CASHEW: Stabilizing Multimodal Reasoning via Iterative Trajectory Aggregation

Anonymous ACL submission

Abstract

Vision-language models achieve strong performance across a wide range of multimodal understanding and reasoning tasks, yet their multi-step reasoning remains unstable. Repeated sampling over the same input often produces divergent reasoning trajectories and inconsistent final predictions. To address this, we introduce two complementary approaches inspired by test-time scaling: (1) **CASHEW**, an inference-time framework that stabilizes reasoning by iteratively aggregating multiple candidate trajectories into higher-quality reasoning traces, with explicit visual verification filtering hallucinated steps and grounding reasoning in visual evidence, and (2) **CASHEW-RL**, a learned variant that internalizes this aggregation behavior within a single model. CASHEW-RL is trained using Group Sequence Policy Optimization (GSPO) with a composite reward that encourages correct answers grounded in minimal yet sufficient visual evidence, while adaptively allocating reasoning effort based on task difficulty. This training objective enables robust self-aggregation at inference. Extensive experiments on 13 image understanding, video understanding, and video reasoning benchmarks show significant performance improvements, including gains of up to +26.2 percentage points on ScienceQA and +9.1 percentage points on EgoSchema.

1 Introduction

Vision-language models (VLMs) have become a dominant paradigm for multimodal understanding, enabling unified models that reason jointly over images, videos, and text (Zhang et al., 2024b; Bai et al., 2025b; Zhang et al., 2025a). While recent VLMs achieve strong performance on tasks such as visual question answering and multimodal dialogue, their reasoning processes remain fragile. Multi-step predictions are often sensitive to sampling noise, prone to visual hallucinations (Li et al., 2025a), and inconsistent across inference runs (Li et al., 2023b). This instability is especially evident in complex reasoning scenarios, where small perceptual or interpretive errors can propagate through longer chains of thought.

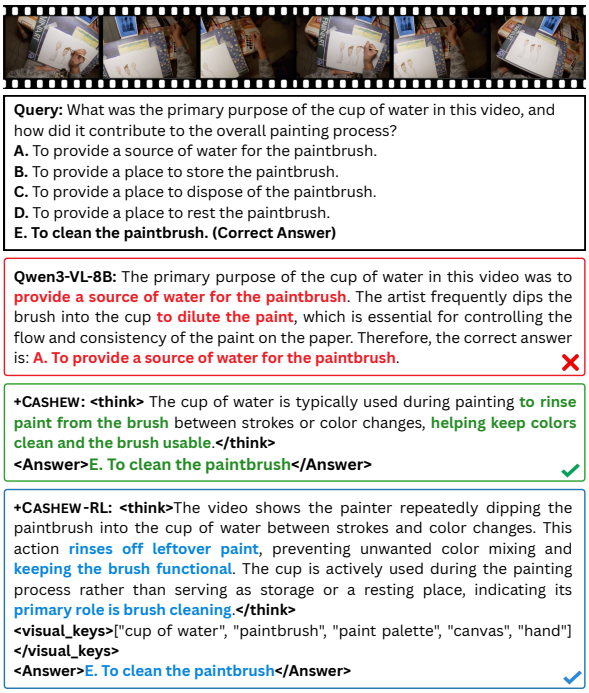

Figure 1: **CASHEW enables robust reasoning through visually grounded iterative aggregation.** Unlike standard vision-language models that rely on single-path reasoning and are prone to hallucinations, CASHEW aggregates multiple reasoning trajectories with explicit visual verification. **CASHEW-RL** further internalizes this aggregation behavior via reinforcement learning.

In large language models, test-time scaling has emerged as an effective strategy to mitigate such instability by allocating additional inference-time computation, such as sampling multiple reasoning trajectories or extending deliberation depth, to obtain more reliable outputs (Wang et al., 2023, 2025a; Chen et al., 2023). Inspired by this paradigm, recent multimodal methods adopt iterative inference mechanisms that repeatedly refine

spatial-temporal attention and textual predictions across multiple passes to improve reasoning consistency (Yan et al., 2025). However, most existing multimodal test-time scaling approaches follow a "sample-and-select" paradigm. They generate multiple independent reasoning chains and attempt to choose the best one, discarding the partial insights contained in rejected trajectories. Moreover, simply increasing reasoning length or sampling count in VLMs carries a fundamental risk. Without explicit grounding, early perceptual errors can be amplified rather than corrected, leading models to reason more without reasoning better. In other words, these methods encourage models to think *longer*, but not necessarily to think *together* or to verify their conclusions against visual evidence.

To address these limitations, we propose CASHEW (**C**andidate **A**ggregation and **S**ynt**H**esis of **E**vidence and re**W**ards), a framework for visually grounded iterative aggregation. Instead of selecting a single trajectory, CASHEW treats reasoning as an evolutionary process: at each iteration, it synthesizes a population of candidate trajectories into a higher-quality aggregate. To prevent hallucinations, object- and attribute-level claims are verified against the visual input, and only grounded evidence guides subsequent aggregation steps. This ensures the resulting consensus is anchored in visual evidence. Building on this, we introduce CASHEW-RL, a learned variant that internalizes aggregation during post-training. CASHEW-RL allows the VLM to integrate multiple reasoning trajectories with visual evidence through learned parameters, thereby improving aggregation efficiency under test-time sampling. The model is trained using Group Sequence Policy Optimization (GSPO) with a composite reward that encourages correct answers grounded in sufficient visual evidence while adaptively allocating reasoning effort based on task difficulty. In summary, our contributions are:

- CASHEW: an inference-time framework that stabilizes multimodal reasoning by iteratively aggregating candidate trajectories with explicit visual verification to filter hallucinations and ground reasoning in evidence.
- CASHEW-RL: a learned variant that internalizes trajectory aggregation during post-training via GSPO and a composite reward, enabling visually grounded aggregation of multiple reasoning trajectories with adaptive control of reasoning effort.
- Extensive experiments on 13 benchmarks and multiple backbone families demonstrate consistent and significant performance improvements, including gains of up to +26.2 percentage points on ScienceQA and +9.1 percentage points on EgoSchema.

2 Related Work

Multimodal Reasoning. Vision-language models (VLMs) (Alayrac et al., 2022; Li et al., 2023a; Liu et al., 2024a; Bai et al., 2023) provide unified architectures for reasoning over images and videos, supporting tasks such as captioning, question answering, and visually grounded dialogue. Recent work further improves multimodal reasoning via explicit chain-of-thought prompting and process-level supervision (Chen et al., 2024; Zhang et al., 2023), showing that modeling intermediate reasoning steps outperforms direct answer prediction (Zhou et al., 2025a). However, most existing VLMs generate reasoning trajectories independently, without mechanisms for iterative refinement or aggregation across multiple paths. In this work, we propose CASHEW and CASHEW-RL, two methods that address this limitation by iteratively aggregating multiple reasoning trajectories to produce more reliable and grounded answers.

Test-Time Scaling. Test-time scaling improves reasoning without additional training (Wang et al., 2023, 2025a) by allocating more inference-time computation to produce consistent outputs (Shinn et al., 2023; Madaan et al., 2023; Yao et al., 2023a; Wu and Xie, 2024; Luo et al., 2025; Yuan et al., 2025). LongPerceptualThoughts (Liao et al., 2025) extends reasoning budgets to generate longer, self-corrective chains, while VideoChat-R1.5 (Yan et al., 2025) iteratively refines spatial-temporal attention and textual predictions for video understanding. These approaches increase reasoning depth but still operate on a single trajectory. In contrast, CASHEW iteratively aggregates multiple reasoning trajectories into a unified process.

Reinforcement Learning for VLMs. Reinforcement learning aligns multimodal models with human or process-level rewards. Methods such as RLHF (Ouyang et al., 2022), RLAIF (Lee et al., 2024), DPO (Rafailov et al., 2023), GRPO (Kulkarni and Fazli, 2025; Shao et al., 2024), and GSPO (Zheng et al., 2025a) improve factuality and reasoning in language and vision-language models. Recent work (Ong et al., 2025) introduces

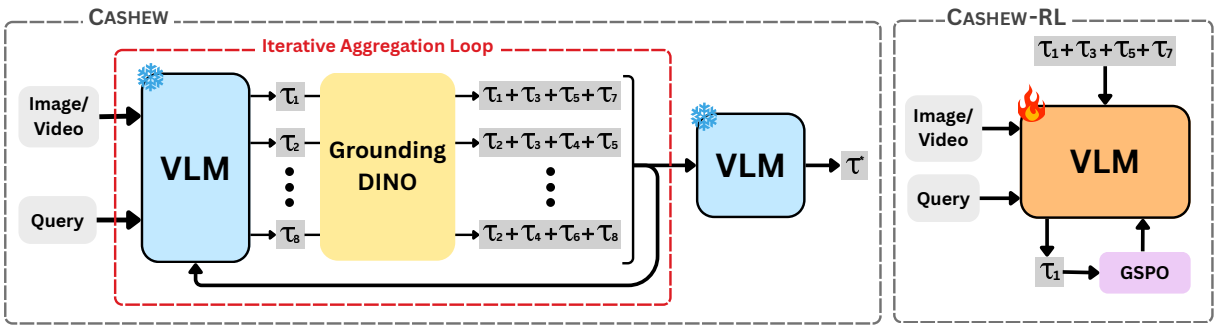

Figure 2: **Overview of the CASHEW and CASHEW-RL frameworks. Left:** CASHEW performs test-time iterative aggregation by generating a population of candidate trajectories from a frozen VLM and synthesizing subsets into refined trajectories over multiple iterations to produce a consolidated trajectory τ^*. A lightweight visual verification module (instantiated with Grounding DINO in our implementation) provides consistency signals to suppress unsupported object-level claims during aggregation. **Right:** CASHEW-RL extends this framework via post-training with GSPO, teaching the VLM to internally aggregate multiple candidate trajectories into high-quality, visually grounded reasoning traces.

step-wise rewards to guide intermediate reasoning, while GFlowVLM (Kang et al., 2025) models distributions over reasoning trajectories using generative flow networks. In contrast, our RL formulation trains an aggregation policy that fuses and refines multiple reasoning trajectories using correctness and consistency rewards, enabling the model to internalize iterative aggregation.

3 Problem Formulation and Preliminaries

Given a multimodal input $\mathbf{x} = I, q$, where I is an image or video and q is a textual query, a vision-language model aims to produce both a final answer y and a reasoning trajectory τ. A reasoning trajectory is defined as a sequence of intermediate textual reasoning steps that interpret visual observations and progressively support the predicted answer. Under standard inference, the model generates a single trajectory per input, which can be sensitive to sampling noise and brittle in complex multimodal reasoning scenarios.

Instead, we consider a multi-trajectory inference setting, in which the model produces a set of candidate reasoning trajectories, $\mathcal{T} = \{\tau_1, \ldots, \tau_N\}$, for the same input. Rather than selecting one trajectory in isolation, we formulate multimodal reasoning as an *iterative aggregation problem*, where an aggregation operator progressively maps $\mathcal{T}$ to a refined trajectory τ^*. This process integrates complementary information across trajectories, resolves inconsistencies, and emphasizes reasoning steps that are consistent with the visual evidence. The goal is to obtain a reasoning trace that is more stable and visually grounded than any individual trajectory.

Building on this formulation, we present **CASHEW**, an iterative aggregation framework for improving multimodal reasoning in VLMs. CASHEW functions as a plug-and-play test-time scaling method that aggregates multiple sampled reasoning trajectories at inference time. We further introduce CASHEW-RL, a learned variant of CASHEW that internalizes aggregation behavior during post-training using Group Sequence Policy Optimization (GSPO). While CASHEW performs aggregation solely at inference time, CASHEW-RL learns a robust aggregation policy that enables the model to combine reasoning trajectories and verified visual evidence through learned parameters, improving aggregation efficiency under test-time sampling. Further details on data generation are provided in Appendix B.

4 CASHEW

CASHEW is a test-time framework that models reasoning as an evolving population of candidate trajectories that are iteratively synthesized and refined. We describe its stages below, with pseudocode provided in Appendix A.

Population Initialization. Let $\mathcal{T}_t$ denote the population of candidate reasoning trajectories at iteration t. The base VLM p_θ first produces an initial population of N candidate trajectories:

$$\mathcal{T}_1 = \{\tau_i^{(1)} \sim p_\theta(\cdot \mid \mathbf{x})\}_{i=1}^{N}. \quad (1)$$

Each trajectory $\tau_i^{(1)}$ consists of a reasoning text $r_i^{(1)}$ and a predicted answer $a_i^{(1)}$.

Subset Sampling. At iteration t, to generate the i-th trajectory $\tau_i^{(t)}$, we sample a random subset of peer trajectories $S_i^{(t)} \subset \mathcal{T}_{t-1}$ of size M as reference candidates. This sampling strategy introduces diversity and mitigates collapse to a single reasoning mode.

Visual Verification Module. To mitigate hallucination, we incorporate a lightweight visual-text grounding verification step before aggregation. For each candidate trajectory $\tau_j^{(t)} \in S_i^{(t)}$, we extract mentioned objects using a text parser $\mathcal{E}$:

$$\mathcal{O}_j^{(t)} = \mathcal{E}(r_j^{(t)}). \tag{2}$$

Each object $o_k \in \mathcal{O}_j^{(t)}$ is then evaluated by a visual verification function $\mathcal{G}$ to assess its presence in the visual input I:

$$v(o_k) = \mathbf{1}\big[\mathcal{G}(I, o_k) > \delta_g\big], \tag{3}$$

where $\mathcal{G}(I, o_k)$ produces a visual grounding confidence score and δ_g is a confidence threshold. In this work, we instantiate $\mathcal{G}$ using Grounding DINO (Liu et al., 2024c), though the framework is agnostic to the specific grounding model and supports alternative verification signals (e.g., attribute checks or relational consistency). We collect all verified objects as grounded visual evidence:

$$\mathcal{V}_j^{(t)} = \{o_k \in \mathcal{O}_j^{(t)} \mid v(o_k) = 1\}. \tag{4}$$

The verified evidence serves as auxiliary consistency guidance during trajectory aggregation rather than as a direct reasoning component.

Grounded Aggregation. Finally, the model synthesizes the sampled trajectories and their corresponding visual verification results to generate an updated reasoning trajectory. We explicitly condition the generation on both the reasoning texts and the corresponding verified visual evidence:

$$\tau_i^{(t+1)} \sim p_\theta\big(\cdot \mid \mathbf{x}, S_i^{(t)}, \{\mathcal{V}_j^{(t)}\}_{\tau_j^{(t)} \in S_i^{(t)}}\big). \tag{5}$$

By providing $\{\mathcal{V}_j^{(t)}\}$, the model can distinguish which parts of the candidate reasoning are visually supported, encouraging it to anchor the new trajectory $\tau_i^{(t+1)}$ in verified evidence. The population is then updated as $\mathcal{T}_{t+1} = \{\tau_i^{(t+1)}\}_{i=1}^N$.

Final Aggregation. After T iterations, all trajectories are merged to produce a single final reasoning trajectory:

$$\tau^* \sim p_\theta(\cdot \mid \mathbf{x}, \mathcal{T}_T) \tag{6}$$

5 CASHEW-RL

To equip the model with robust multimodal aggregation capabilities, we introduce CASHEW-RL, a two-stage post-training framework: (1) *supervised fine-tuning (SFT) for trajectory aggregation*, which teaches the model to combine multiple trajectories into coherent outputs, and (2) *reinforcement learning (RL) via GSPO*, which further optimizes the aggregation policy using reward signals reflecting aggregation quality.

5.1 Data Construction and Aggregation Format

Our post-training framework relies on a unified multimodal aggregation format and two datasets derived from a pool of seven image/video benchmarks (details in Appendix B). These datasets serve distinct purposes for SFT and RL stages, while sharing a consistent output structure.

Structured Aggregation Format. At both the SFT and RL stages, the model is trained to output a standardized, interpretable aggregation composed of three elements:

<think> r </think>
<visual_keys> K </visual_keys>
<answer> a </answer>

Each tag serves a specific role:

- **<think>** contains an intermediate reasoning trace r, revealing how evidence from the candidate trajectories is aggregated.
- **<visual_keys>** contains a set K of object-level entities that are relevant to answering the question. Concretely, these correspond to $\mathcal{V}_j^{(t)}$ from the grounded aggregation step.
- **<answer>** contains the final response a, which is evaluated against the ground truth.

This unified representation provides explicit supervision for reasoning and verified visual evidence, and supports reward computation in the RL stage.

SFT Data. We first construct a 30k-instance dataset for SFT. Each instance includes an image or video frames $\mathbf{x}$, a question q, a ground-truth answer y^*, a reasoning chain generated by Qwen3-VL-30B-Thinking (Bai et al., 2025a), and a set of visual keys that are extracted from the reasoning chain. These keys are validated against the visual

input using Grounding DINO and manually verified for quality control. The model is trained to imitate the structured aggregation format, learning to produce coherent reasoning, identify relevant visual entities, and generate a well-formed final answer. This stage establishes the reference policy π_{ref} and provides reliable formatting behavior for subsequent RL training.

RL Data. We curate a 200k-instance corpus from the same source datasets. Each instance contains the original annotation $(\mathbf{x}, q, y^*)$, three diverse candidate trajectories $\{\tau_i\}_{i=1}^{3}$ generated using Qwen3-VL-30B-Thinking, and a shared set of visual keys extracted from the candidates. These keys are validated against the visual input using Grounding DINO and manually verified to ensure label reliability. Each τ_i provides a candidate answer and a reasoning chain, which may be correct, incorrect, or incomplete. During RL training, the candidate pool is constructed as a curriculum-based mixture of offline teacher-generated trajectories and on-policy trajectories sampled from the current model π_θ. This hybrid setup exposes the policy to both clean teacher signals and noisier self-generated trajectories, reducing distribution mismatch and improving robustness.

Data Split and Isolation. All SFT and RL training data are constructed exclusively from the official training splits of the respective source datasets. For datasets that also appear in our evaluation benchmarks, we explicitly remove any overlapping evaluation instances to prevent train-test leakage. The exact split usage and overlap filtering protocol are described in Appendix B.

5.2 Stage I: SFT

The goal of the supervised stage is to teach the model the *structure* of multimodal aggregation: how to articulate intermediate reasoning, verbalize evidence via visual keys, and format the final answer. Given $(\mathbf{x}, q, \{\tau_i\})$, where $\{\tau_i\}$ are reference trajectories in the structured aggregation format, the model is trained to imitate a structured aggregation output (r, K, a). This stage does not primarily optimize correctness; rather, it provides: (1) a stable and structured textual interface for subsequent RL training, with a well-defined and parseable output format, and (2) a reference policy π_{ref} used for KL regularization in GSPO. Without this SFT initialization, RL training often collapses into malformed or degenerate trajectories.

5.3 Stage II: RL via GSPO

While SFT establishes the structural format of multimodal aggregation, it relies primarily on imitation and does not explicitly teach the model to distinguish correct evidence from hallucinated or noisy candidate trajectories. To explicitly optimize for effective aggregation, we employ reinforcement learning to train the model as an aggregator.

Aggregation Policy. During RL, the model is treated as a policy

$$\pi_\theta(a, K, r \mid \mathbf{x}, S),$$

where π_θ is initialized from the reference policy π_{ref} learned during SFT and $S = \{\tau_i\}_{i=1}^{M}$ denotes a set of candidate trajectories drawn from a mixture of offline cached trajectories and on-policy rollouts, following a curriculum-based teacher-on-policy trajectory mixing strategy. Each rollout consists of a reasoning trace r, a set of predicted visual keys K, and the final aggregated answer a.

Curriculum-Based Teacher-On-Policy Trajectory Mixing. The candidate pool S evolves during RL training and is constructed as a mixture of teacher-generated trajectories and on-policy trajectories sampled from the current model π_θ. We employ a staged curriculum that gradually increases the proportion of on-policy candidates: early stages rely primarily on teacher trajectories to provide clear and reliable evidence, while later stages increasingly incorporate noisier and more diverse self-generated trajectories that the policy must ultimately aggregate at inference time. This dynamic mixture mitigates distribution shift and improves the robustness of the learned aggregation policy. The specific curriculum schedule and mixing ratios are detailed in Section 6.

Reward Design. Each rollout is evaluated by a composite reward that accounts for answer correctness, evidence selection quality, and adaptive reasoning efficiency:

$$R(a, K, r \mid \mathbf{x}) = w_{\text{acc}}\, R_{\text{acc}} + w_{\text{key}}\, R_{\text{key}} + R_{\text{len}}. \tag{7}$$

(1) *Answer correctness (R_{acc}).* We measure answer accuracy using an exact-match criterion against the ground-truth answer y^*:

$$R_{\text{acc}} = \mathbf{1}[a = y^*]. \tag{8}$$

This binary signal provides stable, task-agnostic supervision during RL optimization.

(2) *Evidence selection quality (R_{key})* To explicitly encourage faithful visual grounding, we reward the selection of relevant visual evidence via a balanced precision-recall formulation. Let G denote the ground-truth visual-key set derived from our RL training data annotations, and let K be the model-predicted key set. We define:

$$R_{\text{key}} = (1-\alpha)\frac{|K \cap G|}{|G| + \epsilon} + \alpha\frac{|K \cap G|}{|K| + \epsilon}. \quad (9)$$

Unlike recall-only objectives that incentivize indiscriminate key generation, this weighted formulation enables explicit control over the precision-recall trade-off. In practice, it discourages uncontrolled key proliferation while preserving sensitivity to missing critical evidence.

(3) *Difficulty-Aware Length Penalty (R_{len}).* To regulate the amount of intermediate reasoning without collapsing to trivial traces or encouraging excessive verbosity, we introduce a difficulty-aware length penalty inspired by adaptive computation allocation in reasoning models (Xiang et al., 2025). For each prompt, we sample J rollouts and estimate an empirical solve rate:

$$\hat{p}_{\text{solve}}(q) = \frac{1}{J}\sum_{j=1}^{J} \mathbf{1}[a_j = y^*], \quad (10)$$

which is smoothed across training steps using an exponential moving average (EMA):

$$\tilde{p}_t \leftarrow \gamma\tilde{p}_{t-1} + (1-\gamma)\hat{p}_{\text{solve}}. \quad (11)$$

Let N_{tok} denote the number of tokens within the <think> region of the rollout. The length penalty is defined as:

$$R_{\text{len}} = -\beta\, N_{\text{tok}} \cdot \max(\tilde{p}_t,\, 1/J). \quad (12)$$

Intuitively, easy prompts that are consistently solved (high $\tilde{p}_t$) incur stronger penalties for extended reasoning, encouraging concise aggregation, while difficult prompts retain flexibility for longer computation. The lower bound $1/J$ prevents the penalty from vanishing entirely when no rollout succeeds, ensuring stable optimization. The scalar β controls the relative strength of this term.

Overall, this composite reward encourages the policy to produce correct answers grounded in minimal yet sufficient visual evidence, while allocating reasoning effort adaptively based on task difficulty.

Group Sequence Policy Optimization (GSPO). GSPO trains the aggregation policy by comparing multiple rollouts generated for a fixed prompt and candidate set. For each $(\mathbf{x}, S)$, we sample a group of J aggregation rollouts $\{(r_j, K_j, a_j)\}_{j=1}^{J} \sim \pi_\theta$. Each rollout is evaluated using the composite reward in Eq. 7, and the resulting scores are converted into normalized within-group weights:

$$\tilde{w}_j = \frac{\exp(\lambda R_j)}{\sum_{k=1}^{J}\exp(\lambda R_k)}, \quad (13)$$

where λ controls the sharpness of the relative preference distribution. This intra-group competition provides a relative quality signal well suited for aggregation, since aggregated answers are best evaluated by comparison with alternative attempts over the same evidence pool. The policy is then updated to increase the likelihood of higher-reward rollouts while remaining close to the reference policy learned during SFT:

$$\mathcal{L}_{\text{GSPO}} = -\sum_{j=1}^{J}\tilde{w}_j \log \pi_\theta(r_j, K_j, a_j \mid \mathbf{x}, S) + \alpha_{\text{KL}}\text{KL}[\pi_\theta(\cdot \mid \mathbf{x}, S) \parallel \pi_{\text{ref}}(\cdot \mid \mathbf{x}, S)]\,. \quad (14)$$

The KL regularization stabilizes optimization and preserves the aggregation format learned during SFT. Combined with the reward design in Eq. 7, GSPO encourages the policy to prefer aggregation strategies that are both correct and well grounded, while adaptively regulating reasoning length.

6 Experiments

We apply CASHEW at test time on multiple backbone models, including InternVL-3.5 (Wang et al., 2025b), and Qwen3-VL (Bai et al., 2025a). For each input, we sample a population of N=8 reasoning trajectories and form aggregation groups of size M=4 (i.e., M responses are grouped to synthesize one candidate); the iterative aggregation runs for T=3 iterations. Decoding is performed with temperature $= 0.8$ and $top_p = 0.95$. Visual grounding verification is provided by a frozen Grounding DINO model. All inference experiments use 8 H100 GPUs.

For CASHEW-RL, we fine-tune Qwen3-VL-4B, Qwen3-VL-8B, and InternVL3.5-8B using LoRA. RL fine-tuning is performed on 16 H100 GPUs with a global batch size of 64 and a learning rate of 1×10^{-6}, using the ms-swift (Zhao et al., 2025)

Table 1: **Results of CASHEW and CASHEW-RL on image benchmarks.** T denotes the number of iterations. I: SEED-Bench results are reported only for the image subset. Improvements from CASHEW and CASHEW-RL are highlighted in green and reported as mean values. 95% confidence intervals (CIs, ±) were computed via bootstrap resampling. Improvements are statistically significant at $\alpha = 0.05$ if the corresponding CI excludes zero. § denotes non-significant improvements ($p > 0.05$; 95% CI includes zero).

Model	ScienceQA	MME	POPE	SEED-Bench^I
LLaVA-1.5-7B (Liu et al., 2024a)	66.8	302.1/1506.2	85.9	66.1
Qwen-VL-Chat-7B (Bai et al., 2023)	68.2	392.1/1467.8	74.9	58.2
VILA1.5-13B (Lin et al., 2024)	79.1	288.9/1429.3	84.2	62.8
LLaVA-Next-7B (Liu et al., 2024b)	73.0	308.9/1512.3	87.3	72.4
LLaVA-OneVision-7B (Li et al., 2024a)	95.4	415.7/1577.8	87.4	75.4
InternVL3.5-8B (Wang et al., 2025b)	95.9	663.2/1686.6	88.1	77.7
+ CASHEW	**97.8** (+1.9)	**685.7/1700.2** (+22.5 / +13.6)	**89.7** (+1.6)	**79.0** (+1.3)
+ CASHEW-RL	**97.9** (+2.0)	**691.8/1707.0** (+28.6 / +20.4)	**90.2** (+2.1)	**80.5** (+2.8)
Qwen3-VL-4B (Bai et al., 2025a)	69.5	638.6/1693.7	88.0	78.7
+ CASHEW	**93.1** (+23.6)	**710.4/1756.0** (+71.8/+62.3)	**89.1** (+1.1)	**79.8** (+1.1)
+ CASHEW-RL	**95.7** (+26.2)	**719.7/1764.1** (+81.1/+70.4)	**89.6** (+1.6)	**81.2** (+2.5)
Qwen3-VL-8B (Bai et al., 2025a)	92.9	643.2/1720.3	88.9	78.7
+ CASHEW *(w/o Grounding DINO)*	*96.3* (+3.4)	*711.0/1748.2* (+67.8/+27.9)	*89.3* (+0.4)§	*79.7* (+1.0)
+ CASHEW	**97.7** (+4.8)	**738.2/1772.0** (+95.0/+51.7)	**89.9** (+1.0)	**80.3** (+1.6)
+ CASHEW-RL	**97.8** (+4.9)	**740.1/1769.8** (+96.9/+49.5)	**90.2** (+1.3)	**80.8** (+2.1)

Table 2: **Results of CASHEW and CASHEW-RL on video understanding and reasoning benchmarks.** T denotes the number of iterations. † Results on Video-MME are reported without subtitles. ‡ Results on Video-TT are reported only for multiple-choice questions. Improvements from CASHEW and CASHEW-RL are highlighted in green and reported as mean values. 95% confidence intervals (CIs, ±) were computed via bootstrap resampling. Improvements are statistically significant at $\alpha = 0.05$ if the corresponding CI excludes zero. § denotes non-significant improvements ($p > 0.05$; 95% CI includes zero).

	Video Understanding					Video Reasoning			
Model	Video-MME†	LongVideoBench	EgoSchema	MVBench	NExT-QA	VideoMMMU	VSI-Bench	Video-TT‡	TOMATO
LLaVA-NeXT-Video-7B (Zhang et al., 2024a)	–	43.5	43.9	46.5	–	36.1	35.6	41.8	24.9
VILA1.5-40B (Lin et al., 2024)	60.1	–	58.0	–	67.9	34.0	31.2	–	24.7
LLaVA-OneVision-7B (Li et al., 2024a)	58.2	56.4	60.1	56.7	79.4	33.9	32.4	–	25.5
VideoLLaMA3-7B (Zhang et al., 2025a)	66.2	59.8	63.3	69.7	84.5	47.0	–	–	–
InternVL3.5-8B (Wang et al., 2025b)	63.2	61.3	62.0	71.4	78.6	50.0	53.2	43.4	23.8
+ CASHEW	**63.9** (+0.7)§	**62.9** (+1.6)	**69.9** (+7.9)	**73.0** (+1.6)	**80.0** (+1.4)	**50.8** (+0.8)	**54.5** (+1.3)	**44.2** (+0.8)§	**24.6** (+0.8)
+ CASHEW-RL	**64.8** (+1.6)	**63.2** (+1.9)	**71.1** (+9.1)	**73.7** (+2.3)	**80.4** (+1.8)	**51.3** (+1.3)	**54.6** (+1.4)	**44.6** (+1.2)	**24.9** (+1.1)
Qwen3-VL-4B (Bai et al., 2025a)	64.2	61.0	67.6	65.7	73.8	46.0	56.6	40.4	27.6
+ CASHEW	**65.5** (+1.3)	**63.6** (+2.6)	**73.0** (+5.4)	**68.1** (+2.4)	**78.8** (+5.0)	**47.2** (+1.2)	**60.3** (+3.7)	**41.2** (+0.8)	**28.1** (+0.5)§
+ CASHEW-RL	**67.1** (+2.9)	**64.5** (+3.5)	**74.1** (+6.5)	**68.6** (+2.9)	**79.1** (+5.3)	**48.0** (+2.0)	**61.2** (+4.6)	**42.9** (+2.5)	**30.0** (+2.4)
Qwen3-VL-8B (Bai et al., 2025a)	66.9	63.3	71.2	66.2	75.6	47.3	58.8	43.3	31.5
+ CASHEW *(w/o Grounding DINO)*	*67.5* (+0.6)§	*64.3* (+1.0)	*72.5* (+1.3)	*68.5* (+2.3)	*80.0* (+4.4)	*48.0* (+0.7)	*59.5* (+0.7)	*44.0* (+0.7)§	*32.9* (+1.4)
+ CASHEW	**68.3** (+1.4)	**64.8** (+1.5)	**74.7** (+3.5)	**69.3** (+3.1)	**80.5** (+4.9)	**48.4** (+1.1)	**61.2** (+2.4)	**44.2** (+0.9)§	**33.4** (+1.9)
+ CASHEW-RL	**68.9** (+2.0)	**65.4** (+2.1)	**75.5** (+4.3)	**69.8** (+3.6)	**80.5** (+4.9)	**49.0** (+1.7)	**61.4** (+2.6)	**44.8** (+1.5)	**34.0** (+2.5)

framework. For each RL instance, we construct an evidence pool of $M = 4$ candidate trajectories. To balance stability and robustness, we adopt a staged curriculum that gradually increases the proportion of on-policy candidates generated by the current policy π_θ: (3:1) in early training, (2:2) in the middle stage, and (1:3) in the final stage. This curriculum exposes the model to increasingly realistic evidence distributions and substantially improves robustness when aggregating trajectories during inference.

6.1 Results

CASHEW Evaluation. Across image and video benchmarks, CASHEW consistently improves performance for diverse VLM backbones, demonstrating robustness across model scales and modalities, with most gains statistically significant under bootstrap-based confidence intervals. On image understanding tasks (Table 1), CASHEW yields substantial gains for all evaluated models. For example, Qwen3-VL-4B improves from 69.5 percentage points to 93.1 percentage points on ScienceQA, while Qwen3-VL-8B improves from 643.2/1720.3 to 738.2/1772.0 on MME. These results indicate that multi-trajectory aggregation effectively enhances both factual accuracy and perceptual grounding, regardless of backbone capacity. On video benchmarks (Table 2), CASHEW consistently improves understanding and reasoning across all tested backbones. Notably, it achieves +7.9 percentage points on EgoSchema for InternVL3.5-8B , highlighting its effectiveness on long-horizon tasks. Consistent gains are also observed on VSI-Bench,

including +3.7 percentage points for Qwen3-VL-4B. Even on more challenging benchmarks such as Video-TT and TOMATO, CASHEW shows positive and often statistically significant trends. Overall, these results indicate that CASHEW strengthens temporal coherence and refines noisy reasoning trajectories, leading to more reliable and grounded video understanding and reasoning.

To understand the role of visual grounding, we ablate the Grounding DINO verification step while keeping all other aggregation mechanisms unchanged (N=8, M=4, T=3). Even without grounding, CASHEW substantially improves over the base VLM, e.g., ScienceQA increases from 92.9 to 96.3 and NeXT-QA from 75.6 to 80.0, indicating that iterative trajectory aggregation is the primary driver of performance gains, with visual grounding further enhancing reliability and evidence alignment. The additional analysis is provided in Appendix D.

CASHEW-RL Evaluation. We evaluate CASHEW-RL under $T = 3$ to assess the impact of GSPO on aggregation behavior. Across image and video benchmarks, CASHEW-RL consistently achieves further improvements over both the baseline and the corresponding CASHEW results, with most gains statistically significant. On image understanding tasks (Table 1), CASHEW-RL yields substantial gains across backbone scales and model families. For example, on Qwen3-VL-8B, ScienceQA improves from 92.9 to 97.8 and POPE increases to 90.2. Importantly, similar trends are observed for the smaller Qwen3-VL-4B backbone and the non-Qwen InternVL3.5-8B backbone, indicating that the learned aggregation policy generalizes across model capacities and architectures. On video benchmarks (Table 2), CASHEW-RL further enhances both video understanding and reasoning. For Qwen3-VL-8B, EgoSchema increases from 71.2 to 75.5 and TOMATO from 31.5 to 34.0. Comparable improvements are observed on Qwen3-VL-4B and InternVL3.5-8B, showing consistent gains on all benchmarks over both the baseline and CASHEW. Moreover, results in Appendix D show that CASHEW-RL reaches performance comparable to higher-depth CASHEW configurations with fewer aggregation iterations, suggesting improved aggregation efficiency. Overall, these results indicate that GSPO enables CASHEW-RL to internalize multi-step aggregation, leading to more stable and reliable multimodal reasoning across different backbone capacities.

Table 3: Comparison with state-of-the-art test-time scaling methods. Best scores are shown in **bold**, and second-best scores are underlined.

Model	ScienceQA	MME	EgoSchema	MVBench	VideoMMMU
Qwen3-VL-8B	92.9	643.2/1720.3	71.2	66.2	47.3
+ Self-Consistency	94.2	669.3/1702.1	71.1	66.8	46.4
+ Self-Selector	87.3	508.2/1431.2	70.1	67.8	46.7
+ Self-Synthesizer	95.4	690.0/1689.4	72.0	67.6	47.1
+ CASHEW	**97.7**	**738.2/1772.0**	**74.7**	**69.3**	**48.4**

Comparison with State-of-the-Art Test-Time Scaling Methods. We compare CASHEW with three widely used test-time scaling baselines under an N=8 multi-sample setting. (1) *Self-Consistency* (Wang et al., 2023) selects a final answer via majority voting over multiple sampled responses, effective when outputs are discrete and comparable. (2) *Self-Selector* (Parmar et al., 2025) replaces majority voting with model-based judgment, using the VLM to evaluate and select a single trajectory. (3) *Self-Synthesizer* (Li et al., 2025b,c) goes beyond selection by generating a new response that integrates information from multiple candidate trajectories. As shown in Table 3, CASHEW consistently achieves the best performance across benchmarks. On the MME benchmark, it improves over the strongest baseline (*Self-Synthesizer*) by +48.2/+82.6 points on perception and cognition scores, respectively. On EgoSchema, CASHEW outperforms *Self-Synthesizer* by +2.7 percentage points, demonstrating the effectiveness of iterative aggregation for complex video reasoning. Notably, even with a single aggregation pass ($T = 1$), CASHEW attains superior accuracy at a computational cost comparable to existing multi-sample test-time scaling methods. A detailed latency-performance analysis across different aggregation settings is provided in Appendix D.

7 Conclusion

We present CASHEW, an inference-time framework that stabilizes multimodal reasoning through iterative aggregation of candidate trajectories with visual verification, and CASHEW-RL, a learned variant that internalizes this aggregation behavior. Using a composite reward within GSPO, CASHEW-RL produces evidence-grounded answers while adaptively allocating reasoning effort based on task difficulty. Experiments on image and video benchmarks show that both methods improve accuracy and reasoning consistency, demonstrating the effectiveness of visually grounded iterative aggregation.

Limitations

In this work, we use a fixed aggregation budget to enable controlled comparisons across models and benchmarks, rather than optimizing the compute budget for each individual example. This setting is appropriate for studying the effect of iterative aggregation, but it may be suboptimal in resource-constrained applications where many easy examples may not require the same amount of aggregation. CASHEW-RL partially addresses this issue by internalizing aggregation behavior and improving aggregation efficiency, but a systematic study of adaptive budget allocation remains future work.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. In *Proceedings of the Thirty-Sixth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. 2024. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. 2017. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. 2025. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haoqiang Kang, Enna Sachdeva, Piyush Gupta, Sangjae Bae, and Kwonjoon Lee. 2025. GFlowVLM: Enhancing multi-step reasoning in vision-language models

with generative flow networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yogesh Kulkarni and Pooyan Fazli. 2025. Avatar: Reinforcement learning to see, hear, and reason over video. *arXiv preprint arXiv:v*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *Proceedings of the Forty-First International Conference on Machine Learning (ICML)*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024b. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chaoyu Li, Eun Woo Im, and Pooyan Fazli. 2025a. Vidhalluc: Evaluating temporal hallucinations in multimodal large language models for video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13723–13733.

Cheryl Li, Tianyuan Xu, and Steven Y. Guo. 2025b. Reasoning-as-logic-units: Scaling test-time reasoning in large language models through logic unit alignment. In *Proceedings of the Forty-second International Conference on Machine Learning (ICML)*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the Fortieth International Conference on Machine Learning (ICML)*.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024c. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zichong Li, Xinyu Feng, Yuheng Cai, Zixuan Zhang, Tianyi Liu, Chen Liang, Weizhu Chen, Haoyu Wang, and Tuo Zhao. 2025c. Llms can generate a better answer by aggregating their own responses. *arXiv preprint arXiv:2503.04104*.

Yuan-Hong Liao, Sven Elflein, Liu He, Laura Leal-Taixé, Yejin Choi, Sanja Fidler, and David Acuna. 2025. Longperceptualthoughts: Distilling system-2 reasoning for system-1 perception. *arXiv preprint arXiv:2504.15362*.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024. Vila: On pretraining for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llava-next: Improved reasoning, ocr, and world knowledge.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. 2024c. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

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

Wenjie Luo, Ruocheng Li, Shanshan Zhu, and Julian Perry. 2025. Coherent multimodal reasoning with iterative self-evaluation for vision-language models. *arXiv preprint arXiv:2508.02886*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder,

Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Proceedings of the Thirty-Seventh Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Brandon Ong, Tej Deep Pala, Vernon Toh, William Chandra Tjhi, and Soujanya Poria. 2025. Training vision-language process reward models for test-time scaling in multimodal reasoning: Key insights and lessons learned. *arXiv preprint arXiv:2509.23250*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the Thirty-Sixth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Mihir Parmar, Xin Liu, Palash Goyal, Yanfei Chen, Long Le, Swaroop Mishra, Hossein Mobahi, Jindong Gu, Zifeng Wang, Hootan Nakhost, Chitta Baral, Chen-Yu Lee, Tomas Pfister, and Hamid Palangi. 2025. PlanGEN: A multi-agent framework for generating planning and reasoning trajectories for complex problem solving. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of the Thirty-Seventh Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. 2025. Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models. In *Proceedings of the Thirteenth International Conference on Learning Representations*.

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

Mingjia Shi, Yinhan He, Yaochen Zhu, and Jundong Li. 2026. Saliency-aware multi-route thinking: Revisiting vision-language reasoning. *arXiv preprint arXiv:2602.16702*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Proceedings of the Thirty-Seventh Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Weiqin Wang, Yile Wang, and Hui Huang. 2025a. Ranked voting based self-consistency of large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025b. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *Proceedings of the Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Penghao Wu and Saining Xie. 2024. V*: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. 2025. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa:next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ziang Yan, Xinhao Li, Yinan He, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. 2025. Videochat-r1.5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. In *Proceedings of the Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Proceedings of the Thirty-Seventh Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*.

Yangshu Yuan, Heng Chen, Xinyi Jiang, Christian Ng, and Kexin Qiu. 2025. Cimr: Contextualized iterative multimodal reasoning for robust instruction following in lvlms. *arXiv preprint arXiv:2507.22074*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. 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 (CVPR)*.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. 2025a. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

Yuanhan Zhang, Yunice Chew, Yuhao Dong, Aria Leo, Bo Hu, and Ziwei Liu. 2025b. Towards video thinking test: A holistic benchmark for advanced video reasoning and understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024a. Llava-next: A strong zero-shot video understanding model.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024b. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. In *Proceedings of the Transactions on Machine Learning Research (TMLR)*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2025. Swift:a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025a. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025b. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Jingqi Zhou, Sheng Wang, Jingwei Dong, Kai Liu, Lei Li, Jiahui Gao, Jiyue Jiang, Lingpeng Kong, and Chuan Wu. 2025a. Proreason: Multi-modal proactive reasoning with decoupled eyesight and wisdom. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sheng Zhou, Junbin Xiao, Qingyun Li, Yicong Li, Xun Yang, Dan Guo, Meng Wang, Tat-Seng Chua, and Angela Yao. 2025b. Egotextvqa: Towards egocentric scene-text aware video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Appendix

A CASHEW Pseudocode

Algorithm 1 presents the pseudocode for CASHEW, detailing its iterative aggregation and grounded object verification.

Algorithm 1 Iterative Aggregation with Grounded Verification (CASHEW)

Require: Visual input I, question q, base model p_θ, iterations T, population N
function CHECKCONSENSUS(P)
  Let $a_1, \ldots, a_{|P|}$ be the predicted answers from trajectories in P
  return $\big(a_1 = a_2 = \cdots = a_{|P|}\big)$
end function
$P_1 = \{\tau_i^{(1)} \sim p_\theta(\cdot \mid I, q)\}_{i=1}^{N}$
if CHECKCONSENSUS(P_1) **then**
  return the common answer in P_1
end if
for $t = 1$ to $T-1$ **do**
  for $i = 1$ to N **do**
    Sample subset $S_i^{(t)} \subset P_t$ of size M
    for $\tau_j^{(t)} \in S_i^{(t)}$ **do**
      Extract objects $\mathcal{O}_j^{(t)} = \mathcal{E}(r_j^{(t)})$
      Verify with Grounding DINO to obtain verified objects $\mathcal{V}_j^{(t)}$
    end for
    Aggregate with grounding hints:
    $\tau_i^{(t+1)} \sim p_\theta(\cdot \mid I, q, S_i^{(t)}, \{\mathcal{V}_j^{(t)}\}_{\tau_j^{(t)} \in S_i^{(t)}})$
  end for
  Form new population $P_{t+1} = \{\tau_i^{(t+1)}\}_{i=1}^{N}$
  if CHECKCONSENSUS(P_{t+1}) **then**
    return the common answer in P_{t+1}
  end if
end for
Final aggregation:
$\tau^* \sim p_\theta(\cdot \mid I, q, P_T, \{\mathcal{V}_j^{(T)}\}_{\tau_j^{(T)} \in P_T})$
return τ^*

B CASHEW-RL Data Generation

Table 4: CASHEW-RL training dataset distribution for SFT and RL stages.

Dataset	SFT (# Samples)	RL (# Samples)
Image Datasets		
GQA (Hudson and Manning, 2019)	8,300	76,000
MathVista (Lu et al., 2024)	400	5,000
ScienceQA (Lu et al., 2022)	2,000	10,000
Video Datasets		
SSV2 (Goyal et al., 2017)	8,500	40,000
NExT-QA (Xiao et al., 2021)	600	2,500
Ego4D (Grauman et al., 2022)	9,800	65,000
EgoTextVQA (Zhou et al., 2025b)	400	1,500
Total	**30,000**	**200,000**

As summarized in Table 4, we construct the CASHEW-RL training set from a diverse collection of image and video datasets to expose the model to a wide range of visual inputs and reasoning scenarios. The image datasets include general visual question answering, mathematical reasoning, and science-oriented tasks, while the video datasets cover short-term actions, long-horizon temporal reasoning, and egocentric understanding. This diverse composition encourages robust, generalizable aggregation behaviors rather than specialization to a single task or modality. Table 4 also reports the data distribution for the SFT stage (30k instances) and the RL stage (200k instances).

All SFT and RL samples are drawn exclusively from the official training splits of the respective source datasets. To ensure strict train-test isolation, we explicitly remove any instances that overlap with evaluation benchmarks. Concretely, for ScienceQA, we exclude all image IDs appearing in the evaluation split; for NExT-QA, we remove all video IDs used in evaluation; and for Ego4D, we exclude any video IDs that appear in EgoSchema. This filtering is performed prior to trajectory generation and visual key construction, ensuring that no evaluation samples or associated visual content are exposed during post-training.

C Additional Implementation Details

We provide the complete implementation details and hyperparameter configurations used in both the inference and training phases of CASHEW and CASHEW-RL in Table 5.

Table 5: Complete Hyperparameter Configuration.

Parameter Description	Symbol	Value
CASHEW/CASHEW-RL: ***Inference Phase***		
Candidate population size per iteration	N	8
Iteration steps	T	3
Candidates aggregated per group	M	4
Grounding DINO confidence threshold	δ_g	0.35
CASHEW-RL: ***Training Phase***		
Global batch size	B	64
Learning rate	η	1×10^{-6}
Number of rollouts per prompt	J	4
CASHEW-RL: ***Reward Function Design (Eq.7 - Eq.12)***		
Weight for answer accuracy	w_{acc}	1.0
Weight for visual evidence overlap	w_{key}	0.35
F1-score balancing coefficient	α	0.5
Epsilon for division stability	ϵ	1e-8
Adaptive length penalty coefficient	β	0.001
EMA decay for solve rate	γ	0.9
CASHEW-RL: ***GSPO (Eq. 13 - Eq. 14)***		
Inverse temperature for preference	λ	1.0
KL divergence coefficient	α_{KL}	0.02

Table 6: Study on verifier robustness in CASHEW under Qwen3-VL-8B.

Model	Verifier	ScienceQA	SEED-Bench	NExT-QA	EgoSchema	VSI-Bench
Qwen3-VL-8B	–	92.9	78.7	75.6	71.2	58.8
+ CASHEW	VLM self-check	96.7	79.7	80.2	73.2	60.0
+ CASHEW	GLIP	97.5	**80.5**	**80.6**	74.3	61.0
+ CASHEW	Grounding DINO	**97.7**	80.3	80.5	**74.7**	**61.2**

Table 7: Study on grounding without trajectory aggregation in CASHEW.

Model	ScienceQA	SEED-Bench	NExT-QA	EgoSchema	VSI-Bench
Qwen3-VL-8B	92.9	78.7	75.6	71.2	58.8
+ Grounding DINO	93.2	78.3	76.0	71.6	58.2
+ CASHEW ($T = 1$)	94.6	78.9	76.8	72.4	58.6
+ CASHEW ($T = 3$)	**97.7** (+4.8)	**80.3** (+1.6)	**80.5** (+4.9)	**74.7** (+3.5)	**61.2** (+2.4)

D Ablation Studies

RQ1: What is the role of visual grounding verification in CASHEW? We further analyze the role of visual grounding verification in CASHEW from two complementary perspectives: (1) replacing the Grounding DINO with other verifiers, and (2) introducing grounding without trajectory aggregation.

(1) *Effect of replacing Grounding DINO*. We further replace Grounding DINO with alternative verification signals, including VLM self-check and GLIP (Li et al., 2022). Both variants consistently improve over the no-grounding setting, showing that CASHEW benefits from visual verification signals independent of the specific verifier. Grounding DINO yields the strongest overall results, further boosting performance from 96.3 to 97.7 on ScienceQA and from 72.5 to 74.7 on EgoSchema. These results suggest that grounding acts as a stabilizing mechanism that reduces hallucinations. However, the additional gains from grounding are smaller than those from aggregation itself, confirming that aggregation remains the dominant driver of improvement.

(2) *Effect of grounding without aggregation*. To further disentangle the effect of grounding from aggregation, we evaluate a single-rollout variant with grounding verification but without trajectory aggregation (i.e., no population sampling or iterative synthesis). Table 7 shows that incorporating grounding alone provides only marginal improvements over the base VLM (e.g., +0.3 percentage points on ScienceQA and +0.4 percentage points on EgoSchema), whereas multi-trajectory aggregation results in much larger gains (+4.8 percentage points and +3.5 percentage points, respectively). This confirms that the performance gains arise primarily from the aggregation mechanism guided by verification, rather than from the detector itself.

RQ2: How robust is CASHEW to the visual grounding verification? We examine how sensitive CASHEW is to the design and quality of visual grounding verification from two complementary perspectives: (1) varying the grounding confidence threshold, and (2) perturbing detection outputs with simulated noise.

Table 8: Study on the sensitivity of CASHEW and CASHEW-RL to the grounding confidence threshold δ_g.

Model	δ_g	ScienceQA	EgoSchema
Qwen3-VL-8B	–	92.9	71.2
+ CASHEW	0.25	97.0	74.0
+ CASHEW	0.30	97.4	74.5
+ CASHEW	0.35	**97.7**	**74.7**
+ CASHEW	0.45	97.3	74.4
+ CASHEW-RL	0.25	97.5	75.2
+ CASHEW-RL	0.30	97.7	75.3
+ CASHEW-RL	0.35	**97.8**	**75.5**
+ CASHEW-RL	0.45	97.6	75.3

(1) *Sensitivity to Grounding Confidence Threshold*. If CASHEW heavily relied on precise detector calibration, small changes in the grounding confidence threshold δ_g would cause large performance fluctuations. To test this, we vary $\delta_g \in \{0.25, 0.30, 0.35, 0.45\}$ on ScienceQA and EgoSchema. In all main experiments, we use $\delta_g = 0.35$ as a default setting. As shown in Table 8, performance remains stable across practical thresholds. For CASHEW, variations are within ± 0.7 percentage points on ScienceQA and ± 0.5 percentage points on EgoSchema. CASHEW-RL shows even smaller fluctuations (within ± 0.3 per-

Table 9: Study on the robustness of CASHEW and CASHEW-RL under simulated detection noise.

Model	Noise Type	ScienceQA	EgoSchema
Qwen3-VL-8B	No Noise	92.9	71.2
+ CASHEW	Remove 10% Objects	96.8	73.4
+ CASHEW	Add 10% Objects	96.9	73.6
+ CASHEW	Remove 10% + Add 10%	96.2	72.9
+ CASHEW	No Noise	**97.7**	**74.7**
+ CASHEW-RL	Remove 10% Objects	97.1	74.6
+ CASHEW-RL	Add 10% Objects	97.2	74.8
+ CASHEW-RL	Remove 10% + Add 10%	96.7	74.1
+ CASHEW-RL	No Noise	**97.8**	**75.5**

centage points). Only extreme thresholds result in moderate degradation, which is expected when verification becomes overly permissive or overly restrictive. These results show that CASHEW is not hypersensitive to grounding confidence tuning, and does not rely on delicate detector calibration.

(2) *Robustness to Detection Noise*. To assess dependency on perception quality, we simulate realistic detector noise by randomly removing 10% of verified objects (false negatives) and injecting 10% additional low-confidence objects (false positives). As shown in Table 9, CASHEW degrades only slightly (approximately 1.0 percentage points), while CASHEW-RL shows even smaller degradation (approximately 0.8 percentage points). These results indicate that aggregation integrates grounding cues as soft evidence rather than rigid constraints, and does not critically depend on precise detector outputs.

Table 10: Stability ablation with repeated sampling. Acc. Std.: accuracy standard deviation across runs; Ans. Agr.: average answer agreement.

Metric	Method	ScienceQA	SEED-Bench	NExT-QA	EgoSchema	VSI-Bench
Acc. Std. ↓	Qwen3-VL-8B	0.74	0.42	0.93	1.05	0.96
	+ CASHEW ($T = 1$)	0.50	0.36	0.70	0.79	0.82
	+ CASHEW ($T = 2$)	0.41	0.33	0.58	0.69	0.72
	+ CASHEW ($T = 3$)	**0.33**	**0.31**	**0.51**	**0.64**	**0.63**
Ans. Agr. ↑	Qwen3-VL-8B	94.1	88.3	83.5	80.9	77.4
	+ CASHEW ($T = 1$)	96.0	89.4	86.7	84.3	80.1
	+ CASHEW ($T = 2$)	96.9	90.3	88.1	85.6	81.5
	+ CASHEW ($T = 3$)	**97.8**	**90.5**	**89.1**	**86.0**	**82.4**

RQ3: Does CASHEW improve reasoning stability? A central motivation of CASHEW is that standard VLM reasoning can be unstable: the same input may produce different reasoning trajectories and inconsistent final answers across runs. To directly evaluate this effect, we conduct a repeated-sampling stability analysis with five independent stochastic decoding runs under the same temperature and top-p settings. We report two complementary metrics: accuracy standard deviation across runs, which measures performance variability, and average answer agreement, which measures prediction consistency. As shown in Table 10, CASHEW consistently improves stability across all benchmarks. Even with a single aggregation iteration ($T = 1$), CASHEW reduces run-to-run accuracy variance and increases answer agreement over the Qwen3-VL-8B baseline, indicating that aggregation suppresses unstable reasoning trajectories. Increasing the number of aggregation iterations further strengthens this effect. With $T = 3$, accuracy standard deviation decreases from 0.74 to 0.33 on ScienceQA, and from 1.05 to 0.64 on EgoSchema. Similarly, answer agreement improves from 94.1 to 97.8 on ScienceQA, and from 83.5 to 89.1 on NExT-QA. These results show that CASHEW does not merely improve average accuracy; it also makes multimodal reasoning more reliable under repeated stochastic sampling.

RQ4: How do SFT and RL contribute to aggregation behavior? CASHEW-RL is trained via a two-stage post-training pipeline, consisting of supervised fine-tuning (SFT) followed by reinforcement learning with GSPO. We analyze the contribution of each component from both a stage-wise and a mechanism-level perspective.

(1) *Stage-wise contribution of SFT and RL.* We first evaluate three variants: (1) the base VLM without post-training, (2) SFT-only, and (3) SFT + RL (i.e., CASHEW-RL) with a single aggregation ($T = 1$).

Results in Table 11 show that SFT alone yields mixed effects across benchmarks. While it improves performance on ScienceQA and NExT-QA, it leads to slight regressions on reasoning-oriented benchmarks such as EgoSchema and VSI-Bench. This suggests that SFT primarily enforces structural consistency in aggregation outputs but does not reliably improve evidence selection or reasoning quality. In contrast, adding GSPO-based RL produces consistent gains across all benchmarks, including those where SFT alone underperforms. For example, SFT leads to a drop of 1.1 percentage points on VideoMMMU and a 0.7 percentage point gain on VSI-Bench, whereas introducing RL results in gains of 3.4 and 1.2 percentage points on these benchmarks, respectively. Similar improvements are observed on VideoMMMU, where performance increases only after RL is applied. These results indicate that RL is essential for learning an effective aggregation policy that distinguishes informative trajectories from noisy ones, rather than merely imitating the aggregation format.

(2) *Disentangling RL and aggregation effects.*

Table 11: Stage-wise ablation study of supervised fine-tuning (SFT) and GSPO-based reinforcement learning (RL). SFT + RL corresponds to CASHEW-RL with a single aggregation step ($T = 1$).

Model	ScienceQA	Video-MME	NExT-QA	EgoSchema	VideoMMMU	VSI-Bench
Qwen3-VL-8B	92.9	66.9	75.6	71.2	47.3	58.8
+ SFT	93.6	66.5	76.1	70.8	46.2	58.1
+ SFT + RL (Ours)	**96.9** (+4.0)	**67.8** (+0.9)	**78.6** (+3.0)	**74.6** (+3.4)	**47.9** (+0.6)	**60.0** (+1.2)

Table 12: Ablation study on the effect of RL with and without trajectory aggregation. Single rollout evaluates the RL-tuned model without aggregation, while CASHEW-RL corresponds to aggregation with one iteration ($T = 1$).

Model	ScienceQA	Video-MME	NExT-QA	EgoSchema	VideoMMMU	VSI-Bench
Qwen3-VL-8B	92.9	66.9	75.6	71.2	47.3	58.8
+ CASHEW-RL (single rollout)	93.4	67.1	76.9	71.4	47.4	58.8
+ CASHEW-RL	**96.9** (+4.0)	**67.8** (+0.9)	**78.6** (+3.0)	**74.6** (+3.4)	**47.9** (+0.6)	**60.0** (+1.2)

While the stage-wise analysis shows that RL significantly improves performance when aggregation is enabled, it remains unclear whether these gains stem from improved intrinsic quality of individual trajectories or from more effective aggregation behavior. To isolate these factors, we evaluate CASHEW-RL under a single-rollout setting without aggregation, which measures standalone reasoning quality after RL tuning. We then compare it with CASHEW-RL with aggregation enabled ($T = 1$).

Results in Table 12 show that CASHEW-RL under single-rollout yields only modest improvements over the baseline model, indicating limited gains in intrinsic trajectory quality. In contrast, enabling aggregation ($T = 1$) leads to substantially larger improvements across tasks, with notable gains on benchmarks such as ScienceQA (+4.0 percentage points) and EgoSchema (+3.4 percentage points). These findings suggest that aggregation remains the dominant source of performance gains, while RL primarily enhances how candidate trajectories interact within the aggregation and visual verification framework. Rather than simply producing stronger individual trajectories, RL improves how informative trajectories are selected and integrated during aggregation.

RQ5: How does reinforcement learning affect the iterative behavior of CASHEW? We compare CASHEW and CASHEW-RL across iteration numbers $T \in \{1, 2, 3\}$ on Qwen3-VL-8B under identical decoding settings to examine how RL training alters the aggregation trajectory. Figure 3 shows that CASHEW-RL consistently outperforms CASHEW at every iteration. While both models benefit from increasing the iteration number T, CASHEW-RL starts from a stronger initial performance and maintains a higher improvement throughout the iterative aggregation process. Notably, its performance at iteration T is often comparable to or exceeds that of CASHEW at iteration $T + 1$, indicating that reinforcement learning training effectively improves the aggregation behavior.

Meanwhile, increasing T improve performance for CASHEW-RL, suggesting that RL does not eliminate the benefit of iterative aggregation. Instead, RL strengthens individual rollouts and enhances their compatibility with aggregation. Overall, RL reshapes the iterative behavior of CASHEW by improving per-iteration effectiveness while preserving the benefits of multi-step aggregation.

RQ6: How does CASHEW/CASHEW-RL compare to existing test-time scaling methods in the latency-performance tradeoff? We compare CASHEW and CASHEW-RL against three representative TTS baselines: Self-Consistency, Self-Selector, and Self-Synthesizer, measuring both average performance and end-to-end latency. All methods are evaluated on Qwen3-VL-8B under identical configurations ($N = 8$). For a fair comparison, CASHEW and CASHEW-RL are evaluated with a single aggregation iteration ($T = 1$). As shown in Figure 4, both CASHEW and CASHEW-RL achieve higher average performance than all baselines while maintaining comparable latency. Notably, CASHEW-RL reaches the highest accuracy with even lower latency than the other TTS methods. Importantly, as shown in Figure 3, increasing the iteration number T further improves the performance of both CASHEW and CASHEW-RL when additional time budget is available. This

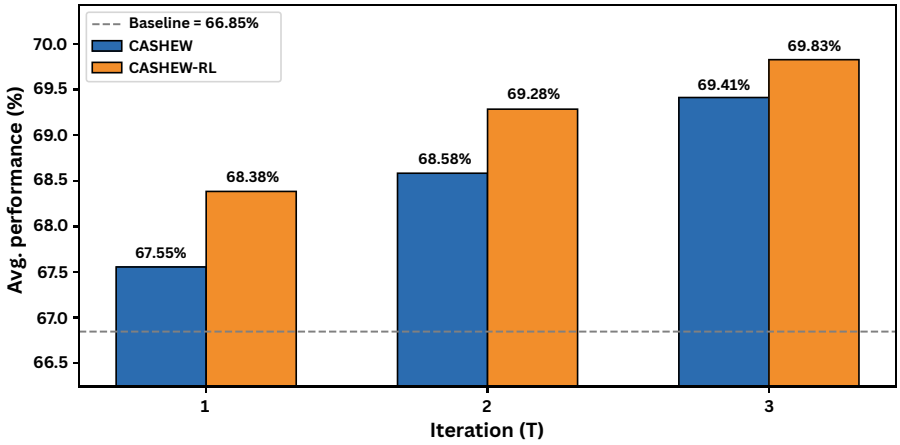

Figure 3: Performance comparison between CASHEW and CASHEW-RL across different T, with fixed population size $N = 8$ and group size $M = 4$. Performance is averaged over 13 benchmarks. For MME, scores are normalized to percentages for consistency.

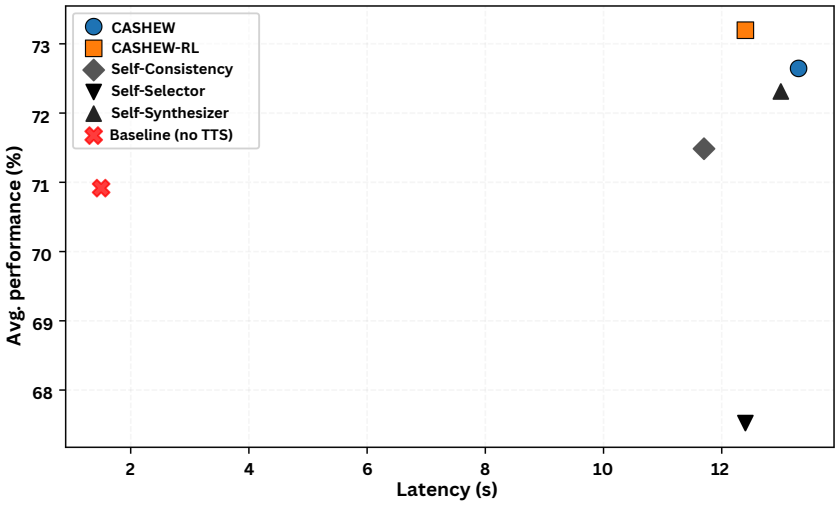

Figure 4: Latency-performance comparison between CASHEW, CASHEW-RL, and baselines TTS methods. CASHEW and CASHEW-RL are evaluated with $T = 1$. Performance is averaged over the five benchmarks reported in Table 3 in the main paper, and latency is measured as end-to-end inference time per sample.

demonstrates that CASHEW offers a favorable latency-performance tradeoff at $T = 1$, while remaining scalable under larger inference budgets.

Table 13: Comparison with recent training-free visual reasoning frameworks. MME: percentage scores.

Model	MME	MMMU	MathVista	HallusionBench	POPE	SEED-Bench
GPT-4o-mini	79.2	48.4	53.0	56.0	81.2	80.9
+ ReAct	87.3	54.8	49.3	51.1	–	–
+ PROREASON	91.9	**61.6**	54.9	59.9	–	–
+ CASHEW	**92.6**	61.4	**57.2**	**61.4**	**83.9**	**82.7**
Qwen3-VL-8B	84.4	62.4	77.2	61.1	88.9	78.7
+ SAP	87.3	62.3	–	–	**89.9**	77.3
+ CASHEW	**89.6**	**62.8**	**77.6**	**63.7**	**89.9**	**80.8**

RQ7: How does CASHEW compare with recent training-free visual reasoning frameworks? In addition to the canonical TTS baselines compared in the main paper, we further compare CASHEW with recent training-free visual reasoning frameworks. Specifically, we consider ReAct (Yao et al., 2023b), which interleaves reasoning traces with action-style steps; PROREASON (Zhou et al., 2025a), which decomposes multimodal reasoning into proactive visual perception and textual reasoning; and SAP (Shi et al., 2026), which performs saliency-guided multi-route inference. As shown in Table 13, CASHEW achieves strong performance across both backbone settings. Under GPT-4o-mini, CASHEW outperforms ReAct and PROREASON on 3 out of the 4 commonly reported benchmarks. Compared with the strongest recent baseline, PROREASON, CASHEW improves MME from 91.9 to 92.6, MathVista (Lu et al., 2024) from 54.9 to 57.2, and HallusionBench (Guan et al., 2024) from 59.9 to 61.4. Under Qwen3-VL-8B,

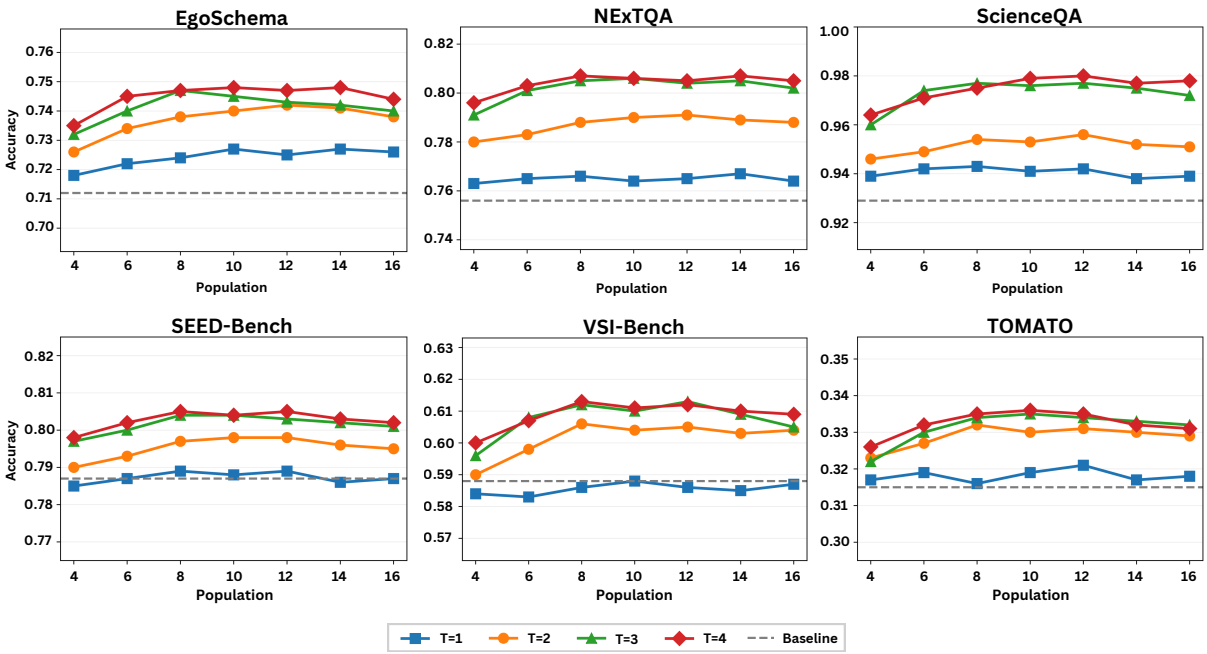

Figure 5: Performance across CASHEW population (N) for different values of T. All results with fixed $M = 4$.

Table 14: **Comparison of GSPO and GRPO for CASHEW-RL on Qwen3-VL-8B.** I: SEED-Bench results are reported only for the image subset.

Model	ScienceQA	SEED-Bench^I	EgoSchema	MVBench	VideoMMMU	VSI-Bench
Qwen3-VL-8B	92.9	78.7	71.2	66.2	47.3	58.8
+ CASHEW-RL (GRPO)	97.4	80.1	**75.5**	69.3	48.4	61.1
+ CASHEW-RL (GSPO)	**97.8**	**80.8**	**75.5**	**69.8**	**49.0**	**61.4**

CASHEW also consistently improves over SAP. For example, CASHEW improves MME from 87.3 to 89.6, MMMU (Yue et al., 2024) from 62.3 to 62.8, and SEED-Bench from 77.3 to 80.8. These results show that CASHEW's iterative trajectory aggregation remains effective not only against canonical TTS baselines, but also against recent training-free visual reasoning frameworks.

RQ8: How do the population size N and iteration number T affect CASHEW's performance? We study the effect of key aggregation hyperparameters in CASHEW by varying the population size $N \in \{4, 6, 8, 10, 12, 14, 16\}$ and the iteration number $T \in \{1, 2, 3, 4\}$, while fixing the group size to $M = 4$. Figure 5 reports results across six benchmarks, with all other decoding parameters held constant. Across tasks, increasing the population size N from small to moderate values (e.g., $4 \rightarrow 8$) yields consistent gains for $T = 2, 3, 4$, while further increases provide only marginal improvements or lead to slight regressions. Performance also improves as T increases from 1 to 3, with $T = 3$ offering a strong and stable operating point across benchmarks. Increasing to $T = 4$ can yield marginal gains, primarily at the cost of increased computational overhead and latency. Overall, the results indicate that a moderate population size of $N = 8$ combined with $T = 3$ iterations offers a favorable trade-off between performance and computational cost.

RQ9: How robust is CASHEW-RL across different RL training algorithms? In the main paper, CASHEW-RL is trained using GSPO, which has been reported to show stable optimization behavior in prior work (Zheng et al., 2025b). To evaluate whether CASHEW-RL depends on a specific RL algorithm, we additionally train CASHEW-RL using GRPO under identical settings, and compare it with the GSPO variant on benchmarks across different tasks. Table 14 shows that GSPO achieves slightly higher performance on most benchmarks, although the differences are small. For example, GSPO outperforms GRPO by 0.7 on SEED-Bench and 0.6 on VideoMMMU, while performance is nearly identical on EgoSchema. Overall, these results indicate that the gains of CASHEW-RL are not tied to a particular RL optimizer. The aggregation-aware training objective remains effective across

different policy optimization methods.

Table 15: Effect of chain-of-thought reasoning in CASHEW under identical aggregation settings.

Model	ScienceQA	EgoSchema	VideoMMMU	VSI-Bench
Qwen3-VL-8B	92.9	71.2	47.3	58.8
+ CASHEW (w/o CoT)	96.6	72.3	47.6	59.7
+ CASHEW	**97.7**	**74.7**	**48.4**	**61.2**

RQ10: What is the contribution of chain-of-thought (CoT) in CASHEW? To isolate the contribution of explicit chain-of-thought reasoning, we compare CASHEW with and without CoT under identical settings. As shown in Table 15, CASHEW without CoT already yields substantial gains over the base model (e.g., 92.9 → 96.6 on ScienceQA), indicating that iterative aggregation with visual verification is the primary source of improvement. Adding CoT further improves performance, particularly on reasoning-oriented benchmarks such as VideoMMMU and VSI-Bench, suggesting that explicit step-by-step reasoning is especially beneficial for temporal and multi-step inference tasks. Overall, aggregation remains the dominant driver of performance gains, while CoT provides additional benefits on tasks requiring structured reasoning.

E Evaluation Benchmarks

We evaluate CASHEW across three categories of multimodal benchmarks: (1) image understanding, (2) video understanding, and (3) video reasoning. Image understanding benchmarks include ScienceQA (Lu et al., 2022), MME (Fu et al., 2024), POPE (Li et al., 2023b), and SEED-Bench (image subset) (Li et al., 2024b). Video understanding benchmarks cover both short- and long-form video comprehension, including Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), EgoSchema (Mangalam et al., 2023), MVBench (Li et al., 2024c), and NExT-QA (Xiao et al., 2021). Finally, video reasoning benchmarks consist of VideoMMMU (Hu et al., 2025), VSI-Bench (Yang et al., 2025), Video-TT (Zhang et al., 2025b), and TOMATO (Shangguan et al., 2025).

F Prompt Templates

Figure 6 shows the prompt templates used in CASHEW. The **population initialization prompt** generates initial candidate trajectories from the input media (image or video) and query. The **grounded aggregation prompt** guides iterative aggregation by incorporating verified visual objects, while the **final aggregation prompt** synthesizes information from a larger candidate set to produce a single final answer. All prompts enforce a consistent output structure, ensuring reliable aggregation.

Figure 7 shows the prompt template used for supervised fine-tuning (SFT). The SFT aggregation prompt trains the model to consolidate multiple candidate trajectories into a single grounded answer by producing a reasoning chain, identifying relevant visual objects, and generating a concise final response. This prompt enforces the same structured output format as the test-time aggregation prompts, providing consistent supervision for learning aggregation and visual grounding behaviors.

Figure 8 shows the prompts used by the Self-Synthesizer baseline. Unlike CASHEW, Self-Synthesizer performs a single-round aggregation without explicit grounding signals. The population initialization prompt is shared to ensure fairness, with only the aggregation prompt differing in structure and the information it receives.

G Qualitative Examples

Figure 9 compares the Qwen3-VL-8B baseline (Bai et al., 2025a) with CASHEW and CASHEW-RL. **In the first example (top)**, the baseline observes "careful alignment to ensure the frame is level" but over-generalizes, incorrectly assuming window installation. This reflects a factual hallucination from single-path reasoning. CASHEW corrects this by iteratively aggregating multiple trajectories and grounding intermediate object-level claims such as "a long, narrow channel", "repeatedly adjusting it", and "frequently places a spirit level on the channel", rejecting the unsupported hypotheses. CASHEW-RL further internalizes this behavior, identifying informative visual cues like the cable channel, screws, and marker, and producing a structured, grounded conclusion. **In the second example (bottom)**, the Qwen3-VL-8B baseline fails temporally, focusing on right-hand actions despite the video showing alternating left- and right-hand use. CASHEW grounds hand usage across multiple temporal segments and aggregates these observations into a consistent interpretation. CASHEW-RL internalizes this reasoning, attending to key objects such as hands, chips, basket, and table, producing a concise, temporally grounded explanation that correctly tracks both hands. Figure 10 provides a detailed view of this iterative aggregation process for the first example.

Prompt for Population Initialization

You are given a {media_type} and a question.

Question: {question}

Carefully reason step by step inside <think>...</think> tag, then output only one concise final answer inside <answer>...</answer> tag.

Prompt for Grounded Aggregation

You are given a {media_type}, a question, a group of candidate answers, and some objects that are already verified and important to answer the question. The candidates may be wrong or incomplete. Review them carefully and generate a new answer.

Question: {question}

Candidate Answers:
Candidate #1 ###: {candidate1}
Candidate #2 ###: {candidate2}
Candidate #3 ###: {candidate3}
Candidate #4 ###: {candidate4}

Key Objects: [{visual_keys}]

Write your reasoning inside <think>...</think> tag and end with exactly one concise answer inside <answer>...</answer> tag.

Prompt for Final Aggregation

This is the final aggregation round. You are given a {media_type}, a group of candidate answers, and some objects that are already verified and important to answer the question. Read **all** candidate answers carefully and aggregate useful information from them to produce **exactly one** final answer.
Question: {question}

Candidate Answers:
Candidate #1 ###: {candidate1}
Candidate #2 ###: {candidate2}
...
Candidate #7 ###: {candidate7}
Candidate #8 ###: {candidate8}

Key Objects: [{visual_keys}]

Even if the information from the candidates and the visual input is insufficient, please make your best possible guess based on the question. Write your reasoning inside <think>...</think> tag and end with **exactly one** final answer and place it inside <answer>...</answer> tag.

Figure 6: Prompt templates for different stages of CASHEW, including population initialization, grounded aggregation, and final aggregation.

Prompt for supervised fine-tuning (SFT)

You are given a `{media_type}`, a question, and a group of candidate answers. The candidates may be wrong or incomplete. Aggregate the useful ideas and produce a single, high-quality answer. Be concise and correct.

Question: `{question}`

Candidate Answers:
Candidate #1 ###: `{candidate1}`
Candidate #2 ###: `{candidate2}`
Candidate #3 ###: `{candidate3}`
Candidate #4 ###: `{candidate4}`

Output Requirements:
1. In <think></think>, write a single coherent reasoning chain that compares and aggregates the candidate answers, and discards incorrect or unsupported claims.
2. In <visual_keys></visual_keys>, output a Python-style list of objects from the visual input that are most relevant for answering the question. Only include objects that provide useful visual evidence.
3. In <answer></answer>, provide one concise and correct final answer.

Output Format:

<think>
Your reasoning chain here.
</think>

<visual_keys>
["object_1", "object_2", ...]
</visual_keys>

<answer>
Your final answer here.
</answer>

Figure 7: Prompt used for supervised fine-tuning (SFT) in CASHEW-RL. The prompt enforces a structured output format consisting of a reasoning chain, a list of visual keys, and a final answer, enabling the model to learn aggregation and grounding behaviors from demonstrations.

Prompt for Population Initialization

You are given a {media_type} and a question.

Question: {question}

Carefully reason step by step inside <think>...</think> tag, then output only one concise final answer inside <answer>...</answer> tag.

Prompt for Self-Synthesizer Aggregation

This is the final aggregation round. You are given a {media_type}, and several candidate answers. Read **all** candidate answers carefully and aggregate useful information from them to produce **exactly one** final answer.
Question: {question}

Candidate Answers:
Candidate #1 ###: {candidate1}
Candidate #2 ###: {candidate2}
Candidate #3 ###: {candidate3}
Candidate #4 ###: {candidate4}
Candidate #5 ###: {candidate5}
Candidate #6 ###: {candidate6}
Candidate #7 ###: {candidate7}
Candidate #8 ###: {candidate8}

Figure 8: Prompt templates for Self-Synthesizer in ablation study.

Query: Based on the actions in the video, what was the primary objective that the person was trying to accomplish and what tool did he use repeatedly to check the accuracy of his work?
A. The person was trying to install a new window. he used a spirit level to make sure the window was level.
B. The person was attempting to fix a leak located in the wall. he utilized a spirit level tool to ensure the applied patch was perfectly level.
C. The person was trying to attach a cable channel to the wall. he used a spirit level to check the accuracy of his work. (Correct Answer)
D. The person was attempting to hang a picture carefully. utilizing a spirit level, he ensured the picture was perfectly level and straight.
E. The person was attempting to construct a shelf. diligently, he utilized a spirit level to ensure the shelf remained perfectly level and even.

Qwen3-VL-8B: The person is working against a wall opening and repeatedly using a spirit level, which could logically suggest **they are installing a window**, since window installation requires **careful alignment to ensure the frame is level**. Therefore, the correct answer is: **A. The person was trying to install a new window. he used a spirit level to make sure the window was level.** ✗

+CASHEW : <think> The video shows the person **positioning a long, narrow channel along the wall** and **repeatedly adjusting it** before fixing it in place. Throughout this process, he **frequently places a spirit level** on the channel to ensure it is straight and properly aligned.**</think>**
<Answer>C. The person was trying to attach a cable channel to the wall. he used a spirit level to check the accuracy of his work. </Answer> ✓

+CASHEW-RL : <think> At the start of the clip, the person measures and aligns **a long, narrow plastic / metal channel** against the wall, marking positions and holding it temporarily in place. He then **repeatedly adjusts the channel, places a spirit level on top, and repositions** until the bubble indicates perfect alignment before securing it with screws. There is **no indication of a window, picture frame, shelf, or wall repair**; the actions and object shape are consistent with **installing a cable channel**.**</think>**
<visual_keys>["cable channel", "screws", "measuring tape", "marker"]**</visual_keys>**
<Answer>C. The person was trying to attach a cable channel to the wall. he used a spirit level to check the accuracy of his work. </Answer> ✓

Query: How do the people in the video eat potato chips?
A. Left-handedly.
B. Right-handedly.
C. Using both hands. (Correct Answer)
D. Using a fork.

Qwen3-VL-8B: The video shows the people repeatedly reaching for potato chips and bringing them to their mouths **using their right hands**. There is **no instance of left-hand use, both hands together**, or any utensils like a fork. The consistent hand choice across multiple bites indicates right-handed eating. Therefore, the correct answer is: **B. Right-handedly**. ✗

+CASHEW : <think> In this video, the person alternates between using the left hand and the right hand to pick up potato chips from the paper-lined basket. The action shows switching hands over time, which rules out "both hands" as a simultaneous eating method and also rules out exclusive left- or right-handed eating.</think>
<Answer>C. Using both hands.</Answer> ✓

+CASHEW-RL: <think>In the video, the person eats potato chips by **alternately using the left and right hands to pick chips** from a basket. At different moments, **either hand reaches into the pile of chips**, but each individual eating action involves only one hand at a time.**</think>**
<visual_keys>["watch", "person's hand", "chips", "basket", "table"]**</visual_keys>**
<Answer>C. Using both hands.</Answer> ✓

Figure 9: **Qualitative examples of CASHEW and CASHEW-RL**. **Red** text indicates errors or hallucinations produced by the baseline Qwen3-VL-8B. **Green** and **blue** text highlight correct, visually grounded reasoning generated by CASHEW and CASHEW-RL, respectively.

Query: Based on the actions in the video, what was the primary objective that the person was trying to accomplish and what tool did he use repeatedly to check the accuracy of his work?
A. The person was trying to install a new window. he used a spirit level to make sure the window was level.
B. The person was attempting to fix a leak located in the wall. he utilized a spirit level tool to ensure the applied patch was perfectly level.
C. The person was trying to attach a cable channel to the wall. he used a spirit level to check the accuracy of his work. (Correct Answer)
D. The person was attempting to hang a picture carefully. utilizing a spirit level, he ensured the picture was perfectly level and straight.
E. The person was attempting to construct a shelf. diligently, he utilized a spirit level to ensure the shelf remained perfectly level and even.

visual_keys: ["cable channel", "long narrow channel", "wall", "spirit level", "drill", "screws", "marker", "measuring tape", "window frame", "hand"]

T=0

P1: <answer>A</answer>
<think> The person is working beside a window frame and repeatedly checks alignment with a spirit level. Since window installation requires careful leveling, the task may be installing or adjusting a new window. **</think>**

P2: <answer>C</answer>
<think> The person holds and positions a long narrow channel against the wall near the window. He marks the placement and uses a drill while checking the straightness with a spirit level. This is consistent with attaching a cable channel to the wall. **</think>**

P3: <answer>E</answer>
<think> The person uses a long straight object and a spirit level, which could indicate shelf construction. However, the video does not clearly show a shelf board or brackets. **</think>**

P4: <answer>C</answer>
<think> A narrow cable trunking/channel is being aligned vertically and then fixed with a drill. The spirit level appears near the work area, suggesting it is used to keep the channel straight. **</think>**

P5: <answer>A</answer>
<think> Because the person works around the window and curtain area, the objective might be related to installing or adjusting window hardware. The spirit level is likely used for alignment. **</think>**

P6: <answer>C</answer>
<think> The person appears to install a long plastic cover or channel along the wall. He checks alignment and then drills it into place, so the repeated tool is the spirit level. **</think>**

P7: <answer>B</answer>
<think> The wall has visible marks and the person works along the wall with tools. This could be a wall repair task, and the spirit level may be used to keep the repaired area even. **</think>**

P8: <answer>C</answer>
<think> The video shows a channel being handled, aligned, and fastened. The person uses a drill and keeps checking straightness with a spirit level. This supports cable-channel installation rather than window installation. **</think>**

Round 1 Aggregation

S1 = {P1, P2, P3, P5}, S2 = {P1, P2, P4, P6}, S3 = {P2, P4, P6, P8}, S4 = {P3, P5, P7, P8}, S5 = {P1, P4, P5, P7}, S6 = {P2, P3, P6, P7}, S7 = {P1, P6, P7, P8}, S8 = {P3, P4, P5, P8}

T=1

P1 from S1: <answer>C</answer>
<think> The person aligns a long narrow channel on the wall and repeatedly checks it with a spirit level before fixing it. This suggests he is attaching a cable channel to the wall. **</think>**

P2 from S2: <answer>C</answer>
<think> The video shows a channel-like object being positioned, leveled, and secured along the wall. The repeated checking tool is the spirit level, so the task is installing a cable channel. </think>

P3 from S3: <answer>C</answer>
<think> A long narrow piece is adjusted several times and checked with a spirit level before being fixed in place. This matches attaching a cable channel to the wall. **</think>**

P4 from S4: <answer>C</answer>
<think> The object being handled is a narrow channel rather than a window or shelf. The person uses a spirit level repeatedly, indicating he is installing a cable channel straight on the wall. **</think>**

P5 from S5: <answer>A</answer>
<think> The work happens next to a window, and the person repeatedly checks level. This suggests he is installing a window-related structure. **</think>**

P6 from S6: <answer>C</answer>
<think> A narrow channel is placed against the wall, leveled, and then secured. These actions are most consistent with attaching a cable channel. **</think>**

P7 from S7: <answer>C</answer>
<think> The person lines up a channel on the wall and checks it with a spirit level before fastening it. This indicates cable-channel installation. **</think>**

P8 from S8: <answer>C</answer>
<think> The person repeatedly adjusts a long narrow channel and checks whether it is level before fixing it. This is best explained by attaching a cable channel to the wall. **</think>**

Round 2 Aggregation

S1 = {P1, P2, P3, P5}, S2 = {P1, P3, P4, P6}, S3 = {P2, P4, P6, P8}, S4 = {P1, P5, P7, P8}, S5 = {P1, P3, P6, P7}, S6 = {P3, P4, P5, P8}, S7 = {P1, P2, P4, P7}, S8 = {P2, P5, P6, P8}

T=2

P1 from S1: <answer>C</answer>
<think> The person aligns a cable channel against the wall and repeatedly checks it with a spirit level before securing it. This indicates the task is attaching a cable channel. **</think>**

P2 from S2: <answer>C</answer>
<think> A long channel is measured, leveled, and fixed along the wall. The repeated use of the spirit level shows it is being used to check the installation's accuracy. **</think>**

P3 from S3: <answer>C</answer>
<think> The person works with a narrow channel-like object rather than a window or shelf. He checks it with a spirit level and then secures it, which supports cable-channel installation. **</think>**

P4 from S4: <answer>C</answer>
<think> The repeated pattern is align the long channel, check level, adjust it, and then fasten it. This matches attaching a cable channel to the wall with a spirit level used for checking. **</think>**

P5 from S5: <answer>C</answer>
<think> The key action is installing a narrow channel along the wall and repeatedly checking whether it is straight. The tool is the spirit level. **</think>**

P6 from S6: <answer>C</answer>
<think> The installed object is a separate long narrow channel on the wall. The person repeatedly uses a spirit level to align it before securing it. **</think>**

P7 from S7: <answer>C</answer>
<think> The video consistently shows the person adjusting a cable channel and checking its alignment with a spirit level. This indicates he is attaching the channel to the wall. **</think>**

P8 from S8: <answer>C</answer>
<think> The person installs a long narrow channel along the wall and checks its position with a spirit level before fixing it permanently. This is consistent with cable-channel installation. **</think>**

Final Aggregation

<answer>C</answer>
<think> The video shows the person positioning a long, narrow channel along the wall and repeatedly adjusting it before fixing it in place. Throughout this process, he frequently places a spirit level on the channel to ensure it is straight and properly aligned.**</think>**

Figure 10: **Example correction process of CASHEW.** Starting from eight trajectories with mixed predictions, CASHEW iteratively aggregates random subsets of four trajectories with verified visual keys.