Temporal Scene Understanding using Contextually Unique Identification

Sanjiv S. Jha*, Kimberly Garcia*, Yasmine Sheila Antille* and Marc Elias Solèr*, Simon Padua†, Simon Mayer*
*Institute of Computer Science, University of St.Gallen, Switzerland
†University of Zurich, Switzerland
Email: sanjiv.jha@unisg.ch

2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3315-2723-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICTAI62512.2024.00145

***Abstract*—Humans can easily comprehend and explain the dynamics of a scene by observing the evolution of *relationships* among identified objects *over time*— while research towards Hybrid Intelligence promotes the integration of human and machine capabilities, this ability is currently beyond the capabilities of automated systems. Toward realizing it in automated scene understanding systems, we present interpretable *Object Identification using Contextual Information System* (OIC). Given a video, OIC detects, identifies, and tracks objects and their relationships over time to answer questions about the analyzed scene. Moreover, OIC makes predictions and infers the actors' intentions in a scene. To achieve this, our approach generates a scene graph containing classified objects and their semantic relationships. It then computes a *Frame Graph* by adding *Contextually Unique IDentifiers* (CUIDs) to each of the detected objects in the scene graph; the CUIDs permit tracking multiple object instances over time, even if the objects are visually identical. The CUIDs are then used to connect objects across a sequence of *Frame Graphs*, generating a *Temporal Graph*. This graph is exported as a cue for a pretrained Large Language Model to provide assistance and answer user questions. OIC's modular architecture enables simple comprehension and swapping of its components, making OIC more interpretable and maintainable than end-to-end scene understanding systems. Our quantitative and qualitative evaluation results demonstrate the effectiveness of OIC as a viable next step toward interpretable automated scene understanding systems.**

***Index Terms*—scene understanding, question and answering, temporal graphs, object identification, hybrid intelligence**

I. Introduction

In recent years, the focus on advancements in Artificial Intelligence (AI) has led to a renewed research agenda on Hybrid Intelligence (HI) [1]: The core objective is to create adaptive intelligent systems capable of augmenting human capabilities by leveraging and appropriately integrating the respective strengths of AI systems and people. Aligned with such research on HI, Digital Companions (DCs) have been proposed [2]–[4], using hypermedia environments to discover and interact with information, connected devices, digital services, and other DCs. The main objective of those DCs is to provide suitable and timely assistance and protection for users. A DC interacts with a user to provide such assistance or protection through different human interfaces (e.g., a mixed reality interface, a mobile application, etc.), and it can directly actuate connected devices in the user's environment (e.g., configuring the ambient heating system to keep a comfortable environment). Similar to the classic agent decision cycle [5], DCs operate in four main phases, namely: *Perceive*, *Interpret*, *Decide*, and *Interact* [6]. In the *Perceive* phase, a DC acquires relevant information (through sensors) about the current state of a user's environment. In the *Interpret* phase, a DC contextualizes the symbols perceived in the previous phase (through technologies such as knowledge graphs [7] or Large Language Models; LLM, cf. [8]). In the *Decide* phase, the DC computes suitable assistance or protection for the users; and it finally actuates on virtual services and physical objects in the environment to realize this in the *Interact* phase.

In this paper, we present *OIC* (Object Identification using Contextual Information; *Oh-I-See*), a novel DC system that exploits temporal scene understanding for providing user support by answering diverse types of questions. To do so, OIC utilizes a scene graph generation algorithm to obtain a symbolic representation of objects in a scene and their relationships. Then, OIC introduces the concept of *Contextually Unique Identifiers* (CUIDs) to follow the evolution of a scene with visually similar objects. Finally, OIC integrates this symbolic representation of a temporally evolving scene with commonsense knowledge of an LLM to decide on a follow-up action or respond to user questions. This paper describes the following contributions:

1) A novel method to assign unique identifiers to scene objects is introduced. These CUIDs consider the visual context of an object (e.g., neighboring objects) to discriminate among visually similar or equal scene objects. CUIDs are used to create what we refer to as *Frame Graphs* (see Section III-B).
2) A novel method to record changes in a scene's graph representation over time in so-called *Temporal Graphs*. This is accomplished by matching object CUIDs across Frame Graphs (see Section III-C). Temporal Graphs remain interpretable by humans and, in contrast to other approaches that consider solely the combination of the visual features of an object and its position in every frame of a video (e.g., by creating tubes [9] or using discriminative correlation filters [10]), they consider semantic relationships among scene objects as well.
3) Three validations of the OIC system, which integrates the above contributions (see Section IV): First, we quantitatively test OIC's scene interpretation capability using the video question-answering LifeQA [11] bench-

2375-0197/24/$31.00 ©2024 IEEE
DOI 10.1109/ICTAI62512.2024.00145

mark. Second, we quantitatively compare OIC's scene interpretation performance with versions of OIC that have been deprived of CUIDs and Temporal Graphs, respectively. Finally, we present a qualitative evaluation in which we asked 34 participants to evaluate OIC's scene interpretations by answering questions based on two videos shot in an office and an outdoor scenario. Our evaluations shows that OIC achieves acceptable scene interpretation performance, that its performance strongly depends on CUIDs and Temporal Graphs, and that people strongly agree that OIC's answers about a scene match answers they would have formulated themselves.

4) Unlike current end-to-end scene understanding systems that sacrifice interpretability (cf. [12]), OIC's modular neuro-symbolic architecture preserves people's ability to comprehend (and correct) intermediate system outputs. This continues recent research (e.g., [12]–[14]) on the trade-offs between end-to-end scene understanding systems' accuracy and their interpretability. While OIC's architecture is application-agnostic, it permits component-level error detection and allows swapping out the Scene Graph (SG) generation algorithm and the LLM to achieve desired accuracy/interpretability trade-offs in specific use cases.

This paper is structured as follows: After presenting relevant related work in Section II, we describe the OIC system and its implementation in Section III. Section IV presents OIC's validation in the setups described above; and Section V offers a discussion of our approach to lead to the conclusion and future work in Section VI.

II. Related Work

In the following, we describe relevant works on object identification and re-identification in Section II-A. We elaborate on SG generation algorithms for scene understanding in Section II-B. Finally, Section II-C gives an overview of the current research that leverages LLMs for visual question-answering tasks.

A. Object Identification and Re-Identification

Object identification is a fundamental task in many domains, including medicine, logistics, and manufacturing. It also involves deciding whether two detected objects are the same, which is essential for tracking and understanding objects [15]. Recent research on object identification has focused on utilizing object features such as shape, texture, and class [16] as well as pre-assigned identifiers, such as serial numbers and barcodes. For example, MotionTrack [17] and Caesar [9] use an object's location (bounding box) and the feature vectors inside the bounding box across consecutive frames (as *tubes*) for object identification and tracking. However, these methods are susceptible to abrupt object motion, changing appearance patterns, object-to-object and object-to-scene occlusions, and camera movements [9]. To overcome the occlusion issue, Somers et al. [18] discuss the importance of part-based representation learning. Meanwhile, Amiri et al. [19] propose a scene analysis approach for robot planning (SARP) that identifies the objects in the path of a robot using the labels and the locations of the objects extracted from a scene graph. The current state of research in the object (re-)identification space, its focus on object features, and its current expansion into multi-modal methods motivates more research on using information from an object's scene context and temporal information for object re-identification.

B. Deeper Understanding through Scene Graphs

In addition to re-identifying objects in dynamic scenes, DCs, and similar assistance systems require a deeper understanding of the scene to be able to derive protection or assistance measures for users. One way to achieve this understanding is by observing the interactions that occur between objects. For example, by recognizing their *semantic relationships* (e.g., `(object1, next-to, object2)`, or `(object1, is-holding, object2)`. Considering these visual relationships is possible for instance through SG generation algorithms, which correspond to a symbolic representation of a scene including objects, their attributes, and relationships with other objects [20]. In a SG, the nodes correspond to detected objects and the edges represent their pairwise relationships while the SG typically contains object-bounding boxes as well. Gu et al. [21] classify SG generation algorithms in two categories: a) methods that detect objects first and then recognize the relationships between them (e.g., [22], [23]), and b) methods that identify objects and their relationships at the same time based on combined object-and-relationship proposals and training data (e.g., [24], [25]). Furthermore, to maximize the number of relationships that SG generation algorithms can detect, it has been proposed to take advantage of symbolic knowledge, which is often referred to as "external knowledge." Chen et al. [26] propose a framework that incorporates a Knowledge Graph (KG) for visual reasoning that *predicts* objects in an image although these are not recognizable by common object detectors (e.g., due to occlusion). In [27], an object recognition algorithm is enhanced with knowledge-grounding modules for spatial relationship detection and validation of a recognized object. One of the most recent approaches to generate SGs is *Relation Transformer* (RelTR) [28], which infers a set of *subject-predicate-object* triplets from an image. RelTR outperforms existing SG generation methods offering fast inference time. However, SG generation algorithms are still in their infancy, and due to the dual task of recognizing objects and relationships, they tend to exhibit lower object recognition accuracy compared to pure object detectors [19], [29].

C. Large Language Models for Question Answering

A symbolic representation of a scene (including, but not limited to, SGs) may be contextualized through common-sense knowledge of an LLM. This could allow, for example, answering questions about a scene and making predictions about its evolution. Several visual scene-understanding systems have

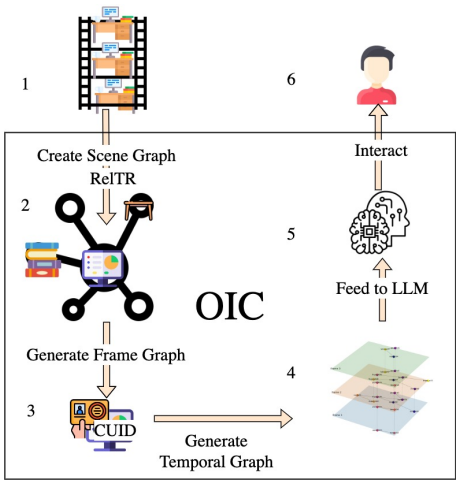

Fig. 1. Given a sequence of still images, OIC uses an *SG Generator* (1) which produces triples of detected objects and their relationships (2); these triples are used by a *Frame Graph Generator* to assign contextually unique identifiers (CUIDs) (3) to the objects. All *Frame Graphs* are then integrated into a *Temporal Graph* (4) to track the evolution of objects and relationships in the scene over time. Next, the *Temporal Graph* is exported as text and fed into an LLM which enriches the description of the scene with external knowledge (5), allowing it to make decisions, such as answering user questions (6).

increasingly leveraged LLMs' common-sense knowledge in performing visual question-answering and vision-centric tasks, including object detection, segmentation, and grounding. For example, Guo et al. [30], propose Img2Prompt. This plug-and-play module creates captions for the given images to bridge the disconnection between vision and language modality, enabling LLMs to understand a scene and provide a zero-shot visual Q&A service for the users. VisualBERT [31] uses a layered transformer and attention mechanism to generate precise image captions for grounding language elements of an LLM to specific image regions. This grounding enhances the LLM's responses to user queries. Alongside, VisionLLM [32] addresses vision-centric open-ended tasks (e.g., object detection and grounding) using a system that combines a language-guided image tokenizer and an LLM-based task decoder, broadening the application range of vision-based LLMs. To further enhance scene understanding using LLM-based visual question-answering systems, researchers are exploring the addition of different types of data (e.g., audio [33]) and training the LLMs to be able to offer causal reasoning [34], despite the computational expensiveness of LLMs (specifically GPT-based systems) in performing reasoning for visual tasks [35].

III. OIC: Modular Scene Understanding

In this section, we introduce OIC and its four main components (see Figure 1): a *Scene Graph Generator*, a *Frame Graph Generator*, a *Temporal Graph Builder*, and an *LLM*. The *Scene Graph Generator* (see Section III-A) receives as input images from a video and outputs a set of triples denoting detected objects and their relationships; this SG together with additional visual information is then processed by the *Frame Graph Generator* (see Section III-B) to add CUIDs to distinct objects; we refer to the result of this process as a *Frame Graph*. Next, the *Temporal Graph Builder* (see Section III-C) integrates multiple *Frame Graphs* to create a temporal graph representation that captures the evolution of the identified objects in a scene: a *Temporal Graph*. This graph is then converted into a natural language and used as a cue to a pre-trained *LLM* (see Section III-D), which contextualizes the temporal scene representation and provides assistance to OIC users through question-answering and contextual services.

Within OIC, these components are loosely coupled and can be swapped to optimize for different non-functional requirements (e.g., computational cost, degree of interpretability, etc.) in a variety of settings. This modularity allows for better generality (i.e., to be used for applications other than Q&A), and it also caters to the interpretability of the system: For instance, the SG generation algorithm can be easily replaced by other algorithms, such as a Graph Region-based Convolutional Neural Network (Graph R-CNN) [36], or Iterative Scene Graph Generation with Generative Transformers (IS-GGT) [37]—specific SG generation algorithms are expected to be more beneficial depending on the target application. Similarly, the LLM module can be switched from GPT [8] to Llama [38], or any other model, again depending on application-specific requirements. This approach simplifies addressing of shortcomings that are due to limited interpretability, limited perception [19], or computational cost [35].

A. Scene Graph Generator

SG generation algorithms, such as RelTR [28], Graph R-CNN [36], and IS-GGT [37] produce symbolic representations of objects in an image and their relationships, making them suitable for creating *CUIDs* and *temporal graphs* using scene graphs. OIC uses RelTR with pre-trained weights on the Visual Genome dataset [39]. After processing a given image (or video frame), RelTR produces triples in the form (s, p, o), e.g., `(woman, holding, bottle)`, where the subject and object in each triple are associated with the object's bounding box and the predicate corresponds to the predicted relationship between the two objects.

B. Frame Graph Generator

OIC assigns CUIDs, which are contextually unique labels, to each scene object to enable disambiguation between similar-looking objects. For example, an object detected but not uniquely identified in an image can be part of several triples in a scene graph, e.g., in `(woman, next to, table)` and `(phone, on, table)` the `table` may be the same entity, or there may be several tables in the scene. CUIDs permit disambiguating this situation: to uniquely identify an object in a scene, OIC considers three measures: (1) **class similarity**– the *class label* that the object was classified as by the object detector of the SG generation algorithm (e.g., `phone`), (2) **neighbor similarity**– the object's *relationships with the other objects* in the frame, and (3) **spatial similarity**– the object's *location* in the frame (i.e., the center of its bounding box).

OIC aggregates these three aspects to compute an overall numerical *similarity score* between pairs of detected objects

Fig. 2. **Left**: Individual frame with the overlay of the identified objects and their CUIDs. OIC computes different CUIDs for the two visually identical bottles in the scene by using their relationships with other detected scene objects, such as the woman `holding` one of the bottles (i.e., object `bottle_12df`). **Right**: The environment has changed (as the woman is not holding the bottle anymore), but the bottle is still detected and assigned the same CUID by our system.

in a scene. We consider two objects identical if their similarity score deviates less than $\varepsilon = 0.3$ from full congruence i.e., 1. This optimal value of ε was found after an evaluation process in which OIC was deployed for ten different scenes while varying the value of ε in the range $[0, 1]$. Through manual inspection of the resulting CUIDs across these trials, the value $\varepsilon = 0.3$ was found to be most suitable for re-identifying objects in the scene. Based on the similarity scores, our system creates unique CUID labels. CUIDs consist of a random alphanumeric string (e.g., `12df`) that is concatenated to the class label (e.g., bottle) of an object. Figure 2 illustrates CUIDs in a typical visual scene and highlights the ability of this approach to disambiguate between visually identical objects (i.e., `bottle_12df` and `bottle_8f26`): while the scene contains two bottles, OIC associates only one of them with the bottle being held by the woman (i.e., `woman holds bottle`) in the left frame. Since OIC aggregates three similarity measures to assign CUIDs, these remain stable with respect to changes in a subset of these measures. For instance, even though the relation `woman holds bottle` ceases to exist in the right frame in Fig. 2, the bottle retains its CUID. We elaborate more on this object's similarity check and re-identification across subsequent frames in Section III-C.

C. Temporal Graph Builder

Using CUIDs, OIC can re-identify objects in subsequent frames and hence track the evolution of a scene. This is done through the creation of *Temporal Graphs*. A Temporal Graph is built by iterating over all CUID-assigned objects in the *Frame Graph* of the current scene, the objects are then matched to appropriate nodes in the *Temporal Graph*, which is gradually built across the frames of a video. The underlying process is shown in Algorithm 1. As input, the algorithm receives:

- the current TemporalGraph$_t$ at time step t,
- the FameGraph$_t$ of the current frame,
- a weighting parameter $\alpha \in [0, 1]$,
- a threshold for minimum confidence Confidence$_{min}$, and
- a set S = {'has', 'attached to', 'covering', 'growing on', 'hanging from', 'made of', 'on', 'painted on', 'part of', 'wears', ...} of predicates which we considered stable (unlikely to change) in immediately succeeding frames.

The process then iterates over all nodes of the *Frame Graph* f (i.e., over every detected object in an image; see Line 3 in Algorithm 1), looking for a node in the *Temporal Graph* $g \in G$ that matches the current node. If no similar node is found, a new node is added to the *Temporal Graph* (Line 15).

Algorithm 1 Algorithm for inserting a Frame Graph into a Temporal Graph.

1: **function** INSERTFRAMEGRAPH(TemporalGraph$_t$, FrameGraph$_t$, α, Confidence$_{min}$, S)
2: $G \leftarrow$ TemporalGraph$_t$.nodes ▷ We store the nodes in a set to reduce the search space after successful matches
3: **for** $f \in$ FrameGraph$_t$.nodes **do**
4: **for** $g \in G$ **do** ▷ Calculate similarity scores between f and each g in the temporal graph
5: Similarity$_f(g) \leftarrow \alpha$ NEIGHSIMILARITY(f, g) + $(1 - \alpha)$ SPATSIMILARITY(f, g) + CLASSSIMILARITY(f, g)
6: **end for**
7: **for** $n \in G$ **do** ▷ Compare f to each node n in the temporal graph
8: **if** max Similarity$_f(n) <$ Confidence$_{min}$ **then**
9: **if** node.predicate$_f \in S$ **then**
10: BestMatch$_f \leftarrow$ argmax$_n$ Similarity$_f(n)$ ▷ The node in the temporal graph that is most likely to be same in f
11: TemporalGraph$_{t+1}$.update(BestMatch$_f$, f)
12: $G \leftarrow G-$ BestMatch$_f$ ▷ Remove the best match from G as it is already assigned
13: **else**
14: CUID$_f \leftarrow$ GETCUID(f)
15: TemporalGraph$_{t+1}$.add(f, CUID$_f$)
16: **end if**
17: **else**
18: BestMatch$_f \leftarrow$ argmax$_n$ Similarity$_f(n)$ ▷ The node in the temporal graph that is most likely to be the predecessor of f
19: TemporalGraph$_{t+1}$.update(BestMatch$_f$, f)
20: $G \leftarrow G-$ BestMatch$_f$
21: **end if**
22: **end for**
23: **end for**
24: **for** $e \in$ Framegraph$_t$.edges **do**
25: **if** $e \in$ TemporalGraph$_t$.edges **then**
26: TemporalGraph$_{t+1}$.append(e)
27: **else**
28: TemporalGraph$_{t+1}$.add(e)
29: **end if**
30: **end for**
31: **return** TemporalGraph$_{t+1}$
32: **end function**

The decision of whether a new node should be created for a detected object or whether this object should be matched to an existing node from the previous scene hence depends

on a similarity score (Line 5). This score is calculated by considering the number of common neighbors of the objects in the SG (NEIGHSIMILARITY), the overlap of the objects' bounding boxes (SPATSIMILARITY), and the similarity in their classes (CLASSSIMILARITY). Each similarity score yields a value in the range $[0, 1]$.

The weighing parameter α in the range $[0, 1]$ (see Line 5) emphasizes neighbor or spatial similarity. If objects move smoothly from frame to frame, α should be set closer to 0 (emphasizing spatial similarity) while neighbor similarity should be emphasized when more movement of objects is expected (or when the frame rate is low). If one of the three similarity scores is insufficient for ensuring unique identification and tracking (e.g., in case of inaccurately positioned bounding boxes), using the other two similarity scores can help keep tracking the object. This strategy makes the OIC system resilient to visual occlusions in contrast to traditional pixel-matching-based methods, such as using *tubes* [9], [17].

To ensure the accuracy of the re-identifying process using the temporal graph, it is crucial to address the issue of having objects in the temporal graph with multiple CUIDs. This could be due to wrongly detected or undetected relationships between an object and other objects in the environment. Therefore, if the similarity score of a detected object is less than the minimum confidence value of 0.6 (in Line 8 and obtained after a quantitative evaluation), OIC checks if the object's relationships (predicate) with other objects in the environment belong to the set S (see Line 9). S contains those relationships that are detectable by RelTR that we consider *sufficiently stable*: they tend to remain unchanged across immediate successive frames (e.g., the relationship `part-of` or `has`). In case the predicate of the evaluated node belongs to set S, it is matched with an existing object that was present in the time step t. If not, a new CUID is assigned to the object (see lines $13 - 15$).

Finally, the edges (i.e., relationships among objects) of the *Frame Graph* are inserted into the *Temporal Graph* by either creating a new edge (Line 28) between objects or appending the new (evolved) edge to the list of already-existing edges (Line 26). Unlike node insertion, existing edges are not updated but appended to an edge list to store the evolution of the relationship between objects over the considered scene. With this step, *TemporalGraph*$_t$ is integrated with *Frame Graph*$_t$, which returns *TemporalGraph*$_{t+1}$.

Using the generated *Temporal Graphs* from videos, OIC captures a large amount of information about the evolution of a scene; as the resulting format is compact (object CUIDs and relationships as semantic triples), the system can capture information over a considerable timeframe of several days or weeks at typical frame rates. In principle, this information is sufficient to respond to queries that include a temporal and an identification component, such as "Which of these bottles has been used by a person?" However, to go one step further than enabling the system to answer such *locally contextualized* queries, we propose to enrich OIC's *Temporal Graphs* with general knowledge from an external source.

D. Contextualizing the Temporal Graph using an LLM

To this end, OIC exports the generated *Temporal Graph* by converting triples into prose using a simplified version of the TEKGEN method (cf. [40]) that is designed to verbalize knowledge graphs. This output is then used as a cue for an LLM that has been pre-trained separately from the OIC system. To preserve information about the temporal evolution of a scene, we take a frame-based approach by inserting temporal statements into the generated prose text. This is best illustrated with an example: The input for the LLM from a video in which a person with a hat sits next to a table, then takes off their hat and drinks from a cup before leaving without picking up the hat from the table is described as:

> The scene opens with: person wears shirt, person wears hat, hat on head; After 1 second: person sits on chair, hat in hand; After 4 seconds: person holds cup, hat on table; After 7 seconds: person near door; hat on table;

OIC's LLM component may be fine-tuned with scene-specific knowledge to integrate this information for scene understanding, corresponding to a DC's *Interpret* phase. The LLM is furthermore used in a DC's *Decide* and *Interact* phases—for answering user questions and interacting with external components on a user's behalf, respectively. To enable this, our implementation contains a Web-based user interface to interact with OIC.

IV. VALIDATION OF THE OIC SYSTEM

In this section, we show the results of three different validations of OIC. Section IV-A describes a quantitative validation of OIC with the LifeQA dataset [11], Section IV-B presents different versions of OIC to highlight the impact of *CUIDs* and *Temporal Graphs*, and Section IV-C explains a qualitative user evaluation of OIC's output.

A. Validating OIC on the LifeQA benchmark

After surveying state-of-the-art Q&A benchmarks to find the best one to compare OIC with, we found out that existing benchmarks [34], [41], [42] present several limitations, such as restricted question types, lack of visual diversity, and hyper-specialization on the dataset the benchmarks have been trained on (e.g., a specific TV show). These restrictions limit the fair comparison with our generalist approach. Furthermore, several of the surveyed benchmarks use only short video clips that do not contain enough contextual information to permit answering complex questions about entire scenes of considerable length, which was the motivation behind developing OIC.

For evaluating OIC, we finally selected the LifeQA benchmark [11] as the best proxy to demonstrate the capabilities of OIC. LifeQA consists of 275 YouTube video clips of average length $\approx$ 2 minutes and 2,326 associated multiple-choice questions (each video clip corresponds to 3-5 questions). We considered 237 of these clips because the other 38 were unreachable through the links that the dataset provides which reduced the total number of multiple-choice questions to

TABLE I
EVALUATION OF OIC ON THE LIFEQA DATASET.

Input to System	Approach	Accuracy
Questions, MC Answer Options, Video Stream	ST-VQA-Tp. [43]	45.0
	ST-VQA-Sp.Tp. [43]	44.6
	Human [11]	48.5
	OIC with GPT-4	**33.14**
	OIC with Llama 3	**31.56**

1,917. While this benchmark is generally used for evaluating systems that use video and voice data (transcriptions), it has also been assessed for only video-based QA models, such as the two variants of ST-VQA [43]: *ST-VQA-Tp* and *ST-VQA-Sp.Tp*, where (*Sp.*) indicates spatial attention and (*Tp.*) refers to the used temporal attention mechanism.

To facilitate the comparison of OIC's responses with LifeQA's ground truth, we prompted OIC to select answers without producing any additional description. Thus, for a LifeQA question and multiple-choice (MC) answers such as: *Who is the first to have breakfast? (1) father, (2) mother, (3) girl, (4) boy*. OIC reformulated this question as a prompt to the LLM: "*Who is the first to have breakfast? Choose the most likely answer among these four options: (1) father, (2) mother, (3) girl, (4) boy. Respond only with a single integer value between 1 and 4. Do not produce any other output*."

Table I shows the results obtained by OIC on the LifeQA benchmark with only video data. We ran the benchmark with two LLMs, resulting in overall accuracies of 33.14% (OpenAI GPT-4 [8]) and 31.56% (Llama 3 [38]). These results are lower compared to the other models evaluated on LifeQA [11] (and human participants), but it is important to highlight that– unlike those systems–OIC was not trained or fine-tuned on the LifeQA dataset and its accuracies hence can be considered to be much more generalizable than the ST-VQA variants.

Furthermore, 34% of LifeQA's questions focus on colors and object counts. However, RelTR (i.e., our implementation's SG generation algorithm) does not have a general concept of color and only few objects (e.g., "Black Car") are annotated with their color—this is why the generated SG does not retain most object colors. Finally, RelTR, being focused on the creation of scene *graphs*, does not consider isolated objects that do not have visible relationships to other objects.

B. Validating the Impact of CUIDs

We provide a separate evaluation of the effect of CUIDs and information about the temporal evolution of a scene on OIC's scene interpretation performance. To this end, we set OIC (with the Llama3 LLM) as the baseline and compare it to two alternative configurations: Configuration A describes RelTR (i.e., SG with neither temporal information nor CUIDs), and Configuration B describes OIC with temporal information (i.e., retaining the per second scene information as described in Section III-D) but without CUIDs. Both A and B hence produce SGs with object relations such as `man using laptop` and `man wearing shirt`, but they do not disambiguate between different objects of the same type.

TABLE II
EVALUATION OF CUIDS WITH ALTERNATIVE CONFIGURATIONS.

Approach (Llama 3)	Accuracy
Baseline: OIC	31.56
Configuration A: RelTR [28]	**25.23**
Configuration B: OIC without CUID	**27.15**

We used the same evaluation approach as before, i.e., the LifeQA benchmark, and Table II shows the results of our experiment. While all three systems can answer some of the questions successfully,[1] the lack of contextual identification with CUIDs leads to markedly lower performance in both alternative systems: 25.23% for SGs from RelTR and 27.15% for OIC without CUIDs, demonstrating the positive impact of contextual identities on scene interpretation performance (comparison between configurations A and B) and their positive impact on temporal graph construction (comparison between Configuration B and Baseline).

C. Validating OIC in Real World Settings

In contrast to the previous two validations and to evaluate OIC on scenes and situations that are closer to its intended application in the context of DCs, we used two real-world scenarios: An *Office* scenario and an *Outdoors* scenario. Inspired by the STAR benchmark [42], we posed seven question categories: Interaction (e.g., *What is a person doing with the laptop*?), Sequence (e.g., *What happened to the bottle on the desk?*), Prediction (e.g., *What tool might the person require next?*), Feasibility (e.g., *What can the person do with the laptop?*), Contextual Search (e.g., *Where is the book that the woman was holding?*), Preferences (e.g., *What might be a person's favorite/preferred object?*), and Ownership (e.g., *Who might the notebook belong to?*).

We conducted an online survey with 34 participants, recruited through our university channels and who were not compensated for their participation. An evaluation of our study with our institution's ethics committee confirmed that no ethics review was required for our study. 12% of participants are between 21 and 30 years (average 25 years) of age and the rest are between 30 and 49 years (average 36 years) of age. Participants were asked to watch videos of each scenario, starting with the Office scene. After watching the first video, they answered questions about the scene from each of the seven categories; answers were provided as free text. Participants were subsequently presented with answers computed by the OIC system (using GPT-4) for each question. They then rated their agreement with the statement *"This is an answer that I might have formulated for this question"* on a 5-point Likert scale (i.e., 1/Strongly Disagree; 2/Disagree; 3/Undecided, 4/Agree, and 5/Strongly Agree). After finishing the first video, participants were asked to watch the video of the *Outdoor* scene and follow the same steps. The survey was performed online and participants completed it in 10-15min.

Table III summarizes the responses of our 34 survey participants with respect to the similarity of their own answers

[1]Note that random guessing already yields an accuracy of 25%

TABLE III
EVALUATION OF OIC WITH HUMAN USERS.

Type of Question	Agree/Strongly Agree		Disagree/Strongly Disagree	
	Office	Outdoors	Office	Outdoors
Interaction	88.2%	76.4%	5.8%	17.6%
Sequence	58.8%	46.1%	32.3%	26.4%
Prediction	55.9%	55.9%	23.5%	23.5%
Feasibility	79.4%	47.1%	5.9%	32.4%
Contextual Search	35.3%	44.1%	50%	26.5%
Preference Inference	64.7%	38.2%	20.6%	38.3%
Ownership Inference	58.8%	61.8%	20.6%	26.5%
Overall (Average)	63.01%	52.8%	22.67%	27.31%

and those given by the OIC system. The numbers show overwhelming agreement of the participants with the system's answers. In this sense, OIC performed worst in the *Contextual Search* questions with only 35.3% and 44.1% agreement (agree and strongly agree) on average in either scenario.

V. DISCUSSION

The validation of OIC on the LifeQA dataset enables us to compare OIC to other scene understanding systems in principle. However, the dataset's bias on voice data (as subtitles) and their pre-training on the LifeQA dataset put OIC—which uses only visual data and is not trained on the dataset—at an a-priori disadvantage. Still, considering no dataset-specific training and using simple prompt engineering for GPT-4 and Llama 3, we consider the obtained results as valuable for (interpretable) scene understanding systems; to improve on these results, we plan to consider the integration of further modalities, predominantly voice/sound information. The poor results using the SG generation algorithm only (i.e., Configuration A in Section IV-B) also hint at the system's poor object and relation detection model, which causes a *limited perception bottleneck* (cf. [19]) due to the limited number of detected object and relation categories in RelTR. However, OIC permits switching SG generation algorithms and LLM components to address these limitations, making the approach future-proof.

OIC's modularity, furthermore permits navigating the trade-off between accuracy and interpretability, in contrast to other current scene understanding systems (e.g., [12]–[14]). These studies have highlighted the trade-off between a scene understanding model's accuracy in detecting objects and relationships, and its interpretability for human experts. For example, Hofmarcher et al. [12] conclude that their proposed tiered approach for interpreting a visual scene understanding system gains higher interpretability at the cost of the lower overall performance of the system. However, our OIC system permits accessing the best of both worlds: It employs black-box systems—the scene graph generator (i.e., RelTR) and the LLM (e.g., Llama 3)—as well as highly interpretable components—specifically, the frame graph and temporal graph generators. These components generate a Temporal Graph for cueing the LLM with a textual representation of the evolving scene observed across Frame Graphs. This intermediate human-readable representation can be easily monitored to find lost details and correct wrong information in the Temporal Graph, thereby making the CUID generation, Frame Graph, and Temporal Graph components of OIC inherently interpretable. Moreover, the modular design of OIC permits easy swapping of the high-accuracy black-box models to more interpretable models or models that can be explained using external explainability tools, such as SHapley Additive exPlanations SHAP [44] or Graph Local Interpretable Model-Agnostic Explanations (GraphLIME) [45].

VI. CONCLUSIONS AND FUTURE WORK

We introduced OIC, a modular and interpretable system for temporal object identification and tracking in dynamic environments. OIC combines a sub-symbolic SG generation algorithm with contextual grounding to create a frame graph by assigning CUIDs to objects. It then generates a Temporal Graph to track and re-identify objects in an easily interpretable way. Then, this Temporal Graph is exported as a natural language to enable scene Q&A using an LLM. We validated OIC on the LifeQA benchmark, resulting in acceptable scene understanding performance, and highlighted the merit of assigning CUIDs and tracking temporal relationships. Unfortunately, we cannot evaluate the performance of OIC in its core domain—deeper interpretation of long scenes—using current benchmarks, and we consider the establishment of such a *scene interpretation benchmark* a promising area for future research. Finally, a qualitative evaluation of OIC with 34 users in real-world settings yielded encouraging results, with a majority of participants accepting OIC's answers as sufficiently similar to their own. We expect that OIC's modular approach opens new possibilities for the system toward interpretable multimodal scene understanding that involves, for instance, audio and environmental sensors.

REFERENCES

[1] Z. Akata, D. Balliet, M. De Rijke, F. Dignum, V. Dignum, G. Eiben, A. Fokkens, D. Grossi, K. Hindriks, H. Hoos, H. Hung, C. Jonker, C. Monz, M. Neerincx, F. Oliehoek, H. Prakken, S. Schlobach, L. Van Der Gaag, F. Van Harmelen, H. Van Hoof, B. Van Riemsdijk, A. Van Wynsberghe, R. Verbrugge, B. Verheij, P. Vossen, and M. Welling, "A Research Agenda for Hybrid Intelligence: Augmenting Human Intellect With Collaborative, Adaptive, Responsible, and Explainable Artificial Intelligence," *Computer*, vol. 53, pp. 18–28, Aug. 2020.

[2] K. García, S. Mayer, A. Ricci, and A. Ciortea, "Proactive Digital Companions in Pervasive Hypermedia Environments," in *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, pp. 54–59, 2020.

[3] M. Kritzler, J. Hodges, D. Yu, K. Garcia, H. Shukla, and F. Michahelles, "Digital Companion for Industry," in *Companion Proceedings of The 2019 World Wide Web Conference*, (San Francisco USA), pp. 663–667, ACM, May 2019.

[4] J. Spirig, K. Garcia, and S. Mayer, "An Expert Digital Companion for Working Environments," in *Proceedings of the 11th International Conference on the Internet of Things*, IoT '21, (New York, NY, USA), p. 25–32, Association for Computing Machinery, 2022.

[5] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT press, 1999.

[6] K. Garcia, J. Vontobel, and S. Mayer, "A Digital Companion Architecture for Ambient Intelligence," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 2, pp. 1–26, 2024.

[7] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda,

S. Staab, and A. Zimmermann, "Knowledge Graphs," *ACM Computing Surveys*, vol. 54, pp. 1–37, May 2022.

[8] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.

[9] X. Liu, P. Ghosh, O. Ulutan, B. S. Manjunath, K. Chan, and R. Govindan, "Caesar: Cross-Camera Complex Activity Recognition," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, (New York New York), pp. 232–244, ACM, Nov. 2019.

[10] Z. Yang, J. Wu, and C. Long, "Learning Spatial-Corrected Regularized Correlation Filters for Visual Tracking," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1229–1236, IEEE, 2019.

[11] S. Castro, M. Azab, J. Stroud, C. Noujaim, R. Wang, J. Deng, and R. Mihalcea, "LifeQA: A Real-Life Dataset for Video Question Answering," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020.

[12] M. Hofmarcher, T. Unterthiner, J. Arjona-Medina, G. Klambauer, S. Hochreiter, and B. Nessler, "Visual Scene Understanding for Autonomous Driving using Semantic Segmentation," *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 285–296, 2019.

[13] Y. Sun, X. Wang, Y. Zhang, J. Tang, X. Tang, and J. Yao, "Interpretable End-to-End Driving Model for Implicit Scene Understanding," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2874–2880, IEEE, 2023.

[14] S. Yang, W. Wang, C. Liu, and W. Deng, "Scene Understanding in Deep Learning-based End-to-End Controllers for Autonomous Vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 53–63, 2018.

[15] T. Huang and S. Russell, "Object Identification in a Bayesian Context," in *IJCAI*, vol. 97, pp. 1276–1282, Citeseer, 1997.

[16] P. Pushkar, L. Aggarwal, M. Saad, A. Maheshwari, H. Awasthi, and P. Nagrath, "Object Identification in Satellite Imagery and Enhancement using Generative Adversarial Networks," in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 2*, pp. 643–657, Springer, 2021.

[17] Z. Qin, S. Zhou, L. Wang, J. Duan, G. Hua, and W. Tang, "MotionTrack: Learning Robust Short-term and Long-term Motions for Multi-Object Tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17939–17948, 2023.

[18] V. Somers, C. De Vleeschouwer, and A. Alahi, "Body Part-based Representation Learning for Occluded Person Re-Identification," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1613–1623, 2023.

[19] S. Amiri, K. Chandan, and S. Zhang, "Reasoning with Scene Graphs for Robot Planning under Partial Observability," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5560–5567, 2022.

[20] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image Retrieval using Scene Graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

[21] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, "Scene Graph Generation with External Knowledge and Image Reconstruction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1969–1978, 2019.

[22] B. Dai, Y. Zhang, and D. Lin, "Detecting Visual Relationships with Deep Relational Networks," in *Proceedings of the IEEE conference on computer vision and Pattern recognition*, pp. 3076–3086, 2017.

[23] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual Relationship Detection with Language Priors," in *European conference on computer vision*, pp. 852–869, Springer, 2016.

[24] Y. Li, W. Ouyang, X. Wang, and X. Tang, "ViP-CNN: Visual Phrase Guided Convolutional Neural Network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1347–1356, 2017.

[25] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable Net: An Efficient Subgraph-based Framework for Scene Graph Generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 335–351, 2018.

[26] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta, "Iterative Visual Reasoning Beyond Convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7239–7248, 2018.

[27] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved Image Captioning via Policy Gradient Optimization of Spider," in *Proceedings of the IEEE international conference on computer vision*, pp. 873–881, 2017.

[28] Y. Cong, M. Y. Yang, and B. Rosenhahn, "RelTR: Relation Transformer for Scene Graph Generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[29] A. Zareian, S. Karaman, and S.-F. Chang, "Bridging Knowledge Graphs to Generate Scene Graphs," in *European conference on computer vision*, pp. 606–623, Springer, 2020.

[30] J. Guo, J. Li, D. Li, A. M. H. Tiong, B. Li, D. Tao, and S. Hoi, "From Images to Textual Prompts: Zero-shot Visual Question Answering with Frozen Large Language Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10867–10877, 2023.

[31] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "VisualBERT: A Simple and Performant Baseline for Vision and Language," *arXiv preprint arXiv:1908.03557*, 2019.

[32] W. Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu, J. Zhou, Y. Qiao, *et al.*, "VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[33] J. B. Li, J. S. Michaels, L. Yao, L. Yu, Z. Wood-Doughty, and F. Metze, "Audio-Journey: Efficient visual+ LLM-aided Audio Encodec Diffusion," in *Workshop on Efficient Systems for Foundation Models@ ICML2023*, 2023.

[34] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "CLEVRER: Collision Events for Video Representation and Reasoning," *arXiv preprint arXiv:1910.01442*, 2019.

[35] J. Yang, X. Chen, S. Qian, N. Madaan, M. Iyengar, D. F. Fouhey, and J. Chai, "LLM-Grounder: Open-Vocabulary 3D Visual Grounding with Large Language Model as an Agent," *arXiv preprint arXiv:2309.12311*, 2023.

[36] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph R-CNN for Scene Graph Generation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

[37] S. Kundu and S. N. Aakur, "IS-GGT: Iterative Scene Graph Generation with Generative Transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6292–6301, 2023.

[38] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.

[39] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual Genome: Connecting Language and Vision using Crowdsourced Dense Image Annotations," *International journal of computer vision*, vol. 123, pp. 32–73, 2017.

[40] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou, "Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3554–3565, Association for Computational Linguistics, june 2021.

[41] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "MovieQA: Understanding Stories in Movies through Question-Answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.

[42] B. Wu, S. Yu, Z. Chen, J. B. Tenenbaum, and C. Gan, "STAR: A Benchmark for Situated Reasoning in Real-World Videos," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[43] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim, "TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[44] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[45] Q. Huang, M. Yamada, Y. Tian, D. Singh, and Y. Chang, "GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks," *IEEE Transactions on Knowledge and Data Engineering*, 2022.