Wonjun Kang, Byeongkeun Ahn, Christian Classen, Minghao Yan, Hyung Il Koo, and Kangwook Lee. In-batch ensemble drafting: Robust speculative decoding for LVLMs. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025. URL https://openreview.net/forum?id=ffDhpmwqdu.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Bohan Li, Hankun Wang, Situo Zhang, Yiwei Guo, and Kai Yu. Fast and high-quality auto-regressive speech synthesis via speculative decoding. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*, 2024b.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*, 2025b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Fangcheng Liu, Yehui Tang, Zhenhua Liu, Yunsheng Ni, Kai Han, and Yunhe Wang. Kangaroo: Lossless self-speculative decoding via double early exiting. *arXiv preprint arXiv:2404.18911*, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024c.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Bradley McDanel, Sai Qian Zhang, Yunhai Hu, and Zining Liu. Pipespec: Breaking stage dependencies in hierarchical llm decoding. *arXiv preprint arXiv:2505.01572*, 2025.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating generative large language model serving with tree-based speculative inference and verification. *arXiv preprint arXiv:2305.09781*, 2023.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 947–952. IEEE, 2019.

Jie Ou, Yueming Chen, and Wenhong Tian. Lossless acceleration of large language model via adaptive n-gram parallel decoding. *arXiv preprint arXiv:2404.08698*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rui Pan, Yinwei Dai, Zhihao Zhang, Gabriele Oliaro, Zhihao Jia, and Ravi Netravali. Specreason: Fast and accurate inference-time compute via speculative reasoning. *arXiv preprint arXiv:2504.07891*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31, 2018.

Lawrence Stewart, Matthew Trager, Sujan Kumar Gonugondla, and Stefano Soatto. The n-grammys: Accelerating autoregressive inference with learning-free batched speculation. *arXiv preprint arXiv:2411.03786*, 2024.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan J. Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *ArXiv*, abs/2009.01325, 2020. URL `https://api.semanticscholar.org/CorpusID:221665105`.

Hanshi Sun, Zhuoming Chen, Xinyu Yang, Yuandong Tian, and Beidi Chen. Triforce: Lossless acceleration of long sequence generation with hierarchical speculative decoding. *arXiv preprint arXiv:2404.11912*, 2024.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented rlhf. *ArXiv*, abs/2309.14525, 2023. URL `https://api.semanticscholar.org/CorpusID:262824780`.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. *ArXiv*, abs/2504.20571, 2025b. URL `https://api.semanticscholar.org/CorpusID:278171513`.

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.

xAI. Realworldqa dataset. `https://huggingface.co/datasets/xai-org/RealworldQA`, 2024. Version 1, CC BY-ND 4.0 License.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3909–3925, 2023.

Heming Xia, Yongqi Li, Jun Zhang, Cunxiao Du, and Wenjie Li. Swift: On-the-fly self-speculative decoding for llm inference acceleration. *arXiv preprint arXiv:2410.06916*, 2024.

LLM Xiaomi, Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, et al. Mimo: Unlocking the reasoning potential of language model–from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*, 2025.

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*, 2025a.

Nan Yang, Tao Ge, Liang Wang, Binxing Jiao, Daxin Jiang, Linjun Yang, Rangan Majumder, and Furu Wei. Inference with reference: Lossless acceleration of large language models. *arXiv preprint arXiv:2304.04487*, 2023.

Qi Yang, Bolin Ni, Shiming Xiang, Han Hu, Houwen Peng, and Jie Jiang. R-4b: Incentivizing general-purpose auto-thinking capability in mllms via bi-mode annealing and reinforce learning. *arXiv preprint arXiv:2508.21113*, 2025b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Honglin Yu, Weinan Dai, Yuxuan Song, Xiang Wei, Haodong Zhou, Jingjing Liu, Wei Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yong-Xu Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv*, abs/2503.14476, 2025. URL https://api.semanticscholar.org/CorpusID:277104124.

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*, pp. 9556–9567, 2024.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *ArXiv*, abs/2503.18892, 2025. URL https://api.semanticscholar.org/CorpusID:277940848.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. Draft & verify: Lossless large language model acceleration via self-speculative decoding. *arXiv preprint arXiv:2309.08168*, 2023.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593, 2019. URL https://api.semanticscholar.org/CorpusID:202660943.

A Appendix

B Implementation Details

B.1 Evaluation Setup

All experiments are conducted on NVIDIA L40S GPUs. The target model is deployed on four L40S GPUs with AWQ INT4 quantization, while each draft model runs on two L40S GPUs. We use a fixed speculative lookahead window of 4 and set the acceptance threshold α to 0.7.

B.2 Training Setup

Table 3 summarizes the key hyperparameters used in DREAM-R. For decoding, we fix the acceptance threshold to 0.7 and limit the maximum prompt and response lengths to 2048 and 8192 tokens, respectively, ensuring stable speculative verification while bounding the cost of long reasoning traces.

All models are trained with a global batch size of 64 and a learning rate of 1×10^{-6} using the Adam optimizer. Training is performed for 15 epochs with BF16 precision, which provides a good balance between numerical stability and computational efficiency.

SAPO employs a composite reward with equal weighting across its components. Specifically, the outcome reward encourages final answer correctness, the draft alignment reward promotes agreement with target verification, and a length penalty discourages excessively long responses through an overlong buffer mechanism. Unless otherwise stated, these hyperparameters are shared across all experiments.

Table 3: Key hyperparameters used in DREAM-R. Rollout parameters control speculative reasoning behavior, training parameters define draft model optimization, and reward parameters specify the SAPO objective.

Category	Hyperparameter	Value	Description
Rollout	Acceptance threshold	0.7	Confidence threshold for accepting draft steps
Rollout	Max prompt length	2048	Maximum input context length
Rollout	Max response length	8192	Maximum generated reasoning length
Training	Batch size	64	Number of samples per optimization step
Training	Learning rate	1×10^{-6}	Step size for parameter updates
Training	Epochs	15	Number of full passes over training data
Training	Precision	BF16	Numeric precision for training
Reward	Outcome reward weight	1.0	Weight on final answer correctness
Reward	Draft alignment reward weight	1.0	Weight on target verification alignment
Reward	Length penalty reward weight	1.0	Weight on discouraging long reasoning

B.3 Training Cost

Training a single Qwen3-VL-4B draft model with SAPO requires approximately 74 hours on 8×NVIDIA H200 GPUs. The target model is queried via API during rollout and verification. This setup provides a practical balance between training stability and computational cost, enabling scalable reinforcement learning for multimodal speculative reasoning.

C Prompts for DREAM-R

We design two complementary prompts in DREAM-R to separate reasoning generation from verification. The *Reasoning Prompt* (Prompt C) guides both the draft and target models to produce structured reasoning, while the *Scoring Prompt* (Prompt C) is used exclusively by the target model to evaluate correctness and alignment.

Reasoning Prompt

You are a reasoning agent responsible for generating a single coherent reasoning step toward solving the given problem.

Input

{problem}
{image}, {options}

Instructions

- Produce exactly one reasoning step.
- The step must logically follow from all previous steps.
- Do not generate the final answer unless explicitly required.

Output Format

```
<reasoning_step>
```

Scoring Prompt

You are a verification agent evaluating the correctness of the final reasoning step.

Input

- Full problem description
- All previous reasoning steps
- The final candidate step

Decision Rules

- Reply `positive` only if the step is factually correct and logically valid.
- Reply `negative` otherwise.

Output Format

```
positive | negative
```

D PSEUDOCODE OVERVIEW OF FULLY PARALLEL SPECULATIVE REASONING

Algorithm 1 Fully Parallel Speculative Reasoning

Initialize confirmed steps $C \leftarrow []$, generation id $g \leftarrow 0$
Events: `can_draft`=True, `rollback`=False
Queues: draft queue Q, pending tasks P
procedure DRAFTER
    while `can_draft` **do**
        if `rollback` **then** $L \leftarrow []$; `rollback`=False
        end if
        $ctx \leftarrow C \cup L$; draft $s \leftarrow M_s(ctx)$
        push $(s, ctx, |ctx|, g)$ into Q; append s to L
    end while
end procedure
procedure VERIFY(s, ctx, i, g')
    Run verifier v and corrector c in parallel
    if v positive **then** cancel c; return (ok,s,i,g')
    else`can_draft`=False; wait c; return (bad,c,i,g')
    end if
end procedure
procedure MANAGER
    $k \leftarrow 0$
    while True **do**
        if Q not empty **then** pop payload; launch VERIFY; store in P
        end if
        if $k \in P$ and $P[k]$ done **then**
            read result $(flag, step, i, g')$; remove $P[k]$
            if $flag$=ok **then**
                append step to C; $k \leftarrow k + 1$
            else
                append step; $k \leftarrow k + 1$; $g \leftarrow g + 1$
                cancel all P; clear P and Q
                `rollback`=True; `can_draft`=True
            end if
        end if
    end while
end procedure