TRUTH IN FOCUS: SCENE GRAPH GROUNDING FOR FAITHFUL VISUAL QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in large vision–language models (VLMs) have substantially improved Visual Question Answering. Nevertheless, reliable answer generation remains a persistent challenge, as VLMs may produce plausible responses that are insufficiently grounded in the visual content. Scene graphs provide a structured representation of visual content by organizing object entities and their relationships. However, a complete scene graph often contains information beyond what a given question requires, which can distract the model from relevant evidence and lead to factually unsupported answers. We therefore shift from global scene modeling to question-conditioned local grounding and propose a scene graph grounding framework that extracts a compact subgraph containing the visual information needed to answer each question. Specifically, we introduce a dual-differential causal attention module that jointly models node–node dependencies and node–query associations, enabling the relative contribution of graph elements to be assessed for subgraph prediction. To mitigate the scarcity of fine-grained grounding supervision, we further construct a factual–counterfactual dataset by leveraging the generative capabilities of large models. Extensive experiments on multiple benchmarks demonstrate that our method achieves superior performance against state-of-the-art approaches.

1 INTRODUCTION

Visual Question Answering (VQA) (Antol et al., 2015) has been a cornerstone task in multimodal artificial intelligence with a wide range of real-world applications, including autonomous robotics (Marcu et al., 2024), drone navigation (Shah et al., 2019), and embodied agents (Lin et al., 2022). Given an image and a natural-language question, VQA aims to generate a response that is consistent with the visual content (Wu et al., 2017), requiring models to move beyond object recognition and reason about object attributes, spatial relationships, and broader visual context (Shih et al., 2016; Sima et al., 2024). Recent breakthroughs in large vision-language models have opened new avenues for VQA (Cocchi et al., 2025). Through large-scale pre-training on massive web data, these models have demonstrated strong zero-shot generalization and reasoning capabilities, significantly pushing the performance boundaries of VQA to new horizons.

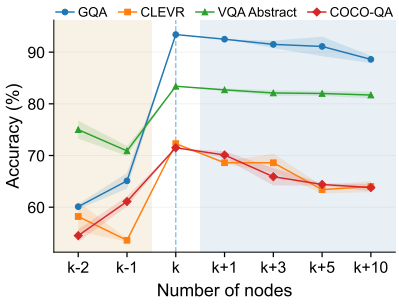

Figure 1: We vary the subgraph size around the budget k using GPT-4.1 Mini as the answering model. Accuracy declines when nodes are removed and degrades gradually when nodes are added.

Despite the remarkable progress, VLMs may fail to anchor their predictions to the visual content that is relevant to a given question, resulting in inconsistent responses with the underlying scene, a phenomenon referred to as model hallucination (An et al., 2025). Existing state-of-the-art methods often address this issue through elaborate inference-time strategies, including multi-turn self-correction mechanisms (He et al., 2025; Huang et al., 2024) and chain-of-thought prompting (Wei et al., 2022; Mitra et al., 2024). Although effective, they typically process the global visual scene without explicitly extracting question-relevant evidence, leaving models vulnerable to distracting background context (Shi et al., 2023; Liu et al., 2024; Ge et al., 2025). When models initially rely on such distracting cues, early errors can propagate through lengthy reasoning traces, causing answers to drift away from

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

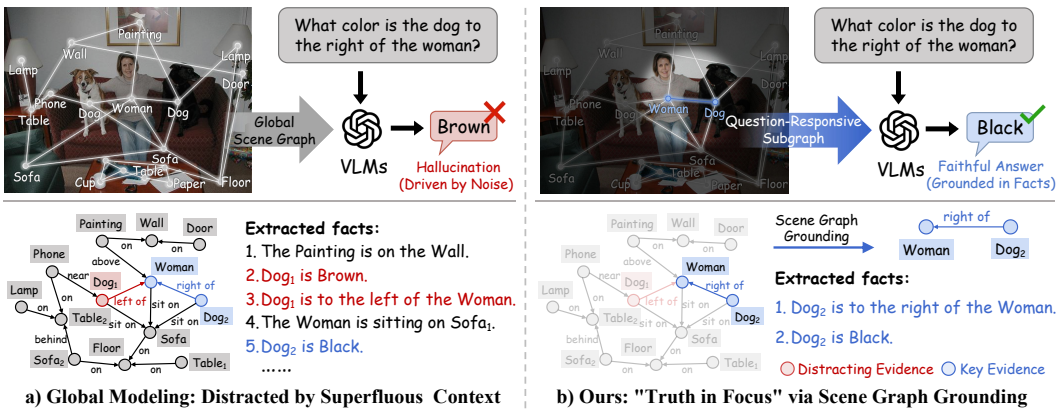

Figure 2: Scene graphs may introduce redundant context and visual noise, weakening the model's focus on visual evidence and increasing the risk of unsupported responses. We propose a scene graph grounding framework that extracts a compact subgraph to improve factual reliability.

factual visual evidence. These limitations motivate a direct grounding paradigm that explicitly selects supporting visual facts before answering, rather than relying solely on complex inference procedures.

Scene graphs support evidence grounding through interpretable representations of visual entities, attributes, and relationships. However, full scene graphs often encode far more information than a given question requires. As shown in Figure 1, randomly removing or adding nodes to a subgraph consistently reduces VQA accuracy, motivating the need to select a compact subgraph that maintains the necessary elements. The difficulty of node selection lies not merely in filtering out irrelevant background noise, but in disambiguating semantically similar candidates. As seen in Figure 2, a scene may contain multiple objects of the same category, and accurately localizing the correct object requires reasoning about explicit spatial or semantic relationships. Thus, whether a graph element supports the answer depends on both its relations to other elements and its relevance to the question. We therefore address two questions: *1) how to model node-node dependencies to capture the topological relevance of graph elements, and 2) how to measure the fine-grained contribution of each node with respect to the semantics of the question?*

In this paper, we propose a scene graph grounding framework that adaptively "zooms in" on the visual facts, as illustrated in Figure 2. The framework builds on the assumption that visual reasoning must account for both the structure of the scene and the semantics of the question. Standard graph modeling, however, tends to pick up spurious correlations from the training distribution, such as structural biases toward highly connected nodes and linguistic biases driven by language priors, both of which we quantify in Section 3. To address this, we formulate evidence selection from a causal intervention perspective and introduce a dual differential causal attention module, dubbed DDCA. DDCA scores each node by contrasting its observed score with the expected score obtained under randomized interventions on either the graph structure or the scene context. This discards the component attributable to the two biases and leaves a measure of each node's own contribution, so the resulting scores favor nodes that are informative and aligned with the question. By using these scores, the model iteratively grows a compact subgraph, conditioning each selection step on the subgraph accumulated so far, so that the final output is a coherent chain of evidence rather than an unordered set. Finally, since ground-truth answers lack explicit node-level annotations to supervise selection, we construct a factual-counterfactual dataset that provides the node-level supervision to separate true evidence from plausible distractors.

Our main contributions are summarized as follows:

- We propose a graph grounding approach that shifts the VQA paradigm from global graph modeling to subgraph selection, reducing the influence of structural and background noise.
- We introduce a dual differential causal attention mechanism to jointly model node-node and query-node dependencies. By evaluating relative causal contributions, it mitigates structural and linguistic biases to preserve relevant evidence and suppress distractors.
- We propose a factual-counterfactual data construction strategy that pairs evidence subgraphs for factual and counterfactual answers, providing fine-grained supervision to distinguish key evidence. Experiments on multiple benchmarks validate the effectiveness of our method.

2 Related Work

2.1 Visual Question Answering

Visual Question Answering requires models to jointly understand visual scenes and natural language questions. Early studies mainly explored vision-language alignment and cross-modal interaction through iterative attention mechanisms (Yang et al., 2016; Lu et al., 2016; Nam et al., 2017) and bilinear pooling-based fusion (Fukui et al., 2016; Kim et al., 2018). Subsequent research shifted towards object-centric representations based on region proposals, as exemplified by the Bottom-Up and Top-Down attention paradigm (Anderson et al., 2018). Recently, large Vision-Language Models have advanced VQA through large-scale pre-training (Li et al., 2023a; Liu et al., 2023). Despite these advances, VLMs remain prone to hallucinations (Li et al., 2023b), producing responses that contradict visual evidence. Existing mitigation methods fall into two categories, including *training-time interventions*, such as RLHF-V (Yu et al., 2024) and V-DPO (Xie et al., 2024), leverage preference-based optimization for generation; and *inference-time approaches* (Wang et al., 2025; Chen et al., 2025) instead intervene in the decoding process to reinforce the influence of visual information. For example, VISTA (Li et al., 2025d) reinforces visual evidence during generation, and SECOND (Park et al., 2025) mitigates perceptual hallucinations via multi-scale visual contrast. However, raw visual inputs are inherently dense and redundant, easily distracting models with irrelevant context. Transforming them into explicit, structured representations provides an effective way to highlight genuine visual facts.

2.2 Scene Graph Modeling

Scene Graphs represent visual scenes as structured collections of objects, attributes, and semantic relationships. In VQA, they provide an interpretable bridge between visual perception and reasoning. Existing scene graph-based methods broadly follow three directions. *Modular-reasoning methods* decompose visual reasoning into explicit functional modules or sequential steps over the graph, progressively combining intermediate results to derive an answer (Shi et al., 2019; Hudson & Manning, 2019a; Koner et al., 2021). *Query-guided methods* use the question to guide information aggregation and relational reasoning across the graph (Liang et al., 2021; Nuthalapati et al., 2021). *Knowledge-augmented methods* (Zhang et al., 2021) enrich scene graphs with external knowledge to support complex and commonsense reasoning. Recent studies have further incorporated scene graphs into LLMs and VLMs as structural guidance or auxiliary context. For example, LLMs can directly interpret scene graphs for downstream QA tasks (Yang et al., 2025) or use them to resolve complex queries (Taluzzi et al., 2025). Scene graphs have also been introduced as structural constraints during generation to improve visual grounding (Wu et al., 2024a; Lu et al., 2025). However, most existing methods process the scene graph as a whole, requiring the model to consider far more information than is needed for a given question. Such representations weaken the influence of supporting evidence and increase the risk of factual errors. In contrast, our method shifts from global scene modeling to question-conditioned local grounding, explicitly extracting a compact subgraph that retains the exact visual evidence needed for reliable prediction.

3 Methodology

3.1 Problem Formulation

We formulate visual evidence selection in VQA as a scene graph grounding problem. Formally, we consider an image I, a question Q, and its corresponding scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{v_i\}_{i=1}^{N}$ and edge set $\mathcal{E} = \{e_{ij}\}$, where each e_{ij} represents a relation between nodes v_i and v_j. Our goal is to extract a compact and task-relevant subgraph $\mathcal{G}_S \subseteq \mathcal{G}$ that retains the visual evidence needed to answer the question. The ideal selection objective is

$$\mathcal{G}_S^* = \arg\max_{\mathcal{G}_S \subseteq \mathcal{G}} P_\phi(A \mid \mathcal{G}_S, Q, I). \tag{1}$$

Here, A denotes the ground-truth answer and $P_\phi(A \mid \mathcal{G}_S, Q, I)$ denotes the likelihood of the answer and can be estimated using a pre-trained VLM. To achieve this goal, we train a grounding model $P_\theta(\mathcal{G}_S \mid \mathcal{G}, Q, I)$ to evaluate each node based on its structural associations and semantic relevance to the question, producing scores for subgraph selection.

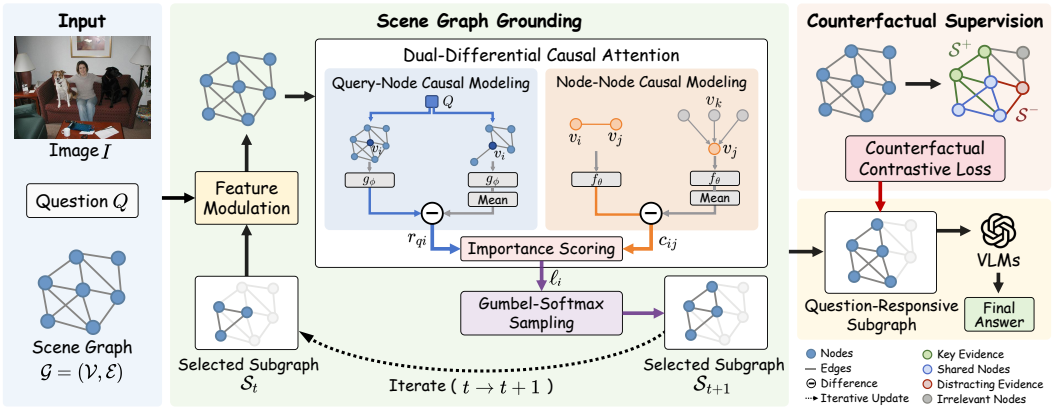

Figure 3: Illustration of the scene graph grounding framework. Given an image I, a question Q, and a scene graph $\mathcal{G}$, our framework produces a question-responsive subgraph. We first employ a dual-differential causal attention module to estimate node–query and node–node causal scores. Then, we adopt the Gumbel-Softmax sampling to augment the subgraph across iterations. To supervise graph grounding, we apply contrastive learning with factual ($\mathcal{S}^+$) and counterfactual ($\mathcal{S}^-$) graphs to focus on visual facts that support the correct answer while suppressing distracting context.

3.2 Scene Graph Grounding

As shown in Figure 3, extracting a compact subgraph $\mathcal{G}_S$ is essential to reduce interference from unnecessary visual information (Zhang et al., 2024; Peng et al., 2024). However, feature similarity alone cannot indicate whether a graph element provides valid evidence for a given question (Li et al., 2025a;b). For example, when asked about *a dog to the right*, a similarity-based model may attend to a salient *brown dog on the left* because of their strong feature resemblance, despite inconsistency with the spatial constraint specified in the question. To reduce the influence of superficial correlations on evidence selection, we formulate scene graph grounding from a causal intervention perspective. By comparing scores under observed and reference inputs, we estimate the contributions of node-node associations and the relevance of individual nodes to the question, while mitigating the effects of shared biases.

As illustrated in Figure 4, we perform causal modeling along two dimensions:

1. **Node-Node Causal Modeling.** Highly connected nodes may receive inflated association scores. We replace one node, keep the other fixed, and compare scores before and after intervention to estimate its influence and mitigate structural bias.
2. **Query-Node Causal Modeling.** Language priors may favor nodes associated with common answers. We modify the scene, keep the query and evaluated node fixed, and compare scores before and after intervention to estimate query-node relevance and mitigate linguistic bias.

We combine node-node and query-node differential scores to capture node relationships and their relevance to the question, guiding the selection of a compact subgraph that preserves the evidence needed to answer the question.

Node-Node Causal Modeling. For two nodes v_i and v_j connected by an edge e_{ij}, let $f_\theta(v_i, v_j, e_{ij})$ denote a scoring function that encodes their relational association. Although this score captures the affinity between the two nodes, it may not reflect the individual contribution of v_i to v_j due to confounding factors. For example, a target node v_j may receive high scores because it aggregates multi-hop information. Consequently, a high value of $f_\theta(v_i, v_j, e_{ij})$ may arise from the structural bias of v_j rather than a causal link from v_i to v_j.

Inspired by Pearl's intervention framework (Pearl, 2009), we introduce a debiasing mechanism that estimates the model-decision causal influence by contrasting a factual observation with randomized reference perturbations. Specifically, we define the following differential score:

$$\Delta(x; p_{\text{ref}}) = \mathbb{E}\left[Y \mid do(X = x)\right] - \mathbb{E}_{\tilde{x} \sim p_{\text{ref}}}\left[\mathbb{E}\left[Y \mid do(X = \tilde{x})\right]\right], \tag{2}$$

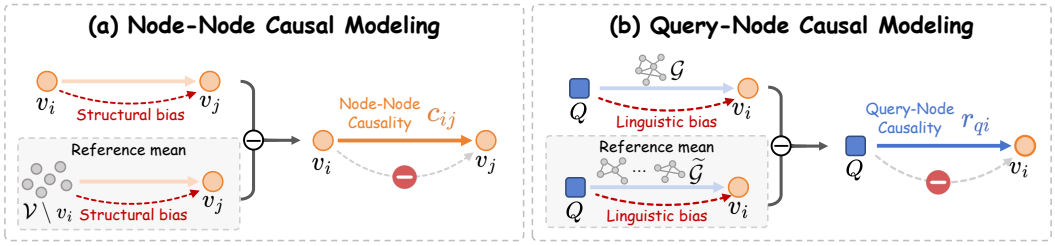

Figure 4: Causal modeling of graph grounding. (a) Node-node modeling keeps the target v_j fixed and subtracts the mean association score over reference nodes to estimate c_{ij}, mitigating shared structural bias. (b) Query-node modeling keeps Q and v_i fixed and subtracts the mean alignment score over counterfactual reference graphs to estimate r_{qi}, mitigating shared linguistic bias.

where Y denotes the output of the scoring function, x denotes the observed input, and $\tilde{x}$ is sampled from a predefined reference distribution p_{ref}. Following this interventional contrast, we quantify the relative contribution c_{ij} of node v_i to its association score with v_j as

$$c_{ij} = f_\theta(v_i, v_j, e_{ij}) - \mathbb{E}_{v_k \sim p_{\text{ref}}(\cdot|v_j)}[f_\theta(v_k, v_j, e_{kj})]. \tag{3}$$

Here, the first term measures the observed association between v_i and v_j. The second term forms a reference baseline by replacing v_i with reference node v_k and taking the expectation of the resulting association score over the reference distribution. As shown in Appendix D, subtracting the baseline mitigates the shared structural bias of v_j, highlighting the association specific to the node pair (v_i, v_j).

Query-Node Causal Modeling. While node-node modeling mitigates structural biases, query-node alignment may still be dominated by language priors, arising from co-occurring visual concepts and question terms. Directly constructing counterfactual questions is nontrivial, because modifying the queried content may change the task semantics. We therefore intervene on the visual conditions encoded by the scene graph while keeping the question and target node fixed. Specifically, we construct reference graphs $\widetilde{\mathcal{G}}$ by randomly deleting nodes other than v_i. We define the differential relevance score r_{qi} as

$$r_{qi} = g_\theta(Q, \mathcal{G}, v_i) - \mathbb{E}_{\widetilde{\mathcal{G}} \sim p_{\text{ref}}(\cdot|Q,\mathcal{G},v_i)}[g_\theta(Q, \widetilde{\mathcal{G}}, v_i)]. \tag{4}$$

Here, $g_\theta(Q, \mathcal{G}, v_i)$ quantifies the alignment of node v_i with question Q in the observed scene $\mathcal{G}$. The first term captures query-node alignment under the factual context, while the second term estimates the expected response under the edited scene conditions. Their difference mitigates shared linguistic bias, highlighting the relevance of v_i to Q in the observed scene.

Dual-Differential Causal Attention. For end-to-end optimization, we instantiate the preceding intervention-based principles as a unified Dual-Differential Causal Attention (DDCA) module. DDCA combines node-node structural debiasing with query-node linguistic debiasing, producing differentiable relevance scores that guide the construction of the grounded subgraph $\mathcal{G}_S$. Specifically, for a node pair (v_i, v_j), we define the attention score u_{ij} as

$$u_{ij} = \text{sim}(v_i, v_j) + \lambda_1 c_{ij} + \lambda_2 r_{qj}, \tag{5}$$

where $\text{sim}(v_i, v_j)$ denotes the cosine similarity between the representations of nodes v_i and v_j. For node pairs without an edge, we set $c_{ij} = 0$. The learnable coefficients λ_1 and λ_2 balance structural coherence and question relevance. The resulting attention scores combine node associations with intervention effects and are normalized over all other nodes using softmax to obtain weights a_{ij}. These weights aggregate the differential scores from both branches to compute node-selection logits during iterative grounding:

$$\ell_i = \sum_{j \neq i} a_{ij} \left(\lambda_1 c_{ij} + \lambda_2 r_{qi}\right). \tag{6}$$

DDCA module is further trained with the counterfactual dataset introduced in Section 3.3. Counterfactual supervision helps the model distinguish supporting evidence from distractors through factual and counterfactual subgraphs.

Iterative Reasoning via Modulation. Building on the dual-differential causal mechanism, we formulate scene graph grounding as a sequential selection process:

$$P(\mathcal{G}_S \mid Q, I, \mathcal{G}) = \prod_{t=1}^{T} P(s_t \mid \mathcal{S}_{t-1}, Q, I, \mathcal{G}), \tag{7}$$

where $\mathcal{S}_{t-1} = \{s_1, \ldots, s_{t-1}\}$ denotes the set of nodes selected before step t and $s_t \in \mathcal{V} \setminus \mathcal{S}_{t-1}$ is selected from remaining candidates. The final subgraph $\mathcal{G}_S$ is the subgraph of $\mathcal{G}$ induced by the selected node set $\mathcal{S}_T$.

We use a frozen Sentence-BERT to encode the question and textual descriptions of nodes and edges as $\mathbf{q}, \tilde{\mathbf{v}}_i, \tilde{\mathbf{e}}_{ij} \in \mathbb{R}^d$, respectively, where d denotes the feature dimension. To condition each decision on the historical context $\mathcal{S}_{t-1}$, we maintain a dynamic representation of the partially grounded subgraph. At step t, we obtain the historical state $\mathbf{g}^{(t-1)} \in \mathbb{R}^d$ by mean-pooling the initial embeddings $\tilde{\mathbf{v}}_j$ of nodes $v_j \in \mathcal{S}_{t-1}$. For $t = 1$, $\mathbf{g}^{(0)} \in \mathbb{R}^d$ is initialized as a zero vector or a learnable state. We then use the question representation $\mathbf{q}$ and the historical state to modulate each node representation:

$$\begin{aligned} (\gamma^{(t)}, \beta^{(t)}) &= \psi([\mathbf{q}; \mathbf{g}^{(t-1)}]) \\ \mathbf{v}_i^{(t)} &= \gamma^{(t)} \odot \tilde{\mathbf{v}}_i + \beta^{(t)}, \end{aligned} \tag{8}$$

where $[\cdot;\cdot]$ denotes feature concatenation, and ψ is a learnable projection that generates the scaling and shifting parameters $\gamma^{(t)}, \beta^{(t)} \in \mathbb{R}^d$, respectively. This state-conditioned modulation updates the representation of each candidate according to both the question and the previously selected visual evidence, encouraging subsequent selection to contribute complementary information.

The modulated representations are processed by DDCA module to obtain the node-selection logits $\ell^{(t)} \in \mathbb{R}^N$. Before sampling, we set the logits of previously selected nodes to $-\infty$ and leave the others unchanged, obtaining the masked logits $\widetilde{\ell}^{(t)}$. To support end-to-end optimization of the discrete selection process, we employ a Gumbel-Softmax relaxation:

$$p^{(t)} \sim \text{Gumbel-Softmax}(\widetilde{\ell}^{(t)}, \tau), \tag{9}$$

where $\tau > 0$ is the temperature parameter. The resulting vector $p^{(t)} \in [0, 1]^N$, with $\sum_{i=1}^{N} p_i^{(t)} = 1$, gives the selection probability of each candidate node. At each step, we select the remaining node with the highest probability and add it to the selected node set $\mathcal{S}_{t-1}$, then proceed to the next iteration for a predefined number of steps. Through the Gumbel-Softmax relaxation, the framework supports gradient-based learning while retaining discrete node selection.

3.3 Counterfactual Contrastive Supervision

Factual-Counterfactual Dataset Construction. To alleviate the lack of supervision, we develop an offline pipeline for constructing a factual-counterfactual dataset. Given a ground-truth answer A, we first generate a counterfactual answer A^- as a plausible but incorrect alternative: For binary questions, A^- is obtained by reversing the ground-truth label. For open-ended questions involving entities, attributes, or relations, we select an alternative value that shares the same semantic type as A, is instantiated in the scene graph, but conflicts with the ground truth. Therefore, A^- constitutes a challenging distractor, requiring the model to distinguish differences among co-occurring objects rather than rely on coarse existence cues.

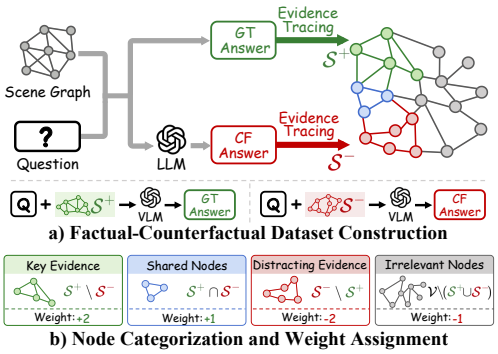

Figure 5: Overview of counterfactual contrastive supervision (a) Constructing factual ($\mathcal{S}^+$) and counterfactual ($\mathcal{S}^-$) evidence subgraphs using an LLM. (b) Node weight assignment based on $\mathcal{S}^+$ and $\mathcal{S}^-$.

Subsequently, we employ an LLM to derive the reasoning trajectories for both factual and counterfactual answers (Xia et al., 2025). Given a triplet $(Q, \mathcal{G}, a)$ where $a \in \{A, A^-\}$, the LLM is prompted to generate a reasoning chain and locate a compact set of supporting nodes $\mathcal{S}_a$. These nodes and their induced edges yield the factual subgraph $\mathcal{S}^+$ and the counterfactual subgraph $\mathcal{S}^-$. The paired subgraphs provide contrastive supervision for DDCA module.

Contrastive Supervision and Objective. To convert the factual-counterfactual partitions into a trainable objective, we define a weight function $\omega(i)$ that reflects the importance of each node in the reasoning process. Nodes supporting the factual answer receive positive weights, while nodes associated with the counterfactual answer or neither reasoning path are assigned negative weights.

Table 1: Accuracy (%) on GQA, CLEVR, VCR, COCO-QA, and VQA-Abstract across Gemini 2.5 Flash Lite, GPT-4.1 Mini, GPT-4o, and Qwen3.5 4B. GT SG and Pred. SG denote ground-truth and predicted scene graphs, respectively. The best result is marked in **bold**.

VLM	Method	GQA		CLEVR		VCR (Pred. SG)			COCO-QA	VQA-Abstract
		GT SG	Pred. SG	GT SG	Pred. SG	Q-A	QA-R	Q-AR	Pred. SG	Pred. SG
Gemini 2.5 Flash Lite	Full SG	81.9	72.4	61.2	60.8	56.1	58.5	33.3	69.3	83.1
	EMRAG (Xue et al., 2024)	83.0	73.1	61.6	61.4	55.9	58.3	33.4	67.5	83.3
	PCST (He et al., 2024)	78.4	74.1	53.9	52.8	56.4	58.9	33.7	68.2	81.5
	SubgraphRAG (Li et al., 2025c)	83.7	72.7	62.1	60.1	56.6	58.8	33.8	69.4	82.7
	Ours	**85.9**	**76.1**	**63.0**	**61.9**	**57.6**	**59.1**	**34.2**	**70.6**	**84.1**
GPT-4.1 Mini	Full SG	85.9	72.3	64.4	63.5	66.1	65.4	44.9	64.1	81.7
	EMRAG (Xue et al., 2024)	83.0	72.6	64.5	63.9	66.7	65.3	44.7	63.7	81.6
	PCST (He et al., 2024)	81.4	68.0	54.3	53.2	64.2	64.0	43.5	64.9	76.0
	SubgraphRAG (Li et al., 2025c)	85.4	70.2	65.5	64.5	66.9	65.7	45.8	67.1	78.4
	Ours	**88.4**	**75.2**	**65.8**	**64.8**	**67.4**	**66.8**	**46.1**	**67.4**	**82.5**
GPT-4o	Full SG	86.9	74.5	67.8	66.9	63.9	68.5	43.4	70.2	80.5
	EMRAG (Xue et al., 2024)	84.3	73.2	67.2	67.4	64.4	66.2	43.0	69.0	80.2
	PCST (He et al., 2024)	81.1	71.0	52.9	52.8	64.1	68.3	43.9	68.5	74.4
	SubgraphRAG (Li et al., 2025c)	86.2	72.1	68.5	67.3	63.6	69.0	43.2	71.2	77.4
	Ours	**89.1**	**75.0**	**70.3**	**67.9**	**64.7**	**69.1**	**44.5**	**71.3**	**81.3**
Qwen3.5 4B	Full SG	76.8	74.3	64.0	64.1	53.4	54.1	29.2	66.0	83.3
	EMRAG (Xue et al., 2024)	80.1	72.7	63.4	63.6	52.5	54.3	28.9	63.6	83.4
	PCST (He et al., 2024)	70.7	71.6	60.3	60.5	52.2	52.4	28.2	61.3	81.2
	SubgraphRAG (Li et al., 2025c)	81.9	72.6	64.7	64.3	53.3	53.8	29.1	65.8	81.5
	Ours	**83.2**	**74.8**	**65.2**	**64.5**	**53.6**	**54.4**	**29.5**	**66.1**	**83.5**

Specifically, we assign weights of $+2$, $+1$, -2, and -1 to nodes in $\mathcal{S}^+ \setminus \mathcal{S}^-$, $\mathcal{S}^+ \cap \mathcal{S}^-$, $\mathcal{S}^- \setminus \mathcal{S}^+$, and $\mathcal{V} \setminus (\mathcal{S}^+ \cup \mathcal{S}^-)$, respectively. Let $\tilde{p} \in [0,1]^{|\mathcal{V}|}$ denote the aggregated selection probabilities over T iterations. The training loss is defined as $\mathcal{L} = -\sum_{i \in \mathcal{V}} \omega(i)\, \tilde{p}_i$.

4 Experiments

We evaluate our method on five VQA benchmarks using four VLMs. Ablation studies are conducted on GQA with Gemini 2.5 Flash Lite. Detailed experimental settings and additional experiments and analyses are provided in Appendices A and B, respectively. The code is released for reproducibility[1].

4.1 Main Results

We compare our method with four baselines across five VQA benchmarks and four VLMs. As shown in Table 1, our method achieves superior performance in all evaluation settings, demonstrating effectiveness across model scales and reasoning tasks. On GQA with ground-truth scene graphs, it outperforms Full SG by 2.2%-6.4%. In contrast, none of the baselines consistently improves upon Full SG, and all the other three perform worse with GPT-4.1 Mini and GPT-4o in this setting. This comparison indicates that reducing graph size is insufficient and the selected subgraph must preserve the evidence concerned with the question.

Importantly, the gains over Full SG are especially pronounced for the two smaller VLMs, reaching 4% with Gemini 2.5 Flash-Lite and 6.4% with Qwen3.5 4B on GQA. This result suggests that grounding can compensate for their limited ability to filter lengthy contextual inputs. Our method also ranks first in settings based on predicted scene graphs, indicating its effectiveness beyond annotated graphs. On VCR, it improves Q-A accuracy over Full SG by 1.5% with Gemini 2.5 Flash-Lite and Q-AR accuracy by 1.2% with GPT-4.1 Mini, suggesting that grounding helps associate commonsense knowledge with the relevant people and events. The gains on COCO-QA and VQA-Abstract are consistent but modest. This may be attributed to the relatively simple clipart scenes in VQA-Abstract and the caption-derived questions in COCO-QA, which rarely require relational disambiguation among similar objects. Consequently, their full scene graphs contain fewer distractors, limiting the potential gains from subgraph grounding.

4.2 Ablation Study

Effect of Components. Table 2 evaluates the contribution of each component. We observe that removing both DDCA module and counterfactual supervision reduces accuracy from 85.9% to 83.6%, confirming their complementary roles in distinguishing supporting evidence from distracting context. Removing DDCA module or counterfactual supervision results in drops of 1.5% and 0.7%, respectively. Excluding feature modulation further lowers accuracy to 84.1%, demonstrating the

[1] https://anonymous.4open.science/r/truth-in-focus

Table 2: Ablation on the proposed components including feature modulation (Feat. Mod.), the dual-differential causal attention (DDCA), and counterfactual supervision.

Feat. Mod.	DDCA	Counterfactual	Acc (%)
✓			83.6
	✓	✓	84.1
✓		✓	84.4
✓	✓		85.2
✓	✓	✓	**85.9**

Table 3: Ablation on Query-Node and Node-Node terms within Dual-Differential Causal Attention.

Method	Acc (%)
w/o Query-Node + w/o Node-Node	84.4
w/o Query-Node	84.7
w/o Node-Node	85.3
Ours	**85.9**

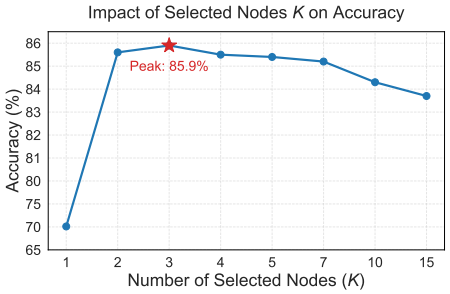

Figure 6: Ablation on the number of iterative steps K. Our model reaches the peak performance at $K = 3$ with the 95-th percentile of ground-truth node counts on the GQA training set.

Table 4: Comparison of different methods on three non-visual graph benchmarks. All four VLMs are evaluated without image inputs. The best result for each VLM and metric is highlighted in **bold**.

VLM	Method	MolecularIQ	MetaQA		MetaHotpotQA		VLM	Method	MolecularIQ	MetaQA		MetaHotpotQA	
		Acc.	Hits@1	F1	EM	F1			Acc.	Hits@1	F1	EM	F1
Gemini 2.5 Flash Lite	Full SG	40.5	48.2	22.6	82.0	87.0	GPT-4o	Full SG	42.0	56.0	31.9	86.9	92.0
	EMRAG	16.0	48.7	22.3	78.9	83.5		EMRAG	19.8	56.2	29.3	87.1	91.2
	PCST	5.3	40.4	17.8	79.1	83.8		PCST	12.2	46.4	23.3	86.5	90.7
	SubgraphRAG	25.2	48.6	25.7	82.4	86.8		SubgraphRAG	27.5	59.6	35.5	87.8	92.1
	Ours	**49.6**	**53.5**	**27.2**	**82.7**	**87.1**		**Ours**	**52.7**	**65.9**	**40.1**	**88.2**	**92.3**
GPT-4.1 Mini	Full SG	41.2	49.5	28.2	83.4	88.7	Qwen3.5 4B	Full SG	36.6	**47.6**	21.0	75.0	79.9
	EMRAG	10.7	49.5	25.0	81.9	85.8		EMRAG	19.1	46.0	20.1	68.1	72.6
	PCST	7.6	43.8	20.7	82.5	86.9		PCST	8.4	34.5	16.7	66.8	71.1
	SubgraphRAG	19.1	50.2	28.5	84.0	88.5		SubgraphRAG	26.0	42.6	20.4	**77.0**	81.3
	Ours	**45.0**	**54.1**	**31.6**	**84.3**	**88.9**		**Ours**	**51.1**	**47.6**	**21.4**	**77.0**	**81.5**

importance of node representations on the question and visual evidence. The complete model achieves the highest accuracy, confirming the complementary contributions of all three components.

Effect of Causal Modeling. We further examine the contributions of query-node and node-node terms in DDCA. As reported in Table 3, removing the differential operations from both the query-node and node-node branches reduces accuracy from 85.9% to 84.4%, demonstrating the overall benefit of causal modeling. Removing the query-node branch decreases accuracy to 84.7%, whereas removing the node-node branch yields 85.3%. This difference in performance drops indicates that while semantic alignment dominates the identification of supporting evidence, node-node modeling offers essential structural context. Consequently, their combination successfully balances task relevance with the relational coherence of the scene graph.

Effect of Node Budget. To balance subgraph compactness with evidence coverage, we determine the node budget K by using the distribution of ground-truth sizes in the GQA training set. Specifically, we set $K = 3$, corresponding to the 95-th percentile of this distribution. As shown in Figure 6, accuracy peaks at 85.9% when $K = 3$. Reducing K to 1 causes a substantial drop to 70.1%, indicating that a single node is insufficient to capture the relational evidence for the question. In contrast, increasing K beyond 3 leads to a gradual performance decrease, as additional selections are more likely to introduce noisy nodes that interfere with reasoning. $K = 3$ strikes a good balance between sufficiency and subgraph compactness.

Effect of Scene Graph Input. We evaluate different scene graph input settings on GQA. Adding scene graphs improves accuracy from 61.2% with the image and question alone to 81.9%. Using ground-truth evidence subgraphs yields 90.6% accuracy, indicating the additional benefit of selecting supporting evidence for the question. Our method achieves 85.9% accuracy through automatic grounding, providing a further gain of 4.0% over the full scene graph and demonstrating the contribution of evidence grounding to VQA performance.

4.3 More Analyses

Graph Grounding Evaluation. We further evaluate the grounding strategy on three non-visual graph benchmarks without image inputs. As shown in Table 4, our method achieves the best result across all datasets and evaluation metrics. SubgraphRAG improves over Full SG on MetaQA and MetaHotpotQA, but its MolecularIQ accuracy drops by 10.6%-22.1%. This decline suggests that

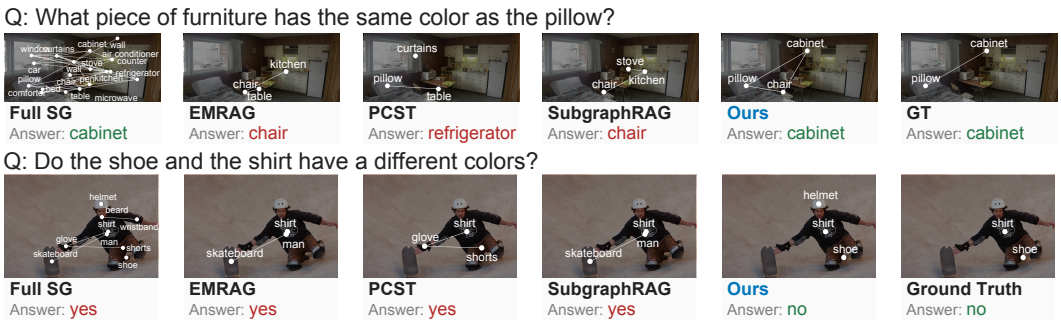

Figure 7: Qualitative comparisons on GQA. Green and red answers indicate correct and incorrect predictions.

Table 5: Accuracy (%) on the POPE-GQA benchmark under Adversarial, Popular, and Random settings.

Method	Adversarial	Popular	Random
Full SG	76.8	76.9	87.9
EMRAG	76.7	76.3	88.0
PCST	78.1	77.3	87.5
SubgraphRAG	77.0	77.1	84.4
Ours	**78.2**	**77.8**	**88.9**

Table 6: Evaluation of subgraph quality measured using Complete, Precision, Recall, and F1 Score (%).

Method	Complete	Precision	Recall	F1
EMRAG	74.9	43.2	84.0	54.8
PCST	12.2	44.0	27.1	32.4
SubgraphRAG	76.7	48.6	83.7	59.7
Ours	**86.7**	**52.3**	**93.4**	**65.0**

selecting highly relevant triples is inadequate for molecular questions, which require joint reasoning over structural relations. By jointly modeling node-node dependencies and query-node relevance, our method improves MolecularIQ accuracy over Full SG by 3.8%-14.5%. The consistent gains across knowledge-graph and molecular-graph benchmarks demonstrate that our grounding mechanism generalizes beyond visual scenes and remains effective across graph structures and reasoning tasks.

Hallucination Evaluation. We evaluate model hallucination on the GQA subset of POPE (Li et al., 2023b), which measures object-level hallucination under random, popular, and adversarial sampling settings. As shown in Table 5, our method achieves the highest accuracy, reaching 78.2% on the challenging Adversarial split, 77.8% on the Popular split, and 88.9% on the Random split. Compared with Full SG, our method improves accuracy by 1.4%, 0.9%, and 1.0%, respectively. These results indicate that selective grounding is more effective for object-level hallucination than the model using the complete scene graph, whose additional context may introduce unnecessary information and interfere with evidence selection (Zhou et al., 2024; Wu et al., 2024b; Tong et al., 2024).

Evaluation on Subgraph Quality. We assess the quality of the grounded subgraphs by measuring their alignment with the ground-truth subgraphs using Complete, Precision, Recall, and F1-score. Complete denotes the percentage of questions for which the selected subgraph contains all ground-truth evidence nodes. As shown in Table 6, our method achieves a Complete score of 86.7%, indicating that it retrieves the full set of supporting evidence for most questions. Compared with SubgraphRAG, it improves Precision by 3.7% and F1-score by 5.3%, while also increasing Recall by 9.7%. These results indicate that our method provides a better balance between retrieving evidence and excluding extraneous nodes, yielding higher-quality grounded subgraphs overall.

Qualitative Analysis. As shown in Figure 7, the compared methods sometimes miss key evidence needed to answer the question. Although Full SG contains all scene information, irrelevant context may mislead the model when answering more complex questions. Our method preserves the evidence needed to answer the question in compact subgraphs while reducing irrelevant information.

5 Conclusion

In this paper, we have presented a scene graph grounding framework for factual visual question answering. We introduce a dual-differential attention mechanism, which jointly models node-node dependencies and query-node relevance to select a compact subgraph while reducing the influence of task-irrelevant information. To alleviate the scarcity of grounding supervision, we further construct a factual-counterfactual dataset that differentiates supporting evidence from misleading visual cues. Extensive experiments across multiple benchmarks demonstrate the effectiveness of the proposed method. In future work, we will investigate its generalization to open-domain visual reasoning and more complex real-world scenarios.

AI USE STATEMENT

In this work, we used generative AI tools to polish the manuscript, including correcting grammatical errors and improving clarity and readability. We also used generative AI tools to generate counterfactual training data, following the procedure described in Section 3.3. We have reviewed AI-assisted work for accuracy and consistency. We take responsibility for the final content of this work, including the text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, QianYing Wang, Ping Chen, Xiaoqin Zhang, and Shijian Lu. Mitigating object hallucinations in large vision-language models with assembly of global and local attention. In *CVPR*, pp. 29915–29926, 2025.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pp. 6077–6086, 2018.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, pp. 2425–2433, 2015.

Christoph Bartmann, Johannes Schimunek, Mykyta Ielanskyi, Philipp Seidl, Günter Klambauer, and Sohvi Luukkonen. MolecularIQ: Characterizing chemical reasoning capabilities through symbolic verification on molecular graphs. In *ICLR*, 2026.

Kejia Chen, Jiawen Zhang, Jiacong Hu, Kewei Gao, Jian Lou, Zunlei Feng, and Mingli Song. Token-level inference-time alignment for vision-language models. *arXiv preprint arXiv:2510.21794*, 2025.

Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Changwen Chen. Gpt4sgg: Synthesizing scene graphs from holistic and region-specific narratives. *arXiv preprint arXiv:2312.04314*, 2023.

Federico Cocchi, Nicholas Moratelli, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Augmenting multimodal llms with self-reflective tokens for knowledge-based visual question answering. In *CVPR*, pp. 9199–9209, 2025.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, pp. 457–468, 2016.

Junqi Ge, Ziyi Chen, Jintao Lin, Jinguo Zhu, Xihui Liu, Jifeng Dai, and Xizhou Zhu. V2pe: Improving multimodal long-context capability of vision-language models with variable visual position encoding. In *ICCV*, pp. 21070–21084, 2025.

Jiayi He, Hehai Lin, Qingyun Wang, Yi R Fung, and Heng Ji. Self-correction is more than refinement: A learning framework for visual and language reasoning tasks. In *ACL*, pp. 6405–6421, 2025.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *NeurIPS*, volume 37, pp. 132876–132907, 2024.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *CVPR*, pp. 13418–13427, 2024.

Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. In *NeurIPS*, volume 32, pp. 5901–5914, 2019a.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pp. 6700–6709, 2019b.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pp. 2901–2910, 2017.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *NeurIPS*, volume 31, pp. 1571–1581, 2018.

Rajat Koner, Hang Li, Marcel Hildebrandt, Volker Tresp, and Stephan Günnemann. Graphhopper: Multi-hop scene graph reasoning for visual question answering. *ISWC*, 12922:111–127, 2021.

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *CVPR*, pp. 9098–9108, 2025a.

Jiaming Li, Jiacheng Zhang, Zequn Jie, Lin Ma, and Guanbin Li. Mitigating hallucination for large vision language model by inter-modality correlation calibration decoding. *arXiv preprint arXiv:2501.01926*, 2025b.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pp. 19730–19742, 2023a.

Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *ICLR*, volume 2025, pp. 6061–6089, 2025c.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, pp. 292–305, 2023b.

Zhuowei Li, Haizhou Shi, Yunhe Gao, Di Liu, Zhenting Wang, Yuxiao Chen, Ting Liu, Long Zhao, Hao Wang, and Dimitris N Metaxas. The hidden life of tokens: Reducing hallucination of large vision-language models via visual information steering. In *ICML*, 2025d.

Weixin Liang, Yanhao Jiang, and Zixuan Liu. Graphvqa: Language-guided graph neural networks for graph-based visual question answering. In *MAIWorkshop*, pp. 79–86, 2021.

Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. Revive: Regional visual representation matters in knowledge-based visual question answering. *NeurIPS*, 35:10560–10571, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, volume 36, pp. 34892–34916, 2023.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *TACL*, 12:157–173, 2024.

Fan Lu, Wei Wu, Kecheng Zheng, Shuailei Ma, Biao Gong, Jiawei Liu, Wei Zhai, Yang Cao, Yujun Shen, and Zheng-Jun Zha. Benchmarking large vision-language models via directed scene graph for comprehensive image captioning. In *CVPR*, pp. 19618–19627, 2025.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *NeurIPS*, volume 29, pp. 289–297, 2016.

Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *ECCV*, pp. 252–269, 2024.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *CVPR*, pp. 14420–14431, 2024.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, pp. 299–307, 2017.

Sai Vidyaranya Nuthalapati, Ramraj Chandradevan, Eleonora Giunchiglia, Bowen Li, Maxime Kayser, Thomas Lukasiewicz, and Carl Yang. Lightweight visual question answering using scene graphs. In *CIKM*, pp. 3353–3357, 2021.

Woohyeon Park, Woojin Kim, Jaeik Kim, and Jaeyoung Do. Second: Mitigating perceptual hallucination in vision-language models via selective and contrastive decoding. In *ICML*, 2025.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding multimodal large language models to the world. In *ICLR*, 2024.

Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *NeurIPS*, volume 28, pp. 2953–2961, 2015.

Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. Kvqa: Knowledge-aware visual question answering. In *AAAI*, volume 33, pp. 8876–8884, 2019.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *ICML*, pp. 31210–31227, 2023.

Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *CVPR*, pp. 8376–8384, 2019.

Kevin J Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, pp. 4613–4621, 2016.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *ECCV*, pp. 256–274, 2024.

Agnese Taluzzi, Davide Gesualdi, Riccardo Santambrogio, Chiara Plizzari, Francesca Palermo, Simone Mentasti, and Matteo Matteucci. From pixels to graphs: using scene and knowledge graphs for hd-epic vqa challenge. *arXiv preprint arXiv:2506.08553*, 2025.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, pp. 9568–9578, 2024.

Flavio Valerio, Lucia Siciliani, Eleonora Ghizzota, Pierpaolo Basile, and Pasquale Lops. Metahotpotqa: A schema-aware multi-hop knowledge graph question answering benchmark, 2026.

Zihu Wang, Boxun Xu, Yuxuan Xia, and Peng Li. VEGAS: Mitigating hallucinations in large vision-language models via vision-encoder attention guided adaptive steering. *arXiv preprint arXiv:2512.12089*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, volume 35, pp. 24824–24837, 2022.

Junjie Wu, Tsz Ting Chung, Kai Chen, and Dit-Yan Yeung. Unified triplet-level hallucination evaluation for large vision-language models. *arXiv preprint arXiv:2410.23114*, 2024a.

Mingrui Wu, Jiayi Ji, Oucheng Huang, Jiale Li, Yuhang Wu, Xiaoshuai Sun, and Rongrong Ji. Evaluating and analyzing relationship hallucinations in large vision-language models. In *ICML*, pp. 53553–53570, 2024b.

Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Visual question answering: A survey of methods and datasets. *CVIU*, 163:21–40, 2017.

Jiaer Xia, Bingkui Tong, Yuhang Zang, Rui Shao, and Kaiyang Zhou. Bootstrapping grounded chain-of-thought in multimodal llms for data-efficient model adaptation. *arXiv preprint arXiv:2507.02859*, 2025.

Yuxi Xie, Guanzhen Li, Xiao Xu, and Min-Yen Kan. V-dpo: Mitigating hallucination in large vision language models via vision-guided direct preference optimization. In *EMNLP*, pp. 13258–13273, 2024.

Junxiao Xue, Quan Deng, Fei Yu, Yanhao Wang, Jun Wang, and Yuehua Li. Enhanced multimodal rag-llm for accurate visual question answering. *arXiv preprint arXiv:2412.20927*, 2024.

Dongil Yang, Minjin Kim, Sunghwan Mac Kim, Beong-woo Kwak, Minjun Park, Jinseok Hong, Woontack Woo, and Jinyoung Yeo. Llm meets scene graph: Can large language models understand and generate scene graphs? a benchmark and empirical study. In *ACL*, pp. 21335–21360, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pp. 2369–2380, 2018.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *CVPR*, pp. 21–29, 2016.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *CVPR*, pp. 13807–13816, 2024.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, pp. 6720–6731, 2019.

Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Leizhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *ECCV*, pp. 19–35, 2024.

Yifeng Zhang, Ming Jiang, and Qi Zhao. Explicit knowledge incorporation for visual reasoning. In *CVPR*, pp. 1356–1365, 2021.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *AAAI*, volume 32, pp. 6069–6076, 2018.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. In *ICLR*, 2024.

A Experimental Settings

A.1 Datasets

We conduct experiments on five VQA benchmarks and one object-hallucination benchmark. **GQA** (Hudson & Manning, 2019b) is a large-scale VQA benchmark. We use its Balanced split, comprising 113k images and 1.7M questions. It evaluates complex reasoning over multiple objects and their relations in realistic scenes. **CLEVR** (Johnson et al., 2017) is a synthetic dataset of 100k rendered images and 850k questions over simple 3D objects in controlled environments. **VCR** (Zellers et al., 2019) provides 110k images and 290k question-answer pairs that require high-level reasoning and commonsense inference about human actions and events. **COCO-QA** (Ren et al., 2015) consists of 78,736 training questions and 38,948 test questions automatically derived from MS COCO captions, each is answered with a single word and targets object category, count, color, or location. **VQA-Abstract** (Antol et al., 2015) offers 50k clipart scenes and 150k questions probing object attributes, human poses, and spatial relations. Finally, **POPE-GQA** is the GQA subset of POPE (Li et al., 2023b), an object-hallucination benchmark. We report accuracy on object-presence questions under the Random, Popular, and Adversarial sampling settings.

To assess grounding beyond visual tasks, we evaluate on three graph-structured benchmarks, each pairing a question with a graph context and no image provided. **MolecularIQ** (Bartmann et al., 2026) identifies atoms or substructures in molecular graphs, with atoms as nodes and chemical bonds as edges. **MetaQA** (Zhang et al., 2018) is a multi-hop question-answering benchmark over a movie knowledge graph. **MetaHotpotQA** (Valerio et al., 2026) grounds HotpotQA (Yang et al., 2018) questions in Wikidata and supplies a local candidate subgraph for each question. We report accuracy on MolecularIQ, Hits@1 and answer F1 on MetaQA, and exact match and token-level F1 on MetaHotpotQA.

A.2 Comparison Methods

We compare our method with four baselines:

- **Full SG** feeds the complete scene graph to the answering model.
- **EMRAG** retrieves textual descriptions of scene-graph objects and relations by semantic similarity to the question and uses them as context for answer generation (Xue et al., 2024).
- **PCST** is the subgraph retrieval module of G-Retriever (He et al., 2024). It assigns prizes to nodes and edges according to their relevance to the question and solves a prize-collecting Steiner tree problem to obtain a compact connected subgraph.
- **SubgraphRAG** scores all triples in parallel with a lightweight multilayer perceptron that combines semantic representations with directional structural distances and assembles the top-scoring triples into a subgraph for answer generation (Li et al., 2025c).

The main comparisons are conducted with four models, including Gemini 2.5 Flash Lite, GPT-4.1 Mini, GPT-4o, and the open-source model Qwen3.5 4B. Within each setting, all methods share the same inputs, answering model, and graph serialization.

A.3 Implementation Details

We run all experiments on eight NVIDIA RTX 3080 GPUs. The model is trained for 10 epochs with AdamW, using a batch size of 200 and a base learning rate of 10^{-4}. The learning rate warms up linearly over the first epoch and then decays. The warmup starting rate and minimum learning rate are both 10^{-5}. Unless otherwise specified, we use Gemini 2.5 Flash Lite as the answering model and conduct all ablations on GQA. We adopt the ground-truth scene graphs provided by GQA and CLEVR. GPT4SGG (Chen et al., 2023) is used to extract scene graphs for VCR, COCO-QA, and VQA-Abstract and to generate predicted graphs for GQA and CLEVR. Questions, node descriptions, and relation descriptions are encoded with a frozen Sentence-BERT, and the answering model is kept frozen during training. For counterfactual supervision, we assign weights of +2, +1, -1, and -2 to key evidence, shared, irrelevant, and distracting nodes. All subgraph selection methods use the same node budget on each dataset, except PCST, whose subgraph size is determined by its prize-collecting objective.

A.4 PROMPTS FOR FACTUAL-COUNTERFACTUAL DATASET CONSTRUCTION.

We use Gemini 2.5 Flash Lite to construct a factual-counterfactual dataset offline, providing supervision for training the graph selector described in Section 3.3. Given a question, its correct answer, and the scene graph, the first prompt generates a plausible but incorrect answer A^-. The second prompt is then applied with the correct answer $a = A$ and the incorrect answer $a = A^-$ to extract $\mathcal{S}^+$ and $\mathcal{S}^-$, respectively. These paired subgraphs determine the node-level training weights used to distinguish supporting evidence from distracting cues.

In the templates below, {Q}, {A}, and {G} denote the question, correct answer, and serialized scene graph, respectively; {a} denotes the answer supplied to the extraction prompt, either the correct answer A or the counterfactual answer A^-.

Prompt 1: Counterfactual Answer Generation

Task: Propose one plausible but incorrect answer to a visual question.

Question: {Q}
Correct answer: {A}
Scene graph: {G}

Rules:

1. Yes/no: reverse the label, even when {G} contains no support for the reversed answer.
2. Counting: return a different non-negative integer. The integer need not appear as an object, attribute, or relation in {G}. Treat digit and word forms of the same count as the same answer (e.g., 2 and two).
3. Other open-ended questions (color, object, attribute, or spatial relation): return another value of the same type that explicitly appears in {G}. Do not invent or use a value absent from {G}.
4. The answer must be incorrect, not merely different from {A}. Reject synonyms, paraphrases, and any alternative wording that still answers the question correctly.
5. If multiple valid incorrect answers are available, choose the most plausible distractor.
6. Output strict JSON containing exactly one non-empty answer string. Do not return `null`, an empty string, additional fields, or explanations.

Output: {`"answer": "..."`}

Prompt 2: Answer-Conditioned Subgraph Extraction

Task: Select the minimal set of scene-graph elements (nodes and relations) that directly supports the given answer.

Question: {Q}
Answer: {a}
Scene graph: {G}

Rules:

1. Extraction Scope: Select only valid node IDs and relation IDs that explicitly exist in {G}.
2. Minimality: Keep the smallest set that still supports the answer. Drop any node or relation that is not required for the answer to remain supported.
3. Attribute Grounding: If an attribute supports the answer, include the node ID of the object that owns it. If objects or attributes alone suffice, leave `"relation_ids"` empty.
4. Structural Integrity: For every selected relation ID, both of its endpoint nodes must be included in `"node_ids"`.
5. If the scene graph contains no support for the answer, return empty lists.
6. Output strict JSON only, with no explanation.

Output: {`"node_ids": [], "relation_ids": []`}

B Additional Experiments and Analyses

B.1 Performance across Question Types

We evaluate all comparison methods across GQA question types using Gemini 2.5 Flash Lite. Each GQA question carries two complementary labels. The structural type reflects the final operation in the functional program underlying the question, including *query, verify, choose, logical, and compare*. The semantic type describes what the question asks about, comprising *object, attribute, category, relation, and global*.

As shown in Figure 8, our method attains the highest accuracy on every structural and semantic question type except *choose* (structural) and *global* (semantic), even though it uses only compact subgraphs. EMRAG performs best on *choose* questions, where the alternatives stated in the question give its retriever explicit cues for semantic matching. Full SG performs best on *global*, retaining the entire graph and the scene-level context that global questions require.

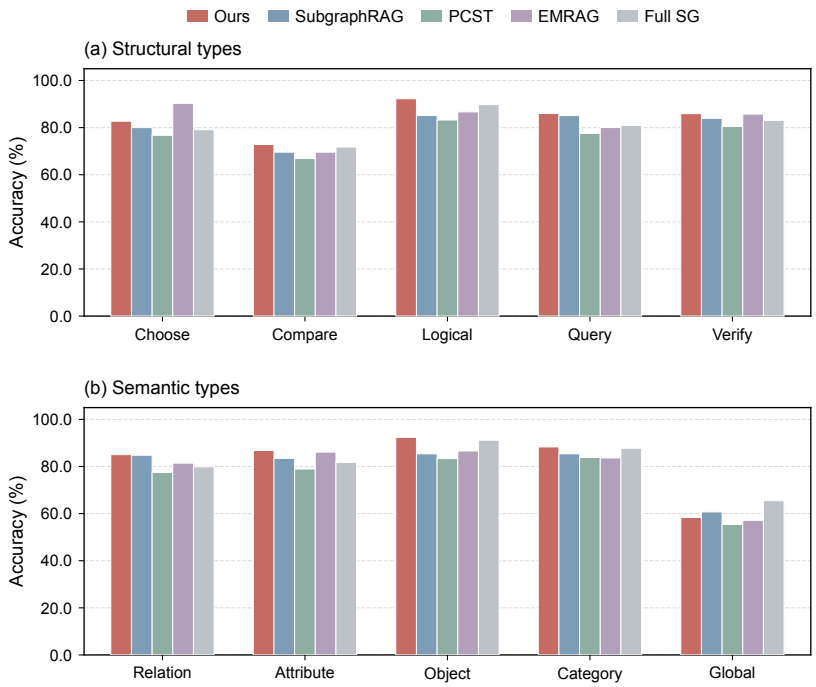

Figure 8: Accuracy (%) across structural and semantic question types on GQA.

We also report mean-per-type accuracy for structural and semantic types. For each group, we compute accuracy within each of its five types and average the five values. As shown in Table 7, our method achieves the highest mean-per-type accuracy in both groups, outperforming EMRAG by 1.5% on structural types and Full SG by 1.0% on semantic types. These results indicate that the advantage holds across both structural and semantic question categories, even though our method uses far more compact subgraphs.

Table 7: Overall and mean-per-type accuracy (%) on GQA.

Method	Overall accuracy ↑	Mean across structural types ↑	Mean across semantic types ↑
Full SG	81.9	80.9	81.2
EMRAG	83.0	82.5	79.0
PCST	78.4	77.0	75.8
SubgraphRAG	83.7	80.8	79.9
Ours	**85.9**	**84.0**	**82.2**

B.2 Zero-Shot Evaluation

To evaluate zero-shot transfer performance, we train the graph selectors of our method and SubgraphRAG on GQA, and conduct testing on CLEVR, COCO-QA, VQA-Abstract, and VCR without further training. Both methods use the node budget K from GQA across all target datasets.

As shown in Table 8, our method outperforms SubgraphRAG on all six metrics, with accuracy gains of 0.4% to 1.1%. These results suggest that our method has stronger cross-dataset transfer ability than SubgraphRAG.

Table 8: Zero-shot accuracy (%) on CLEVR, COCO-QA, VQA-Abstract, and VCR after training on GQA.

Method	CLEVR	COCO-QA	VQA-Abstract	VCR (Q-A)	VCR (QA-R)	VCR (Q-AR)
SubgraphRAG	61.4	64.7	80.4	52.8	53.5	28.9
Ours	**62.5**	**65.1**	**81.4**	**53.5**	**54.1**	**29.9**
Δ	+1.1	+0.4	+1.0	+0.7	+0.6	+1.0

B.3 Robustness to Graph Noise

Distractor nodes. To evaluate robustness to distractor nodes, we randomly copy non-evidence nodes from GQA scene graphs and insert them into each scene graph. Any object whose name appears in the question or the answer, or matches that of an evidence node, is excluded from the candidate pool. Each copied node preserves its original attributes and is attached to a randomly chosen existing node via a relation edge. We vary the number of added distractor nodes from 10% to 300% of the original node count. All methods are evaluated on identical perturbed graphs.

As shown in Figure 9 (left), our method attains the highest accuracy at every noise ratio and exhibits only a marginal decline when distractors are added. In contrast, Full SG and EMRAG degrade as noise increases, and EMRAG falls from third-best to worst. SubgraphRAG and PCST are stable but yield consistently lower accuracy. These results indicate that our method maintains accuracy as distracting information increases, effectively distinguishing key evidence from noise.

Incorrect relations. To evaluate robustness to incorrect relations, we replace spatial relations in GQA scene graphs with their opposites, so that the same pair of objects is assigned a contradictory spatial position, while the nodes, attributes, and number of edges remain unchanged. We increase the proportion of incorrect relations from 10% to 100%, and all methods are evaluated on identical modified graphs.

As shown in Figure 9 (right), accuracy decreases for all methods as the proportion of incorrect spatial relations increases. Our method achieves the highest accuracy and the smallest accuracy drop at every proportion. When all spatial relations are incorrect, it still reaches 80.0%, while the other methods reach 72.4% to 76.7%. These results indicate that our method remains effective when the scene graph contains incorrect relations.

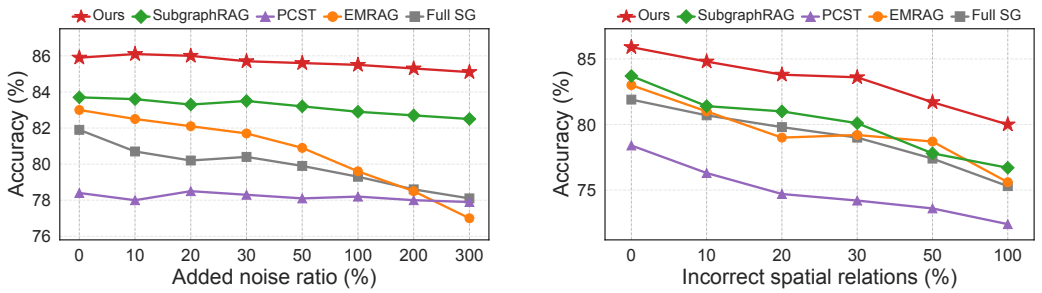

Figure 9: Performance under two types of scene graph noise on GQA. Left: accuracy under added distractor nodes, with noise ratios relative to the original graph size. Right: accuracy under incorrect spatial relations, with noise ratios relative to the number of spatial relations.

B.4 Sensitivity to Missing Evidence

Scene graph generators may omit objects required to answer a question. We simulate this case by randomly deleting the ground-truth evidence nodes of each question from its scene graph, with the deletion probability increasing from 10% to 100%. The image, the question, and all other nodes remain unchanged, and all methods are evaluated on identical modified graphs.

As shown in Table 9, accuracy decreases for all methods as more evidence nodes are deleted. When all evidence nodes are removed, accuracy falls to 49.2% to 54.5%, since the remaining nodes carry no evidence for the question and may mislead the answering model. Nevertheless, our method achieves the highest accuracy at every deletion ratio. These results indicate that our method makes better use of the evidence that remains in incomplete scene graphs, but its advantage is bounded by the evidence available in the graph.

Table 9: Accuracy (%) on GQA under different ratios of deleted evidence nodes.

Method	0%	10%	20%	30%	50%	100%
Full SG	81.9	76.8	72.8	69.6	62.6	49.2
EMRAG	83.0	76.8	74.0	69.7	62.1	53.1
PCST	78.4	74.1	70.7	68.3	61.8	51.3
SubgraphRAG	83.7	78.9	74.8	70.7	62.3	51.2
Ours	**85.9**	**81.2**	**77.0**	**74.1**	**65.6**	**54.5**

B.5 Inference Efficiency

We evaluate inference efficiency on GQA with Gemini 2.5 Flash Lite, measuring latency and input token counts. Restricting the model input to the selected subgraph reduces the average latency from 4,267 ms with the full graph to 3,128 ms. Adding the average grounding time of 73 ms yields a combined latency of 3,201 ms, a 25.0% reduction. Grounding thus accounts for only 2.3% of the combined latency, so the selection cost is small relative to the time saved during answer generation.

Figure 10 compares accuracy against the average number of input tokens on GQA. Our method uses 1,991 tokens on average, versus 4,246 for Full SG, reducing token usage by 53.1% while improving accuracy from 81.9% to 85.9%. PCST is the most token-efficient at 1,702 tokens but reaches only 78.4% accuracy. At the cost of 289 additional tokens over PCST, our method improves 7.5% accuracy, indicating that selectively retaining graph information provides a good balance between token efficiency and accuracy.

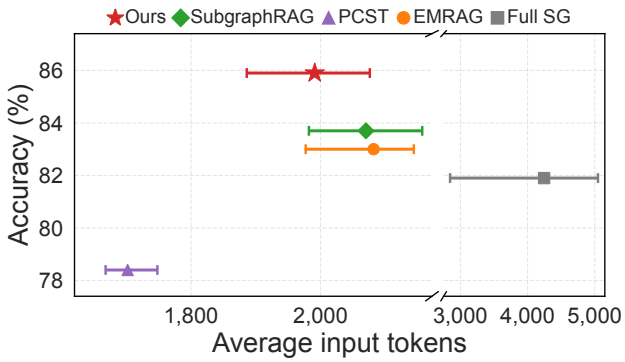

Figure 10: Accuracy versus average total input tokens for different methods on the GQA dataset. Horizontal bars indicate the 25th to 75th percentile range of input token counts.

B.6 Analysis of Counterfactual Supervision

Quality of factual-counterfactual data. To assess the quality of factual-counterfactual supervision, we sample 100 GQA training examples, use an LLM to generate their factual subgraphs $\mathcal{S}^+$,

counterfactual answers A^-, and counterfactual subgraphs $\mathcal{S}^-$, and manually examine the generated results. For each factual subgraph, we check whether $\mathcal{S}^+$ contains the evidence needed to support the correct answer. For each answer, we check whether A^- is a reasonable but incorrect candidate for the question. For each counterfactual subgraph, we check whether its objects and relations in $\mathcal{S}^-$ are consistent with the scene graph and support A^-.

As shown in Table 10, the factual subgraph $\mathcal{S}^+$ is valid in 95.0% of the samples, the counterfactual answer A^- in 94.0%, and the counterfactual subgraph $\mathcal{S}^-$ in 87.0%. These results indicate that most generated samples are valid, providing effective supervision for learning to separate supporting evidence from distracting cues.

Although $\mathcal{S}^-$ contains more noise than $\mathcal{S}^+$, its errors exert a limited impact on training. Under our tiered weighting scheme, $\mathcal{S}^+$ dictates the supervisory sign for each node by assigning positive weights ($+2$ or $+1$) to nodes within $\mathcal{S}^+$ and negative weights (-2 or -1) to the remaining nodes. In contrast, $\mathcal{S}^-$ only modulates the magnitude of these penalties and rewards. Consequently, errors in $\mathcal{S}^-$ merely scale the gradients without altering their direction, whereas noise in $\mathcal{S}^+$ risks flipping genuine evidence nodes into false negatives.

Table 10: Quality of LLM-generated factual subgraphs, counterfactual answers, and counterfactual subgraphs on 100 GQA training examples.

Criterion satisfied	Proportion (%)
Factual subgraph $\mathcal{S}^+$	95.0
Counterfactual answer A^-	94.0
Counterfactual subgraph $\mathcal{S}^-$	87.0

Effectiveness of counterfactual supervision. To evaluate whether counterfactual supervision helps the selector distinguish key evidence from distracting evidence, we compare the node selection logits $\ell_i^{(1)}$ for key evidence ($\mathcal{S}^+ \setminus \mathcal{S}^-$), shared nodes ($\mathcal{S}^+ \cap \mathcal{S}^-$), distracting evidence ($\mathcal{S}^- \setminus \mathcal{S}^+$), and irrelevant nodes ($V \setminus (\mathcal{S}^+ \cup \mathcal{S}^-)$) at the first selection step. For each category, we first average the logits within each question and then average across all questions that contain nodes in that category. As shown in Table 11, key evidence receives a mean logit of 63.0, compared with 17.6 for distracting evidence. Shared nodes have a mean logit between these two values, while irrelevant nodes receive the lowest mean logit. These results are consistent with the objective of counterfactual supervision to encourage the selection of key evidence and suppress distracting evidence.

Table 11: Mean node selection logits for the four node categories at the first selection step.

Node category	Mean logit
Key evidence ($\mathcal{S}^+ \setminus \mathcal{S}^-$)	63.0
Shared nodes ($\mathcal{S}^+ \cap \mathcal{S}^-$)	46.8
Distracting evidence ($\mathcal{S}^- \setminus \mathcal{S}^+$)	17.6
Irrelevant nodes ($V \setminus (\mathcal{S}^+ \cup \mathcal{S}^-)$)	-5.4

B.7 Effect of Expectation Terms in DDCA

We compare the full DDCA model with three variants to evaluate how the intervention-based expectation terms affect evidence selection. The expectation terms $\bar{f}_\theta(v_j)$ and $\bar{g}_\theta(v_i)$ in Eq. (5) are estimated by averaging scores under interventions. We replace only these two terms with the following alternatives.

Without expectation terms. We set both expectation terms to zero and keep the raw MLP scores from two scoring branches. This removes the differential operation while maintaining both learnable branches, thereby testing whether the expectation terms add anything beyond the branch scores themselves.

Graph mean. We replace the expectation terms with the average association score over connected node pairs in the current graph and the average relevance score between the question and all nodes.

These averages reflect the overall score level of the graph, but ignore how individual nodes respond to interventions, testing whether the gain comes from adjusting the overall score level or from comparing each node's scores under the original and intervention conditions.

Shuffled expectation terms. We compute the expectation values as in the full model. Within each branch, we randomly permute them across nodes of the same graph. This preserves the value distribution while breaking the correspondence between a node and its own intervention value, testing whether the gain depends on this correspondence rather than on the distribution alone.

As shown in Table 12, the full DDCA model achieves an F1 score of 65.0%, compared with 64.2% without expectation terms, indicating that the terms improve evidence selection beyond the learned branch scores. The graph mean reaches 64.1%, no better than removing the terms entirely, which suggests that calibrating the overall score level captures little of this signal. Shuffling the expectation terms reduces F1 to 63.2%, below the variant without expectation terms. Preserving the values and their distribution is not sufficient, and what matters is pairing each node's score with its own intervention value. Overall, these results support the DDCA design of comparing each node's scores under the original and intervention conditions.

Table 12: Evidence selection performance (%) on GQA with different expectation terms in DDCA.

Variant	Complete	Precision	Recall	F1
Without expectation terms	85.8	51.6	92.6	64.2
Graph mean	86.1	51.5	92.6	64.1
Shuffled expectation terms	84.3	50.7	91.5	63.2
DDCA (Ours)	**86.7**	**52.3**	**93.4**	**65.0**

B.8 Debiasing Effectiveness of DDCA

We compare the full model against a variant without DDCA to evaluate whether DDCA mitigates structural and linguistic biases. We measure evidence selection performance by using the F1 score between the selected node set and the ground-truth node set.

Structural bias. When evidence nodes tend to have more connections in the training data, the model may use node degree as a shortcut for selecting evidence. We test this bias on GQA, where nodes differ in degree, and define node degree as the number of distinct objects connected to a node. We build a training set with 3,600 samples in which the maximum degree among evidence nodes exceeds that among distracting nodes by at least two, and 400 samples in which it falls short by at least two. This distribution encourages the model to associate higher degree with being evidence. At test time, we evaluate on 800 samples from each group, focusing on whether DDCA can select the correct evidence when distractor nodes have higher degrees.

Linguistic bias. Frequent training answers may likewise guide node selection. For example, a model trained mainly on color questions whose answer is often red may favor red objects, even when they do not match the location named in the question. We use CLEVR color and shape questions, which have clearly defined answer categories. For each question type, we rank the categories by training frequency and treat the top half as common answers and the rest as rare answers. We sample 3,600 training examples with common answers and 400 with rare answers, thereby biasing the model toward nodes associated with common answers. Evaluation uses 800 examples per group, with attention to whether DDCA selects the correct evidence for rare answers.

As shown in Table 13, the two methods achieve similar training F1. On GQA, the biased training distribution misleads the model into associating higher node degrees with being evidence. Consequently, when distractor nodes possess higher degrees at test time, the train-to-test F1 gap reaches 18.7% without DDCA. Incorporating DDCA narrows this gap to 13.7% and improves test F1 from 53.3% to 57.6%, indicating that DDCA mitigates structural bias in evidence selection. Furthermore, on CLEVR, DDCA reduces the train-to-test F1 gap in the rare answer group from 11.9% to 6.4% and raises test F1 from 63.6% to 69.4%, suggesting it also mitigates linguistic bias.

Table 13: Effect of DDCA on evidence selection under structural and linguistic biases. We report training and test F1 (%) between selected and ground-truth evidence nodes. Drop denotes training F1 minus test F1.

(a) Structural bias on GQA

	Higher-degree evidence nodes			Higher-degree distractor nodes		
Method	Train F1	Test F1 ↑	Drop ↓	Train F1	Test F1 ↑	Drop ↓
w/o DDCA	81.1	72.9	8.2	72.0	53.3	18.7
DDCA	81.0	**73.0**	**8.0**	71.3	**57.6**	**13.7**

(b) Linguistic bias on CLEVR

	Common answers			Rare answers		
Method	Train F1	Test F1 ↑	Drop ↓	Train F1	Test F1 ↑	Drop ↓
w/o DDCA	81.6	73.8	7.8	75.5	63.6	11.9
DDCA	82.7	**75.6**	**7.1**	75.8	**69.4**	**6.4**

B.9 Qualitative Comparisons

Figure 11 visualizes the graph context by comparison methods and their resulting answers. Our method preserves the evidence required to answer each question within compact subgraphs, while reducing irrelevant context.

In the fifth example, the question is whether the bowl and wall have different colors, which requires locating both objects and comparing color attributes. EMRAG and SubgraphRAG focus on people and clothing and maintain neither comparison target. PCST requires a connected subgraph and therefore keeps only one of the two objects. Full SG contains both objects yet still gives an incorrect answer, suggesting that the irrelevant context may mislead the answering model. Our method keeps both objects in a compact subgraph and provides the correct answer.

Figure 12 illustrates how the model progressively constructs a subgraph. In the fourth example, the question asks what is around the bench in front of the snow. The model first selects the snow as a reference for locating the bench, then the bench that stands in front of it, and finally the fence, which provides the answer. This order moves from the reference used to localize the target to the object that answers the question. The resulting subgraph preserves both the relations, including snow–bench for localization and bench–fence for answering.

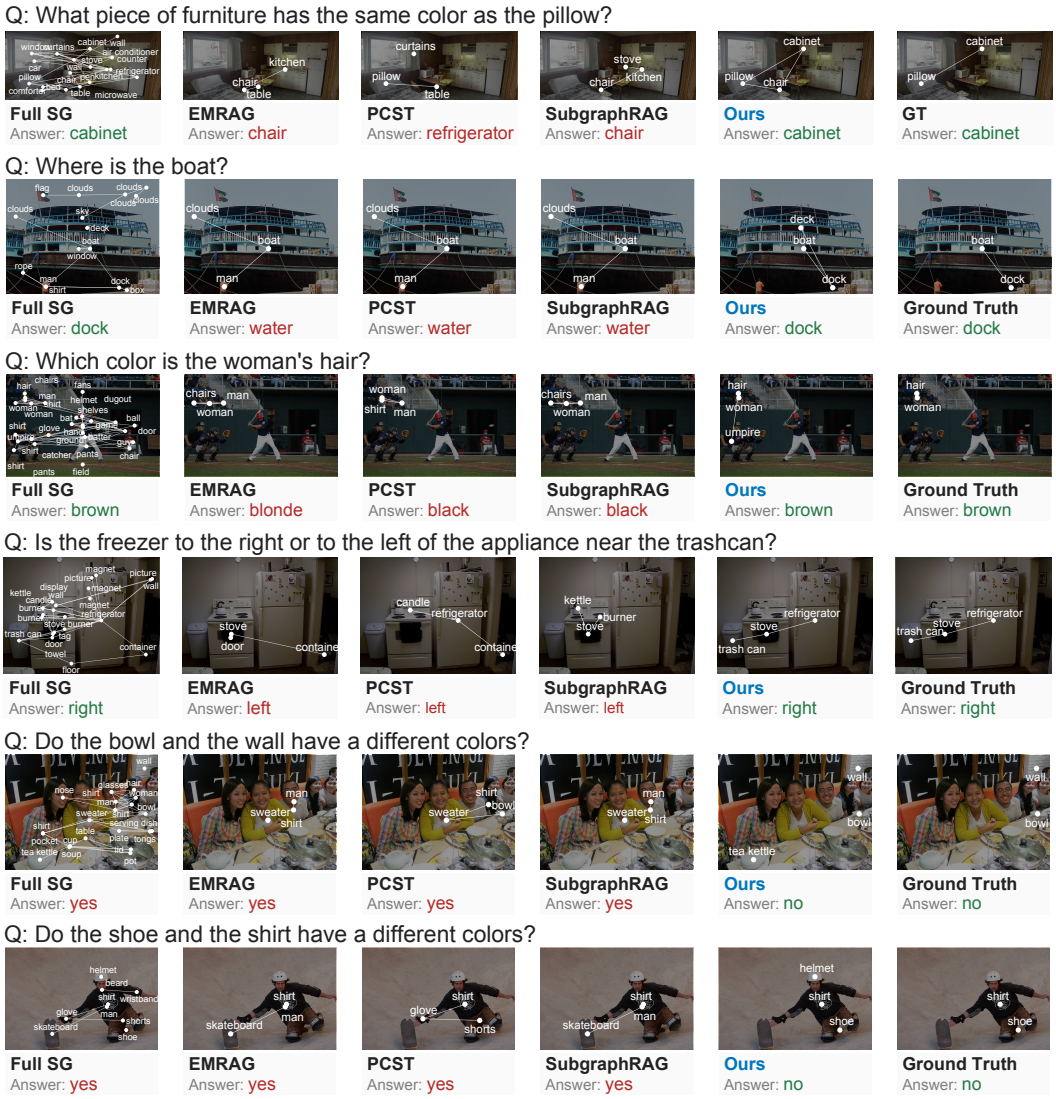

Figure 11: Qualitative comparisons on GQA. Columns show Full SG, EMRAG, PCST, SubgraphRAG, our method, and the ground-truth evidence subgraph. Overlaid nodes and edges visualize graph context; answers in green and red indicate correct and incorrect predictions, respectively. Our selections preserve the objects needed for the question while excluding much of the unrelated context.

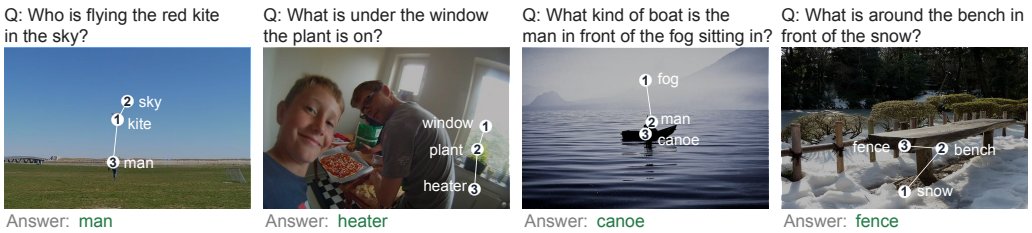

Figure 12: Iterative evidence selection on GQA. Node numbers indicate the selection order, and edges show relations among the selected objects. The examples illustrate the progression from localization references to answer-bearing targets. Answers are shown below each example.

C ALGORITHM PSEUDOCODE

As shown in Algorithm 1, our pipeline is organized into sequential steps. **Step 1** uses an *offline* LLM $\mathcal{M}_{\text{LLM}}$ to construct a factual–counterfactual dataset, producing evidence node sets and node weights. **Step 2** trains only the scene graph grounding model g_Θ on the offline supervision, and this step does not call the VLM. **Step 3** performs inference. We first use g_Θ to select a question-responsive subgraph and then call a frozen VLM $\mathcal{M}$ with $(I, Q, \mathcal{G}_S)$ to generate the final answer.

Algorithm 1 Pipeline of the proposed method

Input: training set $\mathcal{D} = \{(I, Q, \mathcal{G}, A)\}$.
Input: offline LLM $\mathcal{M}_{\text{LLM}}$.
Input: frozen VLM $\mathcal{M}$.
Input: grounding model g_Θ; iterations K; temperature τ.
Output: trained grounding parameters Θ.

Step 1: Offline factual-counterfactual dataset construction (one-time).
Initialize $\mathcal{D}_{cf} \leftarrow \emptyset$.
for all $(I, Q, \mathcal{G}, A) \in \mathcal{D}$ **do**
  Derive counterfactual answer $A^- \leftarrow \text{GETCOUNTERFACTUAL}(Q, \mathcal{G}, A)$.
  $\mathcal{S}^+ \leftarrow \text{LLM_EXTRACTEVIDENCE}(\mathcal{M}_{\text{LLM}}, Q, \mathcal{G}, A)$
  $\mathcal{S}^- \leftarrow \text{LLM_EXTRACTEVIDENCE}(\mathcal{M}_{\text{LLM}}, Q, \mathcal{G}, A^-)$
  $\omega(\cdot) \leftarrow \text{BUILDNODEWEIGHTS}(V, \mathcal{S}^+, \mathcal{S}^-)$.
  $\mathcal{D}_{cf} \leftarrow \mathcal{D}_{cf} \cup \{(I, Q, \mathcal{G}, \mathcal{S}^+, \mathcal{S}^-, \omega)\}$.
end for

Step 2: Train grounding model g_Θ on $\mathcal{D}_{cf}$ (no VLM calls).
repeat
  for all mini-batch $\mathcal{B} \subset \mathcal{D}_{cf}$ **do**
    $\mathcal{L}_{\text{batch}} \leftarrow 0$.
    for all $(I, Q, \mathcal{G}, \mathcal{S}^+, \mathcal{S}^-, \omega) \in \mathcal{B}$ **do**
      $m^{(0)} \leftarrow \mathbf{0}$; $\tilde{p} \leftarrow \mathbf{0}$.
      for $t = 1$ **to** K **do**
        $\ell^{(t)} \leftarrow \text{GROUNDINGLOGITS}(g_\Theta, I, Q, \mathcal{G}, m^{(t-1)})$
        $\widetilde{\boldsymbol{\ell}}^{(t)} \leftarrow \ell^{(t)}$; set $\widetilde{\ell}_i^{(t)} \leftarrow -\infty$ for all i with $m_i^{(t-1)} = 1$.
        $p^{(t)} \sim \text{GUMBELSOFTMAX}(\widetilde{\boldsymbol{\ell}}^{(t)}, \tau)$.
        $\tilde{p} \leftarrow \tilde{p} + p^{(t)} \odot (1 - \tilde{p})$.
        $m^{(t)} \leftarrow m^{(t-1)} \vee \text{ONEHOT}(\arg\max p^{(t)})$.
      end for
      $\mathcal{L} \leftarrow -\sum_{i \in V} \omega(i)\, \tilde{p}_i$.
      $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \mathcal{L}$.
    end for
    Update Θ by backpropagation on $\mathcal{L}_{\text{batch}}$
  end for
until max epochs

Step 3: Inference (grounding → VLM).
Given $(I, Q, \mathcal{G})$, select nodes $\mathcal{S} \leftarrow \text{SELECTNODES}(g_\Theta, I, Q, \mathcal{G}, K, \tau)$.
Construct subgraph $\mathcal{G}_S \leftarrow \text{INDUCESUBGRAPH}(\mathcal{G}, \mathcal{S})$.
$\hat{A} \leftarrow \text{VLM_ANSWER}(\mathcal{M}, I, Q, \mathcal{G}_S)$

D THEORETICAL JUSTIFICATION OF DUAL-DIFFERENTIAL CAUSAL MECHANISM

In this section, we analyze the differential scores in Eqs. (3)–(5) and the node-selection logits in Eq. (6). We show that, when the scoring functions decompose additively, the differential operation cancels the bias term that each score shares with its reference. We first derive the node-node and query-node scores, then analyze the finite-sample estimate of the query-node score and the combination of the two scores in Dual-Differential Causal Attention (DDCA).

D.1 NOTATION AND SETUP

Throughout this section, we fix the question Q, the scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the current selection state, and the model parameters, treat any other randomness in the model, such as dropout, as fixed, and omit the iteration superscript. Let

$$f_{ij} = f_\theta(v_i, v_j, e_{ij}), \qquad g_i(\Gamma) = g_\theta(Q, \Gamma, v_i)$$

denote the association score between nodes v_i and v_j, and the alignment of node v_i with Q in a graph Γ, respectively. We derive the node-node score only for node pairs connected by an edge $e_{ij} \in \mathcal{E}$. For pairs without an edge, we set $c_{ij} = 0$, as in the main text.

For the node-node branch, we fix the target node v_j and replace the source node v_i with reference nodes v_k. Let $\mathcal{V}_{\mathrm{ref}}(v_j)$ denote the predefined set of reference nodes, i.e., the nodes v_k for which $f_\theta(v_k, v_j, e_{kj})$ can be computed; we assume this set is nonempty. We write $p_j = p_{\mathrm{ref}}(\cdot \mid v_j)$ for the reference distribution in Eq. (3) and take it to be uniform over $\mathcal{V}_{\mathrm{ref}}(v_j)$. The reference mean is therefore the average score over all nodes in this set. When every node in the graph can serve as a reference, this set is $\mathcal{V}$.

For the query-node branch, we fix Q and the target entity v_i, and sample reference graphs $\widetilde{\mathcal{G}}$ from $P_i = p_{\mathrm{ref}}(\cdot \mid Q, \mathcal{G}, v_i)$ in Eq. (4). Each reference graph is obtained by randomly deleting nodes other than v_i from $\mathcal{G}$. The reference expectation is estimated from a finite number of independent and identically distributed samples.

The original and reference graphs share the same scoring functions and parameters, while the graph-conditioned representations are computed separately for each graph. We assume that all scores and residual terms below have finite expectations under their reference distributions. The only randomness considered in this section comes from sampling the reference graphs, and all expectations are taken over this sampling.

D.2 MATHEMATICAL PROOF FOR THE NODE-NODE SCORE

Step 1: Additive Decomposition of the Score. We assume that the learned association score between a source node v_i and a target node v_j consists of three components:

1. T_{ij}: the relation-specific component between v_i and v_j.
2. $B_{\mathrm{struct}}(v_j)$: an additive structural bias that depends only on the target node v_j and is shared across reference nodes.
3. ϵ_{ij}: the residual term.

The raw score can thus be written as:

$$f_{ij} = T_{ij} + B_{\mathrm{struct}}(v_j) + \epsilon_{ij}. \tag{10}$$

Since $B_{\mathrm{struct}}(v_j)$ depends only on v_j, it remains unchanged when the source node changes. For a reference node v_k, we therefore have:

$$f_\theta(v_k, v_j, e_{kj}) = T_{kj} + B_{\mathrm{struct}}(v_j) + \epsilon_{kj}. \tag{11}$$

Step 2: Shared Bias in the Reference Baseline. The second term in Eq. (3) is the expected association score of the target node v_j over its reference nodes, which we denote by

$$\bar{f}_j = \mathbb{E}_{v_k \sim p_j}\left[f_\theta(v_k, v_j, e_{kj})\right] = \frac{1}{|\mathcal{V}_{\mathrm{ref}}(v_j)|} \sum_{v_k \in \mathcal{V}_{\mathrm{ref}}(v_j)} f_\theta(v_k, v_j, e_{kj}).$$

Because p_j is uniform and the average includes every node in its support, this average equals the reference expectation exactly. Substituting the decomposition in Eq. (11) and applying the linearity of expectation gives:

$$\begin{aligned}\bar{f}_j &= \mathbb{E}_{v_k \sim p_j}\left[T_{kj} + B_{\text{struct}}(v_j) + \epsilon_{kj}\right] \\ &= \mathbb{E}_{v_k \sim p_j}[T_{kj}] + B_{\text{struct}}(v_j) + \mathbb{E}_{v_k \sim p_j}[\epsilon_{kj}] \\ &= \bar{T}_j + B_{\text{struct}}(v_j) + \bar{\epsilon}_j,\end{aligned} \tag{12}$$

where $\bar{T}_j = \mathbb{E}_{v_k \sim p_j}[T_{kj}]$ denotes the average relation-specific component over the reference nodes, and $\bar{\epsilon}_j = \mathbb{E}_{v_k \sim p_j}[\epsilon_{kj}]$ is the mean residual. The structural bias is unchanged by the averaging, since it takes the same value for every reference node v_k.

Step 3: Bias Cancellation in the Differential Score. Substituting the expanded terms into the definition of c_{ij} in Eq. (3) gives:

$$\begin{aligned}c_{ij} &= f_{ij} - \bar{f}_j \\ &= \left(T_{ij} + B_{\text{struct}}(v_j) + \epsilon_{ij}\right) - \left(\bar{T}_j + B_{\text{struct}}(v_j) + \bar{\epsilon}_j\right) \\ &= T_{ij} - \bar{T}_j + \epsilon_{ij} - \bar{\epsilon}_j.\end{aligned} \tag{13}$$

The shared structural bias $B_{\text{struct}}(v_j)$ therefore cancels out. As a result, c_{ij} measures how the relation-specific component between v_i and v_j deviates from its reference mean, together with the residual difference $\epsilon_{ij} - \bar{\epsilon}_j$. Because it is computed over all reference nodes, the reference mean $\bar{f}_j$ contains no sampling error, while the residual difference remains in the differential score.

D.3 Mathematical Proof for the Query-Node Score

Step 1: Additive Decomposition of the Score. Let ζ_i denote the information about the target that is kept in every reference graph, such as its entity category and lexical description. We assume that the additive language-prior term depends only on Q and ζ_i. The alignment score in the original graph can then be written as:

$$g_i(\mathcal{G}) = A_i(\mathcal{G}) + B_{\text{lang}}(Q, \zeta_i) + \eta_i(\mathcal{G}), \tag{14}$$

where $A_i(\mathcal{G})$ denotes the part of the alignment between v_i and Q that is supported by the scene $\mathcal{G}$, $B_{\text{lang}}(Q, \zeta_i)$ is the additive language prior, and $\eta_i(\mathcal{G})$ is the residual term.

A reference graph $\widetilde{\mathcal{G}} \sim P_i$ keeps the question, the target entity, and ζ_i unchanged, and only removes other nodes at random. Under the same decomposition, we have:

$$g_i(\widetilde{\mathcal{G}}) = A_i(\widetilde{\mathcal{G}}) + B_{\text{lang}}(Q, \zeta_i) + \eta_i(\widetilde{\mathcal{G}}). \tag{15}$$

Since the language prior considered here is determined by Q and ζ_i, it takes the same value in the original and reference graphs.

Step 2: Shared Bias in the Reference Baseline. The second term in Eq. (4) is the expected alignment of the same target entity over the reference graphs, which we denote by

$$\bar{g}_i = \mathbb{E}_{\widetilde{\mathcal{G}} \sim P_i}\left[g_i(\widetilde{\mathcal{G}})\right].$$

Substituting Eq. (15) and applying the linearity of expectation gives:

$$\begin{aligned}\bar{g}_i &= \mathbb{E}_{\widetilde{\mathcal{G}} \sim P_i}\left[A_i(\widetilde{\mathcal{G}}) + B_{\text{lang}}(Q, \zeta_i) + \eta_i(\widetilde{\mathcal{G}})\right] \\ &= \mathbb{E}_{\widetilde{\mathcal{G}} \sim P_i}[A_i(\widetilde{\mathcal{G}})] + B_{\text{lang}}(Q, \zeta_i) + \mathbb{E}_{\widetilde{\mathcal{G}} \sim P_i}[\eta_i(\widetilde{\mathcal{G}})] \\ &= \bar{A}_i + B_{\text{lang}}(Q, \zeta_i) + \bar{\eta}_i,\end{aligned} \tag{16}$$

where $\bar{A}_i = \mathbb{E}_{\widetilde{\mathcal{G}} \sim P_i}[A_i(\widetilde{\mathcal{G}})]$ and $\bar{\eta}_i = \mathbb{E}_{\widetilde{\mathcal{G}} \sim P_i}[\eta_i(\widetilde{\mathcal{G}})]$. The language prior $B_{\text{lang}}(Q, \zeta_i)$ is unchanged by the averaging, since it takes the same value for every reference graph $\widetilde{\mathcal{G}}$.

Step 3: Bias Cancellation in the Differential Score. Substituting the expanded terms into the definition of r_{qi} in Eq. (4) gives:

$$\begin{aligned}r_{qi} &= g_i(\mathcal{G}) - \bar{g}_i \\ &= \left(A_i(\mathcal{G}) + B_{\text{lang}}(Q, \zeta_i) + \eta_i(\mathcal{G})\right) \\ &\quad - \left(\bar{A}_i + B_{\text{lang}}(Q, \zeta_i) + \bar{\eta}_i\right) \\ &= A_i(\mathcal{G}) - \bar{A}_i + \eta_i(\mathcal{G}) - \bar{\eta}_i.\end{aligned} \tag{17}$$

The shared language-prior term $B_{\text{lang}}(Q, \zeta_i)$ therefore cancels out. As a result, r_{qi} measures how the scene-supported alignment between v_i and Q in the original graph deviates from its average over the reference graphs, together with the residual difference $\eta_i(\mathcal{G}) - \bar{\eta}_i$.

D.4 Finite-Sample Estimation of the Query-Node Score

The node-node branch uses the full reference mean derived in Section D.2 and requires no sampling. For the query-node branch, we independently draw $B_{\text{q}} \geq 1$ reference graphs $\widetilde{\mathcal{G}}_i^{(1)}, \ldots, \widetilde{\mathcal{G}}_i^{(B_{\text{q}})}$ from P_i. If the same graph is drawn more than once, each draw is counted separately in the average. The reference mean and the differential score are estimated as:

$$\begin{aligned} \widehat{g}_i &= \frac{1}{B_{\text{q}}} \sum_{b=1}^{B_{\text{q}}} g_i(\widetilde{\mathcal{G}}_i^{(b)}), \\ \widehat{r}_{qi} &= g_i(\mathcal{G}) - \widehat{g}_i. \end{aligned} \tag{18}$$

Under the decomposition and reference-preservation assumptions in Section D.3, every sampled score contains the same target-specific term $B_{\text{lang}}(Q, \zeta_i)$. Since this term has coefficient one in the sample average, it also cancels in the sampled differential score:

$$\begin{aligned} \widehat{r}_{qi} = A_i(\mathcal{G}) - \frac{1}{B_{\text{q}}} \sum_{b=1}^{B_{\text{q}}} A_i(\widetilde{\mathcal{G}}_i^{(b)}) \\ + \eta_i(\mathcal{G}) - \frac{1}{B_{\text{q}}} \sum_{b=1}^{B_{\text{q}}} \eta_i(\widetilde{\mathcal{G}}_i^{(b)}). \end{aligned} \tag{19}$$

This cancellation holds for any set of sampled reference graphs, not only in expectation. Different targets may also have different language priors, as long as each prior remains unchanged in the reference graphs of its own target.

With the question, original graph, current selection state, and parameters fixed, taking expectation over the reference sampling gives:

$$\mathbb{E}[\widehat{g}_i] = \bar{g}_i, \qquad \mathbb{E}[\widehat{r}_{qi}] = g_i(\mathcal{G}) - \mathbb{E}[\widehat{g}_i] = r_{qi}.$$

Thus, $\widehat{r}_{qi}$ is an unbiased estimate of the differential score r_{qi} in Eq. (4). If the reference score also has finite variance

$$\sigma_i^2 = \mathrm{Var}_{\widetilde{\mathcal{G}} \sim P_i}\big[g_i(\widetilde{\mathcal{G}})\big],$$

then, since the samples are independent, we have:

$$\mathbb{E}\big[(\widehat{r}_{qi} - r_{qi})^2\big] = \mathrm{Var}(\widehat{g}_i) = \frac{\sigma_i^2}{B_{\text{q}}}.$$

This error comes only from the variation of $A_i(\widetilde{\mathcal{G}}) + \eta_i(\widetilde{\mathcal{G}})$ across reference graphs. Using more reference graphs therefore reduces the mean squared error, without enumerating all possible reference graphs.

D.5 Combination in Dual-Differential Causal Attention

When the query-node references are sampled, the similarities and the full-reference structural scores c_{ij} remain fixed. For each node v_j, the estimate $\widehat{r}_{qj}$ is computed once and used wherever that node appears in Eqs. (5) and (6). The analysis below assumes that the graph has at least two nodes and that all scores are finite before the node-selection mask is applied.

Eq. (5) combines the two differential scores as $u_{ij} = \mathrm{sim}(v_i, v_j) + \lambda_1 c_{ij} + \lambda_2 r_{qj}$. Using the full node-node reference and the sampled query-node reference, the score computed in practice is:

$$\widehat{u}_{ij} = \mathrm{sim}(v_i, v_j) + \lambda_1 c_{ij} + \lambda_2 \widehat{r}_{qj}. \tag{20}$$

Let

$$\widehat{A}_j = \frac{1}{B_{\text{q}}} \sum_{b=1}^{B_{\text{q}}} A_j(\widetilde{\mathcal{G}}_j^{(b)}), \qquad \widehat{\eta}_j = \frac{1}{B_{\text{q}}} \sum_{b=1}^{B_{\text{q}}} \eta_j(\widetilde{\mathcal{G}}_j^{(b)}).$$

For pairs with $e_{ij} \in \mathcal{E}$, substituting Eqs. (13) and (19) into Eq. (20) gives:

$$\begin{aligned}\widehat{u}_{ij} = \mathrm{sim}(v_i, v_j) + \lambda_1\big(T_{ij} - \bar{T}_j\big) + \lambda_2\big(A_j(\mathcal{G}) - \widehat{A}_j\big) \\ + \lambda_1\big(\epsilon_{ij} - \bar{\epsilon}_j\big) + \lambda_2\big(\eta_j(\mathcal{G}) - \widehat{\eta}_j\big).\end{aligned} \tag{21}$$

Under the additive decomposition and reference-preservation assumptions, the structural differential cancels the term $B_{\mathrm{struct}}(v_j)$ shared by the target node, and the query differential cancels the language-prior term $B_{\mathrm{lang}}(Q, \zeta_j)$ shared by the original and sampled reference graphs. What remains are the relation-specific component relative to the full node reference and the scene-supported alignment between v_j and Q relative to the sampled reference graphs. These two terms supplement the similarity score in Eq. (5), and the residuals appear in the last two terms of Eq. (21).

For fixed weights λ_1 and λ_2, we have $\mathbb{E}[\widehat{u}_{ij}] = u_{ij}$. Under the finite-variance condition above,

$$\mathbb{E}\big[(\widehat{u}_{ij} - u_{ij})^2\big] = \lambda_2^2 \mathbb{E}\big[(\widehat{r}_{qj} - r_{qj})^2\big] = \frac{\lambda_2^2 \sigma_j^2}{B_{\mathrm{q}}}.$$

The node-node term adds no sampling error because it uses the full reference.

Sampling error of the node-selection logits. Let $\mathcal{J}_i = \{j \in \{1, \ldots, |\mathcal{V}|\} : j \neq i\}$, and let ℓ_i and $\widehat{\ell}_i$ denote the node logits in Eq. (6) computed with the exact reference expectation and the sampled reference mean, respectively. We write $\widehat{a}_{ij}$ for the attention weights computed from $\widehat{u}_{ij}$. Since both a_{ij} and $\widehat{a}_{ij}$ sum to one over $\mathcal{J}_i$, Eq. (6) gives

$$\widehat{\ell}_i - \ell_i = \lambda_1 \sum_{j \in \mathcal{J}_i} (\widehat{a}_{ij} - a_{ij}) c_{ij} + \lambda_2 (\widehat{r}_{qi} - r_{qi}).$$

The second term is the sampling error of the query-node score of v_i. The first term arises because the attention weights now depend on the sampled scores $\widehat{r}_{qj}$. Since $\widehat{u}_{ij} - u_{ij} = \lambda_2(\widehat{r}_{qj} - r_{qj})$, every attention score in row i changes by at most $|\lambda_2| \max_{j \in \mathcal{J}_i} |\widehat{r}_{qj} - r_{qj}|$. For the softmax, when every score in a row changes by at most δ, the total change of the weights, $\sum_{j \in \mathcal{J}_i} |\widehat{a}_{ij} - a_{ij}|$, is at most 2δ. With $C_i = \max_{j \in \mathcal{J}_i} |c_{ij}|$, the first term is therefore bounded by $2|\lambda_1 \lambda_2| C_i \max_{j \in \mathcal{J}_i} |\widehat{r}_{qj} - r_{qj}|$. Using $(x + y)^2 \leq 2x^2 + 2y^2$, bounding the maximum by the sum, and applying the finite-variance result in Section D.4 to each node, we obtain

$$\mathbb{E}\big[(\widehat{\ell}_i - \ell_i)^2\big] \leq \frac{2\lambda_2^2}{B_{\mathrm{q}}} \Big(4\lambda_1^2 C_i^2 \sum_{j \in \mathcal{J}_i} \sigma_j^2 + \sigma_i^2\Big). \tag{22}$$

Therefore, at a fixed selection step, the mean squared error of the sampled node logit decreases as $1/B_{\mathrm{q}}$. Since the softmax is nonlinear, the sampled logit is not exactly unbiased, but its bias is at most the square root of this error and also vanishes as B_{q} increases. These results apply to the node logits before the node-selection mask is applied. □