Causal Event Graph-Guided Language-based Spatiotemporal Question Answering

Kaushik Roy*[1], **Alessandro Oltramari***[2], **Yuxin Zi**[1], **Chathurangi Shyalika**[1], **Vignesh Narayanan**[1], **Amit Sheth**[1]

[1]Artificial Intelligence Institute, University of South Carolina
[2]Bosch Center for Artificial Intelligence
kaushikr@email.sc.edu, alessandro.oltramari@us.bosch.com, yzi@email.sc.edu, jayakodc@email.sc.edu, vignar@sc.edu, amit@sc.edu

Abstract

Large Language Models have excelled at encoding and leveraging language patterns in large text-based corpora for various tasks, including spatiotemporal event-based question answering (QA). However, due to encoding a text-based projection of the world, they have also been shown to lack a full-bodied understanding of such events, e.g., a sense of intuitive physics, and cause-and-effect relationships among events. In this work, we propose using causal event graphs (CEGs) to enhance language understanding of spatiotemporal events in language models, using a novel approach that also provides *proofs* for the model's capture of the CEGs. A CEG consists of events denoted by nodes, and edges that denote cause-and-effect relationships among the events. We perform experimentation and evaluation of our approach for benchmark spatiotemporal QA tasks and show effective performance, both quantitative and qualitative, over state-of-the-art baseline methods.

1 Introduction

Large Language Models have emerged as powerful candidates for *world models*, models that succinctly represent knowledge about the world and how it works, by demonstrating excellent performance across several challenging common-sense understanding benchmark tasks (e.g., the Winograd challenge) (Levesque, Davis, and Morgenstern 2012). However, they have yet to demonstrate a robust understanding of some basic physical phenomena, such as affordances (what is possible in a particular physical context, e.g., can you put a coin on a soap bubble?), causality (what events or effects necessarily need to follow a prior causal event?) (Susskind et al. 2021; Browning and LeCun 2023). In this work, we tackle the causality challenge and propose the use of causal event graphs as a mechanism to inform the model about cause-effect relationships among events, specifically within the experimental context of spatiotemporal QA. We work with the benchmark spatiotemporal QA datasets CLEVRER and CLEVRER-Humans (Yi et al. 2019; Mao et al. 2022). The datasets are a compilation of synthetically created videos of objects on a tabletop that can move around on the tabletop and collide with one another (see Section 2 and Figure 1 for dataset details), and the task involves answering questions about spatiotemporal events in the videos. The datasets also contain enough metadata to construct CEGs that capture the cause-effect relationships among the video events.

Prior Work and Gaps on the CLEVRER and CLEVRER-Humans QA Task

Prior Work

Pattern Recognition-Based Approaches. Prior work on the CLEVRER dataset has focused on pattern recognition-based approaches, where either the video and question patterns are compressed into distributed vector-based representations (e.g., using vision models and language models), and fed into a model that predicts different answer choices and their probabilities (Yi et al. 2019).

Toward Utilizing Structured Information The metadata in the CLEVRER-Humans dataset also consists of human-curated CEGs pertaining to each video. Consequently, researchers have since modified the pattern-recognition pipelines to utilize compressed representations of CEGs, e.g., using graph neural network-based methods (Wu et al. 2020)).

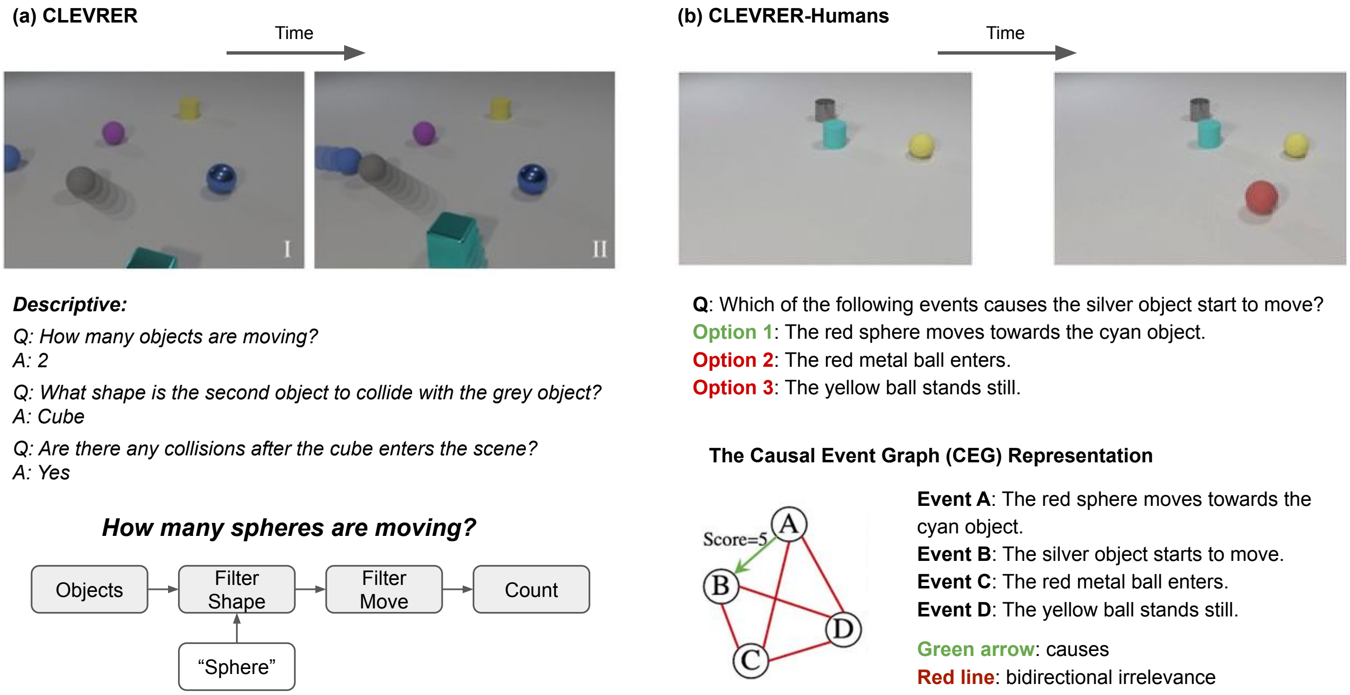
From Black Boxes to Methods with Proofs However, due to the black-box nature of pattern recognition methods, the exact mechanisms behind the model's functioning leave unanswered questions about the robustness of its causality understanding. Therefore researchers have also proposed neurosymbolic approaches that, instead of directly predicting the answer choices, predict a functional program that can then be executed on an interpreter to yield the answer. The program trace then serves as a *proof* that the model's internal structures correlate with explicit mechanisms (the functional programs) for QA (Mao et al. 2022).

Our main contributions in this paper are to address the two gaps discussed below.

Gaps

Intrinsic Knowledge *Proofs* Although prior work has demonstrated methods that possess both the high performance of pattern recognition-based methods and proofs that

*These authors contributed equally.



Descriptive:

Q: How many objects are moving?
A: 2

Q: What shape is the second object to collide with the grey object?
A: Cube

Q: Are there any collisions after the cube enters the scene?
A: Yes

How many spheres are moving?

Q: Which of the following events causes the silver object start to move?
Option 1: The red sphere moves towards the cyan object.
Option 2: The red metal ball enters.
Option 3: The yellow ball stands still.

The Causal Event Graph (CEG) Representation

Event A: The red sphere moves towards the cyan object.
Event B: The silver object starts to move.
Event C: The red metal ball enters.
Event D: The yellow ball stands still.

Green arrow: causes
Red line: bidirectional irrelevance

Figure 1: CLEVRER and CLEVRER-Humans dataset - CLEVRER consists of videos with video-based questions and answer choices for each video. There is also a functional program corresponding to each question which can be executed by an interpreter to get the right answer choice. The CLEVRER-Humans dataset is enhanced with CEG representations- the green arrow depicts the true causal relationship between nodes (events in the video), and the red arrow depicts false ones.

show the model's internal mechanisms correlate with explicit QA mechanisms, they do not provide proof of intrinsic knowledge of causality. In this work, we build on the features of prior works, namely powerful pattern-recognition pipelines for performance, predict functional programs for proof of QA mechanisms, and add a novel method to show proof of intrinsic causal knowledge capture. Specifically, the model trained using our method not only predicts the functional program that solves the QA task for the video but also predicts a CEG, which can then be compared to a ground truth CEG for the video provided in the dataset as proof of causal knowledge capture. This shows that the model's internal mechanisms correlate with explicit QA mechanisms, while also encoding information about causal knowledge capture, visible through the predicted CEGs (see Section 3 for methodology details).

Lack of Framework for Theoretical Analysis Although prior work has addressed leveraging pattern recognition-based methods, and neurosymbolic approaches, the objectives (e.g., loss functions) employed during training for both of these approaches are quite distinctly different from one another. It is not a guarantee that the objectives are synergistic in nature (the combined loss decreases and converges), even if they are demonstrated successfully one two synthetic benchmark datasets. This lack of guarantee is further compounded by our additional objective that constrains the model to predict high-fidelity CEGs (CEGs that closely resemble a ground truth). We, therefore, provide a theoretical analysis of convergence of our proposed method that shows stable model learning and loss convergence in both experimental settings where the objectives are synergistic, and settings where they are not (see Section 4 for analysis details).

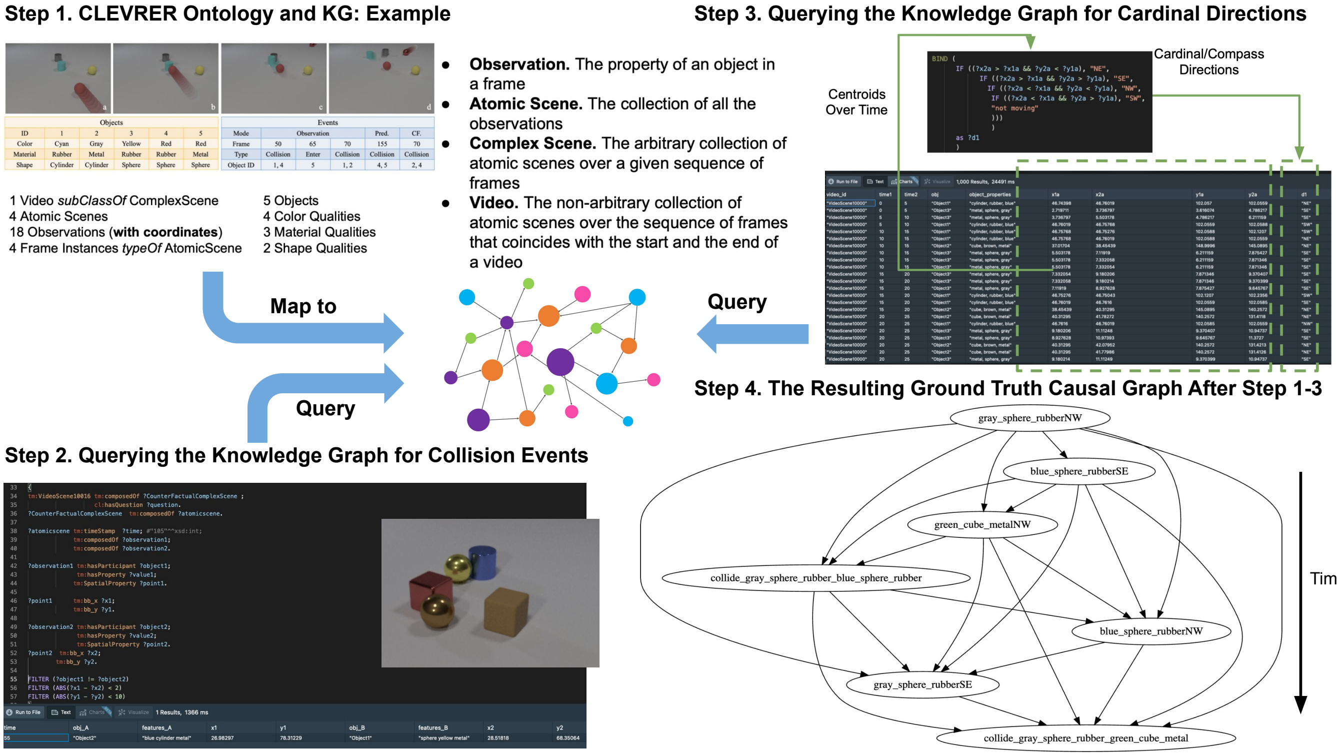
2 The CLEVRER, CLEVRER-Humans Datasets, CEGs and Training Objectives

CLEVRER and CLEVRER-Humans Datasets

The CLEVRER dataset is a compilation of videos and QA sets (questions and answer choices) corresponding to each video. The QA is centered around spatiotemporal events in the videos. Furthermore, the dataset also consists of ground truth-functional programs for each question that can be executed on an interpreter to get the correct answer choice (see Figure 1 (a)). The CLEVRER-Humans dataset consists of QA sets along with human-curated CEGs that show cause-and-effect relationships among events in the videos. The events have natural language descriptions (see Figure 1 (b)). Note that the CLEVRER-Humans dataset does not contain the functional programs, only the answer choices and CEGs.

CEG Enhanced CLEVRER Dataset

Although the CLEVRER dataset does not consist of human-curated CEGs, we use natural language processing and knowledge engineering techniques to extract CEGs from the metadata provided for each video in the dataset. Figure 2 shows the extraction process - we construct a knowledge graph, by mapping CLEVRER dataset-specific videos,



Step 1. CLEVRER Ontology and KG: Example

Objects					
ID	1	2	3	4	5
Color	Cyan	Gray	Yellow	Red	Red
Material	Rubber	Metal	Rubber	Rubber	Metal
Shape	Cylinder	Cylinder	Sphere	Sphere	Sphere

Events					
Mode	Observation			Pred.	CF.
Frame	50	65	70	155	70
Type	Collision	Enter	Collision	Collision	Collision
Object ID	1, 4	5	1, 2	4, 5	2, 4

1 Video *subClassOf* ComplexScene
4 Atomic Scenes
18 Observations (**with coordinates**)
4 Frame Instances *typeOf* AtomicScene

5 Objects
4 Color Qualities
3 Material Qualities
2 Shape Qualities

- **Observation.** The property of an object in a frame
- **Atomic Scene.** The collection of all the observations
- **Complex Scene.** The arbitrary collection of atomic scenes over a given sequence of frames
- **Video.** The non-arbitrary collection of atomic scenes over the sequence of frames that coincides with the start and the end of a video

Step 2. Querying the Knowledge Graph for Collision Events

```
{
tm:VideoScene10016 tm:composedOf ?CounterFactualComplexScene ;
                   cl:hasQuestion ?question.
?CounterFactualComplexScene  tm:composedOf ?atomicscene.

?atomicscene tm:timeStamp  ?time; #"105"^^xsd:int;
             tm:composedOf ?observation1;
             tm:composedOf ?observation2.

?observation1 tm:hasParticipant ?object1;
              tm:hasProperty ?value1;
             tm:SpatialProperty ?point1.

?point1      tm:bb_x ?x1;
             tm:bb_y ?y1.

?observation2 tm:hasParticipant ?object2;
              tm:hasProperty ?value2;
              tm:SpatialProperty ?point2.
?point2  tm:bb_x ?x2;
        tm:bb_y ?y2.

FILTER (?object1 != ?object2)
FILTER (ABS(?x1 - ?x2) < 2)
FILTER (ABS(?y1 - ?y2) < 10)
```

Run to File | Text | Charts | Visualize | 1 Results, 1366 ms

time	obj_A	features_A	x1	y1	obj_B	features_B	x2	y2
55	"Object2"	"blue cylinder metal"	26.98297	78.31229	"Object1"	"sphere yellow metal"	28.51818	68.35064

Step 3. Querying the Knowledge Graph for Cardinal Directions

```
BIND (
    IF ((?x2a > ?x1a && ?y2a < ?y1a), "NE",
        IF ((?x2a > ?x1a && ?y2a > ?y1a), "SE",
          IF ((?x2a < ?x1a && ?y2a < ?y1a), "NW",
           IF ((?x2a < ?x1a && ?y2a > ?y1a), "SW",
           "not moving"
           )))
           )
    as ?d1
    )
```

Run to File | Text | Charts | Visualize | 1,000 Results, 24491 ms

video_id	time1	time2	obj	object_properties	x1a	x2a	y1a	y2a	d1
"VideoScene10000"	0	5	"Object1"	"cylinder, rubber, blue"	46.74398	46.76019	102.057	102.0559	"NE"
"VideoScene10000"	0	5	"Object3"	"metal, sphere, gray"	2.719711	3.736797	3.616074	4.786217	"SE"
"VideoScene10000"	5	10	"Object3"	"metal, sphere, gray"	3.736797	5.503178	4.786217	6.211159	"SE"
"VideoScene10000"	5	10	"Object1"	"cylinder, rubber, blue"	46.76019	46.75768	102.0559	102.0588	"SW"
"VideoScene10000"	10	15	"Object1"	"cylinder, rubber, blue"	46.75768	46.75276	102.0588	102.1207	"SW"
"VideoScene10000"	10	15	"Object1"	"cylinder, rubber, blue"	46.75768	46.76019	102.0588	102.0559	"NE"
"VideoScene10000"	10	15	"Object2"	"cube, brown, metal"	37.01704	38.45439	148.9996	145.0895	"NE"
"VideoScene10000"	10	15	"Object3"	"metal, sphere, gray"	5.503178	7.11919	6.211159	7.875427	"SE"
"VideoScene10000"	10	15	"Object3"	"metal, sphere, gray"	5.503178	7.332058	6.211159	7.871346	"SE"
"VideoScene10000"	10	15	"Object3"	"metal, sphere, gray"	5.503178	7.332054	6.211159	7.871346	"SE"
"VideoScene10000"	15	20	"Object3"	"metal, sphere, gray"	7.332054	9.180206	7.871346	9.370407	"SE"
"VideoScene10000"	15	20	"Object3"	"metal, sphere, gray"	7.332058	9.180214	7.871346	9.370399	"SE"
"VideoScene10000"	15	20	"Object3"	"metal, sphere, gray"	7.11919	8.927628	7.875427	9.645767	"SE"
"VideoScene10000"	15	20	"Object1"	"cylinder, rubber, blue"	46.75276	46.75043	102.1207	102.2356	"SW"
"VideoScene10000"	15	20	"Object1"	"cylinder, rubber, blue"	46.76019	46.7616	102.0559	102.0585	"SE"
"VideoScene10000"	15	20	"Object2"	"cube, brown, metal"	38.45439	40.31295	145.0895	140.2572	"NE"
"VideoScene10000"	20	25	"Object2"	"cube, brown, metal"	40.31295	41.78272	140.2572	131.4118	"NE"
"VideoScene10000"	20	25	"Object1"	"cylinder, rubber, blue"	46.7616	46.76019	102.0585	102.0559	"NW"
"VideoScene10000"	20	25	"Object3"	"metal, sphere, gray"	9.180206	11.11248	9.370407	10.94737	"SE"
"VideoScene10000"	20	25	"Object3"	"metal, sphere, gray"	8.927628	10.97393	9.645767	11.3727	"SE"
"VideoScene10000"	20	25	"Object2"	"cube, brown, metal"	40.31295	42.07952	140.2572	131.4213	"NE"
"VideoScene10000"	20	25	"Object2"	"cube, brown, metal"	40.31295	41.77986	140.2572	131.4126	"NE"
"VideoScene10000"	20	25	"Object3"	"metal, sphere, gray"	9.180214	11.11249	9.370399	10.94737	"SE"

Step 4. The Resulting Ground Truth Causal Graph After Step 1-3

Figure 2: Construction of CEG from CLEVRER dataset videos.

frames, objects, and events to a well-established scene understanding ontology (Qasemi, Francis, and Oltramari 2023; Haller et al. 2019; Tiddi, Lécué, and Hitzler 2020); (2) we design suitable semantic queries to elicit collision events from this knowledge graph, (3) we query this knowledge graph for object movement directions, and (4) we consolidate and visualize the results obtained from the query into a CEG as depicted in the figure.

3 Methodology

We will use the examples in Figure 1(a) and (b) to explain the methodology. As described in the previous section, for the example in (a), the CEG similar to the example in example (b) is obtained through the process illustrated in Figure 2. First, consider the example in (a). For each question Q, and the corresponding functional program, e.g., "How many spheres are moving", and "Count(FilterMove(FilterShape(Objects,*Sphere*)))", we predict the concatenated question and program sequence, e.g., "How many spheres are moving Count FilterMove FilterShape ObjectsSphere", in an autoregressive manner using a feedforward neural network with a position encoder. We will denote this model as $\mathcal{M}(Q, \theta)$, where θ denotes all trainable parameters (e.g., the embedding layers, position encoder layers, and feedforward layers). Next, we perform the following steps (i) For each node in the CEG, we predefine a tokenization structure (e.g., the node for event A denoted is token 0, the node for event B denoted is token 1, and so on ..) and embed the node tokens using a node embedding layer, (ii) To each node embedding, we augment its embedding by adding a representation (embedding) of the question Q obtained from the last layer of $\mathcal{M}(Q, \theta)$. Note that we require that the embedding sizes remain the same for the embedding addition to be valid, and (iii) Using the augmented node embeddings, we reconstruct a directed graph by calculating the sigmoid of the KL divergence between every pair of node embeddings after correcting for domain errors (e.g., log of 0 or negative numbers). We will denote the steps (i)-(iii) by the function $\mathcal{G}(Q, \theta')$, where θ' denotes the trainable parameters relevant to steps (i)-(iii) (e.g., the node embedding layer). We minimize the objective function:

$$\mathtt{CE}(\mathcal{M}(Q, \theta), \mathtt{targets}) + \alpha(Q)\mathtt{MSE}(\mathcal{G}(Q, \theta'), \mathtt{CEG}_{\mathtt{gt}}) \quad (1)$$

In the above Equation 1, the terms in the first summand $\mathtt{CE}$ and $\mathtt{targets}$, maintain their traditional autoregressive training objective definitions, namely the cross-entropy loss and next token, respectively. The terms in the second summand $\mathtt{MSE}$, and $\mathtt{CEG}_{\mathtt{gt}}$, refer to mean squared error and the adjacency matrix for the ground truth CEG, respectively. The intention of the second summand is to minimize the error between the reconstructed directed graph and the ground truth CEG. Since the next token prediction and graph reconstruction losses may not necessarily be minimizable synergistically, we include a question-specific Lagrange multiplier network. The network is a two-layer feedforward network with a ReLU-activated output (because Lagrange multipliers are always positive). The interpretation is that if the value of the multiplier is high, for that Q, the token prediction and graph reconstruction losses can be synergistically minimized.

After a model is trained to minimize Equation 1, reconstructing the graph using $\mathcal{G}(Q, \theta')$ serves as the ***Instrinsic Knowledge Proof*** *of whether or not the causal event knowledge necessary to answer input question Q is being captured. The proof can be compared to the ground truth CEG for verification and interpretation.*

4 Theoretical Analysis

Here, we will make use of the canonical proofs for gradient descent and stochastic gradient descent to prove that the objective in Equation 1 will have a minimum always. For brevity, we will denote $\texttt{CE}(\mathcal{M}(Q, \theta), \texttt{targets})$ by $f(\theta)$, $\alpha(Q)$ by λ, and $\texttt{MSE}(\mathcal{G}(Q, \theta'), \texttt{CEG}_{\texttt{gt}})$ by $g(\theta')$.

Theorem 4.1. *Proof of Convergence using Gradient Descent (GD) for finding a minimizer*

$$\theta^* = \arg\min_{\theta} f(\theta) + \lambda g(\theta') \tag{2}$$

Proof. First, we write GD formula as follows:

$$\frac{\theta_{t+1} - \theta_t}{\delta_t} = -\nabla(f(\theta) + \lambda g(\theta')) \tag{3}$$

Here, g is the squared distance between the graph abstraction (a matrix) and the transitive closure applied on the ground truth graph (an adjacency matrix), and λ is a penalty that is proportional to this distance. This can be seen as a finite difference approximation of the derivative of the continuous function $f(\theta) + \lambda g(\theta')$, i.e., a discretization of the ordinary differential equation

$$\dot{\theta}_t = -\nabla(f(\theta_t) + \lambda g(\theta_t)) \tag{4}$$

Equation (4) evaluated at time t yields iterate θ_t after some steps of GD. Let θ^* be the minimizer of $(f(\theta_t) + \lambda g(\theta_t))$. We denote $f(\theta_t) + \lambda g(\theta_t)$ using the short hand $F(\theta)$. We make two assumptions. First, we assume that F is strongly convex (locally), i.e., $F(x) - F(y) + \nabla F(y)(y - x) \geq \frac{\mu}{2}||x - y||^2$, i.e., for any point of F, there is a quadratic function that bounds its growth. Second, we assume that F is L-Lipshitz (strong smoothness), i.e., $F(x) - F(y) + \nabla F(y)(y - x) \geq \frac{\mu}{2}||x - y||^2 \leq \frac{L}{2}||x - y||^2$. We can also write this as $F(x) - F(y) + \nabla F(y)(y - x) \geq \frac{1}{2L}||\nabla F(x) - \nabla F(y)||^2$. These are not restrictive assumptions as it is generally true (locally - zoomed in at a particular point) for arbitrary neural networks.

We now define an energy function and show that this energy is a Lyapunov function. Finally, we bound the energy and obtain a convergence rate. We define energy as:

$$E(\theta) = \frac{1}{2}||\theta - \theta^*||^2$$

Three out of four properties of a Lyapunov, i.e., (1) E is continuous, (2) $E(\theta_t) = 0$ if and only if $\theta_t = \theta^*$, and (3) $E(\theta_t) > 0$ if and only if $\theta_t \neq \theta^*$ trivially hold. (1) because E is a composition of continuous functions, (2) and (3) because of the definition of a norm (remember that g is also a squared norm between the graphs). Now we prove the fourth property which says that $E(\theta_{t+1}) \leq E(\theta_t),\ \forall t$. After some algebraic manipulation, we get

$$E(\theta_{t+1}) - E(\theta_t) = \frac{1}{2}||\theta_{t+1} - \theta_t||^2 + (\theta_{t+1} - \theta_t) \cdot (\theta_t - \theta^*) \tag{5}$$

Replacing $\theta_{t+1} - \theta_t$ using Equation (3), we get

$$\frac{1}{2}\delta_t^2||\nabla F(\theta_t)||^2 + (-\delta_t \nabla F(\theta_t)) \cdot (\theta_t - \theta^*)$$

We can bound this expression using strong convexity and smoothness to obtain

$$\begin{aligned} & E(\theta_{t+1}) - E(\theta_t) \\ & \leq \delta_t^2(F(\theta_t) - F(\theta^*)) \\ & - \delta_t\left(\frac{\mu}{2}||\theta_t - \theta^*||^2 + (F(\theta_t) - F(\theta^*))\right) \\ & \leq \delta_t(\delta_t L - 1)\Big(F(\theta_t) - F(\theta^*)\Big) - \delta_t\frac{\mu}{2}||\theta_t - \theta^*||^2 \\ & \leq \delta_t(\delta_t L - 1)\Big(F(\theta_t) - F(\theta^*)\Big) - \delta_t \mu E(\theta_t) \end{aligned} \tag{6}$$

Since $\leq \delta_t(\delta_t L - 1)\Big(F(\theta_t) - F(\theta^*)\Big)$ is always negative because $\delta_t \leq 1/L$ and $F(\theta^*) \leq F(\theta_t)$, Equation (6) reduces to:

$$E(\theta_{t+1}) - E(\theta_t) \leq -\delta_t \mu E(\theta_t)$$

Since the learning rate δ_t, the constant μ, and E are always positive, this difference is always negative, proving property four of the Lyapunov. Thus, we conclude the GD is suitable for finding the minimizer θ^* in Equation (2). Note that finding λ is a differentiable part of the GD procedure and therefore does not adversely affect convergence. □

Theorem 4.2. *Proof of Convergence using Stochastic Gradient Descent (SGD) for finding a minimizer*

$$\theta^* = \arg\min_{\theta} f(\theta) + \lambda g(\theta') \tag{7}$$

Proof. Here the proof is similar to the GD case until Equation (5). So, we use the same equation and, this time, make replacements with batch sizes. Thus, we obtain:

$$\frac{1}{2}\delta_t^2||\nabla_b F(\theta_t)||^2 + (-\delta_t \nabla_b F(\theta_t)) \cdot (\theta_t - \theta^*)$$

Here ∇_b denotes batch gradients, i.e., stochastic gradients. We leverage two properties of batch gradients. First, the expected value of batch gradients over all batches is the exact gradient. Second, since the batch gradients are bounded (finite sums), we can compute their variance across batches. Thus, we have:

$$\mathbb{E}[\nabla_b F(\theta_t)] = \nabla F(\theta_t)$$
$$Var[||\nabla F(\theta_t)||] = \sigma^2$$

To remove batch gradients ∇_b from the equation, we

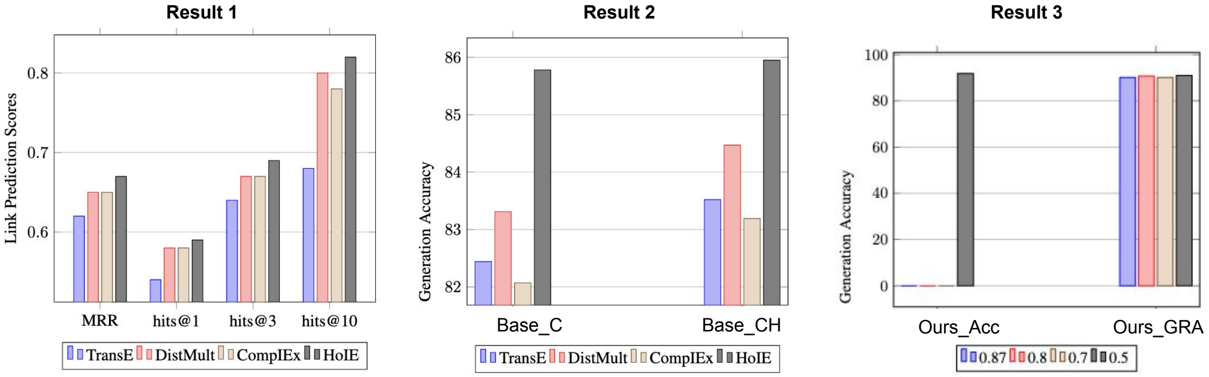


Figure 3: Quantitative Results Graphs

will bound the expected value, $\mathbb{E}[E(\theta_{t+1}) - E(\theta_t)]$, which equates to:

$$\mathbb{E}\left[\frac{1}{2}\delta_t^2||\nabla_b F(\theta_t)||^2 + (-\delta_t \nabla_b F(\theta_t)) \cdot (\theta_t - \theta^*)\right]$$

$$= \frac{1}{2}\delta_t^2(||\nabla F(\theta_t)||^2 + \sigma^2) - (\delta_t \nabla F(\theta_t)) \cdot (\theta_t - \theta^*)$$

We now use strong convexity twice and get:

$$\frac{1}{2}\delta_t^2(||\nabla F(\theta_t)||^2 + \sigma^2) - (\delta_t \nabla F(\theta_t)) \cdot (\theta_t - \theta^*)$$

$$\leq \frac{1}{2}\delta_t^2(M^2 + \sigma^2) - \delta_t \mu ||\theta_t - \theta^*||^2$$

$$= \frac{1}{2}\delta_t^2(M^2 + \sigma^2) - \delta_t 2\mu E(\theta_t)$$

Here we assume that $||\nabla F(\theta_t)||$ is bounded by M, a natural assumption for a discrete algorithm. Plugging in the convergence rate we obtain:

$$\mathbb{E}[E(\theta_{t+1}) - E(\theta_t)]$$

$$\leq \frac{1}{2}\delta_t^2(M^2 + \sigma^2) - \delta_t 2\mu E(\theta_t)$$

$$\leq \frac{1}{2}\frac{1}{\mu^2(t+t_0)^2}(M^2 + \sigma^2) - 2\frac{1}{\mu(t+t_0)}\mu\frac{1}{\frac{2\mu}{M^2+\sigma^2}(t+t_0)}$$

$$\leq -\frac{1}{\frac{2\mu}{M^2+\sigma^2}(t+t_0)^2} \leq 0$$

Thus we have proven that E is a Lyapunov function and can thus conclude that SGD will converge to θ^* when finding the minimizer of Equation (7)*, and therefore confirm the Equation 1 will always have a minimum.* □

5 Experiments and Discussion

Quantitative Experiments

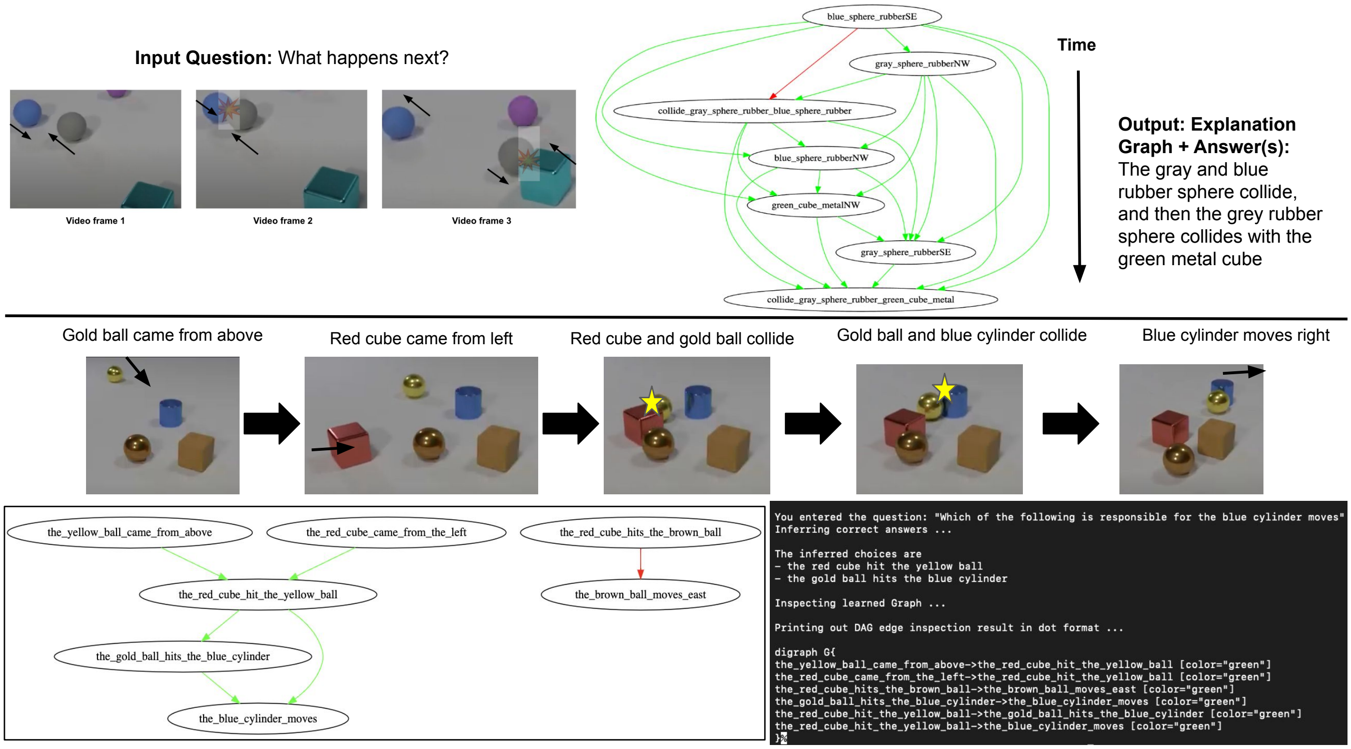
Baseline Method For a competitive baseline, we first construct an autoregressive model similar to the one described in the first summand of Equation 1 $\mathcal{M}(Q, \theta)$. Except we now augment the embedding for Q, by adding graph embeddings of the ground truth CEGs obtained using state-of-the-art (SOTA) graph embedding methods, namely TransE, DistMult, CompIEx, and HoIE (Wang, Qiu, and Wang 2021; Wang et al. 2014; Yang and Liu 2021; Nickel, Rosasco, and Poggio 2016; Trouillon et al. 2016; Yang et al. 2014). We chose this selection as it encompasses different graph geometries (euclidean, hyperbolic, complex) before minimizing the cross-entropy loss. We will denote this augmented embedding for Q as $e_{Q'}$. Thus our baseline model denoted by $M'(e_{Q'}, \beta)$, where β are the trainable parameters (e.g., embedding layers, feedforward layers, and position embedding layers), minimizes the following objective:

$$\texttt{CE}(M'(e_{Q'}, \beta), \texttt{targets})$$

We report the following results, **Result 1.** - The link prediction results for the different graph embedding methods, **Result 2.** - The test set accuracy using the baseline method for the next token prediction of the functional program for the CLEVRER dataset (denoted by Base_C), and the natural language answers for the CLEVRER-Humans dataset (denoted by Base_CH), and **Result 3.** both the test set accuracy averaged across both the CLEVRER and CLEVRER-Humans dataset(denoted by Ours_Acc), and the graph reconstruction accuracy (denoted by Ours_GRA). When measuring graph reconstruction accuracy, we check against the adjacency matrix for the ground truth CEG by thresholding the reconstructed directed graph entries obtained using $\mathcal{G}(Q, \theta')$ (1 if greater than the threshold, and 0 if not). We report the results for four different thresholds of 0.87, 0.8, 0.7, and 0.5. Figure 3 shows the reported results.

Results Summary and Discussion

Results Summary **Result 1.** shows that the link prediction metric of hits@1 of the SOTA graph embedding methods is sub-par (< 0.6) across all models, although substantial improvements are observed when transitioning to hits@3 and hits@10. **Result 2.** shows the accuracy of the baseline method to be quite good $\sim$ 86%. For context, the current leaderboard for the CLEVRER and CLEVRER-Humans dataset shows an accuracy of 95.24%. **Result 3.** shows that our method achieves accuracy scores of 91.85%, and the graph reconstruction accuracy is $\geq 98.3\%$ across all thresholds.


Input Question: What happens next?
Video frame 1
Video frame 2
Video frame 3
blue_sphere_rubberSE
gray_sphere_rubberNW
collide_gray_sphere_rubber_blue_sphere_rubber
blue_sphere_rubberNW
green_cube_metalNW
gray_sphere_rubberSE
collide_gray_sphere_rubber_green_cube_metal
Time
Output: Explanation Graph + Answer(s):
The gray and blue rubber sphere collide, and then the grey rubber sphere collides with the green metal cube
Gold ball came from above
Red cube came from left
Red cube and gold ball collide
Gold ball and blue cylinder collide
Blue cylinder moves right
the_yellow_ball_came_from_above
the_red_cube_came_from_the_left
the_red_cube_hits_the_brown_ball
the_red_cube_hit_the_yellow_ball
the_brown_ball_moves_east
the_gold_ball_hits_the_blue_cylinder
the_blue_cylinder_moves

```
You entered the question: "Which of the following is responsible for the blue cylinder moves"
Inferring correct answers ...

The inferred choices are
- the red cube hit the yellow ball
- the gold ball hits the blue cylinder

Inspecting learned Graph ...

Printing out DAG edge inspection result in dot format ...

digraph G{
the_yellow_ball_came_from_above->the_red_cube_hit_the_yellow_ball [color="green"]
the_red_cube_came_from_the_left->the_red_cube_hit_the_yellow_ball [color="green"]
the_red_cube_hits_the_brown_ball->the_brown_ball_moves_east [color="green"]
the_gold_ball_hits_the_blue_cylinder->the_blue_cylinder_moves [color="green"]
the_red_cube_hit_the_yellow_ball->the_gold_ball_hits_the_blue_cylinder [color="green"]
the_red_cube_hit_the_yellow_ball->the_blue_cylinder_moves [color="green"]
}%
```

Figure 4: Qualitative Results

Discussion Therefore, it is evident that our method enhances quantitative performance across both the intrinsic objective of capturing causal knowledge and the QA objective (predict functional program or answer tokens). While one might be tempted to assume that explicitly minimizing the two losses in Equation 1 would invariably lead to improved outcomes, this is not a given, as the potential for conflict between the two objectives is not always clear. Additionally, even when conflicts are apparent, determining the appropriate values for the Lagrange multipliers to balance objectives is challenging. In response to this challenge, we have proposed utilizing an end-to-end trainable Lagrange multiplier network. Our findings provide empirical support for the synergistic nature of these objectives within the experimental context of this paper. Consequently, our method holds promise as a robust approach to ensure synergistic capture of causal knowledge alongside achieving downstream task objectives if such a synergy exists in other experimental contexts (i.e., tasks other than CLEVRER and CLEVRER-Humans QA).

Qualitative Experiments and Discussion

As mentioned earlier, at inference time, the output from the $\mathcal{G}(Q, \theta')$ part of the trained model, can be visualized based on the chosen threshold t. Green edges indicate those passing the threshold, while red edges represent those that do not. Figure 4 illustrates how this visualization aids human interpretable proof-checking of the model's captured causal ordering of events alongside its QA output. (Top CLEVRER example, and bottom CLEVRER-Humans example).

6 Conclusion, Future Work, and Broader Impacts

We introduce a novel method for capturing and evaluating causal knowledge capture, showcasing its efficacy on benchmark datasets through quantitative and qualitative analyses. Our approach holds promise for causal knowledge-enriched language understanding. Additionally, future work will involve experiments on real-world datasets (e.g., (Yao et al. 2020)), and more complex causal relationship graphs (Blomqvist, Alirezaie, and Santini 2020; Jaimini and Sheth 2022)[1].

Broader Impacts. The gradual rise in adopting AI-systems, particularly in safety-critical industries involving human users (e.g., healthcare and autonomous driving), is notable. In this context, human-AI collaboration is increasingly essential, and graphs can serve as a means to articulate alignment with values encompassing various social dimensions like safety, ethics, social constructs, and legal rules. We take steps towards developing a systematic approach to implement checks and balances, and enhance the interpretability of outcomes by end users of such systems (Purohit, Shalin, and Sheth 2020).

Acknowledgements

This research is partially supported by NSF Award 2335967 ”EAGER: Knowledge-guided neurosymbolic AI

[1] The code is available at: https://github.com/kauroy1994/CEG-QA/tree/main

with guardrails for safe virtual health assistants". (Sheth and Roy 2023; Sheth et al. 2021, 2022; Sheth, Roy, and Gaur 2023). The main ideas and methods described in this manuscript were developed by the first author while interning at Bosch Research and Technology Center in Pittsburgh (USA).

References

Blomqvist, E.; Alirezaie, M.; and Santini, M. 2020. Towards Causal Knowledge Graphs-Position Paper. In *KDH@ ECAI*, 58–62.

Browning, J.; and LeCun, Y. 2023. Language, common sense, and the Winograd schema challenge. *Artificial Intelligence*, 104031.

Haller, A.; Janowicz, K.; Cox, S. J.; Lefrançois, M.; Taylor, K.; Le Phuoc, D.; Lieberman, J.; García-Castro, R.; Atkinson, R.; and Stadler, C. 2019. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10(1): 9–32.

Jaimini, U.; and Sheth, A. 2022. CausalKG: Causal Knowledge Graph Explainability using interventional and counterfactual reasoning. *IEEE Internet Computing*, 26(1): 43–50.

Levesque, H.; Davis, E.; and Morgenstern, L. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Mao, J.; Yang, X.; Zhang, X.; Goodman, N.; and Wu, J. 2022. CLEVRER-Humans: Describing Physical and Causal Events the Human Way. *Advances in Neural Information Processing Systems*, 35: 7755–7768.

Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Purohit, H.; Shalin, V. L.; and Sheth, A. P. 2020. Knowledge graphs to empower humanity-inspired AI systems. *IEEE Internet Computing*, 24(4): 48–54.

Qasemi, E.; Francis, J. M.; and Oltramari, A. 2023. Traffic-domain video question answering with automatic captioning. *arXiv preprint arXiv:2307.09636*.

Sheth, A.; Gaur, M.; Roy, K.; and Faldu, K. 2021. Knowledge-intensive language understanding for explainable ai. *IEEE Internet Computing*, 25(5): 19–24.

Sheth, A.; Gaur, M.; Roy, K.; Venkataraman, R.; and Khandelwal, V. 2022. Process knowledge-infused ai: Toward user-level explainability, interpretability, and safety. *IEEE Internet Computing*, 26(5): 76–84.

Sheth, A.; and Roy, K. 2023. Neurosymbolic Value-Inspired AI (Why, What, and How). *arXiv preprint arXiv:2312.09928*.

Sheth, A.; Roy, K.; and Gaur, M. 2023. Neurosymbolic Artificial Intelligence (Why, What, and How). *IEEE Intelligent Systems*, 38(3): 56–62.

Susskind, Z.; Arden, B.; John, L. K.; Stockton, P.; and John, E. B. 2021. Neuro-symbolic AI: An emerging class of AI workloads and their characterization. *arXiv preprint arXiv:2109.06133*.

Tiddi, I.; Lécué, F.; and Hitzler, P. 2020. Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, 2071–2080. PMLR.

Wang, M.; Qiu, L.; and Wang, X. 2021. A survey on knowledge graph embeddings for link prediction. *Symmetry*, 13(3): 485.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yang, H.; and Liu, J. 2021. Knowledge graph representation learning as groupoid: unifying TransE, RotatE, QuatE, ComplEx. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2311–2320.

Yao, Y.; Wang, X.; Xu, M.; Pu, Z.; Atkins, E.; and Crandall, D. 2020. When, where, and what? A new dataset for anomaly detection in driving videos. *arXiv preprint arXiv:2004.03044*.

Yi, K.; Gan, C.; Li, Y.; Kohli, P.; Wu, J.; Torralba, A.; and Tenenbaum, J. B. 2019. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*.