Generating Multi-Modal Knowledge Clues as an Image: Toward Improving Image-Sequence Reasoning With Assisted Visual Input

Guanghui Ye, *Graduate Student Member, IEEE*, Huan Zhao, *Member, IEEE*, Yixian Shen, *Member, IEEE*, Jiaqi Li, Fengnan Li, Zhihua Jiang, *Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—Recent multi-modal large language models (MLLMs) have exhibited powerful abilities in addressing complex vision-language tasks such as image-sequence reasoning (ISR). However, significant challenges remain, e.g., it is still difficult for the MLLMs to fully capture and represent cross-image visual knowledge such as scene relations, attributes, and entity links between multiple images, which hinders them from better solving ISR. To alleviate these issues, we introduce a novel concept Visualized Knowledge Clue (VizKC) - synthetic images that encode key visual and external knowledge from a sequence of input images and are then used alongside the original input images within a multi-image MLLM to enhance reasoning performance. Accordingly, we propose an accompanying approach named VizKC-ISR, composed of two modules - VizKC generation and VizKC utilization. Specifically, in the generation module, VizKC-ISR follows a *See-Find-Fuse* pipeline: 1) "*See - Scene Perception*", to construct an initial VizKC that incorporates scene relations of key visual entities detected from an original image; 2) "*Find - Knowledge Generation*", to generate enriched image captions with real-world knowledge and fine-grained entity details and then extract structured knowledge tuples from generated captions; 3) "*Fuse - Image Editing*", to introduce relevant knowledge tuples into the VizKC via iterative image editing. In the utilization module, we employ a multi-image MLLM (e.g., mPLUG-Owl3) to solve the VizKC-assisted ISR tasks by reasoning with generated knowledge clues. We evaluate VizKC-ISR on nine ISR benchmarks categorized into three multi-image scenarios. The results show that our VizKC-ISR performs best in all tasks, e.g., obtaining the highest average accuracy of 63.1% and surpassing the mPLUG-Owl3 baseline by 6.4 absolute points, due to the bridge between visually-grounded reasoning and multi-modal knowledge challenges.**

***Index Terms*—Knowledge representation and reasoning, visual knowledge, synthetic image generation, multi-modal large language models, multi-image reasoning.**

I. Introduction

VISUAL knowledge [1], [2], [3] offers a novel form of knowledge representation, capturing visual concepts and their connections in a brief, complete, and understandable way, with strong connections to cognitive psychology [4], [5]. By incorporating visual input into the embedding space of large language models (LLMs), large vision-language models (LVLMs) utilize the robust interpretation abilities of LLMs for complex vision-language tasks like visual question answering (VQA) [6], [7], [8], video understanding [9], [10], grounding [11], *etc*. However, existing LVLMs, sometimes called multi-modal LLMs (MLLMs), focus mainly on single-image scenarios. Meanwhile, advanced image-sequence understanding capabilities [12] are required in practical applications. As an example, real-world multimedia data, like web pages, often have images and text mixed together. Hence, multi-image situations are more practically useful than single-image situations [13], so it is worthwhile to investigate MLLMs' emergent abilities for multi-image inputs when solving vision-language tasks.

Image-sequence reasoning (ISR) [7], [10], [14], normally in the form of multi-image VQA, is a challenging task that requires the model to understand and analyze the relationships between objects or events shown in a given series of input images. Several recent MLLMs have exhibited powerful abilities in handling ISR tasks. For example, Idefics2 [15] pre-trained the Idefics using interleaved image-text data. Mantis [16] performed instruction tuning on multi-modal document-level data. Using both pre-training and fine-tuning, mPLUG-Owl3 [14] improved understanding of long image sequences. However, significant challenges remain, with the focus on **the acquisition and representation of visual knowledge**. For knowledge acquisition, MLLMs still struggle to fully capture a variety of visual knowledge (e.g., scene relations, attributes, and entity links) across multiple images, towards addressing ISR in different scenarios such as logical, temporal, or commonsense reasoning. For knowledge representation,

Received 10 October 2025; revised 5 February 2026; accepted 15 March 2026. Date of publication 20 March 2026; date of current version 6 August 2026. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62076092. This article was recommended by Associate Editor N. Xu. *(Corresponding authors: Huan Zhao; Zhihua Jiang.)*

Guanghui Ye and Huan Zhao are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: yghui@hnu.edu.cn; hzhao@hnu.edu.cn).

Yixian Shen is with the Informatics Institute, University of Amsterdam, 1098 XH Amsterdam, The Netherlands (e-mail: y.shen@uva.nl).

Jiaqi Li is with the Department of Computer Science, University of Warwick, CV4 7AL Coventry, U.K. (e-mail: Jiaqi.Li.16@warwick.ac.uk).

Fengnan Li is with the Department of Biostatistics and Bioinformatics, Duke University, Durham, NC 27710 USA (e-mail: fengnan.li@duke.edu).

Zhihua Jiang is with the Department of Computer Science, Jinan University, Guangzhou 510632, China (e-mail: tjiangzhh@jnu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York at New Paltz, New Paltz, NY 12561 USA, and also with the College of Information Science and Engineering, Hunan University, Changsha 410082, China (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TCSVT.2026.3676204

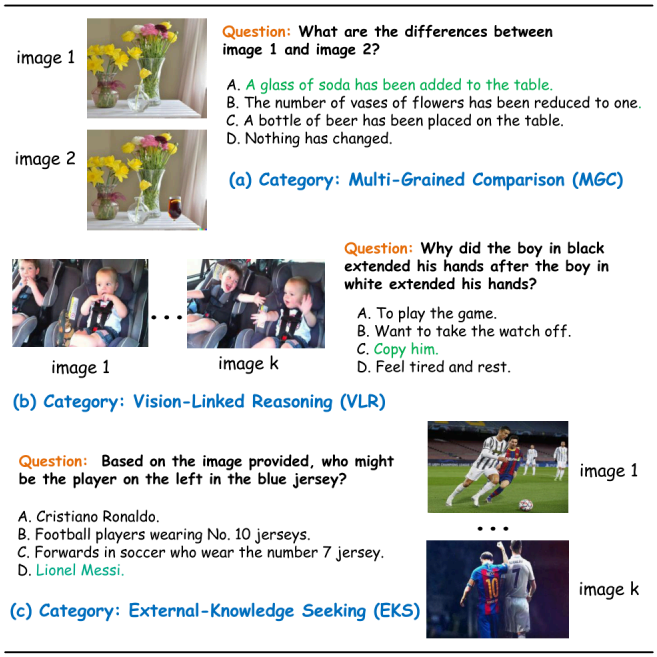


Fig. 1. Motivation illustration of our VizKC concept.

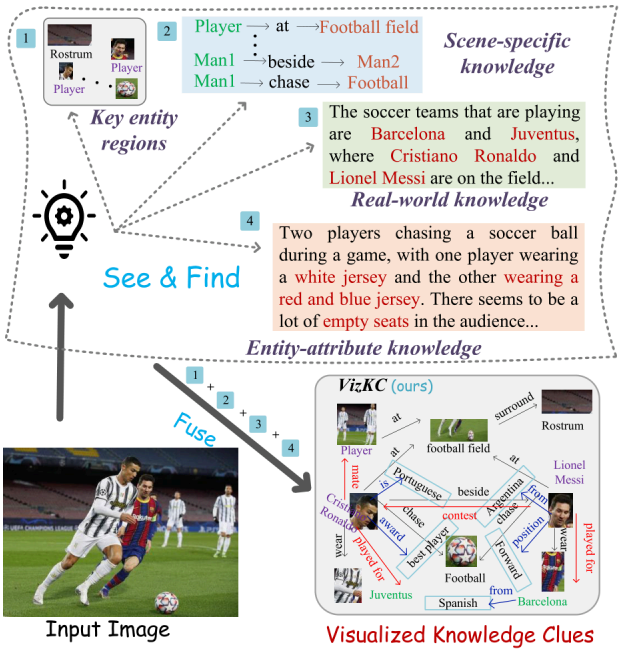


Fig. 2. Motivation illustration of our VizKC concept.

traditional structural tuples facilitate representation learning in natural language processing (NLP), while their textual forms are fundamentally ill-suited to the multi-image abilities of MLLMs.

For better solving ISR, it is a promising direction to capture visual knowledge [17], [18], [19] as useful task clues. As Fig. 1 shows, (a) when we compare different images, scene understanding can provide spatially-grounded relations between visual objects in an image, e.g., a glass of soda on the table is a key clue; (b) when we perform abstract reasoning tasks involving intention or emotion, causal analysis can bring rational predictions, e.g., the boy in black tries to imitate the boy in white when he extends his hands in the same way; (c) when we recognize notable figures or events, querying real-world knowledge can offer more precise answers, e.g., identifying the famous player "Lionel Messi" plays a vital role to answer the question in the last picture. Meanwhile, fine-grained entity details offer key links between similar objects or consecutive events, e.g., even though the figures' faces cannot be seen clearly, a model can still infer that the query player is "Lionel Messi" due to entity details ("wearing a blue jersey") observed from previous images. These task examples motivate us to target the limitations of current MLLMs in capturing key visual knowledge, particularly in cross-image scenarios.

fig Meanwhile, it is another promising direction to represent visual knowledge or make it more understandable, following the development trend of current MLLMs. Traditional knowledge tuples are usually represented with textual features and then inserted into the sequence of input-text embeddings [20], [21], [22], [23], which can easily exhaust the LLM's context window, resulting in significant memory and computational overhead. Some studies [24], [25] constructed knowledge graphs (KGs), a graph-structured collection of textual knowledge tuples, and then trained graph neural networks (GNNs) to represent KGs, which can probably increase the burden of model architecture. These disadvantages hinder the MLLMs from modeling long-vision inputs. Inspired by recent advances in multi-image MLLMs [13], [14], [16], we aim to explore the new perspective for multi-modal knowledge challenges and pose the following question: *Can we fully capture key visual knowledge from the original task and then represent it in a new modality (e.g., images), to further provoke the multi-image ability of MLLMs?*

To this end, we introduce a novel concept - **Visualized Knowledge Clue (VizKC)** - that encodes key visual and external knowledge from the original task (input images, plus the query question) and propose an accompanying framework - **VizKC-ISR** - that improves ISR through the use of VizKC. Our method is aligned with human intuition, where visual diagrams can support complex reasoning. Accordingly, rather than relying solely on textual prompts or large-scale multi-modal pretraining, we propose generating an auxiliary image that fuses scene layouts and entity details extracted from the input image. Such images are then used alongside the original input images within a multi-image MLLM to enhance reasoning performance. As shown in Fig. 2, the proposed system VizKC-ISR follows a modular *See-Find-Fuse* pipeline: construct scene graphs (*See*), extract enriched entity knowledge through captioning models (*Find*), and then synthesize the final image infused with knowledge through iterative editing (*Fuse*). We perform extensive experiments in nine ISR tasks, along with extensive ablation studies and in-depth analysis, demonstrating the effectiveness and performance advantages of our proposal.

Our work bridges multimodal reasoning and knowledge integration with potential applications in education, robotics, and interactive systems. Concretely, we propose the following contributions.

- We present the fresh idea VizKC, which produces visual knowledge clues as an image instead of text or features. It is innovative to expand textual knowledge representation into visually-grounded, multi-modal clues. VizKC also enjoys several advantages: (i) is a vivid image rather than plain text; (ii) is task-independent, which can be applied to any visual task; (iii) is model-agnostic, which can be applicable to a variety of models.
- We propose the novel approach VizKC-ISR, which generates VizKC to improve ISR tasks in MLLMs. The comprehensive integration, which encompasses visual scene perception, caption-sourced knowledge generation, iterative image editing, and multi-image model application, provides a holistic solution to ISR. The methodological innovations focus mostly on: (1) the see-find-fuse generation pipeline methodically arranges and converts visual and external information into structured visual hints; and (2) extending textual triples to include multi-modal, spatially grounded entity descriptions as well as using explicit relation categorization yields a knowledge representation that is interpretable and task-aligned.
- We perform extensive experiments that cover nine ISR benchmarks, along with different model types, comprising task-specific methods and single-/multi-image MLLMs. The results show that our VizKC-ISR beats all others, achieving new state-of-the-art (SOTA) results in all tasks.

II. Related Work

A. Visual Knowledge

Future Artificial Intelligence (AI), according to visual knowledge theory, requires complete expression of visual concepts, detail of attributes, and reasoning about compositions, comparisons, and predictions through a unified, abstract, and interpretable representation, as informed by cognitive studies of human mental imagery [4], [5], [26], [27], [28], [29], [30]. LVLMs have become a potent instrument for finding varied patterns in large datasets that they convert into implicit knowledge [31], [32], [33], eventually abstracting these patterns into an immense set of numeric parameters. For example, Zhao et al. [2] investigated how variations in vision encoder representations influence the comprehension of LVLMs. Chen et al. [3] devised a LION model that empowers MLLMs by injecting visual knowledge (spatial-aware knowledge and semantic visual evidence). However, most MLLMs' image understanding is limited because they use vision encoders (e.g., CLIP [34]) that are pre-trained in coarse-grained image-caption supervision.

Knowledge graphs (KGs) [20], [35] can improve LLMs by providing external knowledge for inference and interpretability. However, KGs' creation and development are naturally challenging, thus testing existing methods in generating new facts and representing unseen knowledge. Some methods have been proposed to unify LLMs and KGs together. Unfortunately, traditional KGs are usually represented with textual features and then inserted into the sequence of input-text embeddings [22], [23], which can easily exhaust the LLM's context window. In addition, graph neural networks (GNNs) [24], [25] have been pre-trained to represent KGs, but it can probably increase the burden of model architecture. These disadvantages hinder the MLLMs from modeling long-vision inputs.

In our work, we propose VizKC which produces visual knowledge as an image instead of text or features, differently from existing studies. Such synthetic images are then used alongside the original input images within a multi-image MLLM to enhance reasoning performance.

B. Image-Sequence Reasoning

In this paper, we categorize distinct ISR tasks into three scenarios for a better summarization: Multi-Grained Comparison (MGC) (our subtasks 1-2), Vision-Linked Reasoning (VLR) (our subtasks 3-7), External-Knowledge Seeking (EKS) (our subtasks 8-9). As Fig. 1 shows, MGC embodies the most basic element of multi-image functionality; VLR requires a model to reason the relationships of objects or events based on its comprehension of the overall content conveyed by input images; and EKS queries external knowledge to better answer the question. We provide a brief introduction to each subtask: (1) The General Comparison (GC) task [36] tests a model's general understanding and comparison skills regarding images; (2) The Subtle Difference (SD) task [37] evaluates the model's capacity to recognize slight variations in comparable images; (3) The Visual Referring (VR) task [38] assesses if the model can use referring data from input images to understand object relationships; (4) The Temporal Reasoning (TR) task [39] tests a model's grasp of time-based links in image sequences; (5) The Logical Reasoning (LR) task [40] requires the model to reason logically and assess causal links in images; (6) The Fine-grained Visual Recognition (FVR) task [41] examines the model's ability to identify objects in query images and is assessed using reference images; (7) The Text-Rich Images (TRI) task [42] tests a model's skill in understanding text-filled images and retrieving pertinent data; (8) The Vision-linked Textual Knowledge (VTK) task [43] demands a model to connect the query image to relevant images and gather helpful information from the related text; (9) The Text-linked Visual Knowledge (TVK) task [44] needs a model to associate the question with pertinent text from reference images and to collect visual data from the appropriate image.

In our work, we perform experiments that span all of these tasks. The results show that our approach is feasible for not only simple tasks (e.g., GC and SD) where objects are visually distinguishable but also more complex reasoning tasks (e.g., VR, LR, and VTK) involving longer image sequences and commonsense knowledge, intent, or emotion reasoning.

C. Multi-Image MLLMs

Based on the power of LLMs, numerous MLLMs have obtained remarkable performance on various vision-language tasks. For example, LLaVA [45] used GPT-4 to produce text-image instruction-following data. Later, LLaVA-1.5 [46] demonstrated stronger performances, using CLIP-ViT. Qwen-VL-Chat, as described in [47], is a Qwen-VL-based visual

language model that is interactive and uses alignment. mPLUG-Owl [48] presented a new training paradigm. This paradigm gives LLMs multi-modal abilities via modular learning of a foundation LLM. However, these MLLMs primarily handle single-image input scenarios, leaving the performance of MLLMs when addressing practical multiple images underexplored. To this end, Liu et al. [13] built the MIBench benchmark to comprehensively evaluate the multi-image capabilities of MLLMs. The results revealed that, when faced with multi-image inputs, most models exhibited significant shortcomings, e.g., limited cross-image reasoning or fine-grained perception. Thus, MLLMs are suggested to be re-trained using specific multi-image data. Therefore, Idefics2 [15] performed a multi-stage pre-training using interleaved image-text documents, image-caption pairs, and PDF documents. Unlike Idefics2 that collected noisy data from the Web, Mantis [16] performed instruction tuning on academic-level data resources. Combining training methods sourced from both Idefics2 and Mantis, mPLUG-Owl3 [14] improved the ability to understand long image sequences.

Inspired by multi-image MLLMs, our work proposes VizKC-ISR which generates VizKC to improve ISR tasks in MLLMs. We also implement VizKC-ISR with multiple multi-image MLLMs (Mantis, Idefics2, mPLUG-Owl3), suggesting the adaptability and model-agnosticism of our proposal.

III. Formalization

In this section, we formally define ISR tasks which are normally depicted in the form of multi-image VQA.

Definition 1: Image-sequence Reasoning (ISR) Task. Given an input instance $< \{V_j\}_J, Q >$, which consists of a set of input images $\{V_j\}_J$ and a textual question Q, the standard ISR task is to generate an answer A based on the information available in the images and text. In this setup, an ISR model M can be represented as $M(\{V_j\}_J, Q) \rightarrow A$.

In NLP, textual knowledge is typically represented as a **structured triple** $< E_h, Re, E_t >$, where E_h, Re, and E_t denote *head entity*, *relation*, and *tail entity*, respectively. However, to formally depict multi-modal knowledge in our VizKC, we extend textual triples to multi-modal, spatially grounded entity descriptions, and adopt explicit categorization of relations (spatial, attribute, linking). These conceptual and methodological innovations provide an interpretable and task-aligned knowledge representation for our work.

Definition 2: Visualized Knowledge Clue (VizKC). VizKC can be formally represented as a collection of **multi-modal structured triples** $< (E_h, R_h), Re, (E_t, R_t) >$, where E_h/E_t denotes *head/tail entity name*, R_h/R_t denotes *head/tail entity region* (possibly empty), and Re denotes a *relation* from (E_h, R_h) to (E_t, R_t). Re can be classified into three cases: (i) *spatial relation (SRe)*, i.e., a triple $< (E_h, R_h), SRe, (E_t, R_t) >$ indicates that the entity (E_h, R_h) has a relative position regarding the entity (E_t, R_t) (e.g., <Lionel Messi, at, Football field>); (ii) *attribute relation (ARe)*, i.e., a triple $< (E_h, R_h), ARe, (E_t, R_t) >$ indicates that the entity (E_h, R_h) has a value (E_t, R_t) on the attribute ARe (e.g., <Lionel Messi, position, Forward>); and (iii) *linking relation (LRe)*, i.e., a triple $< (E_h, R_h), LRe, (E_t, R_t) >$ indicates that there is an linking relation LRe between (E_h, R_h) and (E_t, R_t) (e.g., <Lionel Messi, wear, Barcelona>). All examples mentioned here can be found in Fig. 2.

Definition 3: VizKC-assisted ISR Task. Given an ISR input $< \{V_j\}_J, Q >$, our VizKC-assisted ISR task aims to find a more accurate answer, provided a generated $VizKC_j$ sourced from the image V_j as an additional visual input, i.e., our model can be depicted as $M'(\{(V_j, VizKC_j)\}_J, Q) \rightarrow A$.

IV. VizKC Generation

We formally describe the entire process of VizKC-ISR in Alg.1 and use illustrative examples to show each step of VizKC generation in Fig. 3. The proposed VizKC-ISR system follows a modular *See-Find-Fuse* pipeline There are two key benefits: (i) the generation task at each step is simpler than learning to create a detailed knowledge image directly. Through each step, the search space is restricted to the associated contents of the previously generated components. (ii) the data flow at each step is essentially a collection of knowledge tuples, which can be effectively integrated by advanced technologies such as scene understanding, entity linking, and image editing.

1) See-Scene Perception:

Scene understanding [35], [49] has appeared as a key area in computer vision research, offering a structured depiction of a graph through the identification of objects and their relations. To generate a clear spatial organization of VizKC, we first construct a scene graph G_S using a robust engine, HiKER-SGG [50], which can generate scene graphs in the most challenging settings, including corrupted images. Specifically, we input a raw image V into HiKER-SGG which then outputs a scene graph G_S, i.e., G_S = HiKER-SGG (V) (step 5 in Alg.1). The G_S is a collection of textual triples $< E_h, SRe, E_t >$ where SRe indicates a spatial relation, e.g., <Man1, beside, Man2>. Then, we use a robust entity extraction model, GLEE [11], to detect key visual entities and segment the global image into small region images. Specifically, we input the same V into GLEE which then outputs a set of entity-region pairs $(E, R)_M$ (M is a preset largest number of entities to be detected), i.e., $(E, R)_M$ = GLEE (V) (step 6). Next, we obtain a scene image V_S, using a graph visualization software, graphviz, which represents structural information as abstract graphs or networks. Concretely, we visualize all elements (entity and relation) in G_S via using the layout programs and replace each E with its multi-modal pair (E, R). Consequently, V_S = Graphviz $(G_S, (E, R)_M)$ (step 7) is an extended collection of multi-model triples $< (E_h, R_h), SRe, (E_t, R_t) >$. This detailed breakdown of how HiKER-SGG, GLEE, and Graphviz interact to generate the final knowledge representation can clearly explain *how scene graphs, entity-region pairs, and visualized knowledge are integrated as a synthetic image.*

2) Find-Knowledge Generation:

Following [51] and [52], we employ LLMs as a knowledge generator to extract precise knowledge tuples from a given text. Since our VizKC is designed to capture visual knowledge

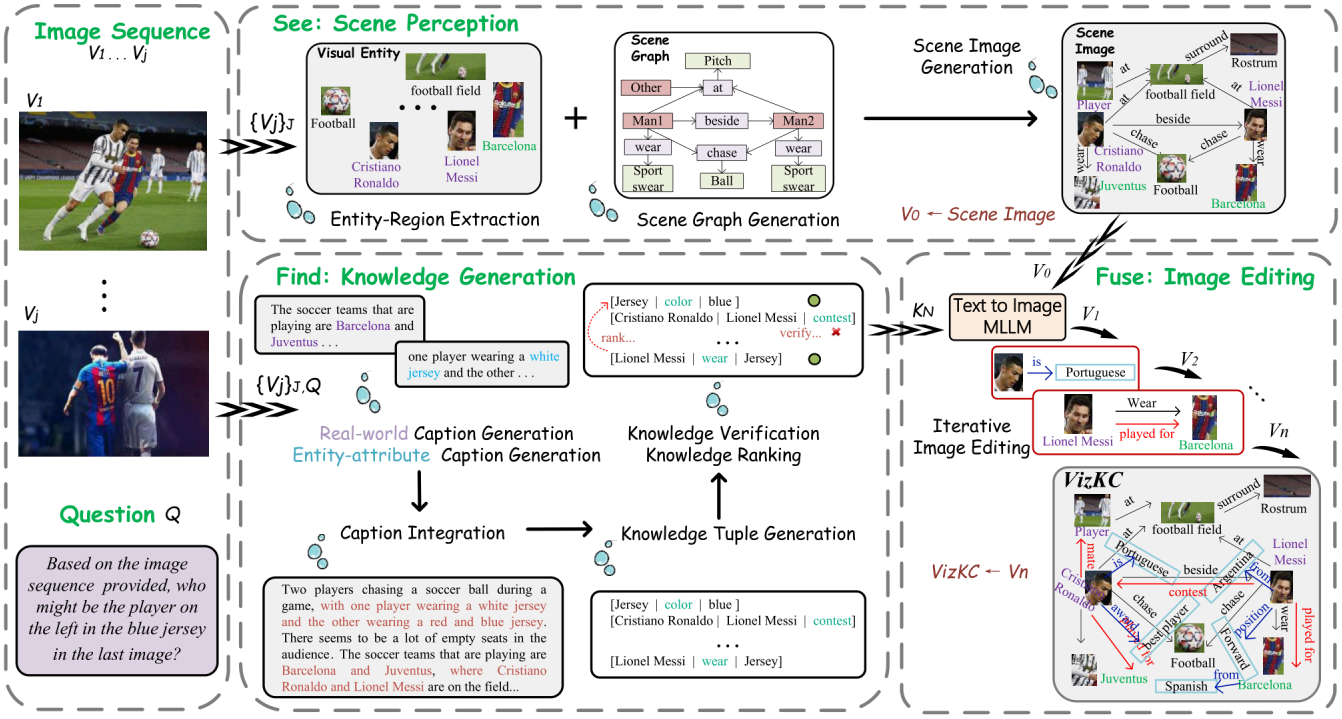


Fig. 3. The overview of VizKC generation. We use illustrative examples to show each generation step and formally describe the entire process of VizKC-ISR (VizKC generation and VizKC utilization) in Alg.1. For generation, we provide a *See-Find-Fuse* pipeline: (1) *See*: given a raw image, we construct its scene graph, segment the raw image, and thus integrate the constructed scene graph and detected entity regions into an initial VizKC. (2) *Find*: we generate two categories of enriched image captions, and employ an LLM to extract knowledge tuples from generated captions. (3) *Fuse*: we integrate selected knowledge tuples into VizKC using an image editor. For utilization, we employ a multi-image MLLM to solve the VizKC-enhanced ISR task.

including entity attributes, we use image captions as visually-based knowledge text. Image captioning [8], [53] aims to automatically explain an image's content in words. A basic caption normally describes only the common objects in an image in a generic manner, while an enhanced caption with real-world knowledge or entity details is often the key to understanding the image. Thus, we generate enhanced image captions.

Concretely, to generate image captions enhanced with real-world knowledge CAP_{wk} (step 9 in Alg.1), we utilize K-Replay [54] which consists of a knowledge predictor that identifies named entities such as "Lionel Messi" and a knowledge distiller that alleviates knowledge hallucination. Meanwhile, to generate image captions enhanced with entity details CAP_{ed} (step 10), we employ TROPE [55] that adds more object-part details such as "a white jersey" to a base caption. For both captioners, we use a general prompt such as "Please generate a caption for the given image". Finally, we generate CAP via concatenation (step 11). Some examples of captions can be seen in Fig. 2.

Based on the generated captions, we construct textual knowledge triples (steps 12 $\sim$ 14). First, a *knowledge generator* prompts the LLM to extract knowledge tuples that are relevant to visual entities detected in the image. Second, to handle conflicts and noise, we introduce a *knowledge verifier* and a *knowledge ranker* after the generator. Specifically, the verifier recognizes and filters the erroneous triples, to promote the precision of generated knowledge. Subsequently, the ranker uses the question as a criterion for tuple selection, to improve the relevance of generated knowledge.

As a result, two categories of triples are generated: (1) $< E_h, ARe, E_t >$, where E_h is a detected visual entity and ARe/E_t is an attribute/value learned from CAP; and (2) $< E_h, LRe, E_t >$, where both E_h and E_t are detected visual entities and LRe is a linking relation learned from CAP. Regarding *the verifier*, we rely on existing criteria mined from open KGs within RuleHub [56]: (i) Format check, e.g., the triple <Cristiano Ronaldo, Lionel Messi, contest> in Fig. 3 does not conform to the correct format $< E_h, LRe, E_t >$, so it should be removed from KT; (ii) Conflict check, e.g., ensuring that a person's age is not a negative number. Regarding *the ranker*, we use SentenceBERT [57] to calculate a semantic similarity score for each candidate in KT_v and the question Q and then select the most relevant tuples, e.g., <jersey, color, blue> in Fig. 3 could be prioritized for inclusion because it is highly relevant to a keyword "jersey" in Q.

Regarding *the generator*, we employ a strong, medium-sized open-source LLM, deepseek-R1-8B [58], to extract knowledge triples KT from CAP using the prompt P_{kg} that specifies the target entity and the caption text given.

Knowledge Text: (We combine captions as knowledge text. See examples in Fig. 3)...

Instruction: Extract the most significant triples related to <Visual Entity> from the above knowledge text.

Output:KT

Algorithm 1 The VizKC-ISR Framework

Input: Input Images and Question $\{\{V_j\}_J, Q\}$.

Output: VizKCs and Answer $\{\{VizKC_j\}_J, A\}$.

Require: G_S denotes scene graph; $(E,R)_M$ is the set of visual entity regions (M in total); V_S denotes scene image via synthetizing G_S and $(E,R)_M$; CAP_{wk} and CAP_{ed} denote image captions enhanced with world knowledge and entity details; $KT/KT_v/K_N$ is the generated/verified/selected knowledge tuples (N in total); V_n is the n-th VizKC via iterative editing, $P_{kg}/P_{t2i}/P_{isr}$ is the prompt for knowledge generation/text-to-image/image-sequence reasoning.

1: # **Module-1 VizKC Generation**
2: **for** $1 \le j \le J$ **do**
3: $V \leftarrow V_j$
4: # **Step-1. See: Scene Perception**
5: $G_S \leftarrow$ `SceneGraphGenerator` (V)
6: $(E,R)_M \leftarrow$ `EntityRegionExtractor` (V)
7: $V_S \leftarrow$ `SceneImageGenerator` $(G_S, (E,R)_M)$
8: # **Step-2. Find: Knowledge Generation**
9: $CAP_{wk} \leftarrow$ `ImageCaptioner-WK` (V)
10: $CAP_{ed} \leftarrow$ `ImageCaptioner-ED` (V)
11: $CAP \leftarrow$ `Combiner` (CAP_{wk}, CAP_{ed})
12: $KT \leftarrow$ `TupleGenerator` (CAP, P_{kg})
13: $KT_v \leftarrow$ `KnowledgeVerifier` (KT)
14: $K_N \leftarrow$ `KnowledgeRanker`(KT_v, Q)
15: # **Step-3. Fuse: Image Editing**
16: $V_0 \leftarrow V_S$
17: **for** $1 \le n \le N$ **do**
18: $V_n \leftarrow$ `ImageEditor` (V_{n-1}, K_n, P_{t2i})
19: **end for**
20: $VizKC_j \leftarrow V_N$
21: **end for**
22: # **Module-2 VizKC Utilization**
23: $A \leftarrow$ `MultiImageMLLM` $(\{(V_j, VizKC_j)\}_J, Q, P_{isr})$
24: **return** $\{\{VizKC_j\}_J, A\}$

3) Fuse-Image Editing:

Pioneering studies [59], [60] empower LLMs with the ability to generate high-quality images from textual prompts. At its core, our VizKC-ISR uses SEED-X [61] to integrate relevant knowledge items K_N into the initial VizKC (steps 16-19 in Alg.1), i.e., image synthesis. To use SEED-X, we design the prompt template P_{t2i}, which includes task instructions and image editing requirements as follows.

**** Adding Entity Attribute Knowledge ****

Instruction: Strictly ensure that other elements of the image remain unchanged, adding the current knowledge $< (E,R)$, attribute, value $>$ to the input image V_{n-1}.

Image Editing Requirements: Add a box containing value next to (E,R) and then point a purple arrow, attached the <attribute> name, from (E,R) to that box.

Output:V_n

**** Adding Entity Linking Knowledge ****

Instruction: Strictly ensure that other elements of the image remain unchanged, adding the current knowledge $< (E_h,R_h)$, relation, $(E_t,R_t) >$ to the input image V_{n-1}.

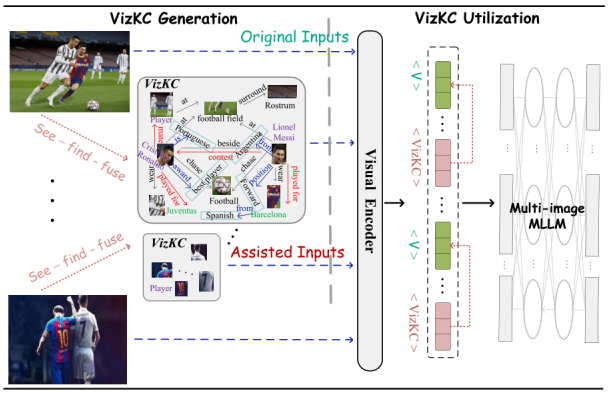


Fig. 4. The utilization of our VizKC in MLLMs.

Image Editing Requirements: Add a red arrow from (E_h,R_h) to (E_t,R_t) and attach the <relation> name to the arrow.

Output:V_n

It should be noted that we utilize an iterative process to edit knowledge items in the image one-by-one rather than all-in, because the latter probably results in cluttered images (e.g., overlapped bounding boxes, as shown in Fig. 9). In addition, to avoid the generated VizKC filled with too many region images, for each new entity (detected from the generation caption rather than from the original image), we introduce its entity name instead of an image that will be randomly generated. These entity names are added as OCR (Optical Character Recognition) within VizKC, with visual connections (e.g., adding a value box or relationship arrow) to relevant image regions, enabling entity names to be visually represented in the VizKC image.

V. VizKC Utilization

As Fig. 4 shows, we employ a multi-image MLLM, mPLUG-Owl3 [14], to solve VizKC-assisted ISR tasks (step 21 in Alg.1). mPLUG-Owl3 uses Siglip-400m [62] as visual encoder and Qwen2 [63] as language model. Given an enhanced ISR input $< \{(V_j, VizKC_j)\}_J, Q >$, mPLUG-Owl3 first extracts visual features of the image sequence $\{(V_j, VizKC_j)\}_J$ and then uses a linear projection to align the dimensions of the visual features with the language model. The projected visual features are denoted by $H_{img} = [I_{V_1}, I_{VizKC_1}, \ldots, I_{V_J}, I_{VizKC_J}]$. Based on the multi-model input, the corresponding text sequence is $S_{text} = [\{T_{img}\}_{2J}, Q]$, where T_{img} is a plain text $< |image| >$ to indicate the original location of an image. We feed the sequence S_{text} into a text encoder to obtain the textual features H_{text}. In the language model, mPLUG-Owl3 fuses H_{img} with H_{text} layer by layer through cross-attention operations in the Transformer architecture.

In summary, P_{isr} describes the enhanced visual input composed of the original images and their VizKC images.

Instruction: Here are several input images, each followed by an additional image which aims to convey visual knowledge (scene, attribute, link) as task clues, generated for you to make a better decision. Image 1: *IMG*. Assisted image 1: *IMG*. Image 2: *IMG*. Assisted image 2: *IMG*.....Based on your observations, answer the following question:Q.

Output:A

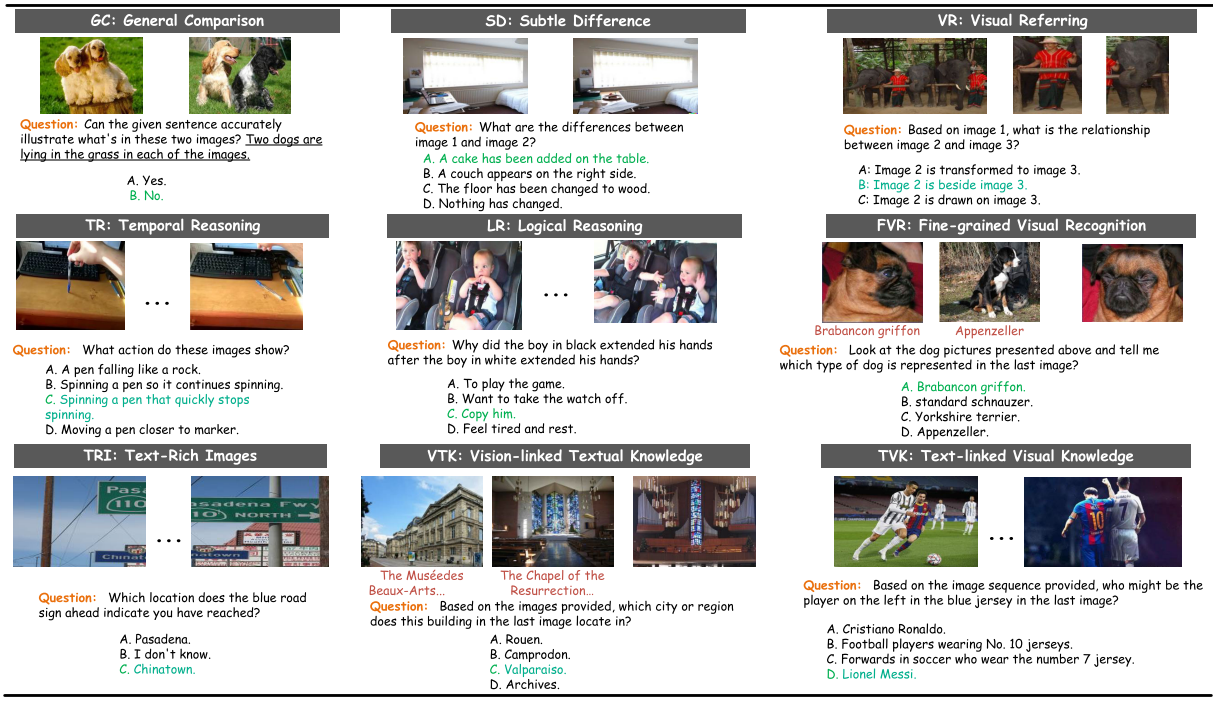


Fig. 5. Specific examples of nine ISR tasks studied in this paper.

VI. Experimental Setup

Datasets. We perform experiments covering nine ISR tasks (see Section II-B for detailed descriptions and Fig. 5 for specific examples): (1) General Comparison (GC) [36]; (2) Subtle Difference (SD) [37]; (3) Visual Referring (VR) [38]; (4) Temporal Reasoning (TR) [39]; (5) Logical Reasoning (LR) [40]; (6) Fine-grained Visual Recognition (FVR) [41]; (7) Text-Rich Images (TRI) [42]; (8) Vision-linked Textual Knowledge (VTK) [43]; (9) Text-linked Visual Knowledge (TVK) [44]. We observe that some tasks are more challenging, e.g., containing abstract reasoning objectives involving commonsense, intent, or emotion. For instance, the VTK task incorporates information searching questions that need to be answered by commonsense knowledge, e.g., "Which city or region does this building locate in?". The LR task involves the model using logical reasoning to analyze intentions or causality among objects in different images, e.g., "Why did the boy in black extend his hands after the boy in white extended his hands?". Our experimental results show that our approach is feasible for not only simple tasks where objects are visually distinguishable but also complex tasks involving commonsense, intent, or emotion.

Compared Methods. We compare VizKC-ISR with task-specific methods and MLLMs of similar size (less than 10B):

(1) The most competitive visual reasoning methods that improve visual encoders or rely on the LLMs, including Mini-Gemini [64], LLaVA-MoD [65], PICa [21], Prophet [22], and VCTP [23]. Specifically, LLaVA-MoD used knowledge distillation and mixture-of-experts (MoE). Mini-Gemini utilized an additional visual encoder for high-resolution refinement. PICa incorporated GPT-3 (175B) with few-shot learning. Prophet further refined the use of GPT-3 via heuristics. VCTP introduced visual Chain-of-Thought in Llama-2-70B.

(2) Open-source single-image MLLMs that improve the capabilities of LLMs by integrating multi-modal data, including Idefics1 (9B) [66], Qwen-VL-Chat (9B) [47], mPLUG-Owl (8B) [48], mPLUG-Owl2 (8B) [67], LLaVA-1.5 (8B) [46]. Specifically, Idefics1 (9B) was trained on a massive dataset of image-text documents from the web. Qwen-VL-Chat (9B) was built upon Qwen-VL, which used alignment methods and allowed for more adaptable interaction. mPLUG-Owl (8B) introduced a new training method that gave LLMs multi-modal skills by modularly training a base LLM. mPLUG-Owl2 (8B) used shared modules to help modalities work together and added a module that adjusted to different modalities, keeping their unique qualities. LLaVA used GPT-4 to produce text-image instruction-following data. LLaVA-1.5 (8B) demonstrated stronger performance than LLaVA by using CLIP-ViT.

(3) Open-source multi-image MLLMs that are pre-trained on multi-image data, including Mantis (8B) [16], Idefics2 (8B) [15], and mPLUG-Owl3 (8B) [14]. Specifically, Idefics2 (8B) performed a multi-stage pre-training using interleaved image-text data. Mantis performed instruction tuning on academic-level data. Integrating pre-training and instruction tuning, mPLUG-Owl3 improved longer-form image understanding.

Evaluation Metric. We use the VQA accuracy [68], the ratio of the number of correct answers to the total sample number.

Implementation Details. Our method only involves the interaction and inference of multiple LLMs/MLLMs and does not require retraining on new datasets. To use the

TABLE I

PERFORMANCE COMPARISON ON NINE ISR TASKS. THE ACCURACY SCORE (%) IS REPORTED. WE USE BOLD TO MARK THE HIGHEST SCORE (OF ALL) AND UNDERLINE TO MARK THE HIGHEST SCORE (EXCEPT FOR OURS) PER EACH TASK. "AVG." INDICATES THE AVERAGED SCORE ON ALL TASKS

Methods	Multi-Grained Comparison		Vision-Linked Reasoning					External-Knowledge Seeking		Avg.
	GC	SD	VR	TR	LR	FVR	TRI	VTK	TVK	
(1) Competitive Task-specific Methods (LLM-based)										
Mini-Gemini (enhancing visual encoder) [64]	19.4	20.5	16.3	19.9	18.5	20.6	19.5	12.9	17.5	18.3
LLaVA-MoD (using knowledge distillation) [65]	20.8	17.6	17.4	23.6	21.2	2.4	19.6	12.0	18.2	19.2
PICa (calling GPT3 API) [21]	13.2	7.7	10.4	12.6	14.5	9.9	10.6	13.5	13.7	11.8
Prophet (calling GPT3 API) [22]	28.7	26.5	17.3	20.2	28.4	30.2	23.5	16.2	18.5	23.3
VCTP (visual CoT & training Llama-2) [23]	22.6	19.2	18.1	17.5	20.6	31.5	24.3	14.2	18.9	20.8
(2) Open-source Single-image MLLMs										
Idefics1 (9B) [66]	37.4	32.5	20.7	22.3	30.7	37.1	29.6	19.5	20.8	27.8
Qwen-VL-Chat (9B) [47]	45.9	22.5	16.3	27.5	36.8	58.8	35.9	22.9	18.1	31.6
mPLUG-Owl (8B) [48]	19.1	4.0	21.7	8.0	29.2	17.3	12.1	14.9	20.6	16.3
mPLUG-Owl2 (8B) [67]	64.2	40.1	35.6	30.7	41.3	13.3	39.0	17.0	25.6	34.1
LLaVA-1.5 (8B) [46]	40.6	14.9	24.1	30.1	44.8	18.2	25.8	16.7	26.3	26.8
(3) Open-source Multi-image MLLMs										
Mantis (8B) [16]	83.0	54.1	37.6	45.5	63.4	16.4	37.7	26.4	41.7	45.1
Idefics2 (8B) [15]	83.1	49.7	32.6	44.8	56.4	42.4	43.9	25.6	39.0	46.4
mPLUG-Owl3 (8B) [14]	86.4	70.1	33.0	46.8	67.2	76.4	50.1	31.1	48.8	56.7
(4) ***Our Method*** *(Implemented with Different Multi-image MLLMs)*										
VizKC-ISR (Mantis-8B)	84.9	56.5	44.6	49.0	71.6	24.8	42.5	38.6	45.0	50.8
VizKC-ISR (Idefics2-8B)	85.4	53.0	40.4	48.7	65.5	45.2	47.3	36.7	45.5	52.0
VizKC-ISR (mPLUG-Owl3-8B)	**88.6**	**72.5**	**48.6**	**51.3**	**77.7**	**78.9**	**54.3**	**40.7**	**55.2**	**63.1**

LLMs/MLLMs, we follow their input/output formats and design effective prompt strategies. Our method is parameter-free, and the tools used in each stage are open-source. More implementation details are: (1) In the *See* stage, we set the maximum number of detected visual entities to 8 when using HiKER-SGG (as used in the original paper); (2) In the *Find* stage, we set the largest number of generated triples for a specific entity to be 5 and that for a specific entity-entity pair to 3; (3) In the *Fuse* stage, after a series of parameter test (See Table XIII), we set the largest number of knowledge tuples added to VizKC to 16 while the number of tuples edited in each iteration to 1; (4) We keep the size of VizKC same with that of a raw image, preventing a large increase in memory workload. We performed all experiments on 2 NVIDIA Tesla A100 40G.

VII. RESULTS AND ANALYSIS

A. Main Results

In this section, we report and analyze the main results. We reproduce all other methods based on their released codes.

VizKC-ISR can be implemented with different MLLMs. As shown in Table I, VizKC-ISR can be implemented with multiple multi-image MLLMs (Mantis, Idefics2, mPLUG-Owl3) and consistently improve each base model, suggesting the adaptability and model-agnosticism of our proposal. Because VizKC-ISR with mPLUG-Owl3 can achieve the best results (average score, plus all scores for subtasks), we use mPLUG-Owl3 as the default MLLM in the utilization module.

VizKC-ISR exhibits better performance than all other methods. Through the results in Table I, we have several valuable observations: (i) VizKC-ISR significantly outperforms the top visual reasoning methods, e.g., VizKC-ISR achieves an average score of 63.1% on nine tasks, surpassing Prophet, which calls GPT3, and VCTP, which fine-tunes Llama-2, by 39.8 and 42.3 absolute points, respectively. (ii) Qwen-VL-Chat performs best among single-image MLLMs, e.g., exceeding LLaVA-1.5 by 4.8. (iii) However, as a whole, single-image MLLMs perform significantly inferior to multi-image MLLMs due to the lack of pre-training or fine-tuning on specific multi-image data. Particularly, mPLUG-Owl3 performs better than mPLUG-Owl and mPLUG-Owl2 and also better than other MLLM families such as Idefics. (iv) Our VizKC-ISR method performs best among all methods.

Our method is effective across distinct or challenging tasks. Observed in Table I, VizKC-ISR achieves the highest score in each of nine tasks, surpassing the second-best model mPLUG-Owl3 (second-best in eight out of nine tasks) by 2.2, 2.4, 15.6, 4.5, 10.5, 2.5, 4.2, 9.6, and 6.4 absolute points in order. This shows that our approach is feasible for not only simple ISR tasks (e.g., GC and SD) where objects are visually distinguishable but also more complex reasoning tasks (e.g., VR and LR) involving commonsense knowledge, intent, or emotion. Compared to mPLUG-Owl3, the top three tasks where VizKC-ISR achieves the greatest performance gain are VR (requiring referring reasoning based on a global image), LR (requiring logical reasoning across multiple images) and VTK (requiring entity linking between relevant images), which justifies the effectiveness of our model in enhancing image reasoning for MLLMs. Furthermore, we calculate the average score over all tasks within the same category (MGC, VLR, EKS) mentioned in Section II-B. The results in Fig. 6 show that our approach achieves a small improvement in simple task groups such as MGC where visual entities can be directly compared, while a large increase in complex task groups such as VLR and EKS where abstract reasoning or external knowledge is necessary.

B. Cost Analysis and Efficiency Comparison

The storage/time overload to address additional visual input is a modest increase. Since each VizKC is added as a new image input, the input sequence doubles in size. We first provide detailed and quantitative comparisons of the total

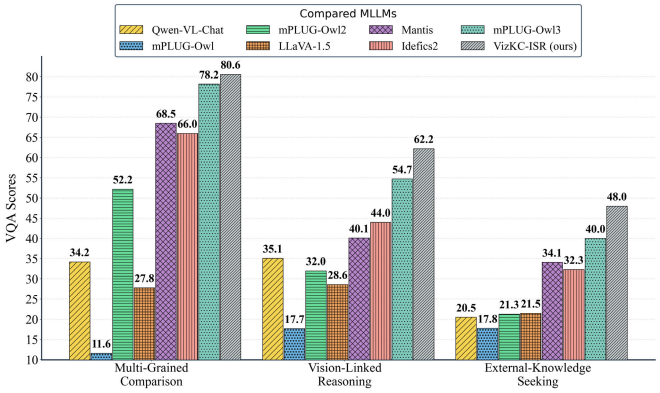


Fig. 6. Performance comparison on three task categories (MGC, VLR, EKS).

TABLE II

COMPARISON OF COMPUTATIONAL COST. (1) STORAGE: THE AVERAGED NUMBER AND SIZE (MB) OF THE VIZKC IMAGES PER TASK; (2) TIME: THE TOTAL INFERENCE TIME (HOURS) PER TASK

Task	GC	SD	VR	TR	LR	FVR	TRI	VTK	TVK	Avg.
(1) Storage Cost										
Avg. #Image	2	2	3	8	8	5	4	5	4	4.6
Avg. size (MB)	1.83	1.54	2.02	1.70	1.65	2.12	1.47	1.39	1.74	1.72
(2) Total Inference Time (hours)										
mPLUG-Owl3	3.1	3.3	2.4	2.6	2.9	2.3	2.0	1.8	1.2	2.4
Mantis	3.4	3.6	2.8	2.8	3.3	2.8	2.7	1.9	1.6	2.8
Idefics2	3.5	3.4	2.7	2.9	3.5	3.5	2.8	2.3	1.7	2.9
GPT-4o	10.8	10.5	8.2	8.3	9.4	8.6	6.9	6.3	6.2	8.4
VizKC-ISR	3.2	3.5	2.6	3.1	3.3	2.4	2.2	1.8	1.3	2.6

inference time and resource usage of VizKC-ISR compared to multiple SOTA approaches, including mPLUG-Owl3, Mantis, Idefics2, and GPT-4o. We also report the averaged number and size (Mb) of the VizKC images per task, showing that most tasks contain longer image sequence input. The findings in Table II are: (1) *Storage:* Our method does not present a significant storage burden. For example, the average size of VizKC on all tasks is 1.72 MB. (2) *Time:* mPLUG-Owl3 has the least inference time (2.4 hours), followed by our method (2.6 hours), while GPT-4o took much longer (8.4 hours) than all others. By using lightweight components, our method achieves high performance with significantly low computational cost.

We then provide the average end-to-end generation time per VizKC image to evaluate practical deployment feasibility.

(1) **Per-VizKC Time Breakdown by Processing Stage:** We further present a detailed breakdown of the time (seconds) required for each stage in the VizKC generation pipeline in Table III. The results reveal that the Fuse phase dominates the generation time (73.2% of total), primarily due to the 16 diffusion iterations in the SEED-X model. This analysis provides transparency into where computational resources are allocated and identifies potential optimization opportunities.

(2) **Standard vs. Fast Version Comparison:** Building on the time breakdown analysis, we develop a fast version of our framework to address deployment feasibility concerns. The comparison in Table IV demonstrates the trade-offs between computational efficiency and performance accuracy. The fast version achieves meaningful improvements through several key optimizations: (i) Simplified Architecture: The fast version reduces the number of components from 6 to 2 by using seed-x-7b for both See and Fuse phases, and deepseek-7b for the Find phase. (ii) Efficiency Gain: It achieves a 28% speedup (1.28×) compared to the standard version, with generation time reduced from 4.1s to 3.2s per VizKC. (iii) Performance Trade-off: Accuracy decreases slightly by 1.9% (from 63.1% to 61.9%), but remains significantly higher than the baseline (56.7%). (iv) Practical Implication: The fast version offers a balanced compromise for scenarios where processing speed is critical, while maintaining competitive accuracy.

TABLE III

THE BREAKDOWN OF AVERAGE END-TO-END GENERATION TIME PER VIZKC IMAGE (SECONDS)

Processing Stage	Components and Functions	Time
See Phase	HiKER-SGG (scene graph) + GLEE (entity extraction)	0.5
Find Phase	K-Replay + TROPE + deepseek-R1 (knowledge extraction)	0.6
Fuse Phase	SEED-X-Edit (16-iteration image editing)	3.0
Total	the complete See-Find-Fuse pipeline	4.1

TABLE IV

COMPARISON OF THE STANDARD & FAST VERSIONS OF VIZKC-ISR AND MPLUG-OWL3 (BASELINE)

Version	# Component	Underlying Components Used
Standard	6	HiKER-SGG, GLEE, K-Replay, TROPE, deepseek-R1, SEED-X
Fast	2	SEED-X-7b (See & Fuse Phases), deepseek-R1-7b (Find Phase)
Baseline	-	-
Version	**VizKC Gen-Time**	**Average ISR Accuracy (%)**
Standard	4.1s	63.1
Fast	3.2s	61.9
Baseline	-	56.7

(3) **Task-Level Total Time Analysis:** To provide a complete picture of the computational requirements, we extend our analysis to the task level, showing how the per-image times scale to complete tasks and comparing with state-of-the-art alternatives. The task-level analysis in Table V demonstrates that while VizKC generation adds computational overhead, the complete pipeline remains competitive. The fast version reduces the average total time from 6.9 to 5.9 hours (14.5% reduction) while maintaining significant accuracy advantages over the baseline. Compared to GPT-4o, our method shows competitive efficiency and specialized capabilities for ISR tasks.

(4) **Modularity and Reproducibility:** The VizKC framework is designed with strict modularity. Each stage (See, Find, Fuse) operates independently with well-defined interfaces. This design not only localizes errors, but also allows components to be updated or replaced individually, significantly reducing integration complexity and enhancing long-term maintainability. The complete code, configuration files, and detailed documentation are available on GitHub to ensure complete reproducibility.

C. Error and Reliability Analysis

Following [30], [69], [70], and [71], we perform a comprehensive quantitative analysis on how these early failures

TABLE V
COMPARISON OF TIME EFFICIENCY ON 2 X A100 GPUS (HOURS)

ISR Task	VizKC-ISR (ours)			mPLUG-Owl3	GPT-4o
	VizKC Gen-Time	Infer-Time	Total Time	Only Infer-Time	Only Infer-Time
GC	5.8	3.2	9.0	3.1	10.8
SD	5.9	3.5	9.4	3.3	10.5
VR	5.0	2.6	7.6	2.4	8.2
TR	4.7	3.1	7.8	2.6	8.3
LR	5.4	3.3	8.7	2.9	9.4
FVR	3.9	2.4	6.3	2.3	8.6
TRI	3.4	2.2	5.6	2.0	6.9
VTK	2.7	1.8	4.5	1.8	6.3
TVK	2.2	1.3	3.5	1.2	6.2
Avg.	**4.3**	**2.6**	**6.9**	**2.4**	**8.4**
Avg. (Fast)	**3.3**↓	**2.6**	**5.9**↓	-	-

negatively impact the VizKC-ISR accuracy across nine ISR tasks. This analysis is crucial for understanding the robustness of the cascaded system and identifying vulnerable components. The errors are categorized into three phases—See (scene perception), Find (knowledge generation), and Fuse (image editing)—with each phase involving specific error types injected at predefined rates. Since there is no ground-truth for the results from middle stages, we artificially inject incorrect entity labels or relationship connections into VizKC, and observe the decline in final ISR accuracy. The error rate (η_{err}) is defined as the proportion of knowledge tuples in which a specific type of error is intentionally injected into the pipeline. It quantifies the intensity of simulated component failures. The propagation rate R_{prop}, calculated as the relative accuracy drop from the base model (e.g., $(A_{\text{base}} - A_{\text{error-injected}})/A_{\text{base}}$), serves as a normalized metric to evaluate error impact. Below, we elaborate on the implementation details of each error type.

1. See Phase Errors

Relation Modification (S1): (i) Implementation Process: During scene graph generation using HiKER-SGG, 20% of spatial relations (e.g., "beside", "above") are randomly modified to incorrect but plausible alternatives. For instance, "person beside car" is altered to "person above car". This is achieved by perturbing the relation predicates in the scene graph triplets. (ii) Mechanism of Impact: Incorrect spatial relations mislead the downstream reasoning process, especially in tasks requiring precise spatial understanding (e.g., VR). The error propagates by corrupting the structural integrity of the scene representation. **Entity Deletion (S2):** (i) Implementation Process: After entity detection using GLEE, 20% of entities with the lowest confidence scores are removed from the output. This simulates missed detections in challenging scenarios (e.g., occluded objects). (ii) Mechanism of Impact: Entity loss leads to incomplete visual knowledge, affecting tasks reliant on object-level details (e.g., GC). The pipeline fails to reason about missing entities, causing cascading errors.

2. Find Phase Errors

False Object (F1): (i) Implementation Process: 10% of the knowledge tuples generated by the LLM are replaced with plausible but image-irrelevant objects. For example, for an image containing a "red car", a tuple like <bus, color, red> is injected. The injection is randomized across samples. (ii) Mechanism of Impact: Introduces semantic noise that misdirects the reasoning process. The LLM's knowledge verification mechanism may partially mitigate this, but persistent errors degrade performance. **Attribute Tampering (F2):** (i) Implementation Process: In 20% of knowledge tuples, attribute values (e.g., color, size) are altered to incorrect values. For instance, <building, height, tall> is changed to <building, height, short>. Attributes are selected based on their frequency in the dataset. (ii) Mechanism of Impact: Distorts entity characteristics, leading to inconsistencies in visual-textual alignment. This affects tasks requiring fine-grained attribute matching (e.g., FVR).

3. Fuse Phase Errors

Localization Bias (U1): (i) Implementation Process: During image editing with SEED-X, the bounding boxes of entities are shifted by 10% of their width/height in a random direction. This misaligns visual elements (e.g., arrows, labels) from their correct positions. (ii) Mechanism of Impact: Misalignment disrupts spatial coherence, confusing the visual encoder. Tasks relying on precise localization (e.g., TR) are most affected. Data Insights: This error has the lowest average propagation rate (1.9%), demonstrating the robustness of the visual encoder to minor positional deviations. **Visual Overlap (U2):** (i) Implementation Process: Intentional overlap is created between 20% of visual elements (e.g., text labels, bounding boxes) during image synthesis. Overlap areas are calculated based on bounding box intersections. (ii) Mechanism of Impact: Overlap occludes critical visual information, impairing the model's ability to parse integrated knowledge. This mimics real-world clutter scenarios.

Based on the comprehensive error impact analysis presented in Table VI, several key findings emerge regarding the robustness of the VizKC framework: **(1) Phase-wise Error Propagation Hierarchy:** The data reveals a clear hierarchy in error susceptibility across pipeline stages. See phase errors exhibit the highest propagation rates (3.93-4.47%), demonstrating that early visual processing inaccuracies have the most substantial cascading effects. Fuse phase errors show moderate impact (2.87-3.38%), while Find phase errors prove least detrimental (1.01-1.55%), validating the effectiveness of built-in knowledge verification mechanisms. **(2) Task-Specific Vulnerability Patterns:** VR (Visual Referring) tasks display exceptional sensitivity to errors, particularly in See phase scenarios (up to 11.90% propagation), attributable to their heavy reliance on precise spatial relationships and entity localization. Conversely, SD (Subtle Difference) tasks demonstrate remarkable robustness (as low as 0.14% propagation), likely due to their dependence on comparative analysis rather than absolute visual features. **(3) Knowledge Processing Resilience:** The minimal impact of Find phase errors (average 1.28% propagation) underscores the efficacy of the knowledge verifier and ranker components. These mechanisms successfully filter erroneous knowledge tuples while preserving valid information, effectively containing errors that originate during knowledge generation. **(4) Error Accumulation Dynamics:** The progressive attenuation of error effects from See→ Fuse→ Find phases demonstrates that while early errors propagate substantially, later-stage processing components possess inherent error-correction capabilities that mitigate downstream impacts, with the knowledge processing stage showing

TABLE VI

RESULTS OF ERROR PROPAGATION ANALYSIS ON NINE ISR TASKS. WE CALCULATE PROPAGATION RATE WITH RESPECT TO VIZKC-ISR (THE DEFAULT MODEL). WE HIGHLIGHT THE COLUMN-WISE HIGHEST AND LOWEST VALUES

Error Type	Error Rate (η_{err})	Task Accuracy (%)									Average
		GC	SD	VR	TR	LR	FVR	TRI	VTK	TVK	
mPLUG-Owl3 (baseline)	-	86.4	70.1	33.0	46.8	67.2	76.4	50.1	31.1	48.8	56.7
VizKC-ISR (default)	-	88.6	72.5	48.6	51.3	77.7	78.9	54.3	40.7	55.2	63.1
See Phase Errors											
-Relation Modification (S1)	20%	87.6	72.0	42.8	49.5	73.0	78.0	53.0	37.0	52.8	60.6
propagation rate (R_{prop})		*1.13%*	*0.69%*	*11.90%*	*3.51%*	*6.05%*	*1.14%*	*2.39%*	*9.09%*	*4.35%*	*4.47%*
-Entity Deletion (S2)	20%	87.8	72.2	43.2	49.8	73.5	78.2	53.3	37.5	53.0	60.9
propagation rate (R_{prop})		*0.90%*	*0.41%*	*11.12%*	*2.92%*	*5.41%*	*0.89%*	*1.84%*	*7.86%*	*3.99%*	*3.93%*
Find Phase Errors											
-False Object (F1)	10%	88.3	72.4	47.0	50.8	77.0	78.7	54.0	40.0	54.7	62.5
propagation rate (R_{prop})		*0.34%*	*0.14%*	*3.29%*	*0.97%*	*0.90%*	*0.25%*	*0.55%*	*1.72%*	*0.91%*	*1.01%*
-Attribute Tampering (F2)	20%	88.1	72.3	46.5	50.5	76.5	78.5	53.8	39.5	54.5	62.2
propagation rate (R_{prop})		*0.56%*	*0.28%*	*4.32%*	*1.56%*	*1.54%*	*0.51%*	*0.92%*	*2.95%*	*1.27%*	*1.55%*
Fuse Phase Errors											
-Localization Bias (U1)	20%	88.0	72.1	44.0	50.2	75.5	78.3	53.5	38.5	53.8	61.5
propagation rate (R_{prop})		*0.68%*	*0.55%*	*9.47%*	*2.14%*	*2.83%*	*0.76%*	*1.47%*	*5.41%*	*2.54%*	*2.87%*
-Visual Overlap (U2)	20%	87.9	72.0	43.5	50.0	75.0	78.2	53.3	38.0	53.5	61.3
propagation rate (R_{prop})		*0.79%*	*0.69%*	*10.49%*	*2.53%*	*3.47%*	*0.89%*	*1.84%*	*6.63%*	*3.08%*	*3.38%*

particularly strong error containment properties. **(5) Consistent Performance Superiority Over Baseline:** Crucially, even under maximum error conditions, VizKC maintains significant performance advantages over the baseline system. The error-injected VizKC achieves average accuracies of 60.6-62.5% across all phases, substantially outperforming the baseline's 56.7% by 3.9-5.8 percentage points. **(6) Error Margin Buffer Analysis:** The substantial performance gap between error-affected VizKC (minimum 60.6%) and baseline (56.7%) represents a 3.9% safety margin, indicating that the system can withstand considerable error accumulation while still delivering superior results. This robustness is particularly valuable for real-world deployments where component failures are inevitable.

These findings collectively validate the VizKC architecture's layered defense mechanism against error propagation while highlighting its consistent performance advantages over conventional approaches, even when facing significant error conditions.

D. Ablation Studies

To verify the contribution of each component, we evaluate six ablation variants of our method: (i) Only ISR: we use the MLLM baseline to solve the original task; (ii) G_S+ ISR: we generate a scene graph G_S and append all textual triples in G_S at the end of the question; (iii) ($G_S \rightarrow V_S$) + ISR: we further convert G_S to V_S and then use V_S as an assisted visual input; (iv) K_N+ ISR: we generate knowledge triples K_N from image captions and append them at the end of the question; (v) G_S + K_N+ ISR: we generate both G_S and K_N and then append them in order at the end of the question; (vi) ($G_S \rightarrow V_S$) + ($K_N \rightarrow V_S$) + ISR: we implement all steps of VizKC-ISR (generating G_S and then converting to V_S, generating K_N and then editing into V_S), thus generating *our full model*. Furthermore, in the setting (K_N+ ISR), we perform more fine-grained ablations such as removing (w/o) diverse caption parts or the verifier/ranker, to validate the effectiveness of key components in our knowledge generation. Notably, in the setting ($G_S + K_N$+ ISR), all knowledge is used in the form of text rather than images, which can show **the necessity of our image-generation decision**.

TABLE VII

ABLATION STUDIES ON MODELS CONSISTING OF DIFFERENT COMPONENTS. WE HIGHLIGHT THE MODALITY (TEXT OR IMAGE) OF KNOWLEDGE TUPLES

Our models (all use mPLUG-Owl3 as the ISR solver)	SD	VR	VTK
VizKC-ISR (the full model)	**72.5**	**48.6**	**40.7**
Ablation variants of VizKC-ISR[1]			
- Only ISR	70.1	33.0	31.1
- G_S (text) + ISR	71.2	33.8	34.2
- $G_S \rightarrow V_S$ (image) + ISR	71.6	35.4	35.3
- K_N (text) + ISR	71.8	36.7	37.6
-w/o Entity-detail Caption	70.7	34.3	33.5
-w/o World-knowledge Caption	70.9	35.4	34.4
-w/o Knowledge Ranker	70.9	36.0	35.2
-w/o Knowledge Verifier	71.2	36.2	35.7
- G_S (text) + K_N (text) + ISR	72.0	39.6	38.2
- $G_S \rightarrow V_S$ (image) + $K_N \rightarrow V_S$ (image) + ISR	**72.5**	**48.6**	**40.7**

[1] $G_S \rightarrow V_S$ denotes that G_S is converted to V_S (via graph visualization). $K_N \rightarrow V_S$ denotes that K_N is added to V_S (via image editing).

Due to task relevance and experimental scale, we only report the results of ablation studies on three representative tasks in Table VII and observe that: (1) Only using K_N achieves better results than only using G_S, showing the relative importance of capturing entity-related knowledge from multi-style image captions. However, combining G_S and K_N together can further improve the model's performance. (2) Generating V_S based on G_S results in a performance gain as expected, indicating the benefits of adding visual input to a multi-image model. (3) The results of ablation studies on caption parts and knowledge components (marked in gray) show that the model performance can degrade if such verification and selection mechanisms are removed. (4) The complete model outperforms all ablation variants, showing that **each component has a unique contribution to the overall performance**.

E. The Necessity and Advantages of VizKC Images

We provide comprehensive justification for the design choice of "image-rendered knowledge". Our core idea is

TABLE VIII

COMPARISON OF PLAIN-TEXT BASELINES THAT FORMAT KNOWLEDGE TUPLES IN VARIOUS TEXT PROMPTS (JSON, LIST, PARAGRAPH)

Our models (all use mPLUG-Owl3 as the ISR solver)	SD	VR	VTK
VizKC-ISR (the full model)	**72.5**	**48.6**	**40.7**
Ablation variants of VizKC-ISR (all use text prompts)[1]			
- K_N (text) + ISR			
- json	71.8	36.7	37.6
- list	71.8	36.7	37.6
- paragraph	71.8	40.2	38.1
- G_S (text) + K_N (text) + ISR			
- json	72.0	39.6	38.2
- list	72.0	39.5	38.2
- paragraph	72.0	42.4	38.6

[1] (i) json: a json file that organizes all knowledge tuples in a structured form. (ii) list: a list of all tuples. (iii) paragraph: a coherent paragraph to depict the semantic of all tuples.

to represent and deliver multimodal knowledge using the visual modality itself, providing the MLLM with an intuitive, structured "knowledge snapshot". Instead of conventional approaches such as improving visual encoders or adding textual prompts, our method pioneers a unique path by compiling multimodal knowledge into a synthetic, visual image. This "knowledge-as-an-image" paradigm not only aligns perfectly with the native multi-image input interface of MLLMs—avoiding complex cross-modal fusion modules—but also offers superior interpretability and powerful spatial relationship modeling.

We summarize the advantages of this design choice as follows. (1) In the ISR tasks studied in this paper, the use of knowledge as images outperforms the use of knowledge as text prompts. We employ deepseek-R1 to convert the collection of knowledge tuples into (i) a json file; (ii) a symbol list; (iii) a coherent paragraph. Based on the results in Table VIII, we find that: (a) using coherent paragraphs slightly outperforms using json files or triple lists; (b) our default model (representing knowledge as images) can significantly outperform these text-prompt baselines. The main reason is that the VizKC images can provide the MLLM with an intuitive, structured knowledge snapshot that highlights significant content of the original input image, which is hard to achieve by using text prompts. (2) VizKC can present the most critical elements as needed. Table IX shows that "only using the VizKC images as visual input" performs better than "only using the original images as visual input". This directly demonstrates the inherent value of VizKC and underscores the superiority of the proposed method. (3) We also estimate the ratio of VizKC that provides answers directly or indirectly. For this, we select 100 correctly answered samples by introducing the VizKC images for each task. In Table X, we observe that the more challenging the task (e.g., with an accuracy less than 60%), the more likely the model will perform cross-image reasoning. (4) Our VizKC images become increasingly beneficial for performance improvement as the need for cross-image reasoning grows. To show this, we calculate the improvement as "the VizKC-ISR accuracy - the mPLUG-Owl3 accuracy". In Table XI, we observe that VizKC-ISR improves the performance of the baseline model better in more complex ISR scenarios—such as VR, TR, LR, VTK, and TVK— where more difficult questions, longer image sequence, or richer clues are often involved.

TABLE IX

COMPARISON OF USING THE ORIGINAL OR VIZKC IMAGES AS INPUT

Our method (using mPLUG-Owl3)	GC	SD	VR	TR	LR
- Only using the original images	86.4	70.1	33.0	46.8	67.2
- Only using the VizKC images	87.1	70.9	38.6	48.9	67.3
- Using both	**88.6**	**72.5**	**48.6**	**51.3**	**77.7**
Our method (using mPLUG-Owl3)	FVR	TRI	VTK	TVK	Avg.
- Only using the original images	76.4	50.1	31.1	48.8	56.7
- Only using the VizKC images	76.6	51.4	33.8	50.5	58.3
- Using both	**78.9**	**54.3**	**40.7**	**55.2**	**63.1**

TABLE X

STATISTICS ON WHETHER VIZKC DISPLAYS ANSWERS DIRECTLY (THE MODEL ONLY NEED TO LOOK UP THE TARGET VIZKC IMAGE) OR INDIRECTLY (THE MODEL NEED TO REASON ACROSS VIZKC IMAGES)

ISR Task	**Task Accuracy**	**Answer (directly)**	**Answer (indirectly)**
Task category: Multi-Grained Comparison			
GC	88.6	78 / 100	22 / 100
SD	72.5	69 / 100	31 / 100
Task category: Vision-Linked Reasoning			
VR	48.6	40 / 100	60 / 100
TR	51.3	23 / 100	77 / 100
LR	77.7	68 / 100	32 / 100
FVR	78.9	64 / 100	36 / 100
TRI	54.3	42 / 100	58 / 100
Task category: External-Knowledge Seeking			
VTK	40.7	26 / 100	74 / 100
TVK	55.2	35 / 100	65 / 100

TABLE XI

ACCURACY IMPROVEMENT OF OUR METHOD AGAINST THE BASELINE. WE RANK ALL TASKS BASED ON THEIR REASONING DIFFICULTY (FROM EASY TO HARD), WHICH IS INVERSELY PROPORTIONAL TO THE IMPROVEMENT

Task[2]	VizKC-ISR Accuracy	mPLUG-Owl3 Accuracy	Improvement[1]	Difficulty (1-9)
Category 1: Multi-Grained Comparison				
GC ♠	88.6	86.4	2.2	1
SD ♠	72.5	70.1	2.4	2
Category 2: Vision-Linked Reasoning				
VR ♢	48.6	33.0	15.6	9
TR	51.3	46.8	4.5	5
LR ♢	77.7	67.2	10.5	8
FVR ♠	78.9	76.4	2.5	3
TRI	54.3	50.1	4.2	4
Category 3: External-Knowledge Seeking				
VTK ♢	40.7	31.1	9.6	7
TVK	55.2	48.8	6.4	6

[1] We calculate the improvement using (the VizKC-ISR accuracy - the mPLUG-Owl3 accuracy).
[2] We use ♠/♢ to highlight the tasks that have a significantly low/high improvement (<3)/(>9).

F. Qualitative Results

We provide some qualitative analysis to understand (i) why the cross-image reasoning is necessary and (ii) how our approach works. As Fig. 7 shows, **our VizKC enjoys a good interpretability by providing a trace of knowledge clues with related visual concepts.** First, when provided only the last image (V_j), the model (mPLUG-Owl3) returned a wrong answer (Option B). The reason is very simple, because V_j only shows the backs of the players so that the model cannot see any of their facial features, thus failing to recognize the notable figures (such as "Messi"). This can be validated by "Man1" and "Man2" instead of "Messi" and "Ronaldo" in $VizKC_j$ (the VizKC image of V_j). Second, even when provided the whole image sequence from V_1 to V_j, the model still outputted the wrong option B. We infer that even though the model successfully identified "Messi" and "Ronaldo" (due to seeing their faces clearly) in the first image V_1, it still failed to infer that "the player on the left in the blue jersey"

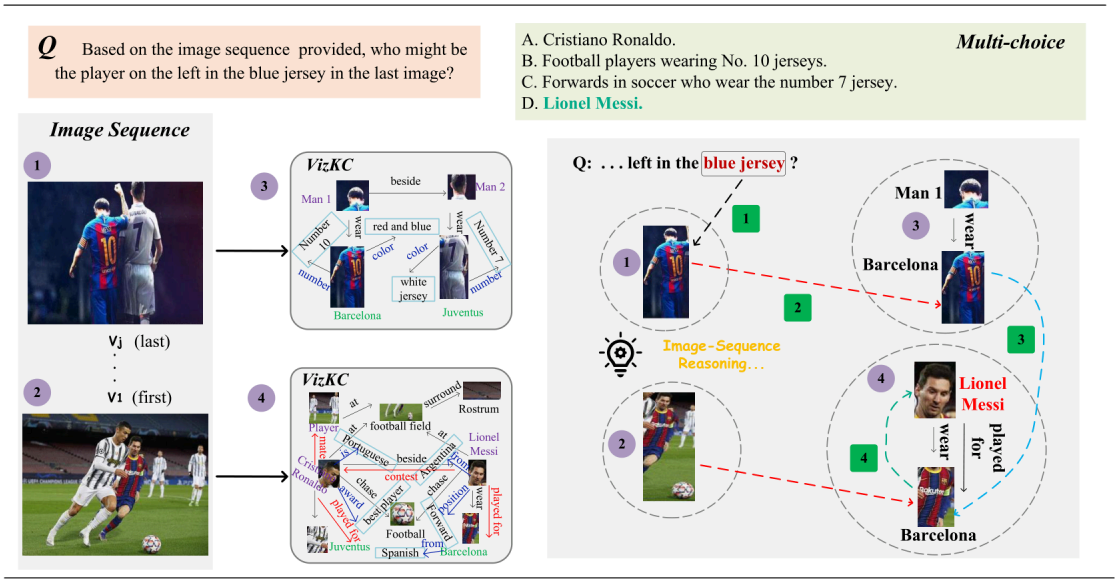


Fig. 7. A case study where our VizKC enjoys a good interpretability by providing a trace of knowledge clues with related visual concepts.

TABLE XII
RESULTS OF USING ALTERNATIVE MODELS IN VARIOUS GENERATION STEPS ACROSS NINE ISR TASKS

Steps in the See-Find-Fuse Pipeline	GC	SD	VR	TR	LR	FVR	TRI	VTK	TVK	Avg.
(1) Scene Graph Generation (SGG)										
Default: HiKER-SGG (456MB)	88.6	72.5	48.6	51.3	77.7	78.9	54.3	40.7	55.2	63.1
Alternative: qwen2.5-VL-7B	87.5	72.3	48.0	52.2	74.9	79.3	54.5	38.6	54.7	62.4
(2) Image Caption Generation										
Default: K-Replay+TROPE (total 2.4B)	88.6	72.5	48.6	51.3	77.7	78.9	54.3	40.7	55.2	63.1
Alternative: qwen2.5-VL-7B	88.2	69.9	47.1	48.4	77.0	78.2	53.0	37.4	52.1	61.2
(3) Knowledge Tuple Generation										
Default: deepseek-R1-8B	88.6	72.5	48.6	51.3	77.7	78.9	54.3	40.7	55.2	63.1
Alternative-1: OPT-66B	86.9	70.7	45.4	50.8	77.3	78.2	52.9	39.5	55.3	61.0
Alternative-2: GPT3-175B	88.6	72.9	48.7	51.8	78.3	78.4	54.5	41.0	55.6	63.3

in the last image V_j is exactly "Messi". The first two tests show that (i) the cross-image reasoning is necessary in this example. Subsequently, we continued to test the model by providing both the original image sequence $<V_1 \ldots V_j>$ and their corresponding VizKC sequence $<VizKC_1 \ldots VizKC_j>$ as joint visual input. As a result, we found that the model outputted the correct answer (Option D) in this VizKC-enhanced scenario. This shows that the VizKC images can provide the MLLM with an intuitive, structured "knowledge snapshot", which not only highlights key visual entities detected from the global image, but also adds explicit, structured knowledge to depict their spatial or attributable relationships. As Fig. 7 (the right square) shows, our method can provide a trace of knowledge clues with related visual concepts. For instance, the black lines stand for multi-modal links between the question and the query image, the red lines stand for visual links between the query image and its VizKC image, the blue lines stand for visual links between different VizKCs, and the green lines stand for visual links between distinct entities within a VizKC. Thus, the model can successfully associate "Man1" in $VizKC_j$ with 'Messi" in $VizKC_1$ and finally return a correct answer. The third test shows (ii) how our approach works, i.e., how the model is learning to perform cross-image reasoning.

Second, we show more case studies, particularly those that involve failure samples, in Fig. 8. These examples vary with diverse task categories (e.g., SD, VR, and TR) and distinct image-sequence lengths (e.g., there are 2, 3, or 4 input images). Case 1 describes a bad situation in which VizKC failed to capture the key clue due to visually unrecognizable objects (only a small part of the baby entity is shown in the lower left corner of V_1). As a result, our method generated the same VizKC for two different images V_1 and V_2. Case 2 reveals a good situation in which VizKC introduced the correct and relevant knowledge (<boy, touch, horse>) derived through the identification of scene relations and the generation of descriptive captions. Fortunately, both the MLLM baseline and our method predicted a correct answer. Case 3 depicts a better situation where VizKC represented a more reasonable rationale (e.g., "the boy in black is acting as the boy in white does") so that our method generated the correct answer ("Copy him") instead of the wrong answer ("To play the game") found by the MLLM baseline based only on its own knowledge. It should be noted that we did not observe a worse situation where the MLLM baseline provided an accurate answer, while our method incorrectly answered. We infer that the MLLM baseline can understand the distinctions between the main input images and the assisted input images, as stated

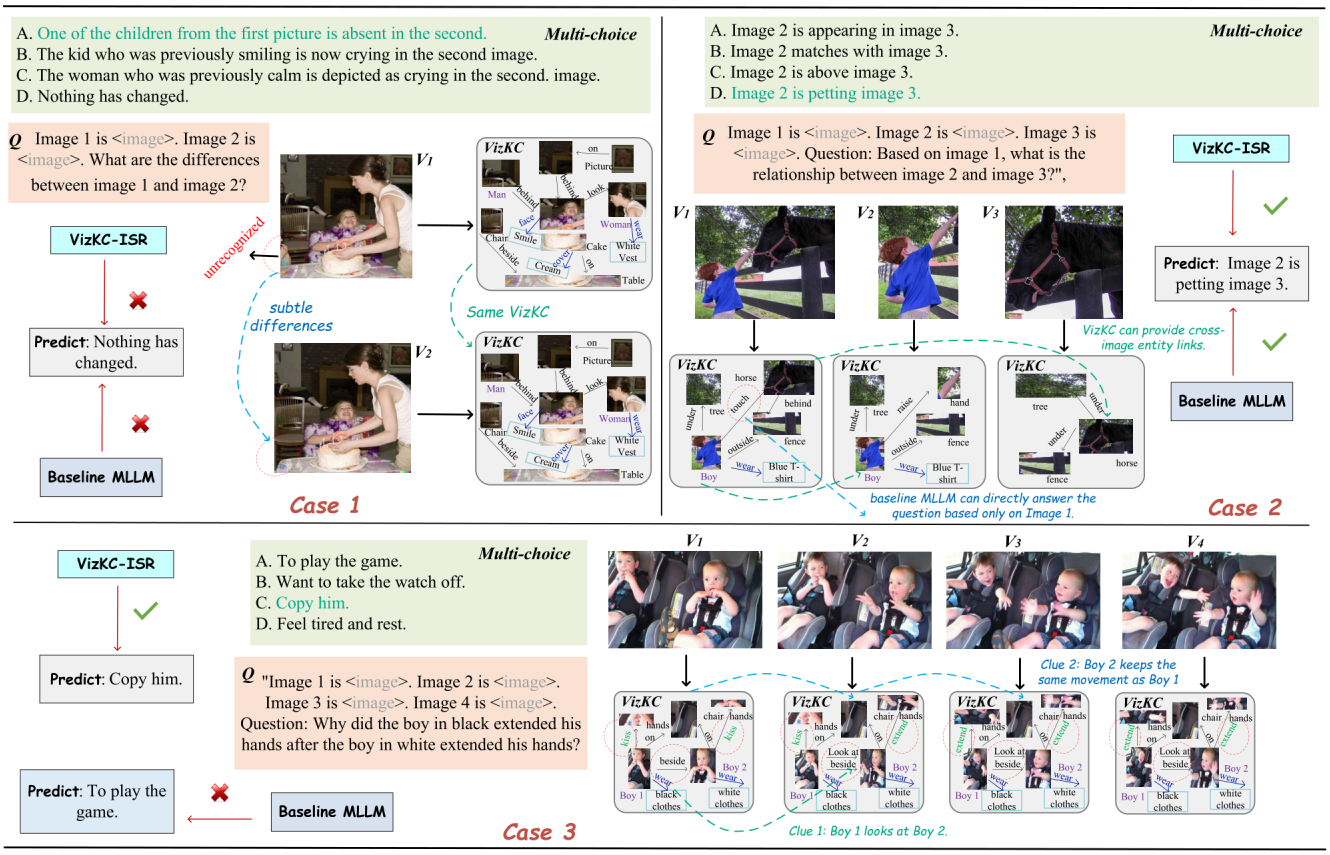


Fig. 8. More case studies with varying task categories and image-sequence lengths. Case 1: a bad situation where VizKC failed to perceive the subtle difference between V_1 and V_2 due to visually unrecognizable objects in V_1. Both the MLLM baseline (mPLUG-Owl3) and our method (VizKC-ISR) gave the wrong answer. Case 2: a good situation where VizKC successfully generated the correct and relevant knowledge (<boy, touch, horse>) sourced from V_1, thus guiding a correct answer to describe the relationship between V_2 and V_3. Fortunately, the MLLM baseline also answered correctly based only on its own knowledge. Case 3: a better situation where VizKC identified key knowledge clues such as <boy1, extend, hands> and <boy2, extend, hands> and captured the more reasonable rationale ("Boy2 looks at Boy1 and keeps the same movement as him") so that our method can correct the wrong answer provided by the baseline.

TABLE XIII
SUPER-PARAMETER TEST ON THE QUANTITY OF KNOWLEDGE TUPLES (#TOTAL-ADDED VS. #EACH-EDITED)

Added \ Edited	#*each*=1	#*each*=2	#*each*=4	#*each*=8
#*total*=4	34.2	33.9	31.2	-
#*total*=8	38.6	38.0	34.9	30.8
#*total*=16	**40.7**	39.2	36.2	32.5
#*total*=32	38.9	38.5	34.5	32.7

in the prompts for VizKC utilization (P_{isr}), i.e., the visual information from the main input images should be prioritized for inclusion.

VIII. IN-DEPTH ANALYSIS

A. Super-Parameter Test

We observe the change in model performance by introducing different quantities of knowledge items into our VizKC. Specifically, we set two variables: (1) the total number of knowledge items *#total-added* and (2) the number of knowledge items *#each-edited* edited at a time. We use the grid search method to examine each combination of *#total-added* and *#each-edited* and report the results in the VTK task in Table XIII. We find that: (i) Under the settings where *#each-edited* = 1, 2, or 4, the model's performance first increases and then decreases as *#total-added* grows, due to possible additions of irrelevant knowledge; (ii) However, in the setting where *#each-edited* = 8, the performance of the model continuously increases as *#total-added* increases, but its final results are still inferior to those with a smaller editing capacity. Finally, we select (*#total-added* = 16, *#each-edited* = 1) as our best setting.

B. Alternative Models

Our method presents an effective systemic framework by integrating several SOTA models. To better understand the contribution of this specific configuration, we perform an analysis on how sensitive the final performance is to the choice of these individual components. For this, we test alternative models that would affect the quality of the VizKC and, consequently, the accuracy of the downstream task, in three key generation steps. Specifically, (1) we use a larger model, qwen2.5-VL-7B, as an alternative scene graph generator; (2) we also use qwen2.5-VL-7B as an alternative image captioner; (3) we use two much larger models, OPT-66B [72] and GPT3-175B [73], as an alternative knowledge tuple generator, respectively. We present the results of using these alternative models in nine ISR tasks in Table XII, which

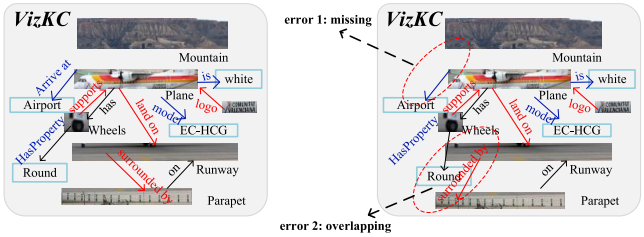


Fig. 9. Quality comparison. Left: the clear image is generated given *#each-edited* = 1. Right: the cluttered image is generated given *#each-edited* = 8.

are compared to the results of using default components. The findings are as follows: (i) for each intermediate generation step, our default model performs better than or on a par with the alternative model, showing the appropriateness of our current configurations. (ii) for SCC and image captioning, specialized and comparatively smaller models (e.g., HiKER-SGG) are more effective than general-purpose MLLMs (e.g., qwen2.5-VL-7B). This is because that HiKER-SGG can create bridging connections between a hierarchical knowledge graph and the initial scene graph, enabling it to accurately capture entity concepts, along with their spatial relationships. (iii) for knowledge tuple generation, although using GPT3-175B can obtain a slight improvement, deepseek-R1-8B enjoys extra advantages, e.g., free of charge and easy to deploy. Thus, we persist in using deepseek-R1-8B.

C. Qualitative Analysis on VizKC Quality

We provide a qualitative comparison of two similar VizKCs generated by assigning the same value to *#total-added* while adjusting different values to *#each-edited*, assuming that all other configurations are equal. As shown in Fig. 9, the VizKC visualization on the right appears cluttered and lacks a clear spatial organization, e.g., overlapping bounding boxes such as "Round" and "Surrounded by". The main reason is that we did not incorporate the explicit spatial organization instructions in the prompt P_{t2i}. The current instructions and editing requirements include: (1) what knowledge elements will be added into the image; (2) how to add an attribute box if the added element represents an entity detail; (3) how to add a relation edge if the added element represents an linking relation. To improve visualizations, we plan to generate images through LLM-generated intermediate code (Python or TikZ) [74], which incorporates explicit spatial layout instructions and then must be compiled into images.

D. Comparisons With Closed-Source MLLMs

We compare with mainstream large-scale closed-source MLLMs (e.g., GPT-4o). We perform this experiment across all nine ISR benchmark. For more comparisons, we also show the results of our baseline model, mPLUG-Owl3. Observed in Table XIV, GPT-4o performs much better than mPLUG-Owl3, particularly in complex tasks such as VR and VTK. However, our method, which improves mPLUG-Owl3 by extracting visual knowledge and making it more explicit in the auxiliary images (VizKC), can surpass GPT-4o in all the datasets tested.

TABLE XIV
PERFORMANCE COMPARISON WITH GPT-4O ON NINE ISR TASKS

ISR Tasks	mPLUG-Owl3	GPT-4o	VizKC-ISR
Multi-Grained Comparison			
GC	86.4	87.2	**88.6**
SD	70.1	72.0	**72.5**
Vision-Linked Reasoning			
VR	33.0	45.9	**48.6**
TR	46.8	47.3	**51.3**
LR	67.2	70.5	**77.7**
FVR	76.4	77.0	**78.9**
TRI	50.1	50.0	**54.3**
External-Knowledge Seeking			
VTK	31.1	38.4	**40.7**
TVK	48.8	51.3	**55.2**
Avg.	56.7	59.9	63.1

TABLE XV
PERFORMANCE COMPARISON ON OTHER MULTIMODAL REASONING TASKS

Methods	GQA	VQAv2	InfoSeek	FM-IQA	IconQA
Competitive LLM-Based Visual Reasoning Methods					
PICa	47.5	46.9	16.9	59.2	57.6
Prophet	55.1	59.8	23.2	62.8	62.3
VCTP	50.4	52.7	21.4	64.3	62.0
Open-source MLLMs					
Qwen-VL-Chat	51.4	52.7	19.7	59.2	60.1
mPLUG-Owl	49.5	49.8	20.0	66.3	65.9
LLaVA-1.5	45.4	50.8	17.2	61.5	59.8
Mantis	50.7	58.6	21.2	67.2	64.9
Idefics2	49.8	54.3	21.8	65.4	68.4
mPLUG-Owl3	57.1	60.4	23.4	68.2	72.3
Our Method *(using mPLUG-Owl3)*					
VizKC-ISR	58.7	62.0	24.2	74.5	77.3

E. Effectiveness on More Multi-Modal Reasoning Tasks

We examine the generalization capabilities of our VizKC-ISR on other multi-modal reasoning tasks. Specifically, we use five datasets as follows. (1) The GQA dataset [75], which automatically generates diverse and semantically rich questions from scene graphs, emphasizing a fine-grained understanding of objects, attributes, relationships, and logical structures in images. (2) The VQAv2 dataset [76], which designs questions where correct answers cannot be inferred from language alone, forcing models to genuinely utilize visual content. (3) The InfoSeek dataset [43], which features questions that require fine-grained object recognition, localization, and commonsense reasoning. (4) The FM-IQA dataset [77], which covers 7 languages and is based on COCO images with professionally translated and culturally localized annotations. (5) The IconQA dataset [78], which focuses on understanding abstract diagrams—such as icons, flow charts, statistical diagrams, and geometric figures—rather than natural images. The results in Table XV show that **our method still outperforms all other compared methods** in these distinct domains, particularly on FM-IQA and IconQA. Compared to mPLUG-Owl3, our method achieves an improvement ratio of (74.5-68.2)/68.2 = 9.2% on FM-IQA and (77.3-72.3)/72.3 = 6.9% on IconQA.

IX. LIMITATIONS

Our work has three primary limitations that point to promising future research directions. (i) Risks of cascade errors. Each sub-module may introduce errors which can propagate through the pipeline. We have clarified the typical error types of key modules and provided the corresponding analysis. (ii) Increase in computational overhead and inference latency. Generating

a VizKC for each input image requires running multiple distinct models. To reduce computational cost and improve practicality, we will consider model compression techniques. (iii) Dependency on multi-image MLLM architectures. We use a strong multi-image MLLM (mPLUG-Owl3) that can effectively process the combined visual input. However, the generalizability of the approach is constrained by the current capabilities of multi-image MLLMs.

X. Conclusion

This paper proposes VizKC, an image-based knowledge representation that captures scene-specific and entity-related knowledge as task clues, and the corresponding framework VizKC-ISR, to address the core challenge of ISR in MLLMs. By generating an auxiliary visual clue image for each input, VizKC-ISR makes visual knowledge more explicit and accessible for ISR tasks. Experimental results show that VizKC-ISR outperforms strong baselines across nine ISR benchmarks.

Acknowledgment

The authors would like to thank the anonymous referees for their invaluable insights.

References

[1] W. Wang, Y. Yang, and Y. Pan, "Visual knowledge in the big model era: Retrospect and prospect," *Frontiers Inf. Technol. Electron. Eng.*, vol. 26, no. 1, pp. 1–19, Jan. 2025.

[2] Y. Zhao et al., "Beyond sight: Towards cognitive alignment in LVLM via enriched visual knowledge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 24950–24959.

[3] G. Chen, L. Shen, R. Shao, X. Deng, and L. Nie, "LION: Empowering multimodal large language model with dual-level visual knowledge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 26530–26540.

[4] B. Jose and A. Thomas, "The illusion of understanding: AI's role in cognitive psychology research," *AI Soc.*, vol. 40, no. 3, pp. 1543–1544, Mar. 2025.

[5] Q. Wu, W. Ji, G. Zhou, and Y. Yang, "Graph knowledge tracing in cognitive situation: Validation of classic assertions in cognitive psychology," *Knowl.-Based Syst.*, vol. 315, Apr. 2025, Art. no. 113281.

[6] L. Luo, H. Lai, Y. Pan, and J. Yin, "Efficient multimodal selection for retrieval in knowledge-based visual question answering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 6, pp. 5195–5207, Jun. 2025.

[7] N. Lan, B. Ou, X. Xie, and G. Shi, "Visual environment-interactive planning for embodied complex-question answering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 7, pp. 6481–6493, Jul. 2025.

[8] Y. Bi, H. Jiang, Y. Hu, Y. Sun, and B. Yin, "See and learn more: Dense caption-aware representation for visual question answering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 2, pp. 1135–1146, Feb. 2024.

[9] Z. You et al., "Toward long video understanding via fine-detailed video story generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 5, pp. 4592–4607, May 2025.

[10] T. Yuan, X. Zhang, B. Liu, K. Liu, J. Jin, and Z. Jiao, "Surveillance video-and-language understanding: From small to large multimodal models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 1, pp. 300–314, Jan. 2025.

[11] J. Wu, Y. Jiang, Q. Liu, Z. Yuan, X. Bai, and S. Bai, "General object foundation model for images and videos at scale," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 3783–3795.

[12] P. Li, Q. Si, P. Fu, Z. Lin, and Y. Wang, "Multimodal hypothetical summary for retrieval-based multi-image question answering," in *Proc. AAAI*, 2025, pp. 4851–4859.

[13] H. Liu et al., "MIBench: Evaluating multimodal large language models over multiple images," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2024, pp. 22417–22428.

[14] J. Ye et al., "mPLUG-Owl3: Towards long image-sequence understanding in multi-modal large language models," in *Proc. ICLR*, 2025, pp. 1–23.

[15] M. Cord, H. Laurençon, V. Sanh, and L. Tronchon, "What matters when building vision-language models?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 87874–87907.

[16] D. Jiang, "MANTIS: Interleaved multi-image instruction tuning," *Trans. Mach. Learn. Res.*, vol. 2024, pp. 1–22, May 2024.

[17] C. Feng, "VQA4CIR: Boosting composed image retrieval with visual question answering," in *Proc. AAAI*, 2025, pp. 2942–2950.

[18] Y. Qing, D. Zeng, S. Xie, K. Huang, and Y. Wang, "Integrating low-level visual cues for enhanced unsupervised semantic segmentation," in *Proc. AAAI*, 2025, pp. 6603–6611.

[19] Y. Zhang, G. Zeng, H. Shen, D. Wu, Y. Zhou, and C. Ma, "Track the answer: Extending TextVQA from image to video with spatio-temporal clues," in *Proc. AAAI*, 2025, pp. 10275–10283.

[20] F. Li et al., "LLaVA-NeXT-interleave: Tackling multi-image, video, and 3D in large multimodal models," 2024, *arXiv:2407.07895*.

[21] Z. Yang et al., "An empirical study of GPT-3 for few-shot knowledge-based VQA," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 3, pp. 3081–3089.

[22] Z. Shao, Z. Yu, M. Wang, and J. Yu, "Prompting large language models with answer heuristics for knowledge-based visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14974–14983.

[23] Z. Chen et al., "Visual chain-of-thought prompting for knowledge-based visual reasoning," in *Proc. AAAI*, 2024, pp. 1254–1262.

[24] W. Liu, Z. Ren, and L. Chen, "Knowledge reasoning based on graph neural networks with multi-layer top-p message passing and sparse negative sampling," *Knowl.-Based Syst.*, vol. 311, Feb. 2025, Art. no. 113063.

[25] Y. Cao et al., "A data-centric framework of improving graph neural networks for knowledge graph embedding," *World Wide Web*, vol. 28, no. 1, p. 2, Jan. 2025.

[26] D. Zhang, Q. Yuan, L. Meng, R. Xia, W. Liu, and C. Qin, "Reinforcement learning for single-agent to multi-agent systems: From basic theory to industrial application progress, a survey," *Artif. Intell. Rev.*, vol. 59, no. 2, pp. 1–43, Nov. 2025.

[27] D. Zhang, C. Yu, Z. Li, C. Qin, and R. Xia, "A lightweight network enhanced by attention-guided cross-scale interaction for underwater object detection," *Appl. Soft Comput.*, vol. 184, Dec. 2025, Art. no. 113811.

[28] C. Qin, X. Ran, and D. Zhang, "Unsupervised image stitching based on generative adversarial networks and feature frequency awareness algorithm," *Appl. Soft Comput.*, vol. 183, Nov. 2025, Art. no. 113466.

[29] C. Qin, S. Hou, M. Pang, Z. Wang, and D. Zhang, "Reinforcement learning-based secure tracking control for nonlinear interconnected systems: An event-triggered solution approach," *Eng. Appl. Artif. Intell.*, vol. 161, Dec. 2025, Art. no. 112243.

[30] C. Qin, M. Pang, Z. Wang, S. Hou, and D. Zhang, "Observer based fault tolerant control design for saturated nonlinear systems with full state constraints via a novel event-triggered mechanism," *Eng. Appl. Artif. Intell.*, vol. 161, Dec. 2025, Art. no. 112221.

[31] X. Lin et al., "CLIPose: Category-level object pose estimation with pre-trained vision-language knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 10, pp. 9125–9138, Oct. 2024.

[32] Y. Wei et al., "DeepMSD: Advancing multimodal sarcasm detection through knowledge-augmented graph reasoning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 7, pp. 6413–6423, Jul. 2025.

[33] D. Zhang, F. Wang, B. Li, Z. Zhao, J. Gao, and X. Li, "KAID: Knowledge-aware interactive distillation for vision-language models," in *Proc. 33rd ACM Int. Conf. Multimedia*, Oct. 2025, pp. 3212–3221.

[34] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[35] Z. Zhan, J. Qin, W. Zhuo, and G. Tan, "Enhancing vision and language navigation with prompt-based scene knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 10, pp. 9745–9756, Oct. 2024.

[36] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," in *Proc. ACL*, Jul. 2019, pp. 6418–6428.

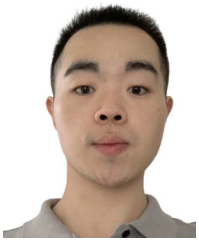
[37] W. Chen, L. Mo, Y. Su, H. Sun, and K. Zhang, "MagicBrush: A manually annotated dataset for instruction-guided image editing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 31428–31449.

[38] Y. Liang, Y. Bai, W. Zhang, X. Qian, L. Zhu, and T. Mei, "VrR-VG: Refocusing visually-relevant relationships," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10402–10411.

[39] R. Goyal et al., "The 'something something' video database for learning and evaluating visual common sense," in *Proc. ICCV*, 2017, pp. 5843–5851.
[40] J. Xiao, X. Shang, A. Yao, and T.-S. Chua, "NExT-QA: Next phase of question-answering to explaining temporal actions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9772–9781.
[41] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop Fine-Grained Vis. Categorization (FGVC)*, 2011, vol. 2, no. 1, pp. 1–8.
[42] R. Tanaka, K. Nishida, K. Nishida, T. Hasegawa, I. Saito, and K. Saito, "SlideVQA: A dataset for document visual question answering on multiple images," in *Proc. AAAI*, 2023, pp. 13636–13645.
[43] Y. Chen et al., "Can pre-trained vision and language models answer visual information-seeking questions?," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 14948–14968.
[44] Y. Chang, G. Cao, M. Narang, J. Gao, H. Suzuki, and Y. Bisk, "WebQA: Multihop and multimodal QA," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16474–16483.
[45] Y. J. Lee, C. Li, H. Liu, and Q. Wu, "Visual instruction tuning," in *Proc. Adv. Neural Inf. Process. Syst. 36*, 2023, pp. 34892–34916.
[46] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 26286–26296.
[47] J. Bai et al., "Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond," 2023, *arXiv:2308.12966*.
[48] Q. Ye et al., "MPLUG-owl: Modularization empowers large language models with multimodality," 2023, *arXiv:2304.14178*.
[49] S. Zheng et al., "Look around before locating: Considering content and structure information for visual grounding," in *Proc. AAAI*, 2025, pp. 1656–1664.
[50] C. Zhang, S. Stepputtis, J. Campbell, K. Sycara, and Y. Xie, "HiKER-SGG: Hierarchical knowledge enhanced robust scene graph generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 28233–28243.
[51] P. A. Alamdari, Y. Cao, and K. H. Wilson, "Jump starting bandits with LLM-generated prior knowledge," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2024, pp. 19821–19833.
[52] Y. Xu et al., "Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2024, pp. 18410–18430.
[53] T. Kim, S. Lee, S. Kim, and D. Kim, "ViPCap: Retrieval text-based visual prompts for lightweight image captioning," in *Proc. AAAI*, 2025, pp. 4320–4328.
[54] K. Cheng, W. Song, Z. Ma, W. Zhu, Z. Zhu, and J. Zhang, "Beyond generic: Enhancing image captioning with real-world knowledge using vision-language pre-training model," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 5038–5047.
[55] J. Feinglass and Y. Yang, "TROPE: TRaining-free object-part enhancement for seamlessly improving fine-grained zero-shot image captioning," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2024, pp. 3644–3655.
[56] N. Ahmadi, T. Truong, L. Dao, S. Ortona, and P. Papotti, "RuleHub: A public corpus of rules for knowledge graphs," *ACM J. Data Inf. Qual.*, vol. 12, no. 4, pp. 21:1–21:22, 2020.
[57] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP*, 2019, pp. 3980–3990.
[58] D. Guo et al., "DeepSeek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning," 2025, *arXiv:2501.12948*.
[59] Y. Ge et al., "Making llama SEE and draw with SEED tokenizer," in *Proc. ICLR*, 2024, pp. 1–26.
[60] Z. Chen et al., "VersaGen: Unleashing versatile visual control for text-to-image synthesis," in *Proc. AAAI*, 2025, pp. 2394–2402.
[61] Y. Ge et al., "SEED-X: Multimodal models with unified multi-granularity comprehension and generation," 2024, *arXiv:2404.14396*.
[62] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 11941–11952.
[63] A. Yang et al., "Qwen2 technical report," 2024, *arXiv:2407.10671*.
[64] Y. Li et al., "Mini-gemini: Mining the potential of multi-modality vision language models," 2024, *arXiv:2403.18814*.
[65] F. Shu et al., "LLaVA-MoD: Making LLaVA tiny via MoE knowledge distillation," in *Proc. 13th Int. Conf. Learn. Represent.*, Singapore, 2025, pp. 1–19.
[66] S. Bekman et al., "OBELICS: An open Web-scale filtered dataset of interleaved image-text documents," in *Proc. Adv. Neural Inf. Process. Syst. 36*, 2023, pp. 71683–71702.
[67] Q. Ye et al., "MPLUG-owl2: Revolutionizing multi-modal large language model with modality collaboration," 2023, *arXiv:2311.04257*.
[68] S. Antol et al., "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2425–2433.
[69] C. Zhang, Y. Liu, H. Fu, and X. Sun, "Position guided dynamic receptive field network: A small object detection friendly to optical and SAR images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 4, pp. 1234–1248, Apr. 2025.
[70] Z. Fan et al., "Combatting dimensional collapse in LLM pre-training data via submodular file selection," in *Proc. 13th Int. Conf. Learn. Represent.*, 2025, pp. 1–25.
[71] L. Chen, Z. Wang, Y. Zhang, and J. Li, "Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2025, pp. 1–9.
[72] S. Zhang et al., "OPT: Open pre-trained transformer language models," 2022, *arXiv:2205.01068*.
[73] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NeurIPS*, 2020, pp. 1–9.
[74] L. Zhang et al., "Scimage: How good are multimodal large language models at scientific text-to-image generation?," in *Proc. ICLR*, 2025, pp. 1–25.
[75] D. A. Hudson and C. D. Manning, "GQA: A new dataset for real-world visual reasoning and compositional question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 6700–6709.
[76] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in VQA matter: Elevating the role of image understanding in visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6325–6334.
[77] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? Dataset and methods for multilingual image question answering," 2015, *arXiv:1505.05612*.
[78] P. Lu et al., "IconQA: A new benchmark for abstract diagram understanding and visual language reasoning," in *Proc. Neural Inf. Process. Syst.*, 2021, pp. 1–14.

Guanghui Ye (Graduate Student Member, IEEE) received the M.S. degree in computer science and technology from the Department of Computer Science, Jinan University, Guangzhou, China, in 2023. He is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University. His publications have been published at top academic conferences, such as the Annual Meeting of the Association for Computational Linguistics (ACL), the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), the Conference on Empirical Methods in Natural Language Processing (EMNLP), and the AAAI Conference on Artificial Intelligence (AAAI). His current research interests include knowledge engineering, natural language processing, large language models, and multimedia computing.

Huan Zhao (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and technology from Hunan University, Changsha, China, in 1989, 2004, and 2010, respectively. She is currently a Professor with the School of Information Science and Technology, Hunan University. She has published over 100 research papers in top-tier international journals and conferences, covering various high-impact academic platforms, including journals, such as IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and IEEE INTERNET OF THINGS JOURNAL; and conferences, such as the Annual Meeting of the Association for Computational Linguistics (ACL), the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), the AAAI Conference on Artificial Intelligence (AAAI), and the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Her current research interests mainly include speech signal processing, cross-media retrieval, and natural language processing.



Yixian Shen (Member, IEEE) received the Ph.D. degree in computer science from the University of Amsterdam, The Netherlands, in 2024. His publications have appeared at leading AI conferences, such as the Conference on Neural Information Processing Systems (NeurIPS), the IEEE/CVF International Conference on Computer Vision (ICCV), the Annual Meeting of the Association for Computational Linguistics (ACL), the Conference on Empirical Methods in Natural Language Processing (EMNLP), the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), and the AAAI Conference on Artificial Intelligence (AAAI). His research interests focus on efficient artificial intelligence, including parameter-efficient fine-tuning and cross-modal reasoning efficiency for large language and multimodal foundation models.

Jiaqi Li received the M.S. degree from The University of Hong Kong in 2023. He is currently pursuing the Ph.D. degree in computer science with the University of Warwick, U.K. His publications have been published at top academic conferences, such as the Annual Meeting of the Association for Computational Linguistics (ACL) and the Conference on Empirical Methods in Natural Language Processing (EMNLP). His research interests include deep learning in computer vision, generative model, and multi-modalities.

Fengnan Li received the B.S. degree in data science from Duke Kunshan University in 2023 and the M.S. degree in biostatistics from Duke University, USA, in 2025, where he is currently pursuing the Ph.D. degree. His publications have been published at conferences, such as the Annual Meeting of the Association for Computational Linguistics (ACL) and the Conference on Empirical Methods in Natural Language Processing (EMNLP). His research interests broadly include clinical natural language processing and machine learning methods for electronic health record data.

Zhihua Jiang (Member, IEEE) received the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2008. She is currently an Associate Professor with the Department of Computer Science, Jinan University, Guangzhou. Her publications have been published at top academic conferences, such as the Annual Meeting of the Association for Computational Linguistics (ACL), the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), and the Conference on Empirical Methods in Natural Language Processing (EMNLP). Her research interests include natural language processing, information retrieval, data mining, and multimedia computing.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University in 1985 and the Ph.D. degree in computer science from the University of Houston in 1990. He is currently a SUNY Distinguished Professor with the State University of New York and a National Distinguished Professor with Hunan University, China. He has authored or co-authored more than 1130 journal articles, book chapters, and refereed conference papers. He holds nearly 80 patents announced or authorized by Chinese National Intellectual Property Administration. Since 2020, he has been among the World's top few most influential scientists in parallel and distributed computing regarding single-year impact (ranked #2) and career-long impact (ranked #4) based on a composite indicator of the Scopus citation database. He is listed in ScholarGPS Highly Ranked Scholars from 2022 to 2024 and is among the top 0.002% out of over 30 million scholars Worldwide based on a composite score of three ranking metrics for research productivity, impact, and quality in the recent five years. He is listed in Scilit Top Cited Scholars from 2023 to 2024 and is among the top 0.02% out of over 20 million scholars Worldwide based on top-cited publications. He is an AAAS Fellow, an AAIA Fellow, an ACIS Fellow, and an AIIA Fellow. He is a member of European Academy of Sciences and Arts and the Academia Europaea (Academician of the Academy of Europe). He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He won the IEEE Region 1 Technological Innovation Award (Academic) in 2023. He was a recipient of the 2022 and 2023 International Science and Technology Cooperation Award and the 2023 Xiaoxiang Friendship Award of Hunan Province, China.