technologies

MDPI

Article

DyGEnc: Encoding a Sequence of Textual Scene Graphs to Reason and Answer Questions in Dynamic Scenes

Sergey Linok [1,*], Vadim Semenov [1], Anastasia Trunova [1], Oleg Bulichev [1,2] and Dmitry Yudin [1,3,*]

[1] Moscow Independent Research Institute of Artificial Intelligence (MIRAI), 117218 Moscow, Russia; semenov.v@miriai.org (V.S.); trunova.a@miriai.org (A.T.); bulichev.o@miriai.org (O.B.)
[2] Research Center of the Artificial Intelligence Institute, Innopolis University, 420500 Tatarstan, Russia
[3] Cognitive AI Systems Lab, 123317 Moscow, Russia
* Correspondence: linok.s@miriai.org (S.L.); yudin.d@miriai.org (D.Y.)

Abstract

Analyzing events in dynamic environments poses a fundamental challenge in the development of intelligent agents and robots capable of interacting with humans. Current approaches predominantly rely on visual–text models; however, these methods often capture information implicitly from images, lacking interpretable and structured spatio-temporal object representations and their relationships. To address this issue, we introduce DyGEnc—a novel method for dynamic graph encoding. This method integrates compressed spatio-temporal representation with the cognitive capabilities of large language models. The purpose of this integration is to enable advanced question answering based on sequences of textual scene graphs. Extensive evaluations on the STAR and AGQA datasets demonstrate that DyGEnc improves large language model performance when addressing queries related to the history of human–object interactions. Furthermore, the proposed method can be extended to process input images by leveraging foundation models to extract explicit textual scene graphs, as validated by the evaluation results. We expect these findings to contribute to the development of robust and compact graph-based memory for long-horizon reasoning in real-world applications, as demonstrated in a robotic experiment conducted using a wheeled manipulator platform.

Keywords: multi-hop question answering (MHQA); spatial–temporal scene graph (STSG); multimodal large language model (MLLM)

1. Introduction

Interpretable object maps for representing a robot's surrounding environment are an active area of research. These maps include descriptions—either explicit textual annotations or implicit representations in the form of extracted features—of scene elements, along with their 2D-3D positions. Such representations are typically used in conjunction with text generative models to enable logical analysis and reasoning over users' queries [1,2].

ConceptGraphs [3], BBQ [4], Search3D [5], and similar approaches construct advanced graph structures from sequences of spatially localized frames using foundational visual–text models. This enables the identification of objects of interest via arbitrary textual queries that specify diverse inter-object spatial relationships. HOV-SG [6], OVIGo-3DHSG [7], and Clio [8] employ a multi-level hierarchy to represent large interior spaces as layered graphs (e.g., floors and rooms), with each node preserving its unique features. This strategy substantially narrows the search space for textual queries and facilitates scaling knowledge maps for intelligent agents and robots to large-scale environments.

Check for updates

Academic Editors: Sotirios K. Goudos, Liviu Marian Ungureanu, Iulian Sorin Munteanu and Péter Galambos

Received: 26 November 2025
Revised: 13 February 2026
Accepted: 17 February 2026
Published: 1 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

The presented methods operate under the assumption that the observed environment is static, which significantly limits their deployment in real-world settings where variability is a key characteristic. PSG4D [9] incorporates an object-tracking procedure, allowing observations to be represented as graphs of objects with edges that depend on the query time. However, when the resulting spatio-temporal scene graph (STSG) representation is conveyed entirely in textual form to the context of a large language model, it may lead to hallucinations during logical reasoning due to the large volume of information generated by continuous changes [10–13]. G-Retriever [14] and related works [15–23] advance the idea of encoding a single graph representation, enabling a concise and implicit depiction of scene descriptions in the form of specialized input tokens with a high compression rate and without loss of information quality. To address context length limitations in dynamic scenes, we propose the DyGEnc method, which extends the encoding concept to sequences of graphs (dynamic graphs).

Thus, our key contributions are as follows:

1. We propose the DyGEnc architecture for encoding sequences of textual scene graphs using the parameter-efficient fine-tuning of a large language model, enabling reasoning and question answering (Section 3);
2. We provide a comprehensive analysis of DyGEnc components and their impact on encoding performance, evaluated on the STAR [24] (Section 4.1) and AGQA [25] (Section 4.2) benchmarks;
3. We introduce a practical approach for deploying DyGEnc in real-world applications by leveraging foundation models to extract textual scene graphs from image sequences, along with an in-depth preliminary discussion of input modality data types (Section 5).

Code is publicly available at github.com/linukc/DyGEnc (accessed on 16 February 2026).

2. Related Work

2.1. Dynamic Scene Graph Generation

A dynamic scene graph (DSG) is a sequence of graphs in which connections are characterized not only by spatial relationships between objects but also by action-based relationships between moving actors and objects. Predicting such graphs from sensory data (primarily image sequences) has been extensively studied. Two broad categories of modern methods can be identified: (i) end-to-end trainable approaches and (ii) graph construction based on foundational models.

The emergence of trainable methods was driven by the availability of manually annotated graph datasets such as Visual Genome [26], GQA [27], PSG [28], Action Genome [29], and STAR [24]. Today, a wide range of approaches exists, including several state-of-the-art transformer-based image graph predictors, such as PSG [28], PVSG [30], EGTR [31], RelTR [32], and OED [33].

In the second group of methods, the use of foundational models [34–36], as well as modern large language models with visual–text input (vLLMs) [37–40], deserves particular attention. These models are actively employed to generate large-scale synthetic graph annotations [41–43]. Compared to trainable methods based on existing graph datasets, which are limited in diversity—a key factor for the proposed algorithm's performance across a wide range of scenarios—this second group of approaches exhibits better generalization to new data but requires human-in-the-loop supervision to correct hallucinations and verify outputs.

2.2. Question Answering with Textual Graphs

Recent research on question answering (QA) with textual graphs primarily focuses on answering questions over explicit graphs or large aggregated knowledge graphs, where

nodes and edges encode entities and relationships extracted from text, databases, or other structured representations [15,16,20]. Approaches such as G-Retriever [14], GNN-RAG [17], and others [18,19] integrate graph neural networks with retrieval-augmented generation, demonstrating that structured graph representations can significantly improve evidence selection and reasoning in large language models. Similarly, GraphGPT [44], GraphTranslator [45], and others [14,21–23] emphasize aligning graph-structured information with LLMs through instruction tuning or representation alignment to enhance multi-hop QA (MHQA) and open-ended reasoning. Collectively, these studies converge on the claim that explicit structure serves as an inductive bias, improving retrieval accuracy, reasoning faithfulness, and robustness compared to purely text-based pipelines.

The graphs considered in existing work are typically static or globally aggregated, often assuming a single, fixed graph per task. As a result, the empirical benefits of graph structure are evaluated only in isolated settings, without considering how these advantages persist or degrade across sequences of graphs. In particular, to the best of our knowledge, none of the surveyed approaches explicitly study temporal, evolving, or sequential graphs, where the structure itself changes over time or across reasoning steps. This raises an open question as to whether the reported gains from structured representations generalize to graph sequences and how QA systems should adapt their retrieval and reasoning mechanisms in such dynamic settings.

In DyGEnc, we explicitly model dynamic scene understanding as a sequence of textual scene graphs, where each graph captures object identities and typed relations at a given time step. Each graph is encoded using a graph neural network that propagates information along object–relation edges, producing structure-aware node embeddings. To capture temporal evolution, these graph-level representations are further processed by a sequence encoder that models dependencies across time steps and compresses the sequence into a compact representation. This design enables DyGEnc to retain both object-level structure and temporal dynamics while remaining compatible with the parameter-efficient fine-tuning of a large language model. The resulting embeddings are integrated as structured tokens into the LLM, allowing it to perform reasoning and question answering over evolving scenes rather than over unstructured textual descriptions.

3. Materials and Methods

We first briefly formalize the notion of textual dynamic scene graphs, which constitute the input to our method. A textual scene graph is a graph derived from an image, where nodes and edges are associated with textual attributes representing objects and their relationships. Formally, a scene graph is defined as $G = (V, E)$, where V and E denote the sets of nodes and edges, respectively. A dynamic scene graph is then defined as an ordered sequence of such graphs $\{G_t\}_{t=1}^{T}$.

Building on this representation, we introduce DyGEnc, a novel encoding architecture designed to transform sequences of textual scene graphs into compact representations suitable for reasoning and question answering with large language models. DyGEnc consists of a graph-based encoder, as shown in Section 3.1, that captures the object–relation structure at each time step and a temporal aggregation module, as shown in Section 3.2, that models the evolution of graph structure over time. An overview of the proposed DyGEnc pipeline is shown in Figure 1.

https://doi.org/10.3390/technologies14030150

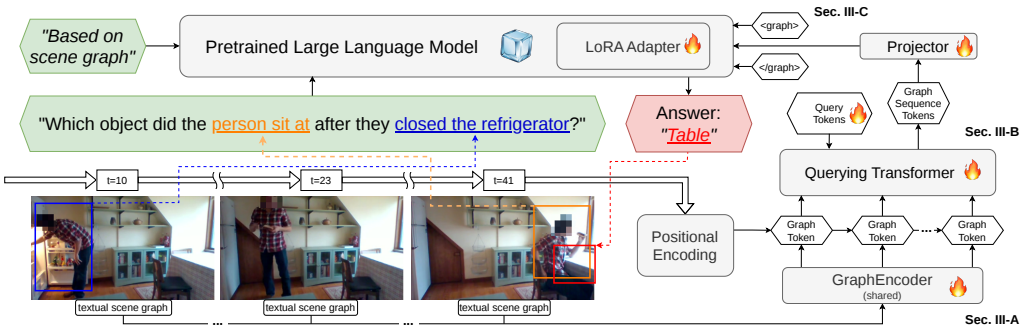


Figure 1. An overview of the DyGEnc pipeline: Given a dynamic scene graph—a sequence of textual scene graphs where nodes and edges carry attributes encoded by a pretrained text encoder—we first pass each encoded graph through a graph neural network to generate an aggregated graph token. To preserve temporal information, each graph token is enriched with positional encoding. Next, the Q-Former module is applied to capture temporal relationships, producing a compact sequence representation in query tokens. Finally, a multilayer perceptron projects these tokens into the embedding space of a large language model, with special tokens marking the start and end of the graph soft prompt. This allows the LLM to ground its reasoning based on input.

3.1. Graph Encoding

Consider x_n to be the text attributes of node n. Utilizing a pretrained text encoder LM, we apply it to x_n, yielding the representation z_n:

$$z_n = LM(x_n) \in \mathbb{R}^d, \tag{1}$$

where d denotes the dimension of the output vector. Similar preprocessing steps are applied to edges. We utilize a pretrained ModernBert [46] base version with 149M parameters, denoted as LM. Additionally, for each node n, Laplacian positional encoding $d_{lpe} \in \mathbb{R}^4$ [47] is added to the text encoder's dimensionality d.

Then latent representation $G_z = (V_z, E_z)$ is used to encode the graph structure with a graph neural encoder GNN:

$$h_g = F_{aggr}(GNN_{\theta_1}(G_z)) \in \mathbb{R}^{d_g}. \tag{2}$$

Here, F_{aggr} denotes the aggregation scheme, and d_g is the output dimension of the graph encoder. For F_{aggr}, we use mean pooling, and for GNN, we use GraphTransformer [48] with 21M parameters.

3.2. Sequence Encoding

To preserve the original timeline, we extend each graph token h_G with a positional encoding vector $p_{tpe} \in \mathbb{R}^{d_g}$:

$$\hat{h}_g = h_g + p_{tpe} \in \mathbb{R}^{d_g}, \tag{3}$$

where t is a graph index in a sequence. We utilize Rotary Positional Encoding [49] for temporal encoding. An ablation study of different positional encoding schemes can be found in Section 4.1.1.

To encode temporal relationships, we utilize a sequence encoder SE with a Q-Former [50] architecture of a cross-attention decoder transformer with 2 layers and 4 cross-attention heads each, resulting in 19M parameters θ_2. For a given sequence of m scene graph tokens $\hat{h}_g$ ($M \in \mathbb{R}^{m \times d_g}$), cross-attention to k learnable query tokens $K \in \mathbb{R}^{k \times d_g}$ ($k \leq m$) is applied to gather sequence information in a compact latent representation of a dynamic graph:

$$\hat{h}_{dg} = SE_{\theta_2}(M, K) \in \mathbb{R}^{k \times d_g}. \tag{4}$$

A fixed number of output tokens K and a theoretically unconstrained number of M in a potential scene graph sequence allow us to apply sequence encoding to an arbitrary sequence while preserving a constant context size for a large language model. An ablation study on the number of learnable query tokens K with respect to the parameters of the Q-Former can be found in Section 4.1.2.

3.3. LLM Fine-Tuning

To align compressed tokens, describing a sequence of graphs, we project K to the vector space of the LLM by incorporating a multilayer perceptron MLP:

$$h_{llm} = MLP_{\theta_3}(\hat{h}_{dg}) \in \mathbb{R}^{k \times d_{llm}}, \tag{5}$$

where d_{llm} is the dimension of the LLM's hidden embedding.

The final stage involves generating the answer A given the list of latent dynamic graph tokens h_{llm}, acting as a soft prompt, and the question Q. These concatenated inputs are fed through the self-attention layers of a pretrained frozen LLM with parameters θ_4:

$$A = LLM_{\theta_4}(Q, h_{llm}). \tag{6}$$

While most θ_4 are frozen, part of the weights $\hat{\theta_4}$ are updated with parameter-efficient training alongside the graph tokens h_{llm} receiving gradients, enabling the optimization of the parameters θ_3 of the projection layer, Q-Former θ_2, and graph encoder θ_1 through backpropagation. More technical and implementation details can be found in Section 3.6.

3.4. Datasets

STAR [24] benchmark for situated reasoning is built upon 9K real-world videos of human actions and their surrounding environments in daily life scenes. Annotations consist of 60K situated questions, divided into four types, interaction, sequence, prediction, and feasibility, for 22K video clips labeled with scene graphs. The lengths of graph sequences exhibit a positively skewed distribution, with a median of 20, an interquartile range of 15, a 5th percentile of 7, and a 95th percentile of 46.

Action Genome Question Answering [51] (AGQA) benchmark for compositional spatio-temporal reasoning consists of visual events, each representing a composition of temporal actions involving actors interacting with objects. We use the latest (second) version of the benchmark [25], which contains 9.6K unique scene graph sequences annotated from real-life videos. The lengths of these sequences exhibit a positively skewed distribution, with a median of 28, an interquartile range of 20, a 5th percentile of 10, and a 95th percentile of 60. A total of 2.27M balanced question–answer pairs are generated from more than 30 diverse templates covering reasoning, structure, and semantic understanding across 16 subcategories.

DRobot is our internally created dataset, collected on a real robot to evaluate the generalization of the trained model to real-world scenarios. Full details are provided in Section 5.3, where we discuss how a trained model can be supplied with input in the form of an image sequence.

3.5. Evaluation Metrics

To evaluate answer quality, we experimented with both *Exact Match* and *Accuracy* (the percentage of correctly answered questions). However, we found the difference between the two metrics to be negligible, primarily due to the short-answer annotation format used in STAR and AGQA, where responses typically consist of only a few words. Therefore, in alignment with prior work and established practices in the corresponding publications, we

report accuracy. Under this metric, a prediction is considered correct if the normalized (e.g., lowercased, with punctuation and extra whitespace removed) ground-truth answer from the dataset contains the normalized response generated by the model.

Additionally, for experiments involving DSG construction on our dataset—where the model primarily works with textual scene graphs from out-of-training distributions and generated answers may be semantically close but differ from the labeled answer—we extend the evaluation with *BLEU* [52], *METEOR* [53], and *BERTScore* [54], implemented via the HuggingFace Evaluate [55] package.

3.6. Implementation Details

3.6.1. Data Preprocessing

Each sequence undergoes a preprocessing step that retains only unique graphs, which helps significantly reduce the context size for the model in low-dynamic environments.

Each textual graph used for prompting follows a scheme that preserves its directional properties. First, a list of indexed node labels is provided, each separated by a newline and a comma, followed by a list of edges in the pattern *subject/predicate/object*, where the *subject* and *object* are represented using the corresponding indices.

To preserve the temporal component—for example, to reason about time intervals and durations—the indices t correspond to the positions of the graphs in the original sequence and are maintained.

3.6.2. LLM Fine-Tuning

For parameter-efficient LLM fine-tuning, a LoRA adapter [56]—a method for efficiently fine-tuning large models by updating only a small subset of parameters instead of the entire model—is used with parameters $r = 8$, $alpha = 16$, and a dropout rate of 0.05, specifically targeting the Q and V components of the LLM's attention modules. In our work, we experiment with the modern open Llama3 [57] model family, selecting the Llama 3.1-8B and Llama 3.2-3B versions to meet the resource constraints of most potential application systems and to align with previous research [14].

To adapt the language model for understanding graph tokens, we add special tokens <graph> and </graph> to represent the start and end of the dynamic graph latent representation, resulting in an input prompt: "*Based on scene graph,* <graph>$/h_{llm}$</graph>*,* Q". For sampling, we use a greedy decoding strategy.

We use the AdamW optimizer with an initial learning rate of 2×10^{-5} and a weight decay of 0.05. The learning rate follows a half-cycle cosine decay after a warm-up period of 1 epoch. The batch size is 32, and the number of epochs is set to 5 for STAR and 1 for AGQA. To prevent overfitting and ensure training efficiency, an early stopping mechanism is implemented. All experiments are conducted on an A100 80GB GPU. With these settings, training on STAR takes approximately ten times less than on AGQA, which is why STAR was chosen for the costly ablation studies. For both datasets, all other training hyperparameters are kept the same.

4. Results

4.1. Ablation of DyGEnc Components

To better understand sequence encoding, we conduct ablation studies of key components: the type of temporal positional encoding, which preserves event time order, and a search over Q-Former hyperparameters to determine the extent to which dynamic scene graph tokens can be compressed without significant information loss for the LLM, using two variations in the number of parameters and without text input. These ablations allow us to identify critical hyperparameters for the encoder and select an appropriate LLM size.

4.1.1. Temporal Encoding Type

We emphasize that DyGEnc does not propose a new positional encoding mechanism. Instead, we evaluate several existing temporal positional encodings as interchangeable components within DyGEnc to analyze their impact on dynamic graph encoding. For temporal positional encoding, we fine-tuned both LLM versions using three different approaches: temporal encoding [58] (TE), Absolute Positional Encoding [59] (APE), and Rotary Positional Encoding [49] (RoPE). For all settings, the number of Q-Former query tokens was set to 1, as we were primarily interested in the impact of encoding at an extreme sequence compression level. The results in Table 1 show that incorporating explicit temporal positional information consistently improves performance within the DyGEnc framework, especially for complex *Prediction* and *Feasibility* splits. This confirms that DyGEnc effectively exploits the temporal structure in dynamic scene graphs while remaining flexible to different positional encoding choices. For further experiments, we adopt the RoPE approach as the default.

In Figures 2 and 3, we depict cross-attention to demonstrate that our method does not simply memorize information during training but instead learns to attend to relevant frames and reason based on a sequence of scene graphs, where each graph is represented as a latent token. We show cross-attention from one Q-Former latent query token to each input graph embedding, with eight blocks representing two layers with four heads per layer. Brighter colors indicate higher model attention. These visualizations reveal a strong correlation between the correct graph in a sequence and the attention weights that activate the corresponding GNN embedding, demonstrating that the sequence encoder effectively compresses the temporal dimension in a meaningful manner.

Table 1. Ablation of temporal positional encoding of DyGEnc architecture on STAR [24] QA validation split. Bold and underlined text indicate the first and second best results, respectively.

Base Model	Encoding	Split				Average
		Interaction	Sequence	Prediction	Feasibility	
Llama3.2-3B	-	**0.97**	**0.90**	0.69	0.59	**0.89**
Llama3.1-8B		**0.97**	0.88	0.68	0.58	0.87
Llama3.2-3B	TE	<u>0.96</u>	<u>0.89</u>	0.7	0.63	<u>0.88</u>
Llama3.1-8B		0.92	0.82	0.68	0.55	0.82
Llama3.2-3B	APE	**0.97**	0.88	**0.77**	**0.69**	**0.89**
Llama3.1-8B		0.95	0.85	<u>0.76</u>	0.65	0.86
Llama3.2-3B	RoPE	**0.97**	**0.9**	**0.77**	0.65	**0.89**
Llama3.1-8B		<u>0.96</u>	0.87	**0.77**	<u>0.67</u>	<u>0.88</u>

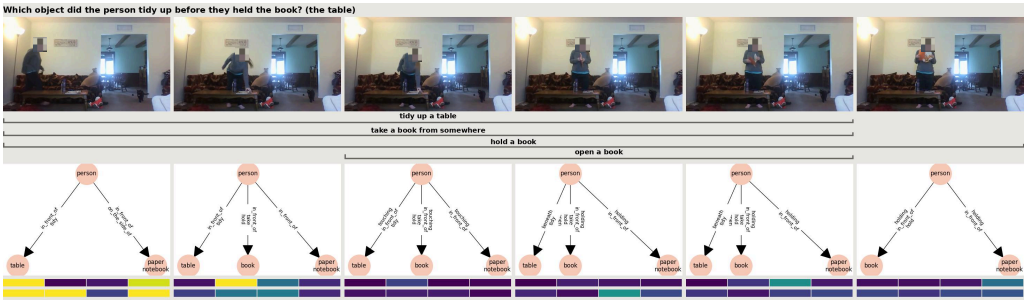


Figure 2. Example of cross-attention visualization from Q-Former sequence encoder on STAR benchmark for text query "*Which object did the person throw before they held the dish*". Brighter colors indicate larger attention weights.

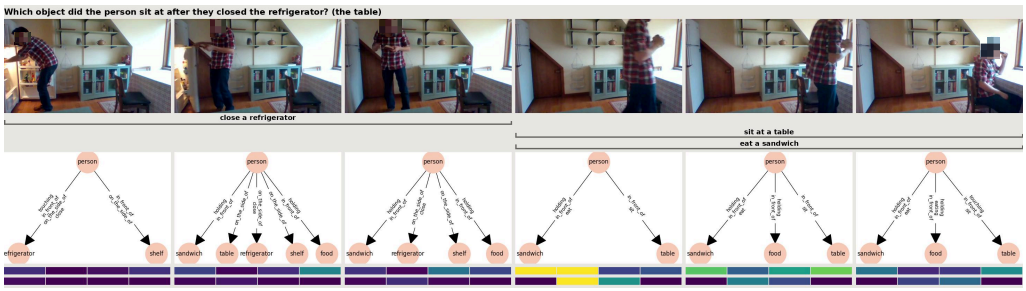


Figure 3. Example of cross-attention visualization from Q-Former sequence encoder on STAR benchmark for text query *"Which object did the person sit at after they closed the refrigerator"*. Brighter colors indicate larger attention weights.

4.1.2. Number of Query Tokens

The Q-Former in DyGEnc introduces a learnable latent bottleneck that compresses dynamic graph sequences before integration into the LLM. To study the impact of sequence compression on reasoning performance, we fine-tuned both LLMs with varying numbers of latent Q-Former query tokens (1, 2, 4, 16) while keeping the number of cross-attention layers fixed at four and the number of heads per layer fixed at four.

Table 2 shows the resulting accuracy across the STAR benchmark splits. The results show that *Interaction* and *Sequence* questions—which require understanding events and their temporal ordering, the main focus of DyGEnc—benefit consistently from additional query tokens, though the gains are smaller than those for *Prediction* and *Feasibility* questions. The latter categories exhibit larger improvements because their answers are less dependent on temporal reasoning and more sensitive to overall sequence representation capacity. Based on this ablation, we set the number of latent Q-Former query tokens to 16 in the subsequent experiments, as it achieves the highest average performance.

Table 2. Ablation of Q-Former query token numbers of DyGEnc architecture on STAR [24] QA validation split. Bold and underlined text indicate the first and second best results, respectively.

Base Model	Num Tokens	Split				Average
		Interaction	Sequence	Prediction	Feasibility	
Llama3.2-3B	1	0.97	0.9	0.77	0.65	0.89
Llama3.1-8B		0.96	0.87	0.77	0.67	0.88
Llama3.2-3B	2	0.97	0.9	0.78	0.7	0.9
Llama3.1-8B		0.97	0.89	0.75	0.64	0.89
Llama3.2-3B	4	0.96	0.9	0.8	0.69	0.9
Llama3.1-8B		**0.99**	<u>0.92</u>	0.86	0.73	0.92
Llama3.2-3B	16	<u>0.98</u>	**0.94**	**0.91**	**0.79**	**0.94**
Llama3.1-8B		<u>0.98</u>	<u>0.92</u>	<u>0.89</u>	<u>0.77</u>	<u>0.93</u>

4.2. Impact of DyGEnc Architecture

AGQA is the most comprehensive benchmark with graph annotations, containing nearly 2 million QA pairs. Therefore, we chose to evaluate our model on AGQA only after performing hyperparameter ablations on the STAR dataset. With the same settings, we trained DyGEnc on the AGQA training split. For training, we limited the sequence length of unique graphs to 60, which preserves more than 95% of the original training data and allows us to discard samples that could negatively impact convergence. This helps stabilize the training process. It should be noted that the test split is not subject to the same limitation.

In our experiments, we consider three configurations to investigate the influence of the DyGEnc architecture. (1) Inference-only: Using a frozen LLM for direct question answering. (2) Frozen LLM with prompt tuning (PT): Keeping the LLM parameters frozen while adapting only the learnable prompt embeddings. (3) Tuned LLM: Fine-tuning the LLM with LoRA [56]. A schematic of the experimental configurations can be found in Figure 4. For subsequent experiments, we use only the 3B model, as it offers a favorable tradeoff between quality and model size, as illustrated by the previous experiments.

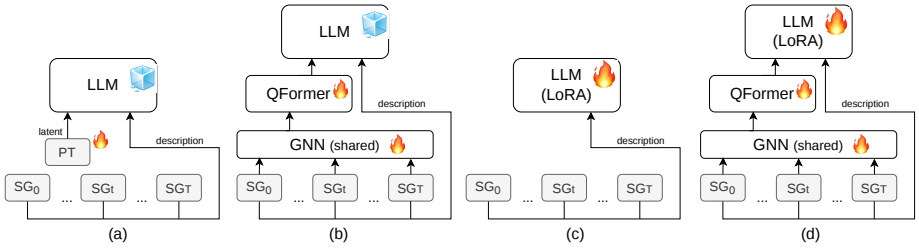


Figure 4. Experiment setups used for impact measurement of DyGEnc architecture on AGQA [25] QA test split. From left to right: (a) frozen LLM with PT (prompt tuning); (b) frozen LLM with DyGEnc; (c) LLM fine-tuning with LoRA; (d) LLM fine-tuning with LoRA based on DyGEnc architecture.

The results of the evaluation can be found in Table 3. For all columns, we report the *Accuracy* metric. The ablation study on the AGQA QA test split demonstrates that DyGEnc consistently enhances reasoning performance across all relational and temporal dimensions. When integrated with a frozen LLM, DyGEnc yields substantial gains over the PT baseline, indicating that structured dynamic graph encoding complements linguistic representations even without model fine-tuning. Under the LoRA setting, DyGEnc achieves further improvements, reaching the best performance across all reasoning categories and confirming that lightweight adaptation benefits from explicit spatio-temporal structure modeling. Notably, the largest gains are observed in the *Action–Recognition* and *Object–Action* splits, with relative improvements exceeding 200% compared to zero-shot performance, highlighting DyGEnc's capacity to capture fine-grained visual dependencies. The model's superior results on *Sequencing* and *Superlative* reasoning underscore its ability to maintain temporal coherence and multi-hop relational consistency across frames. Even in categories with traditionally limited context, such as *Duration*, DyGEnc demonstrates steady improvements.

Table 3. Impact measurement of DyGEnc architecture on AGQA [25] QA test split. Bold and underlined text indicate the first and second best results, respectively.

Split	LLM (Zero-Shot)	Frozen LLM		LoRA [56]	
	Llama3.2-3B	LLM + PT	DyGEnc	LLM	DyGEnc
Reasoning					
Object–Relationship	0.53	0.73	0.82	<u>0.87</u>	**0.88**
Relationship–Action	0.34	0.55	0.76	**0.80**	<u>0.79</u>
Object–Action	0.48	0.52	0.83	**0.85**	<u>0.84</u>
Superlative	0.44	0.52	0.68	<u>0.77</u>	**0.79**
Sequencing	0.45	0.52	0.78	<u>0.79</u>	**0.80**
Exists	0.59	0.68	0.75	<u>0.79</u>	**0.83**
Duration	0.32	0.49	0.50	<u>0.51</u>	**0.54**
Activity–Recognition	0.01	0.07	0.34	<u>0.47</u>	**0.57**
Overall					
Binary	0.50	0.60	0.74	<u>0.78</u>	**0.80**
Open	0.54	0.80	<u>0.89</u>	**0.93**	**0.93**
All	0.52	0.70	0.82	<u>0.85</u>	**0.87**

Overall, DyGEnc improves *Binary* QA accuracy from 0.50 (zero-shot) to 0.80 and *Open* QA accuracy from 0.54 to 0.93, establishing a new benchmark for generalizable video reasoning. Importantly, these gains persist under both frozen and LoRA-adapted LLMs, implying that the improvements stem from DyGEnc's representational bias rather than parameter overfitting. These findings validate the hypothesis that graph-based temporal modeling provides a complementary inductive structure to language-driven reasoning.

5. Discussion

The previously discussed results were obtained using available scene graph annotations. However, for real-world applications of DyGEnc, it is necessary to develop an algorithm capable of constructing textual scene graphs from a sequence of images. In Section 5.1, we first highlight the advantages of using text–graph input compared to visual–text input with given ground-truth annotations. In Section 5.2, we further investigate the extent of the performance degradation of the DyGEnc model by simulating typical inaccuracies that occur during dynamic scene graph construction, representing the lack of ideal annotations in real-world setups. Finally, in Section 5.3, we describe our methodology for generating textual scene graphs using foundation models, providing a complete pipeline for applying DyGEnc to common video input while preserving the structured reasoning capabilities of the developed model.

5.1. Impact of Input Modality

It should be noted that the input data in the presented works (visual–text) and in ours (text–graph) differ. In the absence of direct competitors for comparison (to the best of our knowledge), the motivation for this section is to compare the two input modalities under ground-truth conditions and demonstrate the advantages of reasoning based on a structured representation of textual–graph information.

The results for the STAR benchmark can be found in Table 4. On the validation split, DyGEnc attains 0.99 accuracy in both *Interaction* and *Sequence* reasoning, suggesting that the model effectively captures temporal and relational dependencies. The *Prediction* metric also reaches 0.97, demonstrating that DyGEnc can accurately forecast outcomes based on observed interactions. Similarly, the *Feasibility* metric reaches 0.97, indicating that the model can reliably evaluate whether predicted events are plausible within a given context.

Table 4. Comparison of visual and text modalities on STAR [24] QA validation split. Bold and underlined text indicate the first and second best results, respectively.

Method	Interaction	Sequence	Prediction	Feasibility	Average
Visual–Text modalities					
STEP [60]	-	-	-	-	0.4
Q-ViD [61]	0.48	0.47	0.44	0.43	0.46
MIST [62]	0.56	0.54	0.54	0.45	0.51
SeViLA [63]	0.64	<u>0.71</u>	0.63	0.62	0.65
ViLA [64]	0.7	0.7	0.66	0.62	0.67
VidF4 [65]	0.68	0.7	0.61	0.59	0.68
LRR [66]	<u>0.74</u>	<u>0.71</u>	<u>0.71</u>	<u>0.65</u>	0.71
Text–Graph modalities					
DyGEnc	**0.99**	**0.99**	**0.97**	**0.97**	**0.99**

Compared to the best-known visual method LRR [66] on the benchmark, which achieved 0.74 in *Interaction* and 0.71 in *Sequence*, our method shows a relative improvement of over 30%, highlighting the effectiveness of dynamic graph encoding. Notably, models such as SeViLA [63] and ViLA [64], despite leveraging visual–language alignment,

are unable to capture temporal dependencies as effectively as DyGEnc. VidF4 [65] and Q-ViD [61], which integrate end-to-end video understanding, lag behind considerably, demonstrating that naive multimodal fusion is insufficient for complex reasoning tasks. Scores in *Interaction* indicate that DyGEnc effectively models relational dynamics between objects over time. Similarly, the *Sequence* score reflects the model's ability to maintain temporal consistency and understand event ordering. The improvements in *Prediction* and *Feasibility* demonstrate that DyGEnc generalizes learned dynamics to novel scenarios without task-specific fine-tuning. Moreover, the consistent performance gains across all metrics suggest adaptivity to variations in input content and query complexity. These results imply that dynamic graph modeling with textual input provides a stronger inductive bias for temporal reasoning in sequences of data.

Because the STAR [24] benchmark proved to be relatively easy from the results of the experiments for LLM tuning, potentially due to the limited templates for QA construction, we perform ablations on the larger and more complex AGQA [25] benchmark. Table 5 shows that, compared to existing methods, DyGEnc outperforms most visual-based approaches in the presence of text–graph information, particularly on open-formulated (non-binary) queries, demonstrating its ability to distinguish unique text–graph features and reason over the semantics of questions and context. DyGEnc achieves large gains on *Object–Relationship*, *Relationship–Action*, and *Object–Action* tasks, illustrating the effectiveness of dynamic graph encoding for modeling temporal and relational dependencies among objects and actions in the text–graph modalities. Strong results on *Sequencing* and *Superlative* further highlight DyGEnc's capacity to capture temporal order and comparative relationships. The significant improvement in *Activity–Recognition* indicates the enhanced understanding of evolving event structures.

Table 5. Comparison of visual and text modalities on AGQA [25] QA test split. Bold and underlined text indicate the first and second best results, respectively.

	Visual–Text Modalities					*Text–Graph Modalities*
	MIST [62]	**GF [67]**	**IPRM [68]**	**TGB [69]**	**DeST [70]**	**DyGEnc**
Reasoning split						
Object–Relationship	0.52	0.55	0.58	<u>0.62</u>	0.60	**0.88**
Relationship–Action	0.67	–	–	0.52	<u>0.73</u>	**0.79**
Object–Action	0.69	–	–	0.66	<u>0.75</u>	**0.84**
Superlative	0.42	0.45	0.48	0.54	0.49	**0.79**
Sequencing	0.67	0.53	<u>0.76</u>	0.60	0.74	**0.80**
Exists	0.60	0.59	0.62	0.61	<u>0.63</u>	**0.83**
Duration	<u>0.55</u>	0.53	0.51	0.37	**0.60**	0.54
Activity	0.20	0.14	0.20	0.00	<u>0.28</u>	**0.57**
Overall						
Binary	0.58	0.54	0.62	–	<u>0.63</u>	**0.80**
Open	0.51	0.56	0.59	–	<u>0.61</u>	**0.93**
All	0.54	0.55	0.60	<u>0.62</u>	<u>0.62</u>	**0.87**

Overall, given ground-truth images and graph annotations, text–graph DyGEnc outperforms visual–text methods on the complex multi-hop question answering task, with most gains resulting from the alignment between graph annotations and questions.

5.2. Ablation of DSG Data Quality

To evaluate the robustness of the proposed model, we conducted an inference-time noise injection experiment to assess how DyGEnc handles imperfect annotations containing errors, which more closely reflect real-world usage. Seven types of perturbations were

introduced to simulate various forms of structural and semantic corruption in the input graphs. Experiments were performed on the evaluation STAR [24] benchmark using ground-truth annotations. For each noise type, the perturbation intensity was varied from 0 to 1 with a step size of 0.33, and each setting was repeated with three random seeds to ensure stability. The experimental setups are illustrated in Figure 5.

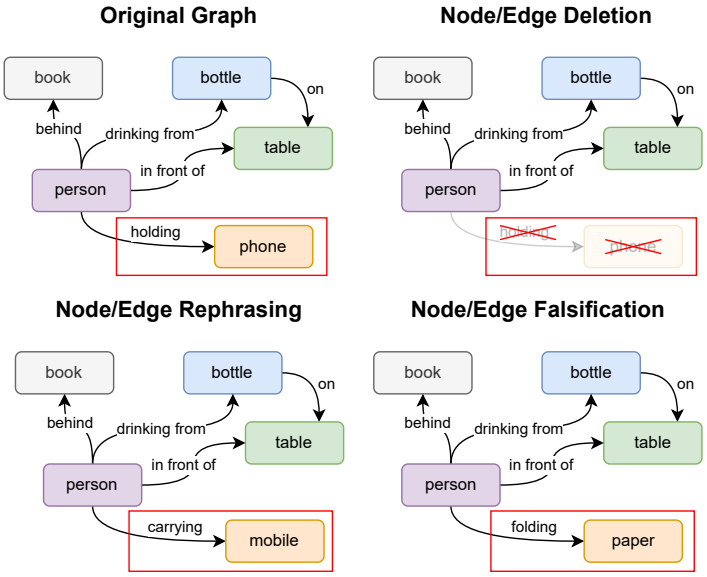


Figure 5. An example of graph modification for the question "What did the person hold?" in the inference-time noise injection experiment on the STAR [24] evaluation split. The types of scene graph alterations are highlighted in the red box.

All experiments modify answer nodes and edges to simulate typical drawbacks of visual foundation models and evaluate their impact on DyGEnc's performance. The first and second experiments simulate structural degradation by progressively removing a certain percentage of answer-related nodes and edges, respectively. The third and fourth experiments introduce semantic noise through paraphrasing, where nodes and edges related to the correct answer are replaced with synonyms at varying proportions. The fifth and sixth experiments correspond to label-level noise, in which the class assignments of answer nodes or associated edges are randomly changed to words unrelated to the original labels. These perturbations collectively assess the model's ability to reason under incomplete or inconsistent contextual information.

The experimental results are summarized in Figure 6. Overall, the model maintains strong performance under mild perturbations, with a mean accuracy above 0.9 for all noise types at low intensity levels. As noise increases, performance gradually declines, with the most severe degradation observed for structural removals and edge paraphrasing. Semantic perturbations cause moderate performance drops, indicating the model's partial reliance on lexical consistency. In most cases, visual foundation models are affected by a combination of factors, with drastic performance degradation occurring only under high levels of noise.

These findings highlight the adaptability of DyGEnc in scenarios where graph-based representations may be noisy, incomplete, or partially corrupted, strengthening the comparison between visual–text models and our text–graph approach in Section 5.1. Even under extreme noise levels, DyGEnc preserves reasonable inference quality, surpassing visual question answering models when reliable textual data is available. In Section 5, we discuss

strategies for obtaining textual scene graph annotations of comparable quality using large language models.

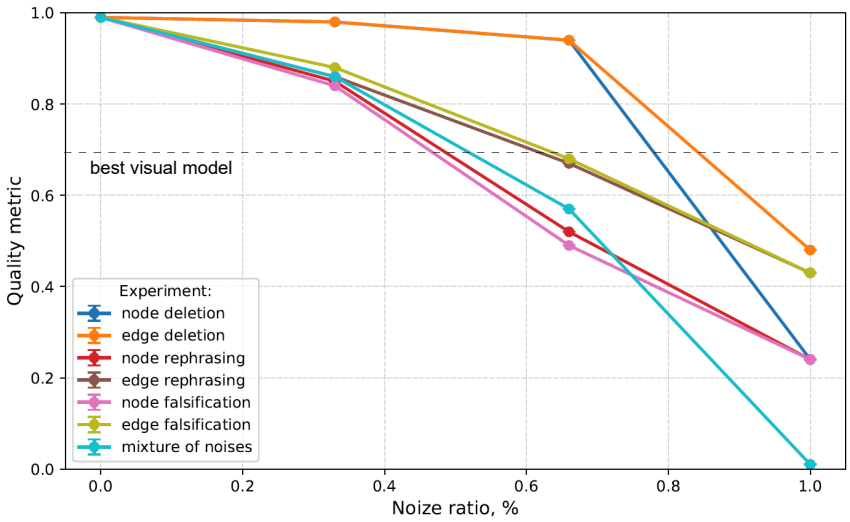


Figure 6. Inference-time robustness of DyGEnc under different types and intensities of graph noise on STAR [24] QA eval split.

5.3. Ablation Study on DSG Construction

To generate textual scene graphs from images, we compare three different approaches. The first approach utilizes Nvila [39] vLLM for image captioning and then employs the Factual [71] model, specifically trained to extract triplets from the input text describing an image. The second approach uses the same captioning method but relies on a GPT [37] LLM to extract triplets from the text. The third approach solely relies on GPT [37] processing with visual input to describe the image. For our experiments, we utilize the 4o-mini model.

To compare these methods, we constructed our DRobot dataset. It consists of 10 scenes, each containing dynamic actions between humans and objects. The benchmark also serves a second goal: to demonstrate the use case of the DyGEnc model on sensory video input from a real robot. For this, we set up a robotic experiment in which a mobile-wheeled robot with a manipulator (depicted in Figure 7, left) is tasked with moving to and picking an object of interest (the target of the answer). We created 50 questions following the template-based approach from AGQA [25].

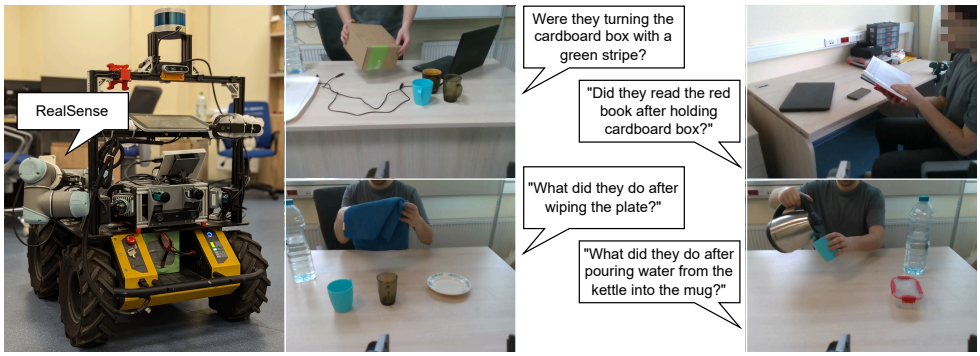


Figure 7. An illustration of the robotic experiment setup: on the **left**, a mobile Husky platform with a UR5 manipulator, equipped to perform a MOVE-AND-PICK task based on DyGEnc output; on the **right**, general scene overviews from our DRobot dataset.

It should be noted that our environment differs significantly in object set from the training distribution (examples are shown in Figure 7, right). For the experiments, we

used the pretrained DyGEnc model on AGQA. The evaluation results in Table 6 show that using GPT-generated textual scene graphs yields higher metrics, although open-source alternatives can also produce reasonable results without significant performance drops. These values are not affected by success rate failures.

Table 6. Ablation of DSD construction algorithms on our DRobot dataset. Bold and underlined text indicate the first and second best results, respectively.

Textualizer	Graph Constructor	Acc.	BLEU	METEOR	BERTScore
Nvila [39]	Factual [71]	0.3	0.32	0.18	**0.94**
Nvila [39]	GPT [37]	**0.34**	<u>0.36</u>	<u>0.2</u>	**0.94**
GPT(v) [37]		**0.34**	**0.41**	**0.21**	<u>0.92</u>

6. Limitations

It should be noted that keyframes must be extracted to apply DyGEnc to a video. In our robotic experiment, this was done using uniform sampling at one-second intervals. For long-term video understanding, a more advanced system should be implemented to capture sparse key events; addressing this is left for future work. DyGEnc already provides two features to support long-term reasoning: the arbitrary shape of input graphs in the Q-Former temporal encoder and a subgraph retrieval algorithm to reduce context size.

To apply the DyGEnc model to a video, a preliminary step of textual scene graph extraction must be performed, which can be demanding in terms of time and computational resources. However, we expect that in the near future, smaller and more efficient models will reduce this demand.

One of the biggest challenges in this two-step approach in practice is the misalignment between graph annotations and QA due to synonyms. Our experiments emphasize the importance of carefully designing instruction prompts for foundation models, using chain-of-thought reasoning and few-shot examples. We investigated error propagation during captioning and demonstrated DyGEnc's strong ability to handle common annotation shortcomings.

Despite achieving strong results on the STAR and AGQA benchmarks, DyGEnc cannot yet be considered a foundational language model for graph encoding beyond its training domain and statistics, as its capabilities remain constrained by the amount and diversity of training data with scene graph annotations compared to typical NLP tasks.

7. Conclusions

With DyGEnc, we extend the boundaries of dynamic scene perception by integrating large language models with a textual graph sequence encoder. The successful outcomes of our experiments on the complex STAR and AGQA benchmarks, as well as on our real-world data, demonstrate the effectiveness of our approach, opening new avenues for more comprehensive and flexible understanding and interaction with dynamic scenes. We hope that these findings will facilitate applications in real-world projects where spatio-temporal understanding is crucial, such as robotics, helping to bridge the communication gap between humans and intelligent autonomous agents.

Author Contributions: Conceptualization, S.L. and D.Y.; methodology, S.L.; software, S.L., V.S.; validation, S.L., V.S.; formal analysis, S.L., V.S.; investigation, S.L., V.S., A.T.; resources, O.B.; data curation, O.B.; writing—original draft preparation, S.L.; writing—review and editing, S.L., D.Y.; visualization, S.L., A.T.; supervision, S.L., D.Y.; project administration, D.Y.; funding acquisition, D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-15-2025-013, dated 20 June 2025, subsidy identifier 000000C313925P4B0002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data used in this study are publicly available and include direct links to their sources. The authors' code implementation and other materials are openly accessible at https://github.com/linukc/DyGEnc (accessed on 16 February 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yudin, D.A. M3DMap: Object-Aware Multimodal 3D Mapping for Dynamic Environments. *Opt. Mem. Neural Netw.* **2025**, *34*, 285–312. [CrossRef]
2. Yudin, D.; Lazarev, A.A.; Bakaeva, E.A.; Kochetkova, A.A.; Kovalev, A.K.; Panov, A. Scene Graph-Driven Reasoning for Action Planning of Humanoid Robot. *Eng. Appl. Artif. Intell.* **2026**, *169*, 114150. [CrossRef]
3. Gu, Q.; Kuwajerwala, A.; Morin, S.; Jatavallabhula, K.M.; Sen, B.; Agarwal, A.; Rivera, C.; Paul, W.; Ellis, K.; Chellappa, R.; et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2024; pp. 5021–5028.
4. Linok, S.; Zemskova, T.; Ladanova, S.; Titkov, R.; Yudin, D. Beyond bare queries: Open-vocabulary object retrieval with 3d scene graph. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA) 2025, Atlanta, GA, USA, 19–23 May 2025; pp. 13582–13589.
5. Takmaz, A.; Delitzas, A.; Sumner, R.W.; Engelmann, F.; Wald, J.; Tombari, F. Search3D: Hierarchical Open-Vocabulary 3D Segmentation. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2558–2565. [CrossRef]
6. Werby, A.; Huang, C.; Büchner, M.; Valada, A.; Burgard, W. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In Proceedings of the First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024, Yokohama, Japan, 13 May 2024.
7. Linok, S.; Naumov, G. Open-Vocabulary Indoor Object Grounding with 3D Hierarchical Scene Graph. *Opt. Mem. Neural Netw.* **2025**, *34*, 323–333. [CrossRef]
8. Maggio, D.; Chang, Y.; Hughes, N.; Trang, M.; Griffith, D.; Dougherty, C.; Cristofalo, E.; Schmid, L.; Carlone, L. Clio: Real-time task-driven open-set 3d scene graphs. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8921–8928. [CrossRef]
9. Yang, J.; Cen, J.; Peng, W.; Liu, S.; Hong, F.; Li, X.; Zhou, K.; Chen, Q.; Liu, Z. 4d panoptic scene graph generation. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 69692–69705.
10. Wei, Z.; Wang, S.; Rong, X.; Liu, X.; Li, H. Shadows in the Attention: Contextual Perturbation and Representation Drift in the Dynamics of Hallucination in LLMs. *arXiv* **2025**, arXiv:2505.16894. [CrossRef]
11. Liu, S.; Halder, K.; Qi, Z.; Xiao, W.; Pappas, N.; Htut, P.M.; John, N.A.; Benajiba, Y.; Roth, D. Towards Long Context Hallucination Detection. *arXiv* **2025**, arXiv:2504.19457. [CrossRef]
12. Cheng, J.; Su, T.; Yuan, J.; He, G.; Liu, J.; Tao, X.; Xie, J.; Li, H. Chain-of-Thought Prompting Obscures Hallucination Cues in Large Language Models: An Empirical Evaluation. *arXiv* **2025**, arXiv:2506.17088.
13. Alansari, A.; Luqman, H. Large Language Models Hallucination: A Comprehensive Survey. *arXiv* **2025**, arXiv:2510.06265. [CrossRef]
14. He, X.; Tian, Y.; Sun, Y.; Chawla, N.; Laurent, T.; LeCun, Y.; Bresson, X.; Hooi, B. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Adv. Neural Inf. Process. Syst.* **2025**, *37*, 132876–132907.
15. Li, Q.; Huang, C.; Li, S.; Xiang, Y.; Xiong, D.; Lei, W. Graphotter: Evolving llm-based graph reasoning for complex table question answering. In Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, United Arab Emirates, 19–24 January 2025; pp. 5486–5506.
16. Zemskova, T.; Yudin, D. 3dgraphllm: Combining Semantic Graphs and Large Language Models for 3D Scene Understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Honolulu, HI, USA, 19–23 October 2025; pp. 8885–8895.
17. Mavromatis, C.; Karypis, G. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv* **2024**, arXiv:2405.20139. [CrossRef]
18. Li, Z.; Guo, Q.; Shao, J.; Song, L.; Bian, J.; Zhang, J.; Wang, R. Graph neural network enhanced retrieval for question answering of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*; Volume 1: Long Papers; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 6612–6633.

19. Wu, Y.; Hu, N.; Bi, S.; Qi, G.; Ren, J.; Xie, A.; Song, W. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv* **2023**, arXiv:2309.11206.
20. He, M.; Shen, Y.; Zhang, W.; Peng, Q.; Wang, J.; Lu, W. Star-sql: Self-taught reasoner for text-to-sql. *arXiv* **2025**, arXiv:2502.13550.
21. Tian, Y.; Song, H.; Wang, Z.; Wang, H.; Hu, Z.; Wang, F.; Chawla, N.V.; Xu, P. Graph neural prompting with large language models. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 26–27 February 2024; Volume 38, pp. 19080–19088.
22. Perozzi, B.; Fatemi, B.; Zelle, D.; Tsitsulin, A.; Kazemi, M.; Al-Rfou, R.; Halcrow, J. Let your graph do the talking: Encoding structured data for llms. *arXiv* **2024**, arXiv:2402.05862. [CrossRef]
23. Selvam, K.P.; Phothilimthana, P.M.; Abu-El-Haija, S.; Perozzi, B.; Brorsson, M. Can LLMs Enhance Performance Prediction for Deep Learning Models? Available online: https://openreview.net/forum?id=bpS4vaOg7q#discussion (accessed on 13 February 2025).
24. Wu, B.; Yu, S.; Chen, Z.; Tenenbaum, J.B.; Gan, C. Star: A benchmark for situated reasoning in real-world videos. *arXiv* **2024**, arXiv:2405.09711. [CrossRef]
25. Grunde-McLaughlin, M.; Krishna, R.; Agrawala, M. AGQA 2.0: An Updated Benchmark for Compositional Spatio-Temporal Reasoning. *arXiv* **2022**. [CrossRef]
26. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.J.; Shamma, D.A.; et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73. [CrossRef]
27. Hudson, D.A.; Manning, C.D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 6700–6709.
28. Yang, J.; Ang, Y.Z.; Guo, Z.; Zhou, K.; Zhang, W.; Liu, Z. Panoptic scene graph generation. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 178–196.
29. Ji, J.; Krishna, R.; Fei-Fei, L.; Niebles, J.C. Action genome: Actions as compositions of spatio-temporal scene graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10236–10247.
30. Yang, J.; Peng, W.; Li, X.; Guo, Z.; Chen, L.; Li, B.; Ma, Z.; Zhou, K.; Zhang, W.; Loy, C.C.; et al. Panoptic video scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 18675–18685.
31. Im, J.; Nam, J.; Park, N.; Lee, H.; Park, S. Egtr: Extracting graph from transformer for scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 24229–24238.
32. Cong, Y.; Yang, M.Y.; Rosenhahn, B. Reltr: Relation transformer for scene graph generation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 11169–11183. [CrossRef]
33. Wang, G.; Li, Z.; Chen, Q.; Liu, Y. OED: Towards one-stage end-to-end dynamic scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 27938–27947.
34. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. Sam 2: Segment anything in images and videos. *arXiv* **2024**, arXiv:2408.00714.
35. Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; Lee, Y.J. Llavanext: Improved Reasoning, Ocr, and World Knowledge, 2024. Available online: https://llava-vl.github.io/blog/2024-01-30-llava-next/ (accessed on 10 February 2025).
36. Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; Shan, Y. Yolo-world: Real-time open-vocabulary object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16901–16911.
37. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
38. Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Wang, G.; Li, H.; Zhu, J.; Chen, J.; et al. Yi: Open foundation models by 01. ai. *arXiv* **2024**, arXiv:2403.04652.
39. Liu, Z.; Zhu, L.; Shi, B.; Zhang, Z.; Lou, Y.; Yang, S.; Xi, H.; Cao, S.; Gu, Y.; Li, D.; et al. NVILA: Efficient frontier visual language models. *arXiv* **2024**, arXiv:2412.04468. [CrossRef]
40. Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. Qwen2. 5-VL Technical Report. *arXiv* **2025**, arXiv:2502.13923.
41. Hsieh, Y.G.; Hsieh, C.Y.; Yeh, S.Y.; Béthune, L.; Pouransari, H.; Vasu, P.K.A.; Li, C.L.; Krishna, R.; Tuzel, O.; Cuturi, M. Graph-Based Captioning: Enhancing Visual Descriptions by Interconnecting Region Captions. *arXiv* **2024**, arXiv:2407.06723. [CrossRef]
42. Zhang, J.; Xue, L.; Song, L.; Wang, J.; Huang, W.; Shu, M.; Yan, A.; Ma, Z.; Niebles, J.C.; Xiong, C.; et al. ProVision: Programmatically Scaling Vision-centric Instruction Data for Multimodal Language Models. *arXiv* **2024**, arXiv:2412.07012.

43. Kim, K.; Yoon, K.; Jeon, J.; In, Y.; Moon, J.; Kim, D.; Park, C. LLM4SGG: Large language models for weakly supervised scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 28306–28316.
44. Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; Huang, C. Graphgpt: Graph instruction tuning for large language models. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington DC, USA, 14–18 July 2024; pp. 491–500.
45. Zhang, M.; Sun, M.; Wang, P.; Fan, S.; Mo, Y.; Xu, X.; Liu, H.; Yang, C.; Shi, C. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 1003–1014.
46. Warner, B.; Chaffin, A.; Clavié, B.; Weller, O.; Hallstrom, O.; Taghadouini, S.; Gallagher, A.; Biswas, R.; Ladhak, F.; Aarsen, T.; et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv* **2024**, arXiv:2412.13663. [CrossRef]
47. Dwivedi, V.P.; Joshi, C.K.; Luu, A.T.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking graph neural networks. *J. Mach. Learn. Res.* **2023**, *24*, 1–48.
48. Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv* **2020**, arXiv:2009.03509.
49. Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **2024**, *568*, 127063. [CrossRef]
50. Li, J.; Li, D.; Savarese, S.; Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 19730–19742.
51. Grunde-McLaughlin, M.; Krishna, R.; Agrawala, M. Agqa: A benchmark for compositional spatio-temporal reasoning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11287–11297.
52. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*; ACL: Stroudsburg, PA, USA, 2002; pp. 311–318.
53. Banerjee, S.; Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72.
54. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
55. Hugging Face. Evaluate: A Library for Easily Evaluating Machine Learning Models. 2022. Available online: https://github.com/huggingface/evaluate (accessed on 16 February 2025).
56. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *ICLR* **2022**, *1*, 3.
57. Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. The llama 3 herd of models. *arXiv* **2024**, arXiv:2407.21783 [CrossRef]
58. Cong, W.; Zhang, S.; Kang, J.; Yuan, B.; Wu, H.; Zhou, X.; Tong, H.; Mahdavi, M. Do we really need complicated model architectures for temporal networks? *arXiv* **2023**, arXiv:2302.11636. [CrossRef]
59. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017.
60. Qiu, H.; Gao, M.; Qian, L.; Pan, K.; Yu, Q.; Li, J.; Wang, W.; Tang, S.; Zhuang, Y.; Chua, T.S. STEP: Enhancing Video-LLMs' Compositional Reasoning by Spatio-Temporal Graph-guided Self-Training. *arXiv* **2024**, arXiv:2412.00161.
61. Romero, D.; Solorio, T. Question-instructed visual descriptions for zero-shot video question answering. *arXiv* **2024**, arXiv:2402.10698.
62. Gao, D.; Zhou, L.; Ji, L.; Zhu, L.; Yang, Y.; Shou, M.Z. Mist: Multi-modal iterative spatial-temporal transformer for long-form video question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14773–14783.
63. Yu, S.; Cho, J.; Yadav, P.; Bansal, M. Self-chained image-language model for video localization and question answering. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 76749–76771.
64. Wang, X.; Liang, J.; Wang, C.K.; Deng, K.; Lou, Y.; Lin, M.C.; Yang, S. ViLA: Efficient video-language alignment for video question answering. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 186–204.
65. Liang, J.; Meng, X.; Wang, Y.; Liu, C.; Liu, Q.; Zhao, D. End-to-End Video Question Answering with Frame Scoring Mechanisms and Adaptive Sampling. *arXiv* **2024**, arXiv:2407.15047.

66. Bhattacharyya, A.; Panchal, S.; Lee, M.; Pourreza, R.; Madan, P.; Memisevic, R. Look, remember and reason: Grounded reasoning in videos with language models. *arXiv* **2023**, arXiv:2306.17778.
67. Bai, Z.; Wang, R.; Chen, X. Glance and focus: Memory prompting for multi-event video question answering. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 34247–34259.
68. Jaiswal, S.; Roy, D.; Fernando, B.; Tan, C. Learning to Reason Iteratively and Parallelly for Complex Visual Reasoning Scenarios. *Adv. Neural Inf. Process. Syst.* **2025**, *37*, 137965–137998.
69. Wang, Y.; Wang, Y.; Wu, P.; Liang, J.; Zhao, D.; Liu, Y.; Zheng, Z. Efficient temporal extrapolation of multimodal large language models with temporal grounding bridge. *arXiv* **2024**, arXiv:2402.16050.
70. Lee, H.Y.; Su, H.T.; Tsai, B.C.; Wu, T.H.; Yeh, J.F.; Hsu, W.H. Learning Fine-Grained Visual Understanding for Video Question Answering via Decoupling Spatial-Temporal Modeling. *arXiv* **2022**, arXiv:2210.03941.
71. Li, Z.; Chai, Y.; Zhuo, T.Y.; Qu, L.; Haffari, G.; Li, F.; Ji, D.; Tran, Q.H. Factual: A benchmark for faithful and consistent textual scene graph parsing. *arXiv* **2023**, arXiv:2305.17497. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.