SVFSearch: A Multimodal Knowledge-Intensive Benchmark for Short-Video Frame Search in the Gaming Vertical Domain

Anonymous authors
Paper under double-blind review

Abstract

Multimodal large language models are increasingly used as agent backbones that understand multimodal inputs, plan retrieval actions, invoke external tools, and reason over retrieved information. Yet existing benchmarks rarely evaluate this ability in short-video applications, where a paused frame is often visually ambiguous and answering requires vertical, long-tail, and fast-evolving domain knowledge. We introduce SVFSearch, the first open benchmark for short-video frame search in the Chinese gaming domain. SVFSearch contains 5,000 four-choice test examples and 4,198 auxiliary training examples, each centered on a paused game scene from a real short-video clip. To support fair and reproducible evaluation, SVFSearch provides a frozen offline retrieval environment with a game-domain text corpus, a topic-linked image gallery, and text, image, and multimodal retrieval interfaces, avoiding reliance on uncontrolled web search APIs. We evaluate representative paradigms ranging from direct QA and RAG workflow to Plan-Act-Replan agents and learned search models. Results reveal a large gap between model-only answering, practical agentic search, and oracle knowledge: the best open-source direct-QA model reaches 66.4%, the best practical agent achieves 79.1%, and oracle knowledge reaches 95.4%. Further analysis exposes bottlenecks in visual grounding, retrieval quality, evidence-grounded reasoning, and tool-use behavior, including over-search, answer-only shortcuts, and retrieval-induced misleading.

1 Introduction

Multimodal large language models (MLLMs) are shifting from passive multimodal predictors to active controllers in agentic systems, as seen in both proprietary systems such as GPT-5, Claude 4, and Gemini 3.1, and open-weight models such as Qwen3-VL and Qwen3.5 (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023; 2024a; Wang et al., 2024a; OpenAI, 2025; Anthropic, 2025; Google AI for Developers, 2026; Bai et al., 2025). They are increasingly used as agent backbones that analyze multimodal context, plan tool use, acquire external evidence, integrate returned information, and dynamically re-plan their reasoning process (Yao et al., 2022; Shinn et al., 2023; LangChain Inc., 2024). Accordingly, recent benchmarks have begun to evaluate search-oriented multimodal behavior, including when to search, which tool to use, and how to process retrieved evidence to produce the final response (Jiang et al., 2024; Tao et al., 2025; Li et al., 2024; Hu et al., 2024).

Despite this progress, a practical short-video interaction remains underexplored: *search from a paused short-video frame*. Users browsing gameplay videos may pause on an unfamiliar entity or scene and seek further information. Here, "short-video" describes the application scenario and user interaction; the benchmark evaluates a single paused frame and question, without temporal video modeling. The frame anchors a search process that requires identifying the relevant game entity, retrieving long-tail domain knowledge, and integrating evidence to answer the question. The contribution is a gaming-domain benchmark with frozen multimodal retrieval resources and controlled evaluation of direct QA, fixed RAG workflows, and agentic search. This setting complements encyclopedic visual QA (Chen et al., 2023; Mensink et al., 2023) and existing multimodal-search benchmarks by emphasizing game-specific visual grounding and external knowledge acquisition.

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

In this work, we instantiate short-video frame search in the specialized Chinese gaming domain, where answering often requires recognizing game scenes, characters, equipment, maps, mechanics, version-specific content, and community knowledge. We introduce SVFSearch, **the first open benchmark** for this setting. Each example is formulated as a four-choice QA task over a paused game frame and a corresponding user question, with the ground-truth answer and an answer rationale. SVFSearch contains 5,000 test items and 4,198 auxiliary train items. We additionally release video-side metadata, including the video title, cover-frame OCR, and ASR transcription, although these fields are not used in the main evaluation.

To support reproducible evaluation, we release an offline retrieval repository with a game-vertical text corpus and a topic-linked image gallery, avoiding dependence on online commercial search APIs. We evaluate representative paradigms on SVFSearch, including direct QA, oracle knowledge QA, RAG workflow, LangGraph-based agents, and MMSearch-R1 (LangChain Inc., 2024; Wu et al., 2025). Results reveal a clear search gap. On the same Qwen3.5-9B backbone, accuracy improves from 59.9% with direct QA to 66.5% with RAG workflow and further to 79.1% with LangGraph-based agents, yielding a 19.2-point gain over the no-search setting. These findings highlight both the value of retrieval and the remaining challenges in evidence acquisition and efficient search.

Our contributions are summarized as follows:

(1) **We introduce SVFSearch, the first agent-facing multimodal search benchmark for short-video frame search in the Chinese gaming vertical domain.** Built from real short-video clips, SVFSearch evaluates game-scene understanding and retrieval-augmented QA over paused frames and user questions, and releases video-side metadata for future multi-source evaluation.

(2) **We provide a reproducible offline retrieval environment.** SVFSearch ships a game-vertical text corpus, a topic-linked image gallery, and unified retrieval interfaces, enabling fair and reproducible evaluation without relying on uncontrolled online search APIs.

(3) **We benchmark representative multimodal-search paradigms and analyze their limitations.** We evaluate direct QA, oracle knowledge QA, RAG workflow, LangGraph-based agents, and MMSearch-R1, revealing a large performance gap and recurring tool-use failures such as over-search, answer-only shortcuts, and retrieval-induced misleading.

2 SVFSearch

2.1 Overview and Task Formulation

SVFSearch is a multimodal search benchmark with an agent-facing offline retrieval environment for game-scene understanding and retrieval-augmented question answering. Unlike standard visual QA benchmarks that mainly evaluate the direct answering ability of MLLMs, SVFSearch supports a broader range of systems, including direct-QA MLLMs, fixed RAG workflows, and autonomous multimodal agents that must decide whether to retrieve evidence and how to use it. Each example is centered on a paused game scene and formulated as a four-choice QA task that requires selecting the correct answer by grounding the visual context and using external evidence when needed.

In SVFSearch, each instance follows the format illustrated in Figure 2: a paused-frame image, a question, four candidate answers, the ground-truth answer, and an answer rationale. We also release video-side metadata, including the video title, cover-frame OCR text, and ASR transcript, to support future studies of noisy video context and short-video information seeking. The benchmark contains a held-out test split and an auxiliary training split for training robust search-aware models. To support reproducible evaluation, SVFSearch further provides a standardized offline retrieval environment repository consisting of a game-vertical text corpus, a topic-linked image gallery, and frozen retrieval indices built over these resources, enabling controlled evaluation of visual grounding, tool selection, evidence retrieval, and final reasoning.

2.2 Data Construction Pipeline

As shown in Figure 1, we construct SVFSearch around *game-specific core elements* rather than generic image captions or web QA pairs. This design grounds each example in both concrete visual

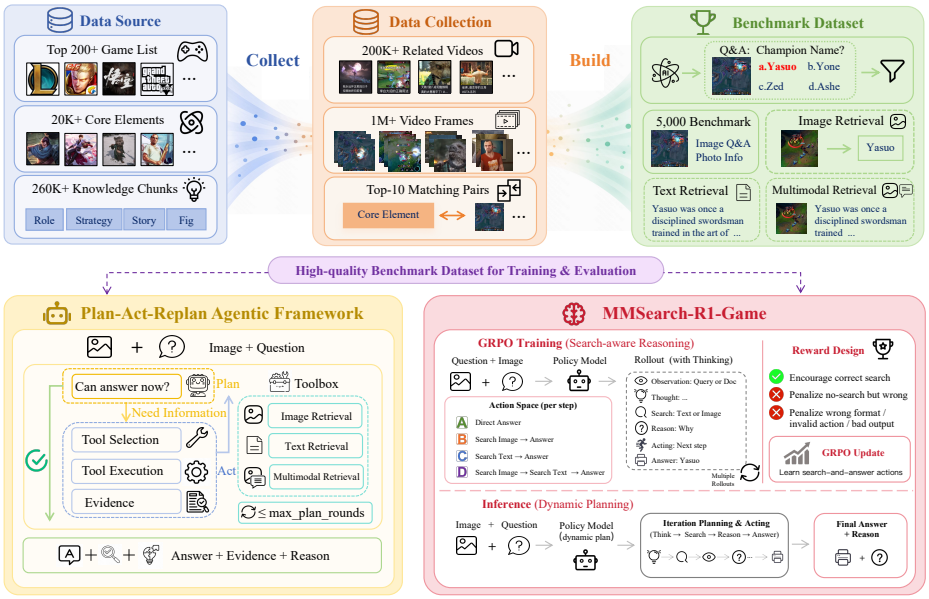


Figure 1: **Overview of SVFSearch.** *Top row:* benchmark construction from game-specific core elements, short-video frames, and web-sourced knowledge to QA splits and frozen retrieval resources. *Bottom left:* a Plan-Act-Replan agent that dynamically decides whether more information is needed, selects retrieval tools, and integrates returned evidence before answering. *Bottom right:* MMSearch-R1-Game, which learns search-and-answer actions through GRPO training in the same frozen retrieval environment.

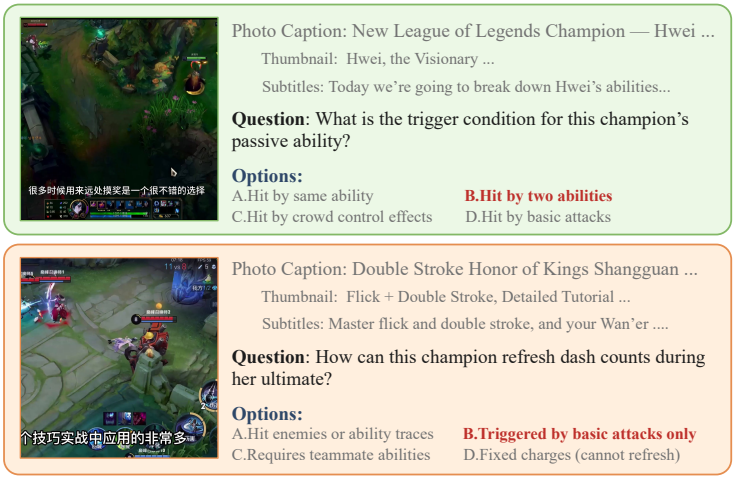


Figure 2: **Representative examples from SVFSearch.** Examples show paused frames, video-side metadata, and multiple-choice QA instances.

scenes and game-domain knowledge. The pipeline has three stages: core-element and knowledge construction, short-video-based visual grounding, and QA generation with quality filtering.

Stage 1: Core Element and Knowledge Construction. We first collect 221 popular games covering diverse genres. Based on in-platform user queries, we mine game-specific *core elements* for each game, including characters, equipment, maps, story events, skills, and gameplay mechanics. This process yields 22,800 core elements in total. For each core element, we retrieve multiple related knowledge sources through search engines. The retrieved content is then cleaned, summarized, and

Table 1: **Comparison with representative multimodal QA benchmarks.** ✓ = present; ✗ = absent. **T-KB/I-KB**: text/image retrieval resources. **Agent**: supports tool-augmented retrieval evaluation. **Dom.**: evaluation domain. † denotes an auxiliary training split for search-aware model development.

Dataset	#Test	#Train	T-KB	I-KB	Agent	Dom.
VQAv2 (Goyal et al., 2017)	214K	443K	✗	✗	✗	General
OK-VQA (Marino et al., 2019)	5.0K	9.0K	✗	✗	✗	General
InfoSeek (Chen et al., 2023)	8.9K	1.35M	Wiki	✗	✗	General
CharXiv (Wang et al., 2024b)	2.3K	–	✗	✗	✗	Charts
GMAI-MMBench (Chen et al., 2024)	26K	–	✗	✗	✗	Medical
Dyn-VQA (Li et al., 2024)	1.5K	–	Web	Web	✓	General
MMSearch (Jiang et al., 2024)	300	–	Web	Web	✓	General
MMSearch-Plus (Tao et al., 2025)	311	–	Web	Web	✓	General
SVFSearch (ours)	5.0K	4.2K†	45K	34K	✓	Gaming

chunked with LLM assistance, producing standardized textual units for retrieval and QA construction. Finally, we obtain a text knowledge base with over 260K knowledge chunks grounded in the mined core elements.

Stage 2: Visual Grounding through Short-Video Retrieval. To connect textual game knowledge with concrete visual scenes, we use the game name and core element as the query to retrieve short-video content. In total, we collect more than 200K short videos related to the mined core elements, and extract over 1M candidate frames using `ffmpeg`. We then use an MLLM to verify whether each candidate frame visually matches the target core element, retaining up to 10 top-ranked frames for each core element. After filtering, we obtain 43,130 reliable core-element–image pairs with strong visual grounding. These pairs provide the visual basis for subsequent QA construction.

Stage 3: QA Generation and Quality Filtering. Given a visually grounded core element, its matched image, and the associated knowledge, an 8B-parameter model generates approximately 80K multiple-choice QA candidates that cover visual, knowledge-grounded, and reasoning-oriented questions. A 32B-parameter model then assesses each candidate's question quality, answer correctness, distractor plausibility, and difficulty. After quality assessment, difficulty annotation, and manual spot checks, we retain 9,198 high-quality QA instances, including 4,198 auxiliary training examples and 5,000 test examples. Each retained instance is linked back to its corresponding video-side metadata.

2.3 Retrieval Resources and Frozen Indices

A central design goal of SVFSearch is to provide a reproducible offline retrieval environment rather than relying on online web search. We therefore release frozen retrieval resources for RAG workflows and multimodal agents. The image pool is built from the 43,130 visually grounded core-element–image pairs. We remove the 9,198 QA images to avoid exact-image retrieval leakage, resulting in 33,932 indexed game images, each associated with its game and core-element information. The text knowledge base contains 45,608 structured entries, segmented into 262,938 retrieval chunks. We build frozen retrieval resources over this corpus, including a dense text index using 512-dimensional Qwen3-Embedding-0.6B representations, a BM25 lexical retrieval tool, an image index using 256-dimensional features from a fine-tuned DINOv3-Base model, and a multimodal index using 512-dimensional Qwen3-VL-Embedding-2B representations for both image entries and text chunks. Together, these resources enable controlled retrieval evaluation and help diagnose failures in visual grounding, entity matching, knowledge retrieval, evidence aggregation, and final reasoning.

2.4 Benchmark Characteristics

As shown in Table 1, SVFSearch differs from prior multimodal QA benchmarks by coupling short-video frame questions with a frozen, domain-specific retrieval environment. The benchmark provides both text and image resources, enabling controlled evaluation of direct-QA, RAG workflows, and multimodal agents under the same offline setting. This design also allows agentic systems to decide

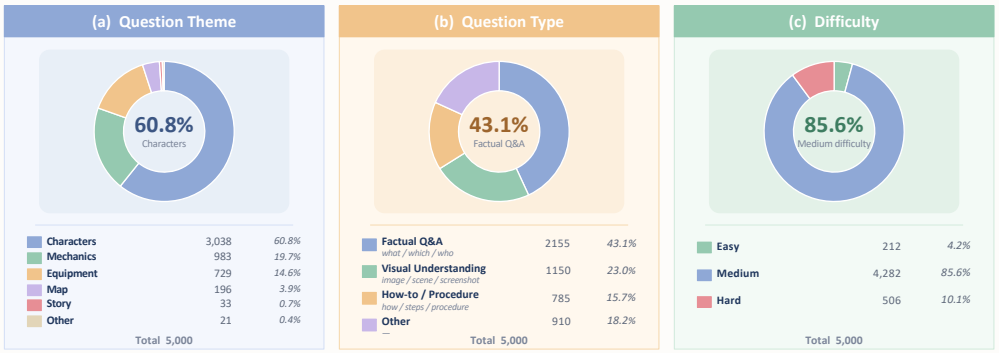


Figure 3: **Distribution analysis of SVFSearch.** Test examples grouped by question theme, question type, and difficulty. The test split is dominated by character questions, factual Q&A types, and medium-difficulty examples, while retaining long-tail themes and harder cases for stratified analysis.

whether additional evidence is needed and which retrieval interface to use, including text, image, and multimodal retrieval.

Figure 3 shows that the 5,000-example test split reflects realistic game-scene search demand. Character and mechanics questions dominate, while equipment, map, story, and other long-tail themes are retained for stratified analysis. The benchmark further covers factual, visual-understanding, procedural, and reasoning-oriented questions across easy, medium, and hard difficulty levels. Each released instance has a unique image–question pair, while the same core element may appear in different visual contexts and support different questions.

3 Evaluation Protocols and Search Baselines for SVFSearch

To evaluate representative paradigms on SVFSearch, we build retrieval backends and search baselines using the frozen offline resources introduced in Section 2.3. We first present the core-element-aware image retrieval backend, and then instantiate the retrieval-dependent evaluation settings, including a fixed RAG workflow, a LangGraph-based Plan-Act-Replan agent, and MMSearch-R1 search models.

All tool-augmented settings operate on the same offline retrieval resources. The available tools include `img_ann`, `text_ann`, `bm25_ann`, and `multimodal_ann`. The outputs of `img_ann` are adapted to different search settings. For MMSearch-R1 models, we keep `img_ann` aligned with the original MMSearch-R1 image-retrieval interface, where an image search returns compact metadata rather than long knowledge passages. Specifically, each image entry in the ANN index is associated with metadata fields containing its game name and core element, so an `img_ann` call returns concise element-level signals. For the LangGraph agent system, retrieved core elements from `img_ann` are further used to invoke `kn_lookup`, which returns detailed structured knowledge passages.

3.1 Core-element-aware Image Retrieval Backend

Image retrieval in SVFSearch identifies the core element in a paused frame. We initialize DINOv3-Base from its official pretrained checkpoint and fine-tune it with a cluster-aware InfoNCE objective using AdamW (learning rate 10^{-5}, batch size 256). For each core element, we encode its images with DINOv3-Large and apply K-means++ to group visually diverse appearances into clusters.

Each epoch samples one positive pair per core element by randomly selecting a cluster and two images within it. In-batch images from other core elements serve as negatives; other clusters of the same element are not treated as negatives. All benchmark QA images are excluded from encoder training and the retrieval index. At inference, we retrieve nearest neighbors using 256-dimensional embeddings, cosine similarity, and FAISS, then map the retrieved images to core elements for knowledge retrieval and answering.

3.2 Fixed RAG Workflow and Plan-Act-Replan Agent

We include a fixed RAG workflow as the simplest tool-augmented baseline. It first calls `img_ann` to retrieve visually similar images, aggregates the associated core elements by voting, and uses the selected element together with the original question to query `text_ann`. The retrieved evidence is then provided to the MLLM along with the image, question, and options. Since the retrieval order is fixed, this workflow cannot adapt its search strategy based on intermediate evidence.

We further instantiate a Plan-Act-Replan (PAR) agent with LangGraph (LangChain Inc., 2024). At each round, the model observes the image, question, options, accumulated evidence, and previous tool outputs, then decides whether to answer or continue searching. When more evidence is needed, it selects one tool from `img_ann`, `text_ann`, `bm25_ann`, and `multimodal_ann`, and executes the corresponding tool call. The returned observation is appended to the evidence pool before the next planning round. We allow up to six rounds, with at most one tool call per round. The final answer is generated from the image, question, options, and accumulated evidence. Detailed algorithms for the fixed RAG workflow and PAR agent are provided in Appendix C.

3.3 MMSearch-R1 Reproduction and Game-domain Adaptation

We evaluate SVFSearch with MMSearch-R1 (Wu et al., 2025) (MS-R1) under three settings. First, we directly test the released MS-R1 model based on Qwen2.5-VL-7B. Second, we train **MS-R1-8B**, a Qwen3-VL-8B model using the MS-R1 codebase, our training data, and the original MS-R1 prompt and reward design. Third, we train **MS-R1-8B-Game**, which uses the same backbone and training framework but adapts the prompt and reward to the game-domain multiple-choice setting. Following the original MS-R1 protocol, all MS-R1-based models use only `text_ann` and `img_ann`.

Unlike PAR, MS-R1 internalizes tool use into generation: the model alternates between tool-call tokens and final-answer tokens, with retrieval observations appended to the context. For MS-R1-8B-Game, we adapt the reward to encourage explicit search behavior and reduce a multiple-choice shortcut where the model skips retrieval and guesses an option. Specifically, we penalize incorrect answers without search and add small rewards for valid image search and image–text search trajectories. Both MS-R1-8B variants are trained with GRPO following the MMSearch-R1 training framework. Detailed prompts are provided in Appendix C.

4 Experiments and Analysis

4.1 Experimental Setup

We evaluate all methods on the 5,000-example test split of SVFSearch, using accuracy as the main metric. Unless otherwise specified, each method receives only the paused-frame image, question, and answer options. The released video title, cover-frame OCR text, and ASR transcript are excluded from the main experiments.

We compare five evaluation settings. Direct QA evaluates MLLMs without retrieval tools. RAG Workflow follows the fixed image-to-text retrieval workflow described in Section 3.2. PAR evaluates the LangGraph-based Plan-Act-Replan agent from the same section. MS-R1 evaluates MMSearch-R1-style models that generate tool calls as part of the decoding process. Oracle Knowledge provides the ground-truth knowledge associated with each question and serves as an upper bound on evidence availability. All retrieval-augmented settings use the same frozen retrieval resources to ensure fair comparison. RAG Workflow and PAR use top-5 retrieval, while MS-R1 uses top-3 image retrieval and top-5 text retrieval. We also report search rate, denoted as SR, which measures the fraction of examples where a method invokes at least one retrieval tool. Direct QA and Oracle Knowledge do not invoke SVFSearch retrieval tools, so their SR is marked as "—". Qwen2.5-VL-7B-CoT uses prompt-induced step-by-step reasoning, while Thinking models use their native thinking mode. Further implementation details are provided in Appendix C.

4.2 Main Results

Table 2 reports results on the 5,000-example test split. Full experimental results and additional analyses are provided in Appendix B. Overall, SVFSearch is challenging when methods rely only on direct

Table 2: **Comprehensive results on SVFSearch.** Accuracy and search rate (SR) are reported on the 5,000-example test split. "—" indicates settings without SVFSearch retrieval tools. † denotes native thinking mode; Boldface indicates the best accuracy within each setting block.

		Overall	Search	Category						Difficulty		
Setting	**Model**	**Acc.**	**SR**	**Char.**	**Equip.**	**Map**	**Story**	**Mech.**	**Other**	**Easy**	**Med.**	**Hard**
Proprietary Models (Direct QA)												
Direct	Claude-Opus-4.7	69.0	—	70.0	69.0	66.8	72.7	66.2	66.7	67.9	68.6	72.9
Direct	GPT-5.4	67.9	—	69.8	67.8	59.7	69.7	63.7	66.7	65.1	67.7	70.4
Direct	**Gemini-3.1-Pro**	**77.5**	—	79.1	79.6	69.9	87.9	72.3	71.4	82.5	78.3	68.2
Open-source Direct QA												
Direct	Qwen2.5-VL-7B	49.8	—	50.8	47.9	47.4	51.5	48.1	57.1	57.1	49.4	50.2
Direct	Qwen2.5-VL-7B-CoT	47.4	—	47.8	47.6	46.4	60.6	45.8	42.9	64.6	47.5	39.3
Direct	Qwen3-VL-8B	54.8	—	56.5	52.4	55.1	66.7	50.9	47.6	58.0	53.8	61.9
Direct	Qwen3-VL-8B-Thinking	53.7	—	54.1	54.0	52.0	60.6	52.0	61.9	74.1	52.8	52.8
Direct	Qwen3-VL-32B	57.1	—	58.7	54.3	53.6	63.6	54.4	71.4	71.2	56.2	58.9
Direct	Qwen3-VL-32B-Thinking	62.2	—	63.4	60.9	58.2	69.7	59.8	76.2	75.5	61.9	59.9
Direct	Qwen3.5-9B	59.9	—	61.1	57.1	57.1	60.6	58.6	66.7	59.9	59.2	65.4
Direct	Qwen3.5-9B†	58.3	—	59.3	55.7	61.2	66.7	56.6	57.1	63.7	57.6	62.3
Direct	**Qwen3.5-27B**	**66.4**	—	68.0	63.1	63.8	75.8	63.9	76.2	73.6	65.5	71.3
Direct	Qwen3.5-27B†	64.8	—	65.4	62.5	61.2	72.7	64.8	66.7	69.8	64.8	62.5
Open-source RAG Workflow												
Workflow	Qwen2.5-VL-7B	57.3	100.0	57.7	58.6	58.2	69.7	54.3	52.4	59.4	57.9	51.0
Workflow	Qwen3-VL-8B	63.5	100.0	64.1	66.5	63.3	60.6	59.6	61.9	66.5	64.3	55.3
Workflow	Qwen3-VL-32B	62.2	100.0	61.6	67.2	62.8	69.7	60.3	61.9	78.3	62.7	51.4
Workflow	Qwen3.5-9B	66.5	100.0	67.5	67.2	64.3	63.6	63.3	61.9	70.8	66.5	64.4
Workflow	**Qwen3.5-27B**	**69.4**	100.0	70.8	69.4	65.8	78.8	65.3	66.7	70.8	69.1	71.0
Open-source Plan-Act-Replan Agent												
PAR	Qwen2.5-VL-7B	59.3	85.6	59.3	61.7	63.3	54.5	56.9	52.4	55.2	59.8	56.5
PAR	Qwen3-VL-8B	63.7	99.7	62.8	66.1	70.9	63.6	63.2	66.7	66.5	64.2	58.3
PAR	Qwen3-VL-32B	71.6	98.4	71.1	75.9	74.5	69.7	69.4	76.2	79.7	72.4	61.5
PAR	**Qwen3.5-9B**	**79.1**	100.0	79.3	82.7	79.1	87.9	76.0	71.4	75.0	79.7	75.9
PAR	Qwen3.5-27B	78.6	96.8	78.3	83.1	81.1	81.8	75.5	81.0	79.7	79.0	74.5
Open-source MMSearch-R1 Search Models												
MS-R1	Qwen2.5-VL-7B	49.4	72.8	49.1	53.8	54.1	60.6	46.2	33.3	68.4	49.4	41.5
MS-R1	Qwen3-VL-8B	63.2	0.02	64.5	60.8	63.3	63.6	61.1	52.4	70.3	62.6	65.6
MS-R1-Game	**Qwen3-VL-8B**	**64.5**	68.2	65.2	66.8	65.3	63.6	60.0	76.2	73.6	64.3	61.9
Open-source Oracle Knowledge (Upper Bound)												
Oracle	Qwen3-VL-8B	86.5	—	88.4	87.5	87.8	78.8	80.3	76.2	73.1	87.5	84.0
Oracle	Qwen3-VL-32B	90.8	—	92.0	94.1	88.3	93.9	85.4	81.0	87.7	91.4	87.0
Oracle	Qwen3.5-9B	90.3	—	90.7	91.9	86.2	97.0	88.4	85.7	91.5	90.7	85.8
Oracle	**Qwen3.5-27B**	**95.4**	—	96.0	96.7	95.9	87.9	93.0	90.5	94.3	95.7	93.5

MLLM answering. Among proprietary models, the strongest Direct QA result is achieved by Gemini-3.1-Pro at 77.5%, while the best open-source Direct QA setting reaches 66.4% with Qwen3.5-27B. In contrast, Oracle Knowledge substantially improves performance, reaching 95.4% with Qwen3.5-27B and 90.8% with Qwen3-VL-32B. This gap shows that many questions are answerable when the relevant game-specific evidence is available, but remain difficult without retrieved evidence.

Retrieval improves open-source methods across most settings. With Qwen3-VL-8B as the backbone, accuracy increases from 54.8% under Direct QA to 63.5% with RAG Workflow, 63.7% with PAR, and 64.5% with MS-R1-Game. For stronger backbones, PAR gives the best practical results: Qwen3.5-9B with PAR reaches 79.1%, surpassing all Direct QA baselines, including Gemini-3.1-Pro. The high SR values under PAR, ranging from 85.6% to 100.0%, show that these agents frequently rely on retrieval, and the corresponding gains indicate that external evidence is often beneficial. Model scaling and version upgrades are most visible on easy examples: under Direct QA, Qwen3-VL improves from 58.0% at 8B to 71.2% at 32B, and Qwen3.5-27B reaches 73.6%. Hard examples still remain far below Oracle Knowledge, suggesting persistent bottlenecks in visual grounding, retrieval quality, evidence selection, and evidence-grounded reasoning.

The MS-R1 results highlight both the promise and the difficulty of learned tool use. MS-R1-Game improves Qwen3-VL-8B to 64.5% with an SR of 68.2%, showing that task-adapted training can recover meaningful retrieval behavior while improving answer accuracy. By contrast, the original MS-R1 training improves Qwen3-VL-8B to 63.2% but yields an SR of only 0.02%, indicating that its

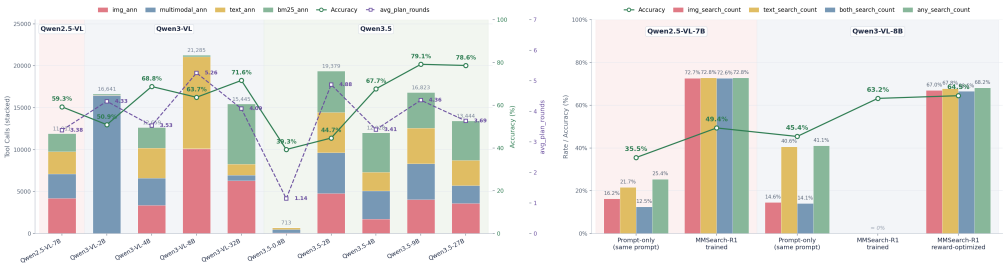


Figure 4: **Tool-use diagnostics.** Left: PAR tool calls, accuracy, and average planning rounds across backbones. Right: item-level search rates and accuracy of MS-R1-style models. Search-rate bars on the right are not mutually exclusive.

gain mainly comes from improved reasoning behavior rather than actual retrieval use. This suggests a limitation of outcome-only RL in multiple-choice settings: a model can improve answer accuracy while learning to bypass search, because guessing can still receive a non-trivial reward. Effective MS-R1 training on SVFSearch therefore requires task-specific reward design that encourages retrieval when evidence is needed while preserving answer correctness.

4.3 Tool-use Diagnostics

PAR tool-use behavior. The left panel of Figure 4 shows that PAR behavior varies substantially across model scales. Qwen3.5-0.8B makes only 713 tool calls, with an average of 1.14 planning rounds, and achieves 39.3%, suggesting that very small models struggle to sustain multi-step search behavior. Qwen3.5-2B shows the opposite pattern: it makes 19,379 tool calls over 4.88 rounds but reaches only 44.7%, indicating excessive search without effective use of evidence. In contrast, Qwen3.5-27B makes fewer calls, 13,444 with 3.69 rounds, yet reaches 78.6%. These results show that PAR performance is not determined by the number of tool calls alone, but also depends on evidence selection and integration.

Tool preference also varies across model families. Qwen3-VL models show uneven tool-use patterns across scales, with different sizes favoring different retrieval channels, while Qwen3.5 models above 4B use multiple channels more consistently. The best PAR setting, Qwen3.5-9B, reaches 79.1% with relatively balanced use of image, text, BM25, and multimodal retrieval. This suggests that effective PAR benefits from coordinating complementary retrieval channels rather than simply increasing the number of calls.

MS-R1 tool-use behavior. The right panel of Figure 4 compares prompt-only and RL-trained MS-R1 variants. Prompt-only tool use is weak: under the same search prompt, Qwen2.5-VL-7B achieves only 35.5% accuracy with a 25.4% any-search rate, and Qwen3-VL-8B achieves 45.4% accuracy with a 41.1% any-search rate. Both are below their Direct QA counterparts in Table 2, showing that long tool-use prompts alone can introduce instability and hurt performance.

RL training changes tool-use behavior, but the effect depends strongly on the task and reward design. The released Qwen2.5-VL-7B MMSearch-R1 model searches on 72.8% of examples, yet remains slightly below its Direct QA baseline, showing that a trained search policy does not transfer automatically to the game-domain QA setting. For Qwen3-VL-8B, the original MS-R1 training improves accuracy to 63.2% but almost eliminates tool use, indicating that the model mainly learns stronger reasoning behavior rather than reliable search. Our reward-adapted MS-R1-Game restores meaningful retrieval, reaching 64.5% accuracy with a 68.2% any-search rate and frequent use of both image and text search. These results support the value of agentic-search RL, while showing that task-specific data and reward design are needed to avoid answer-only shortcuts in multiple-choice settings.

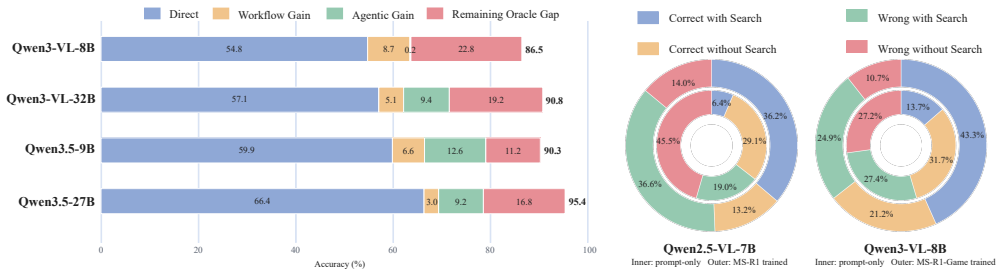


Figure 5: **Retrieval gains and search behavior.** Left: accuracy decomposition from Direct QA to RAG Workflow, PAR, and Oracle Knowledge. Right: correctness and search-usage breakdown for prompt-only and trained MS-R1-style models.

4.4 Retrieval Gains and Search Behavior

Figure 5 summarizes how retrieval changes model performance and search behavior. The left panel decomposes the accuracy from Direct QA to Oracle Knowledge. Across Qwen3-VL and Qwen3.5 backbones, RAG Workflow consistently improves over Direct QA, and PAR can add further gains by allowing adaptive tool use beyond a fixed retrieval chain. For example, PAR adds 9.4 points over RAG Workflow on Qwen3-VL-32B and 12.6 points on Qwen3.5-9B. However, the remaining gap to Oracle Knowledge is still large, including 19.2 points for Qwen3-VL-32B and 16.8 points for Qwen3.5-27B, indicating persistent bottlenecks in visual grounding, query formulation, retrieval quality, and evidence-grounded reasoning.

The right panel compares prompt-only and trained MS-R1 models by correctness and search usage. After training, correct-with-search examples increase from 6.4% to 36.2% for Qwen2.5-VL-7B and from 13.7% to 43.3% for Qwen3-VL-8B, while wrong-without-search examples decrease from 45.5% to 14.0% and from 27.2% to 10.7%, respectively. This shows that training makes search more useful in evidence-demanding cases. At the same time, correct-without-search examples decrease and wrong-with-search examples remain non-negligible, suggesting that learned search policies can still over-search or retrieve unhelpful evidence. These results support learned tool use on SVFSearch, but also show the need for reward and data design that balances search necessity, search quality, and answer correctness.

5 Limitations

SVFSearch focuses on short-video frame search in the Chinese gaming domain, which provides a controlled and realistic setting for evaluating domain-specific multimodal search. This focus also means that the results should be interpreted within this vertical domain rather than as a universal estimate of all short-video applications. Our main evaluation uses four-choice QA and a frozen offline retrieval environment with a February 2026 knowledge cutoff for reproducible comparisons. Although we release video-side metadata such as titles, cover-frame OCR text, and ASR transcripts, their full use is left to future multi-source evaluation. Future work can extend the benchmark to open-ended answering, additional vertical domains, and evolving retrieval corpora.

6 Conclusion

We introduced SVFSearch, a benchmark for short-video frame search in the Chinese gaming domain, together with a frozen offline retrieval environment spanning text, image, and multimodal evidence. The benchmark supports controlled evaluation of direct-QA MLLMs, fixed RAG workflows, planner-based agents, and MS-R1 learned search models under the same retrieval resources. Experiments show that game-specific evidence is critical for many examples, while practical agents still face substantial gaps in visual grounding, tool-use control, retrieval quality, and evidence integration. We hope SVFSearch can serve as a practical testbed for studying retrieval-augmented and agentic multimodal search over paused frames from short videos.

AI USE STATEMENT

Generative AI is used in the benchmark construction pipeline described in Section 2.2: LLMs assist with cleaning, summarizing, and chunking retrieved knowledge; an MLLM checks image–element alignment; an 8B-parameter model generates candidate multiple-choice questions; and a 32B-parameter model assesses their quality, correctness, distractors, and difficulty. The pipeline includes manual spot checks. LLMs and MLLMs also serve as the evaluated models and search-agent backbones described in Section 3 and Appendix C.

REPRODUCIBILITY STATEMENT

Sections 2.2 and 2.3 describe the benchmark construction and frozen retrieval resources. Section 4.1 specifies the evaluation inputs, metric, and retrieval depths. Appendix C provides the retrieval interfaces, algorithms, and prompts, and Appendix B reports additional experimental results. The project page is available at `https://svfsearch.github.io/SVFSearch-page`, with code at `https://github.com/SVFSearch/SVFSearch-code` and the dataset at `https://huggingface.co/datasets/svfsearch/SVFSearchData`.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 2022.

Anthropic. System Card: Claude Opus 4 & Claude Sonnet 4. `https://www.anthropic.com/claude-4-system-card`, 2025.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2015.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

M Chen, L Sun, T Li, H Sun, Y Zhou, C Zhu, H Wang, JZ Pan, W Zhang, H Chen, et al. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025.

Pengcheng Chen, Jin Ye, Guoan Wang, Yanjun Li, Zhongying Deng, Wei Li, Tianbin Li, Haodong Duan, Ziyan Huang, Yanzhou Su, et al. Gmai-mmbench: A comprehensive multimodal evaluation benchmark towards general medical ai. *Advances in Neural Information Processing Systems*, 2024.

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Google AI for Developers. Gemini 3.1 Pro Preview. `https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview`, 2026.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

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

Wenbo Hu, Jia-Chen Gu, Zi-Yi Dou, Mohsen Fayyaz, Pan Lu, Kai-Wei Chang, and Nanyun Peng. Mrag-bench: Vision-centric evaluation for retrieval-augmented multimodal models. *arXiv preprint arXiv:2410.08182*, 2024.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

LangChain Inc. LangGraph Overview. `https://docs.langchain.com/oss/python/langgraph/overview`, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 2023.

Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Philip S Yu, Fei Huang, et al. Benchmarking multimodal retrieval augmented generation with dynamic vqa dataset and self-adaptive planning agent. *arXiv preprint arXiv:2411.02937*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024a.

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

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 2022.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 2019.

Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3113–3124, 2023.

Kartik Narayan, Yang Xu, Tian Cao, Kavya Nerella, Vishal M Patel, Navid Shiee, Peter Grasch, Chao Jia, Yinfei Yang, and Zhe Gan. Deepmmsearch-r1: Empowering multimodal llms in multimodal web search. *arXiv preprint arXiv:2510.12801*, 2025.

OpenAI. GPT-5 System Card. `https://openai.com/index/gpt-5-system-card/`, 2025.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 2023.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

Xijia Tao, Yihua Teng, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. Mmsearch-plus: Benchmarking provenance-aware search for multimodal browsing agents. *arXiv preprint arXiv:2508.21475*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 2024b.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search. *arXiv preprint arXiv:2506.20670*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Qinhan Yu, Zhiyou Xiao, Binghui Li, Zhengren Wang, Chong Chen, and Wentao Zhang. Mramg-bench: a comprehensive benchmark for advancing multimodal retrieval-augmented multimodal generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

A Related Work

Multimodal Benchmarks: from VQA to Multimodal Search. Visual question answering has evolved from open-ended image QA (Antol et al., 2015; Goyal et al., 2017) to more specialized settings such as scene-text understanding, science reasoning, and knowledge-intensive QA (Singh et al., 2019; Lu et al., 2022; Marino et al., 2019). In parallel, broad multimodal benchmarks including MMBench, MME, MM-Vet, and MMMU-Pro (Liu et al., 2024b; Fu et al., 2023; Yu et al., 2023; Yue et al., 2024; 2025) have become common testbeds for evaluating frontier MLLMs. More recently, several benchmarks have moved toward *multimodal retrieval and search*, covering multi-hop web QA (Chang et al., 2022), vision-centric mRAG (Hu et al., 2024), adaptive retrieval planning (Li et al., 2024), and multimodal search engines (Jiang et al., 2024; Tao et al., 2025; Yu et al., 2025). Despite this progress, existing multimodal search benchmarks mainly focus on general-domain image-question inputs or web-search settings. These settings do not fully match short-video frame search, where a user question is grounded in a paused video frame and may be accompanied by noisy video-side context, such as the cover image, title, OCR text, and ASR transcript. The relevant evidence may also be scattered across domain-specific and rapidly changing sources. This gap is especially pronounced in the gaming vertical, where characters, items, mechanics, events, and patches evolve quickly and are rarely covered by stable, structured knowledge bases. SVFSearch targets this setting by evaluating game-scene understanding and retrieval-augmented QA over paused-frame images, with frozen text, image, and multimodal retrieval resources.

Agentic Multimodal Search. Agentic search methods provide a natural evaluation setting for multimodal search benchmarks, since models must decide when and how to use external tools before answering. One line of work follows the ReAct-style paradigm (Yao et al., 2022), decomposing a query into planning, tool invocation, evidence aggregation, and answer generation. This paradigm is well matched to SVFSearch, where questions may require visual grounding, text retrieval, multimodal retrieval, and domain knowledge lookup. We therefore implement a LangGraph-based Plan-Act-Replan baseline with domain-specific retrieval tools. A second line trains search agents with outcome-based RL. Built on R1-style reasoning and GRPO (Guo et al., 2025; Shao et al., 2024), Search-R1 and its variants (Jin et al., 2025; Song et al., 2025; Chen et al., 2025) study text-based search agents, while MMSearch-R1 (Wu et al., 2025) extends this paradigm to on-demand multimodal web search. DeepMMSearch-R1 (Narayan et al., 2025) further relaxes single-call and whole-image constraints. We adapt this line to the game-domain multiple-choice setting by training an MMSearch-R1-style model and using it as a learned search baseline. Together, these baselines provide complementary references for evaluating retrieval-augmented and agentic multimodal search on SVFSearch.

B Additional Experimental Results

B.1 Full Results on SVFSearch

Table 3 extends Table 2 with additional model scales. The same trends remain: Direct QA is limited, retrieval-augmented settings improve performance when the backbone can use external evidence, and Oracle Knowledge provides a much higher upper bound. The results also show that SVFSearch is challenging across model families. Larger and newer models generally perform better, but sizable Oracle gaps remain, indicating persistent failures in visual grounding, retrieval quality, evidence integration, and evidence-grounded reasoning.

The extended results also reveal robustness issues that are less visible in the main table. Very small models are unstable in tool-augmented settings, consistent with the diagnostics in Section 4.3: agentic search requires sufficient visual understanding, instruction following, and evidence integration. Native thinking mode is not uniformly helpful either. It improves some larger models but can degrade smaller ones, likely due to formatting errors or context drift. These observations complement the main analysis by showing how model scale and reasoning mode affect robustness on SVFSearch.

B.2 Planning-round Analysis

Figure 6 groups PAR examples by the planning round at which the agent stops. Accuracy usually improves after a few rounds but drops on late or capped cases. This suggests that moderate search

Table 3: **Comprehensive results on SVFSearch.** Results are reported on the 5,000-example test split. **SR** denotes the fraction of examples where the system invokes at least one retrieval tool; "—" indicates settings without SVFSearch retrieval tools. † denotes native thinking mode. **Bold** marks the best overall accuracy within each setting block.

		Overall	Search	Category						Difficulty		
Setting	**Model**	**Acc.**	**SR**	**Char.**	**Equip.**	**Map**	**Story**	**Mech.**	**Other**	**Easy**	**Med.**	**Hard**
Proprietary Models (Direct QA)												
Direct	Claude-Opus-4.7	69.0	—	70.0	69.0	66.8	72.7	66.2	66.7	67.9	68.6	72.9
Direct	GPT-5.4	67.9	—	69.8	67.8	59.7	69.7	63.7	66.7	65.1	67.7	70.4
Direct	**Gemini-3.1-Pro**	**77.5**	—	79.1	79.6	69.9	87.9	72.3	71.4	82.5	78.3	68.2
Open-source Direct QA												
Direct	Qwen2.5-VL-7B	49.8	—	50.8	47.9	47.4	51.5	48.1	57.1	57.1	49.4	50.2
Direct	Qwen2.5-VL-7B-CoT	47.4	—	47.8	47.6	46.4	60.6	45.8	42.9	64.6	47.5	39.3
Direct	Qwen3-VL-2B	33.1	—	34.2	32.5	27.6	27.3	31.3	38.1	34.9	31.6	44.9
Direct	Qwen3-VL-2B-Thinking	47.8	—	49.2	45.1	49.5	48.5	45.7	33.3	62.3	46.9	49.8
Direct	Qwen3-VL-4B	52.9	—	53.7	52.3	50.0	69.7	51.2	57.1	67.9	52.2	53.2
Direct	Qwen3-VL-4B-Thinking	54.2	—	54.5	51.9	58.7	51.5	54.8	38.1	71.7	53.4	53.8
Direct	Qwen3-VL-8B	54.8	—	56.5	52.4	55.1	66.7	50.9	47.6	58.0	53.8	61.9
Direct	Qwen3-VL-8B-Thinking	53.7	—	54.1	54.0	52.0	60.6	52.0	61.9	74.1	52.8	52.8
Direct	Qwen3-VL-32B	57.1	—	58.7	54.3	53.6	63.6	54.4	71.4	71.2	56.2	58.9
Direct	Qwen3-VL-32B-Thinking	62.2	—	63.4	60.9	58.2	69.7	59.8	76.2	75.5	61.9	59.9
Direct	Qwen3.5-0.8B	34.1	—	34.4	32.2	32.1	30.3	34.8	42.9	45.3	33.7	32.4
Direct	Qwen3.5-0.8B†	42.8	—	45.4	37.2	41.8	27.3	39.4	47.6	52.4	41.5	49.8
Direct	Qwen3.5-2B	56.5	—	55.6	57.9	58.7	51.5	57.7	71.4	63.2	56.1	56.7
Direct	Qwen3.5-2B†	49.3	—	50.2	45.7	51.0	30.3	49.1	61.9	63.2	47.9	54.9
Direct	Qwen3.5-4B	57.3	—	57.8	57.8	53.1	66.7	56.0	61.9	69.8	55.9	63.6
Direct	Qwen3.5-4B†	59.1	—	60.7	55.1	54.6	51.5	58.6	42.8	66.5	58.2	63.6
Direct	Qwen3.5-9B	59.9	—	61.1	57.1	57.1	60.6	58.6	66.7	59.9	59.2	65.4
Direct	Qwen3.5-9B†	58.3	—	59.3	55.7	61.2	66.7	56.6	57.1	63.7	57.6	62.3
Direct	**Qwen3.5-27B**	**66.4**	—	68.0	63.1	63.8	75.8	63.9	76.2	73.6	65.5	71.3
Direct	Qwen3.5-27B†	64.8	—	65.4	62.5	61.2	72.7	64.8	66.7	69.8	64.8	62.5
Open-source RAG Workflow												
Workflow	Qwen2.5-VL-7B	57.3	100.0	57.7	58.6	58.2	69.7	54.3	52.4	59.4	57.9	51.0
Workflow	Qwen3-VL-2B	47.1	100.0	47.1	48.3	49.5	45.5	46.1	28.6	48.6	47.2	45.1
Workflow	Qwen3-VL-4B	57.6	100.0	57.2	60.4	58.2	60.6	56.5	61.9	70.8	58.2	47.2
Workflow	Qwen3-VL-8B	63.5	100.0	64.1	66.5	63.3	60.6	59.6	61.9	66.5	64.3	55.3
Workflow	Qwen3-VL-32B	62.2	100.0	61.6	67.2	62.8	69.7	60.3	61.9	78.3	62.7	51.4
Workflow	Qwen3.5-0.8B	33.0	100.0	33.9	30.3	35.7	30.3	31.6	33.3	40.1	33.0	29.4
Workflow	Qwen3.5-2B	44.0	100.0	44.3	46.6	38.3	45.5	42.5	23.8	56.1	44.8	32.4
Workflow	Qwen3.5-4B	56.5	100.0	57.9	53.5	53.6	63.6	55.0	47.6	61.8	56.8	52.2
Workflow	Qwen3.5-9B	66.5	100.0	67.5	67.2	64.3	63.6	63.3	61.9	70.8	66.5	64.4
Workflow	**Qwen3.5-27B**	**69.4**	100.0	70.8	69.4	65.8	78.8	65.3	66.7	70.8	69.1	71.0
Open-source Plan-Act-Replan Agent												
PAR	Qwen2.5-VL-7B	59.3	85.6	59.3	61.7	63.3	54.5	56.9	52.4	55.2	59.8	56.5
PAR	Qwen3-VL-2B	50.9	100.0	50.3	53.2	53.1	54.5	50.6	42.9	52.4	51.4	45.9
PAR	Qwen3-VL-8B	63.7	99.7	62.8	66.1	70.9	63.6	63.2	66.7	66.5	64.2	58.3
PAR	Qwen3-VL-32B	71.6	98.4	71.1	75.9	74.5	69.7	69.4	76.2	79.7	72.4	61.5
PAR	Qwen3.5-2B	44.7	99.3	44.8	48.1	40.8	48.5	42.6	28.6	55.7	45.5	33.4
PAR	Qwen3.5-4B	67.7	98.1	67.5	69.7	68.4	69.7	66.4	71.4	65.6	68.1	65.4
PAR	**Qwen3.5-9B**	**79.1**	100.0	79.3	82.7	79.1	87.9	76.0	71.4	75.0	79.7	75.9
PAR	Qwen3.5-27B	78.6	96.8	78.3	83.1	81.1	81.8	75.5	81.0	79.7	79.0	74.5
Open-source MMSearch-R1 Search Models												
MS-R1	Qwen2.5-VL-7B	49.4	72.8	49.1	53.8	54.1	60.6	46.2	33.3	68.4	49.4	41.5
MS-R1	Qwen3-VL-8B	63.2	0.02	64.5	60.8	63.3	63.6	61.1	52.4	70.3	62.6	65.6
MS-R1-Game	**Qwen3-VL-8B**	**64.5**	68.2	65.2	66.8	65.3	63.6	60.0	76.2	73.6	64.3	61.9
Open-source Oracle Knowledge (Upper Bound)												
Oracle	Qwen2.5-VL-7B	76.0	—	78.0	78.5	81.6	84.9	67.6	47.6	63.7	77.4	69.4
Oracle	Qwen3-VL-2B	52.3	—	50.3	61.9	61.7	57.6	49.8	38.1	44.3	52.3	55.9
Oracle	Qwen3-VL-4B	84.6	—	86.7	87.7	82.7	90.9	76.6	57.1	77.4	85.3	81.6
Oracle	Qwen3-VL-8B	86.5	—	88.4	87.5	87.8	78.8	80.3	76.2	73.1	87.5	84.0
Oracle	Qwen3-VL-32B	90.8	—	92.0	94.1	88.3	93.9	85.4	81.0	87.7	91.4	87.0
Oracle	Qwen3.5-0.8B	25.8	—	25.4	30.2	22.5	18.2	24.7	33.3	32.1	26.4	18.2
Oracle	Qwen3.5-2B	68.5	—	67.5	75.3	66.8	66.7	66.6	81.0	76.9	69.2	59.5
Oracle	Qwen3.5-4B	87.9	—	88.4	90.7	86.7	81.8	84.9	76.2	86.3	88.4	84.4
Oracle	Qwen3.5-9B	90.3	—	90.7	91.9	86.2	97.0	88.4	85.7	91.5	90.7	85.8
Oracle	**Qwen3.5-27B**	**95.4**	—	96.0	96.7	95.9	87.9	93.0	90.5	94.3	95.7	93.5

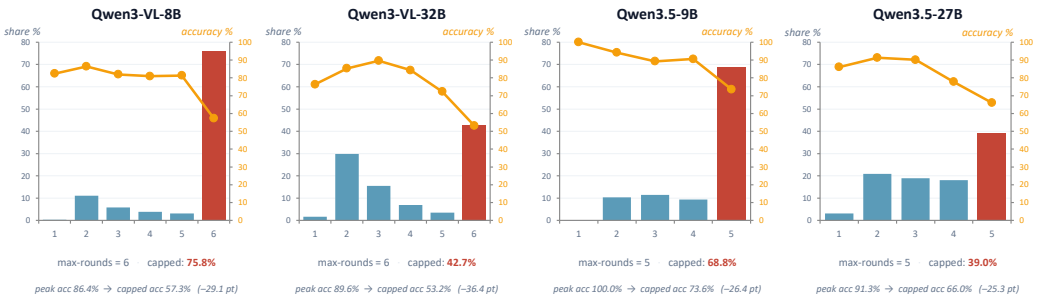


Figure 6: **Planning-round analysis under PAR.** Bars show the share of examples ending at each planning round, and lines show the corresponding accuracy. Red bars indicate examples capped by the maximum number of planning rounds.

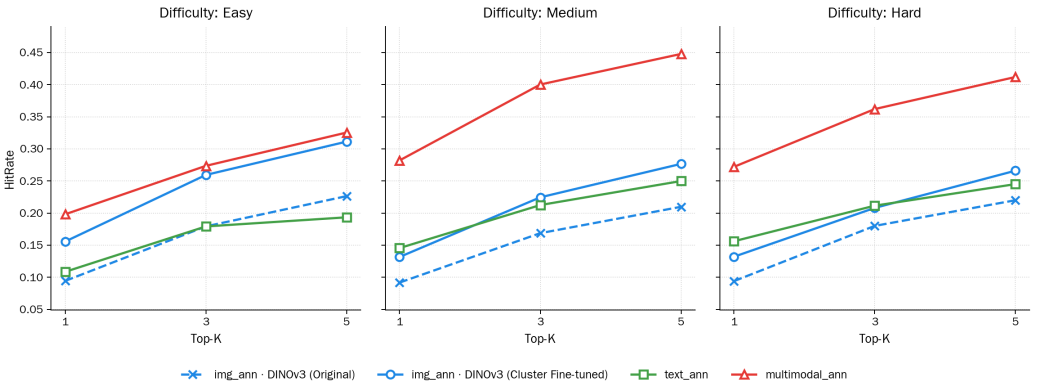


Figure 7: **HitRate@K of retrieval backends across difficulty levels.** We compare original DINOv3 image retrieval, cluster-aware fine-tuned DINOv3 image retrieval, text retrieval, and multimodal retrieval for $K \in \{1, 3, 5\}$.

can help evidence acquisition, while excessive planning often reflects unresolved uncertainty rather than better evidence use. Capped examples are typically harder cases where the agent fails in visual grounding, query formulation, or retrieval. These results suggest that PAR benefits from a limited number of search rounds, but simply extending the search process does not guarantee better answers.

B.3 Retrieval Backend Analysis

Figure 7 evaluates the frozen retrieval backends used by SVFSearch. The cluster-aware DINOv3 fine-tuning described in Section 3.1 consistently improves image retrieval over the original DINOv3 encoder across difficulty levels and top-K values. This shows that core-element-aware training helps align paused game frames with their target elements. The gain is moderate, since the same core element may appear with large visual variation in viewpoint, skin, UI state, background, and combat context, while different elements from the same game can also look similar.

Multimodal retrieval achieves the highest HitRate@K in most settings, especially on medium and hard examples. This indicates that combining visual and textual signals is more effective than relying on a single retrieval modality. The comparison should also be interpreted with model capacity in mind: the multimodal index uses Qwen3-VL-Embedding-2B, while the text index uses Qwen3-Embedding-0.6B and the image index uses the fine-tuned DINOv3-Base encoder. Overall, the results show that image, text, and multimodal retrieval provide complementary signals for SVFSearch.

B.4 Video-side Metadata Analysis

Among the 5,000 test examples, video titles, cover-frame OCR, and ASR transcripts are available for 90.24%, 44.84%, and 67.58%, respectively; 425 examples contain none. At least one metadata source mentions the queried entity in 45.16% of examples, whereas only 71 examples (1.42%) directly contain the correct answer. Thus, metadata can expose the retrieval target without revealing the answer.

Table 4 compares Direct QA and RAG with and without metadata, alongside standard PAR. Metadata improves Direct QA by 1.1–8.8 percentage points and RAG by 0.7–6.9 points. For Qwen3.5-9B, the gap from PAR narrows from 19.2 to 11.1 points for Direct QA and from 12.6 to 6.8 points for RAG.

Table 4: **Effect of video-side metadata.** Accuracy (%) on the 5,000-example test set. +Meta adds available video titles, cover-frame OCR, and ASR transcripts. PAR uses the standard frame-and-question setting without added metadata.

Model	Direct	Direct+Meta	RAG	RAG+Meta	PAR
Qwen2.5-VL-7B	49.8	50.9	57.3	58.0	59.3
Qwen3-VL-8B	54.8	62.4	63.5	66.7	63.7
Qwen3-VL-32B	57.1	65.9	62.2	69.1	71.6
Qwen3.5-9B	59.9	68.0	66.5	72.3	79.1

Videos were collected using known game entities, making their metadata closely associated with the target and simplifying entity identification. The main frame-centric protocol isolates visual grounding and subsequent knowledge retrieval; the released metadata supports separate evaluation with these additional signals.

B.5 Error Analysis under Oracle Knowledge

Oracle Knowledge supplies gold supporting knowledge but does not guarantee correct grounding, reasoning, or formatting. We analyzed the 230 errors made by Qwen3.5-27B, which achieves 95.4% accuracy on the 5,000-example test set. Table 5 reports the categories, with percentages calculated over these 230 errors.

Table 5: **Oracle Knowledge error breakdown for Qwen3.5-27B.** Counts and shares of the 230 incorrect predictions under the original strict matching evaluation.

Failure type	Count	Share (%)
Condition-specific reasoning errors	69	30.0
Evidence selection/integration errors	51	22.2
Answer-format or strict-matching errors	50	21.7
Visual, UI, or spatial grounding errors	37	16.1
Annotation/evidence-boundary cases	14	6.1
Final option-mapping inconsistencies	5	2.2
Numerical or detail-level mistakes	4	1.7
Total	230	100.0

Condition-specific errors involve missing a required trigger, location, state, or visual condition despite identifying the general mechanism. Evidence selection/integration errors involve using the wrong part of a passage containing multiple related entities or rules. Visual grounding still requires connecting the frame to the relevant entity or state.

The 50 format-related errors contain the correct option letter or a semantically correct choice but fail the required full-answer-text format under strict exact matching. They remain incorrect in the reported results and should be distinguished from reasoning failures. The remaining errors show that gold knowledge still requires visual grounding, evidence integration, and reliable answer mapping.

C Implementation Details of Tool-Augmented Search Settings

We provide implementation details for the three tool-augmented settings used in our evaluation: the fixed RAG workflow, the LangGraph-based PAR agent, and MS-R1-style search models. These settings share the same frozen offline retrieval environment, but differ in how retrieval actions are selected, tool observations are accumulated, and final answers are produced.

Because SVFSearch targets the Chinese gaming vertical, the actual implementation uses Chinese prompts and tool cards. The surrounding explanations are written in English, while the original Chinese prompts and skill cards are preserved verbatim in shaded implementation boxes. Machine-readable tags, tool names, endpoints, and JSON fields are kept unchanged throughout.

C.1 Shared Offline Retrieval Tools

All tool-augmented settings access the same frozen retrieval resources. The retrieval tools operate offline and do not access live web search. This design makes the evaluation deterministic and allows us to compare search behavior under identical retrieval conditions.

C.2 Fixed Retrieval-Augmented Generation Workflow

System description. The fixed RAG workflow follows a deterministic image-to-text retrieval chain. It first performs image retrieval to infer the most likely core element in the paused-frame image. The selected element is then concatenated with the original question to form a text retrieval query. Finally, the retrieved evidence is inserted into a multimodal answer prompt together with the image, question, and answer options.

Algorithm 1 Fixed RAG Workflow

Require: question q, options O, image I, retrieval depth k
Ensure: predicted answer a^*

Image retrieval.
$\mathcal{R}_{\text{img}} \leftarrow \text{ImgANN}(I, k)$
Core-element selection.
$C \leftarrow \{r.\text{query} \mid r \in \mathcal{R}_{\text{img}},\ r.\text{query} \neq \emptyset\}$
if $C \neq \emptyset$ **then**
  $Q^* \leftarrow \arg\max_{q' \in C}\ \text{count}(q', C)$ ▷ majority vote over retrieved queries
else
  $r^* \leftarrow \arg\max_{r \in \mathcal{R}_{\text{img}}} r.\text{score}$
  $Q^* \leftarrow r^*.\text{query}$ ▷ fallback to the highest-scoring record
end if
Text retrieval.
if $Q^* \neq \emptyset$ **then**
  $\mathcal{R}_{\text{txt}} \leftarrow \text{TextANN}(Q^* \oplus q, k)$
else
  $\mathcal{R}_{\text{txt}} \leftarrow \text{TextANN}(q, k)$
end if ▷ $\oplus$ denotes string concatenation
Answer generation.
$\mathcal{E} \leftarrow \text{ExtractTopK}(\mathcal{R}_{\text{img}}) \cup \text{ExtractTopK}(\mathcal{R}_{\text{txt}})$
$a^* \leftarrow \text{MLLM}_{\text{answer}}(I, q, O, \mathcal{E})$
return a^*

Prompt template. The RAG answer prompt is intentionally concise. It presents the question, candidate options, and retrieved knowledge, and asks the model to output only the full text of the selected option.

RAG answer prompt: original implementation

```
你是一个多模态游戏问答助手。根据图片和参考知识（如有）回答问题，并从给定选项中选择最正确的一项。
问题:
```

```
{question}

选项:
A. {option_1} B. {option_2}
C. {option_3} D. {option_4}

参考知识:
{knowledge}

只输出正确选项的完整文本，不要解释。
```

C.3 Plan-Act-Replan Agent with LangGraph

System description. Plan-Act-Replan runs search as an explicit loop. Each round, the planner is given the image, question, options, the evidence gathered so far, and a record of past tool calls. It judges whether the current evidence is sufficient for answering. If not, it selects one retrieval tool and prepares the corresponding tool input. The environment executes the tool call, adds the returned observation to the evidence pool, and passes control back to the planner.

Algorithm 2 Plan-Act-Replan Agent

Require: question q, options O, image I, maximum rounds R, retrieval depth k
Ensure: predicted answer a^*

$\mathcal{E} \leftarrow \emptyset$ ▷ evidence pool
$\mathcal{U} \leftarrow \emptyset$ ▷ used tools
$\mathcal{H} \leftarrow \emptyset$ ▷ tool-call history
for $r = 1$ **to** R **do**
  $d \leftarrow \text{LLM}_{\text{plan}}(I, q, O, \mathcal{E}, \mathcal{U}, \mathcal{H}, r)$ ▷ d contains `can_answer_now`, `selected_tool`, `bm25_query`, reason, and confidence
  if d.can_answer_now $=$ true **or** d.selected_tool $=$ `none` **then**
    break
  end if
  $\tau \leftarrow d$.selected_tool
  if $\tau =$ `bm25_ann` **then**
    query $\leftarrow d$.bm25_query ▷ planner-generated keyword query
  else if $\tau =$ `img_ann` **then**
    query $\leftarrow I$
  else if $\tau =$ `text_ann` **then**
    query $\leftarrow q$
  else
    query $\leftarrow (I, q)$ ▷ `multimodal_ann`
  end if
  $\mathcal{R} \leftarrow \text{ANN}_\tau(\text{query}, k)$
  $\mathcal{E} \leftarrow \mathcal{E} \cup \text{ExtractTopK}(\mathcal{R})$
  $\mathcal{U} \leftarrow \mathcal{U} \cup \{\tau\}$
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\tau, |\mathcal{R}|, \text{error}(\mathcal{R}))\}$
  if $\tau =$ `img_ann` **and** $\mathcal{R} \neq \emptyset$ **then**
    $Q_{\text{kn}} \leftarrow \text{MajorityVote}(\{r.\text{query} \mid r \in \mathcal{R}\})$
    $\mathcal{K} \leftarrow \text{KnLookup}(Q_{\text{kn}})$
    if $\mathcal{K} \neq \emptyset$ **then**
      $\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{K}$
    end if
  end if
end for
$a^* \leftarrow \text{LLM}_{\text{answer}}(I, q, O, \mathcal{E})$
return a^*

Tool skill cards. The planner is guided by skill cards that define each tool's use cases, request schema, response fields, and decision hints. The original cards are included below.

SKILL.md: ann-bm25-recall

```
---
name: ann-bm25-recall
description: BM25 keyword retrieval tool skill. Use when the question contains specific
↪  names, terms or IDs that should be matched exactly in the knowledge base, and
↪  semantic embedding may miss them.

---

# ann-bm25-recall

## Tool Name

`bm25_ann`

## When to use

- 题目包含 ** 专有名词、角色名、技能名、地图名、道具名 ** 等需要精确匹配的关键词。
- 语义向量召回 (`text_ann` / `multimodal_ann`) 结果不相关，或分数偏低时。
- 你希望用 ** 关键词组合 ** 直接锁定知识片段，而非依赖语义相似度。
- 图像已能确认某个具体名称，但仍需其文字描述或规则说明。

## 特别说明：query 由大模型生成

与其他 ANN 工具直接使用原始问题作为查询不同，`bm25_ann` 的查询由 **planner 在规划时
生成 **，写入 `bm25_query` 字段。
Query 应是 ** 简短的关键词组合 **，而非完整问句，例如：

| 原始问题                              | 好的 bm25_query                |
| ---------------------------------- | ---------------------------- |
| 原神中雷电将军的终结技叫什么名字？ | `原神 雷电将军 终结技 名称`    |
| 火影忍者手游里鸣人有哪些觉醒？     | `火影忍者手游 漩涡鸣人 觉醒`   |
| CS:GO Dust II 的 A 点炸弹区在哪里？  | `CS:GO Dust2 A 点 炸弹区 位置` |

## Request

- Method: `POST`
- Endpoint: `/bm25_ann`
- JSON payload:

```json
{
  "query": " 原神 雷电将军 终结技 名称",
  "top_k": 5
}
```

> `query` 填入 planner 生成的 `bm25_query`，而非原始问题。

Response

- 读取 `scores` 字段作为召回结果列表。
- 常见字段：`title`、`content`、`chunk_id`、`score` (BM25 分数，越高越相关)。
- 若所有文档均无词语重叠，`scores` 为空列表。

Decision Hints

- 若问题核心是 ** 精确关键词命中 **，优先本工具。
- 若问题需要 ** 语义相似度 ** 或 ** 跨语言理解 **，优先 `text_ann` 或 `multimodal_ann`。
- 本工具与 `text_ann` 互补：两者同时用可提升召回覆盖度。
```

SKILL.md: ann-image-recall

```
---
name: ann-image-recall
description: Image ANN retrieval tool skill. Use when the answer depends on visual
↪  similarity, UI style, iconography, or scene details from the image.

---

# ann-image-recall

## Tool Name

`img_ann`

## When to use

- 题目与图像中的视觉元素高度相关 (界面样式、图标、角色外观、场景布局)。
```

```
- 需要在图库中找“视觉上最像”的样本作为参考。
- 文本召回不足以解释图像细节。

## Request

- Method: `POST`
- Endpoint: `/img_ann`
- JSON payload:

```json
{
  "img": "/path/to/image.jpg",
  "top_k": 5
}
```

Response

- 读取 `scores` 字段作为召回结果列表。
- 常见字段: `img`、`query`、`pid`、`score`。

Knowledge Enrichment (自动触发, 无需额外规划)

调用 `img_ann` 后, pipeline 会自动执行以下步骤:

1. 从召回 top-N 结果中提取每条记录的 `query` 字段 (该字段为与图库图片关联的知识查询文本)。
2. 通过策略 (众数投票或 LLM 相关性判断) 选出最有价值的 query。
3. 调用内部 `kn_lookup` 服务, 检索对应详细知识内容 (游戏地图介绍、武器说明、规则解释等)。
4. 将知识内容追加至证据, 供答题器使用。

因此, `img_ann` 不仅返回视觉相似图像, ** 还会自动补充相关知识文本 **。

Decision Hints

- 当”看图找相似”是主需求时优先本工具。
- 若题目同时依赖文本语义和图像语义, 可优先尝试 `multimodal_ann`。
- 当题目涉及游戏机制、地图规则、武器参数等知识型内容, 且图像有明确游戏/场景标识时, `img_ann` +
↪ 自动知识补充的组合效果更优。
```

SKILL.md: ann-multimodal-recall

```
---
name: ann-multimodal-recall
description: Multimodal ANN retrieval tool skill. Use when both text intent and image
↪  evidence are needed together to recall the most relevant knowledge.

---

# ann-multimodal-recall

## Tool Name
`multimodal_ann`

## When to use
- 问题需要同时结合“图像内容 + 查询文本”才能准确检索。
- 单独文本 ANN 或图像 ANN 的结果都不稳定时。
- 你希望一次召回就兼顾语义和视觉相关性。

## Request
- Method: `POST`
- Endpoint: `/multimodal_ann`
- JSON payload:

```json
{
  "query": " 天涯明月刀 一代宗师毒圈 基本规则",
  "image_path": "/path/to/image.jpg",
  "top_k": 5
}
```

Response
- 读取 `scores` 字段作为召回结果列表。
- 常见字段: `title`、`content`、`source_query`、`source_best_img`、`score`。

Decision Hints
- 当你不确定该选文本还是图像召回时, 优先尝试本工具。
- 若仍无法回答, 可再补一个 `text_ann` 或 `img_ann`。
```

SKILL.md: ann-text-recall

```
---
name: ann-text-recall
description: Text ANN retrieval tool skill. Use when the question needs external textual
↪  knowledge, concept definitions, or rule details not directly visible in the image.

---

# ann-text-recall

## Tool Name

`text_ann`

## When to use

- 题目主要是概念、机制、背景知识，图像本身无法直接给出完整答案。
- 需要通过文本语义找相近知识片段（如术语解释、规则描述）。
- 你已经看过图像，但仍存在明显知识缺口。

## Request

- Method: `POST`
- Endpoint: `/text_ann`
- JSON payload:

```json
{
  "query": " 天涯明月刀 一代宗师毒圈 基本规则",
  "top_k": 5
}
```

Response

- 读取 `scores` 字段作为召回结果列表。
- 常见字段：`title`、`content`、`score`、`chunk_id`。

Decision Hints

- 若问题核心是“文本知识匹配”，优先尝试本工具。
- 若问题核心是“画面视觉细节”，优先考虑 `img_ann` 或 `multimodal_ann`。
```

SKILL.md: kn-lookup

```
---
name: kn-lookup
description: Knowledge base lookup service. Given a list of query strings (obtained from
↪  img_ann results), returns the corresponding long-form knowledge content from the
↪  internal knowledge base. This service is automatically triggered after img_ann and
↪  does not need to be called manually by the planner.
---

# kn-lookup

## Overview

`kn_lookup` 是一个本地知识库查询服务。它根据图像召回（`img_ann`）返回的 `query` 字段，
↪  在预构建的知识字典中检索对应的详细文本内容（如游戏规则、地图介绍、武器说明等）。

该服务在 `img_ann` 调用后 ** 自动触发 **，不需要规划器单独决策调用。
---
## 工作流程
```
img_ann 召回 → 提取 top-N query → 策略选取 → kn_lookup → 知识内容追加到证据
```
1. `img_ann` 返回 `records`，每条记录含 `query` 字段（与图库图片关联的文本查询）。
2. 从 top-N 记录中提取 `query`，通过策略（众数投票 / LLM 相关性判断）选出最有价值的查询。
3. 调用 `kn_lookup` 服务，查出对应的知识内容。
4. 将知识内容拼入证据（`evidence`）中，供答题器参考。

---
## Request

- Method: `POST`
- Endpoint: `/kn_lookup`
- JSON payload:
```

```json
{
  "queries": ["CS:GO Dust II", "CS:GO Mirage"]
}
```

字段	类型	说明
`queries`	`list[str]`	要查询的 query 字符串列表，来自 `img_ann` records 的 `query` 字段

Response

```json
{
  "results": [
    {
      "query": "CS:GO Dust II",
      "found": true,
      "contents": [
        "### 游戏内容介绍：CS:GO Dust II\n\n...",
        " 在 CS:GO 地图 Dust II 中，..."
      ]
    },
    {
      "query": "CS:GO Unknown Map",
      "found": false,
      "contents": []
    }
  ]
}
```

字段	说明
`found`	是否在知识库中找到该 query
`contents`	匹配的知识内容列表（同一 query 在两个文件中均有时，返回两条）

Query 选取策略

众数投票（默认，`majority`）

统计 `img_ann` top-N 结果中各 query 出现次数，选出频率最高的 K 个 query 进行知识查询。

- 优点：零额外 LLM 开销，快速稳定。
- 适用：图像库中同一主题出现频繁时（如同一游戏地图有多张截图）。

LLM 相关性选取（`llm`）

将所有候选 query 连同当前题目发给 LLM，由 LLM 判断哪些 query 对解题最有帮助，输出选中的 query 列表。

- 优点：能结合题意做语义对齐，精度更高。
- 适用：图像库 query 多样，需要语义判断时。

服务启动

```bash
python tools/kn_lookup_server.py          # 默认端口 8002
KN_PORT=9002 python tools/kn_lookup_server.py
KN_FILES=tools/query_rag_kn_part_1.jsonl:tools/qwen_rag_kn_part_2.jsonl python
↪  tools/kn_lookup_server.py
```

健康检查

```
GET /health
→ {"status": "ok", "num_unique_queries": 22800}
```

Decision Hints（供规划器参考）

- `kn_lookup` 由 pipeline 自动触发，无需规划器主动选择。
- 当 `img_ann` 证据中 `query` 字段有明确语义（如游戏名称、地图名、武器名）时，知识库极有可能命中，
↪ 补充知识效果显著。
- 若知识库未命中（`found=false`），不影响其他证据正常使用。

C.4 MMSEARCH-R1 AGENTIC-RL SEARCH MODEL

System description. MS-R1 internalizes tool use into model generation. Instead of relying on an external planner, the model first generates reasoning and then emits either a final-answer tag or a tool-call tag. The environment parses the action, executes the corresponding retrieval tool, appends the returned observation, and continues the generation process.

Algorithm 3 MS-R1 Search Inference and GRPO Training

Require: image I, question q, options O, policy π_θ, reference policy π_{ref}, retrieval depth k, mode $m \in \{\textsc{Infer}, \textsc{Train}\}$
Ensure: answer a^* if $m = \textsc{Infer}$; updated policy π_θ if $m = \textsc{Train}$

Shared three-round search rollout.
function ROLLOUT(π, I, q, O, k, decode)
  Initialize context c_1 with (I, q, O) and the round-1 prompt; initialize trajectory $\tau \leftarrow \emptyset$
  for $t = 1$ **to** 3 **do**
    Generate $y_t \sim \pi(\cdot \mid c_t)$ using decode; append y_t to τ
    $(\alpha_t, z_t) \leftarrow$ ParseAction(y_t) ▷ $\alpha_t \in \{\textsc{Answer}, \textsc{ImageSearch}, \textsc{TextSearch}, \textsc{Invalid}\}$
    if $\alpha_t = \textsc{Answer}$ **or** $t = 3$ **then**
      $a \leftarrow$ ExtractAnswer(y_t); **return** (a, τ)
    else if $\alpha_t \in \{\textsc{ImageSearch}, \textsc{TextSearch}\}$ **then**
      $(o_t, p_t) \leftarrow$ ExecuteTool(α_t, z_t, I, k) ▷ return observation and next-stage prompt
      $c_{t+1} \leftarrow c_t \oplus y_t \oplus o_t \oplus p_t$; append o_t to τ
    else
      Mark y_t as invalid in τ
    end if
  end for
end function

Inference.
if $m = \textsc{Infer}$ **then**
  $(a^*, \tau) \leftarrow \textsc{Rollout}(\pi_\theta, I, q, O, k, \text{greedy})$
  return a^*
end if

Training.
Sample G trajectories $\{(a_i, \tau_i)\}_{i=1}^G$ by $\textsc{Rollout}(\pi_\theta, I, q, O, k, \text{sample})$
$R_i \leftarrow R_{\text{ans}}(a_i) + R_{\text{format}}(\tau_i) + R_{\text{tool}}(\tau_i)$ for each trajectory i
Normalize $\{R_i\}_{i=1}^G$ within the group to obtain advantages $\{A_i\}_{i=1}^G$
Update π_θ with GRPO using $\{(\tau_i, A_i)\}_{i=1}^G$ and KL regularization to π_{ref}
return π_θ

Rollout and reward. IMAGESEARCH and TEXTSEARCH are parsed from `<search><img> </search>` and `<text_search>...</text_search>`. ExecuteTool calls `img_ann` or `text_ann`, wraps the result in `<information>...</information>`, and appends the next-stage prompt. Inference uses one greedy rollout, while training samples trajectories in groups and applies GRPO. The reward combines answer correctness, format validity, and tool-call validity, with a KL penalty against the reference model.

Prompt templates. Three stage-specific prompt templates define the MS-R1 interaction process. The first prompt defines the allowed actions and the three-round interaction constraint. The second prompt is used after image search and asks the model to determine whether the image-search observation is sufficient or whether text search is needed. The third prompt is used after text search and requires the model to produce the final answer. The original Chinese prompt templates are included below.

round1 user prompt

```
请根据给定图片回答用户问题。你正在参与一个多轮推理任务，允许你在必要时先调用搜索工具，再在后续轮次继续作答。

你需要遵循以下总体原则：

* 你的输出必须始终先写在 <reason>...</reason> 中，再决定下一步动作。
* 合法动作只有三种：

  1. 直接给出最终答案：<answer>...</answer>
  2. 调用图搜工具：<search><img></search>
  3. 调用文本搜索工具：<text_search> 你的查询</text_search>
* 最终答案必须放在 <answer>...</answer> 中，答案要简洁明确，不要展开解释。
* 如果题目给了多个候选答案，你在最终答案中必须输出完整的候选内容本身，不要输出缩写、简称、编号、代号，
↪  也不要只输出部分关键词。

你可能出现的合法链路包括：

1. 单轮：直接回答。
2. 双轮：先图搜，再回答。
3. 双轮：先文搜，再回答。
4. 三轮：先图搜，再文搜，最后回答。

请按下面顺序决策：

1. 先判断这个问题是否真的依赖图片。如果问题主要依赖常识或文本知识，而图片不是关键证据，则可以直接回答。
2. 如果问题依赖图片，再判断当前图片是否已经足够让你确认与问题相关的核心游戏元素。
3. 只要你对关键游戏元素、名称、物品、角色、地点、机制、界面名词或其他核心实体不是特别确定，
↪  或者虽然看出一些线索但还不能稳定支撑答案，就优先调用图搜工具，在回复末尾输出
↪  <search><img></search>。不要因为勉强猜到一个可能答案就跳过图搜。
4. 对于明显依赖图片的问题，只要你还不能非常稳定地说出“这张图里最关键的元素到底是什么”，
↪  默认优先图搜通常比直接猜更稳妥。与其冒险直接回答，不如先通过图搜确认元素。
5. 图搜的核心作用通常不是直接给出最终答案，而是帮助你确认图片对应的游戏元素、名称、物品、角色、地点、机制、
↪  界面名词或其他关键实体。
6. 一旦你已经确认了最可能的游戏元素，就判断是否已经可以结合你的已有知识直接回答。如果仍然不能稳定回答，
↪  就必须基于“游戏元素 + 用户问题”构造一个简洁、具体、可检索的文本搜索 query，并在回复末尾输出
↪  <text_search>你的查询</text_search>。

请特别注意：

* 对依赖图片的问题，图搜通常是一个值得优先尝试的步骤；如果你还在犹豫是否已经认准了元素，
↪  优先图搜通常比直接猜更稳妥。
* 即使你已经隐约猜到某个答案，只要关键视觉元素还没有确认清楚，仍然应该优先图搜，而不是直接凭印象作答。
* 图搜结果通常只是帮助你确认“这是哪个游戏元素”，后续真正关键的是利用这个元素信息继续推理。
* 搜索结果可能存在噪声、误召回或不完整信息，不要机械相信返回内容；你必须结合原图、用户问题、
↪  搜索分数和已有知识判断其是否可信。也就是说，可以先搜，再决定是否采用搜索结果。
* 如果图搜结果里至少有部分候选与原图足够接近，这些结果就值得拿来帮助你缩小元素范围，
↪  而不是因为存在误召回就完全放弃搜索结果。
* 如果需要做文本搜索，query 中不要遗漏已经确认的游戏元素名称，这是检索到正确信息的关键。
* 如果题目给了多个候选内容，你应输出你认为正确的那个候选内容的完整文本，不要偷懒省略成简短代称。
* 不要同时输出多个动作。
* 当你准备输出最终答案时，不要再调用任何工具。
* 示例：<answer>Titanic</answer>

所有后续搜索结果都会放在 <information>...</information> 中返回给你。

下面是图片与问题： <image>

问题：
{question}

候选答案：
{options}

请从给定候选答案中选择最正确的一项。请注意：如果你最终作答，必须输出完整候选答案内容本身，不要输出编号、
↪  字母、简称或部分片段。

额外约束：整个任务中，模型最多只能进行 3 次回复（即最多 3 轮 assistant 输出）。如果你已经进入第 3 次回复，
↪  就必须直接给出最终答案，不能再调用任何搜索工具。
```

after image search prompt

```
你已经完成了一轮图搜。现在你处于多轮任务的中间阶段，需要根据图搜结果决定：是直接回答，还是继续进行文本搜索。

你仍然必须遵守统一格式：
* 先在 <reason>...</reason> 中进行分析。
* 然后只能选择以下两种动作之一：
  1. 直接回答：<answer>...</answer>
  2. 继续文本搜索：<text_search> 你的查询</text_search>

请按下面顺序思考：
1. 先结合原图、用户问题以及图搜返回的多个候选结果，判断原图最对应的核心游戏元素是什么。
2. 重点参考返回结果中的元素名称、标题、关键词和分数，同时结合图搜返回的图片与原始 query 图片进行比对，
↪  判断哪个候选在视觉上和语义上都最匹配当前问题。图搜图片的核心价值是帮助你确认元素信息。
3. 注意，图搜结果可能存在误召回，分数高只表示相对更相似，不等于一定正确。你需要检查返回图片中的角色、物体、
↪  场景、UI、文本或关键元素是否真的与原图一致或足够接近；如果不一致，就应降低对该结果的信任。
4. 你可以把分数当作一个较强但不是绝对的参考：
   * 如果某个候选分数大于 0.95，并且视觉上与原图关键元素基本一致，通常可以较高置信地采信。
   * 如果某个候选分数大于 0.90，通常说明它比较可信，但仍要结合原图中的关键细节再确认。
   * 如果候选分数小于 0.90，就要更加仔细辨别；除非视觉证据非常吻合，否则一般不要轻易相信。
5. 即使图搜没有直接给出答案，它仍然可能帮助你缩小元素范围。你需要先判断哪些结果可采信、哪些结果应忽略，
↪  而不是直接照单全收。
6. 如果图搜结果里有一个或几个候选与原图明显更接近，你应优先把这些候选当作后续推理的基础，
↪  而不是继续停留在模糊猜测上。
7. 一旦确认了最可能的游戏元素，就判断：仅凭这个元素提示和你已有的知识，是否已经足以回答用户问题。
8. 如果已经足够回答，就直接给出最终答案。
9. 如果仍然不能稳定回答，就必须围绕“已确认的游戏元素 + 用户问题”构造一个高质量文本搜索 query，
↪  并在回复末尾输出 <text_search>你的查询</text_search>。

请特别注意：
* 图搜通常不能直接给出最终答案，它更重要的作用是帮助你确认原图对应的元素信息。
* 如果图搜结果彼此接近，你应在 <reason> 中说明你为何选择某个元素作为后续推理基础，尤其要说明它与原图、
↪  当前问题和图搜分数的匹配依据。
* 如果图搜结果整体都不可信，不要把误召回结果当成事实；但只要某些结果能帮助你进一步锁定元素范围，
↪  就可以据此继续做更有针对性的文本搜索。
* 当你对元素已经有一定把握，但对题目中的具体机制、数值、条件或隐藏规则还不够确定时，优先继续文本搜索，
↪  而不是仓促直接回答。
* 文本搜索 query 不能只重复用户原问题，必须显式包含你确认出的游戏元素名称或核心实体，
↪  这是文本搜索成功的关键。
* 如果题目给了多个候选内容，最终答案必须写出完整候选内容本身，不要输出缩写、代号或部分片段。
* 这一轮不要再调用图搜。
* 如果给出最终答案，必须使用 <answer>...</answer> 包裹，且答案保持简洁。
* 示例：<answer>Titanic</answer>

额外约束：这通常是任务的第 2 次回复。如果你此时选择继续文本搜索，那么下一次回复必须直接输出最终答案，
↪  不能再调用任何工具。
```

after text search prompt

```
你已经完成文本搜索，现在处于多轮任务的最后收束阶段。此时你的目标是结合已有信息，尽可能稳定地给出最终答案。
你仍然必须遵守统一格式：
* 先在 <reason>...</reason> 中分析。
* 然后输出最终答案：<answer>...</answer>

请结合以下信息完成最终推理：
* 用户原问题
* 之前确认出的游戏元素或核心实体
* 当前返回的文本搜索结果

要求如下：
1. 在 <reason> 中说明文本搜索结果如何支持某个答案，以及你如何利用前面确认的游戏元素缩小范围。
2. 文本搜索结果也可能包含噪声、错页、泛化描述或与当前元素不完全对应的内容。你需要优先采信与已确认游戏元素、
↪  问题语义和候选答案都一致的结果，而不是机械相信任意一条检索信息。
3. 如果文本搜索结果与前面确认的游戏元素明显不一致，或者只支持了一个过于泛化的结论，你应降低其权重，
↪  并结合上下文做更稳妥的判断。
4. 如果多条检索结果彼此冲突，你应优先采用与已确认元素更一致、与问题更贴合、且能更直接支持某个候选答案的证据。
↪
5. 如果检索结果仍不完全充分，也要结合你已有知识和上下文，选择当前最合理、最恰当的答案。
6. 如果题目给了多个候选内容，最终答案必须输出完整候选内容本身，不要输出缩写、代号、部分关键词或不完整片段。
7. 不要输出拒答语句，也不要继续调用任何工具。
8. 最终答案必须放在 <answer>...</answer> 中，答案要简洁、明确，不要额外展开。

请记住：文本搜索阶段的核心不是重新泛化问题，而是利用前面已经确认的游戏元素与检索结果完成收束并最终回答。
额外约束：这已经是最后收束阶段，通常对应任务中的最后一次回复。你这一轮必须直接输出最终答案，不能继续搜索，
↪  也不能输出额外的下一步计划。
```

C.5 Paradigm Comparison

The three tool-augmented settings differ in their level of search autonomy rather than in the retrieval resources they can access. Fixed RAG follows a hard-coded image-to-text retrieval chain and serves as a simple reproducible baseline. PAR exposes the same tools to an explicit planner, allowing the model to adaptively choose between visual, semantic, lexical, and multimodal retrieval. MS-R1 further internalizes tool invocation into the model generation process and learns search behavior through outcome-based reinforcement learning.

Table 6 summarizes the shared retrieval tools used by these settings. Since the tool set is fixed across methods, performance differences mainly reflect search behavior, tool-use planning, and evidence integration under controlled retrieval conditions.

Table 6: Shared retrieval tools used across the three tool-augmented settings.

Tool	**Input**	**Function**
`img_ann`	Image	Retrieves visually similar game images from the frozen image pool. Each retrieved record is associated with a core-element query, enabling downstream knowledge lookup.
`text_ann`	Text query	Retrieves semantically relevant knowledge chunks from the text index. It is used for game rules, concept definitions, equipment effects, background descriptions, and mechanics.
`bm25_ann`	Keyword query	Performs sparse lexical retrieval over the knowledge chunks. It is useful for exact matching of character names, map names, skill names, equipment names, interface terms, and other game-specific entities.
`multimodal_ann`	Image and text query	Retrieves evidence using both visual content and textual intent. It is used when neither image-only nor text-only retrieval is sufficient.
`kn_lookup`	Core-element query	Looks up structured knowledge associated with a core element. It is automatically triggered after `img_ann` and is not exposed as an independent planner action.

D REPRESENTATIVE EXAMPLES

Case 1

Video Information

TITLE

鸣潮 鸣潮相里要 鸣潮吟霖 导电声骸速刷路线（1）

WutheringWaves XiangliYao Yinlin Electro (Conductive) Echo Farming Route (1)

COVER OCR

导电套, 彻空冥雷,COST3 速刷路线,=C3

Electro Set –Void Thunder | COST 3 Speed Farming Route (C3)

ASR TRANSCRIPT

大家好这里是霖宝，本期是导电声骸速刷路线。先传送到北落野信标，往右前方走，依次刷乐师、惊奇猎手；随后传送到燎照之骑全息左侧锚点，右转清理火属性怪、乐师和惊奇猎手。回头爬上高处，继续刷乐师和惊奇猎手……

Hi, this is Linbao. This is an Electro Echo farming route. First, teleport to the Beiluo Field waypoint, move to the front-right, and clear the Musician and Aberrant Hunters along the way. Then teleport to the waypoint left of the Inferno Rider Hologram, turn right, and clear the Fusion-type enemies, Musician, and Aberrant Hunters. Head back and climb up, then continue clearing more Musicians and Aberrant Hunters ...

Query



Query image

QUESTION

该机制 5 件套效果最多可叠加几层导电伤害增益?

How many layers of conductive damage buff can this 5-piece set effect stack at most ?

OPTIONS

A. 1 层
1 stack

B. **2 层**
2 stacks

C. 3 层
3 stacks

D. 4 层
4 stacks

GROUND TRUTH **B. 2 层 / 2 stacks**

Model Outputs

Vanilla Model ✗ Wrong

ANSWER: C. 3 stacks

Thinking Model ✗ Wrong

ANSWER: C. 3 stacks
REASONING: *[Thinking]* The question asks for the maximum number of Electro DMG bonus stacks provided by the 5-piece set effect. The image mainly shows the game interface (character, skills, level), but no explicit evidence related to the stacking mechanic is present…
[Final] Therefore, the most plausible answer is C. 3 stacks.

RAG (Ours) ✓ Correct

ANSWER: **B. 2 stacks**

RETRIEVED EVIDENCE

[R1]



鸣潮 合鸣 · 彻空冥雷
Wuthering Waves Resonance: Void Thunder

[R2]

QUERY: *In Wuthering Waves, under the Resonance: Void Thunder set, what is the maximum…*

DOCUMENT:
Void Thunder (Wuthering Waves)…2-Piece: Electro DMG +10%…this effect can stack up to 2 times, with each stack lasting 15 seconds…

LangGraph (Ours) ✓ Correct

ANSWER: **B. 2 stacks**
REASONING: *[Plan]* Identify the set effect and its stacking rule⋯
[Tool: img_ann] Retrieve related image of "Resonance: Void Thunder" …
[Tool: text_ann] Retrieve set description⋯result shows 5-piece effect grants Electro DMG bonus stacks…
[Synthesize] The description explicitly states the effect stacks up to 2 times, each lasting 15s…thus answer is 2 stacks.

RETRIEVED EVIDENCE

[R1]



鸣潮 合鸣 · 彻空冥雷
Wuthering Waves Resonance: Void Thunder

[R2]

QUERY: *In Wuthering Waves, what is the max Electro DMG stacks from the 5-piece effect?*

DOCUMENT:
Void Thunder (Wuthering Waves)…2-Piece: Electro DMG +10%…this effect can stack up to 2 times, with each stack lasting 15 seconds…

MMSearch-R1 (Ours) ✅ **Correct**

ANSWER: B. 2 stacks

REASONING: *[Round 1]* The question asks for the maximum number of stacks in a 5-piece set effect, which depends on identifying the exact in-game mechanism. Since the image alone does not provide enough reliable evidence about the set or its stacking rule, the model first calls image search.

[Round 2] The returned visual candidates suggest the relevant element may be "Resonance: Void Thunder," but the match is still not reliable enough to answer the stacking question directly. The model therefore continues with text search to verify the 5-piece effect description.

[Round 3] The retrieved text states that the 5-piece effect grants Electro DMG Bonus after certain actions, and that this effect can stack up to 2 times, each lasting 15 seconds. Therefore, the correct answer is 2 stacks.

RETRIEVED EVIDENCE

`[R1]`



鸣潮合鸣 · 彻空冥雷

Wuthering Waves Resonance: Void Thunder

`[R2]`

QUERY: *Electro DMG Bonus 5-piece effect*

DOCUMENT:

Void Thunder (Wuthering Waves)…2-Piece: Electro DMG +10%…this effect can stack up to 2 times, with each stack lasting 15 seconds …

Case 2

Video Information

TITLE

怪物猎人荒野白炽龙套装展示-充能斧 原视频来自 youtube，作者 Pinit

Monster Hunter Wilds –Whiteflame Wyvern Armor Set Showcase | Charge Blade

COVER OCR

Monster Hunter Wilds –Whiteflame Wyvern Armor Set | Charge Blade | Armor Set Showcase

《怪物猎人：荒野》白炽龙套装展示 | 充能斧

ASR TRANSCRIPT

N/A

No ASR transcript available for this sample.

Query



Query image

QUESTION

充能斧在剑模式下如何为瓶子充能?

How are phials generated/charged in Sword Mode with the Charge Blade?

OPTIONS

A. **攻击敌人积累能量**
Build up energy by attacking enemies

B. 使用 R+A 键注入能量
Inject energy using the R + A input

C. 长按 ◯ 键触发 GP
Hold the Circle (◯) button to trigger Guard Point (GP)

D. 消耗瓶子释放技能
Consume phials to unleash attacks

GROUND TRUTH **A. 攻击敌人积累能量 / Build up energy by attacking enemies**

Model Outputs

Vanilla Model ✗ Wrong

ANSWER: B. Inject energy using the R + A input

Thinking Model ✗ Wrong

ANSWER: B. Inject energy using the R + A input
REASONING: *[Thinking]* The question asks how the Charge Blade charges phials in Sword Mode…However, the image provides no explicit control instructions, so the answer cannot be derived purely from visual evidence…Based on typical game mechanics, charging is performed via a specific input…
[Final] Therefore, option B (using the R + A input to inject energy) is selected.

RAG (Ours) ✓ Correct

ANSWER: **A. Build up energy by attacking enemies**

RETRIEVED EVIDENCE

[R1]



怪物猎人　狩猎笛
Monster Hunter: Hunting Horn

[R2]

QUERY: *In Monster Hunter, considering the Hunting Horn and Charge Blade, how does the Charge Blade charge phials…*

DOCUMENT:
1) Phials are charged through sword attacks and later consumed to unleash…
2) The Charge Blade is a morph weapon that switches between Sword and Axe modes, storing energy…

LangGraph (Ours) ✓ Correct

ANSWER: **A. Build up energy by attacking enemies**
REASONING: *[Plan]* Identify how phials are charged in Sword Mode⋯
[Tool: img_ann] Retrieve relevant Monster Hunter weapon visuals⋯
[Tool: text_ann] Retrieve Charge Blade mechanics⋯result shows phials are built up through attacks⋯
[Synthesize] Evidence indicates energy is accumulated via attacking enemies⋯thus, the answer is A. Build up energy by attacking enemies.

RETRIEVED EVIDENCE

[R1]



怪物猎人　狩猎笛
Monster Hunter: Hunting Horn

[R2]

QUERY: *How does the Charge Blade charge phials while in Sword Mode?*

DOCUMENT:
1) In Sword Mode, phials are charged by attacking enemies. Charging progresses in two stages: yellow and red…
2) In Axe Mode, these phials can be consumed to perform Elemental Discharge attacks…

MMSearch-R1 (Ours) ✅ **Correct**

ANSWER: **A. Build up energy by attacking enemies**

REASONING: *[Round 1]* The image shows a Charge Blade in Monster Hunter Wilds, but does not provide explicit charging mechanics. The model first performs image search to identify the weapon.
[Round 2] Visual results confirm the weapon as the Charge Blade (despite some noisy matches such as Hunting Horn). However, no direct evidence about phial charging is obtained, so the model proceeds to text search.
[Round 3] Retrieved knowledge indicates that in Sword Mode, phials are charged by attacking enemies. Therefore, the correct answer is A.

RETRIEVED EVIDENCE

[R1]



怪物猎人　狩猎笛
Monster Hunter: Hunting Horn

[R2]

QUERY: *Monster Hunter Wilds –Charge Blade Sword Mode phial charging method*

DOCUMENT:

1) The Charge Blade, also known as the Switch Axe Shield (or simply Charge Blade), is a hybrid weapon in the Monster Hunter series…
2) The Charge Blade is a weapon that combines both flexibility and burst damage, and holds a prominent role in the Monster Hunter series due to its unique sword–axe switching mechanics and phial system…