Scene graph prediction

Trunova Anastasia
Moscow Institute of Physics and Technology
Moscow, Russia
firstauthor@i1.org

Dmitry Yudin
Moscow Institute of Physics and Technology
Moscow, Russia
secondauthor@i2.org

Abstract

Scene graph prediction over time is a crucial problem in dynamic environment understanding, enabling intelligent agents to reason not only about the current scene but also anticipate future changes. In this work, we propose a novel method for temporal scene graph prediction based on an autoregressive neural architecture. Our model leverages a temporal Transformer encoder to extract dynamic contextual features from object representations across multiple time steps. The predicted future graph is generated autoregressively, node by node, with the corresponding edges predicted conditionally at each step. This allows the model to flexibly handle variable numbers of nodes and relationships, unlike traditional fixed-size graph predictors.

To further improve prediction accuracy, we integrate a memory module and incorporate activity cues that guide the evolution of scene objects and their interactions. Additionally, we enrich the object features with motion-based descriptors such as relative speed and direction, allowing the model to distinguish fine-grained relations like approaching or following.

We evaluate our approach on a real-world scene graph dataset, demonstrating that it effectively captures spatio-temporal dependencies, outperforms baseline models, and provides interpretable predictions for future scene configurations.

1. Introduction

Please follow the steps outlined below when submitting your manuscript to the IEEE Computer Society Press. This style guide now has several important modifications (for example, you are no longer warned against the use of sticky tape to attach your artwork to the paper), so all authors should read this new version.

[4] Understanding and predicting the evolution of dynamic scenes is a key challenge in computer vision, with widespread applications in autonomous driving, robotics, surveillance, and human-computer interaction. Static scene understanding has made substantial progress with the use of scene graphs—graph-based representations in which nodes correspond to objects and edges capture semantic relationships. However, the task of forecasting how such graphs change over time remains relatively underexplored.

Scene graph prediction involves forecasting not only future object representations but also their evolving interactions. This problem goes beyond low-level motion estimation or trajectory prediction by incorporating higher-order semantic reasoning, such as detecting when a person will approach a car or when an object is likely to fall. To effectively model these dynamic interactions, the prediction system must handle variable graph sizes, maintain temporal coherence, and reason about relational structure.

Existing approaches to scene graph prediction often exhibit several limitations. Many models are constrained to a fixed number of nodes and edges, making them unsuitable for real-world applications where objects may appear or disappear. Some methods restrict themselves to reasoning over a single object or target node rather than modeling the complete scene. Others rely on temporally rigid architectures or predefined time steps, which limits their applicability in continuous or event-driven scenarios. Additionally, many current models overlook the influence of human activity, object motion dynamics, and contextual cues that drive relational changes between entities.

To address these challenges, this work formulates the task of temporal scene graph prediction as the mapping:

$$\{G_{t-k}, \ldots, G_t\} \mapsto G_{t+\delta}, \tag{1}$$

where $G_t = (V_t, E_t)$ represents the scene graph at time t, with V_t denoting the set of objects and E_t the set of edges. The goal is to approximate the future state $G_{t+\delta}$ using the sequence of past graphs.

The proposed model adopts an autoregressive architecture that generates the future scene graph node-by-node, predicting each new object and its edges to previously generated nodes. A Transformer-based encoder captures temporal dependencies across the input graph sequence. To improve semantic forecasting, the model design incorporates

additional modalities: (i) object-level activity embeddings to capture agent intentions, (ii) memory modules to store long-term scene context, and (iii) relative motion features such as velocity or direction to assist in detecting relationships like "approaching" or "following".

This framework enables interpretable, flexible, and semantically rich predictions of future scenes. The modular nature of the design allows further enhancement with additional relational cues or domain-specific priors.

2. Related Work

2.1. Scene Graph Generation

A considerable body of work has focused on generating scene graphs from static images or videos. These methods aim to detect objects and infer pairwise relationships, providing structured semantic representations of scenes.

[1] integrates spatial and temporal context using attention mechanisms to improve visual relation detection across video frames. It effectively captures short-term temporal dependencies but does not perform forecasting beyond the current frame.

[7] combines visual and motion-based cues for spatio-temporal scene graph generation, leveraging optical flow and object trajectories. While TEMPURA captures motion-induced relations, it is limited to frame-wise inference and does not model the evolution of scene graphs into the future.

[2] extends transformer-based object detection to dense scene graph generation by jointly reasoning over object queries and their relationships. Its strength lies in its end-to-end architecture and global relational modeling, yet it is constrained to single-frame prediction.

Although these models successfully extract semantic structure from visual input, they do not address the task of scene graph prediction, where the goal is to anticipate changes in object configurations and relationships over time.

2.2. Scene Graph Prediction

A number of recent methods address the problem of long-term scene graph prediction, focusing on different aspects such as temporal dynamics, partial observability, semantic variability, and user activity modeling.

[6] focuses on the semantic variability of objects by modeling the probabilities of changes in their attributes and states. The method demonstrates realistic behavior when predicting the movement of frequently used objects, but it does not utilize historical context or human activity information, which limits its accuracy in complex scenarios.

[4] proposes a mechanism for predicting changes in partially observed graphs. It incorporates both semantic and spatial features of nodes and edges using PCA and sparse graph processing, enabling efficient handling of large-scale sparse graphs. However, it lacks memory components or user-aware modeling.

[5] introduces a spatio-temporal graph neural network to predict user needs based on patterns of everyday activity. While it does not directly predict changes in the structure of the scene graph, it offers valuable insights for predictive planning and could be extended with scene-level information.

[3] is one of the few methods that incorporates an agent activity vector and a memory state updated at each time step. This design allows the model to effectively predict the next scene graph G_{t+1} based on the current state h_t. However, the graph structure is not predicted explicitly, but rather through a probabilistic state transition model.

[8] is the most closely related method to our work. It integrates temporal context, user activity, and Markov chains to predict future object interactions and displacements. The model constructs a tree of possible action sequences using a language model (LLM) and performs probabilistic forecasting of future scene configurations. However, its high computational complexity limits real-time applicability.

Generative graph models, such as *GraphRNN* and *GRAN*, while not originally designed for dynamic scene understanding, offer powerful autoregressive generation capabilities. These models are particularly effective at step-wise generation of both nodes and edges, which makes them potentially well-suited for modeling scene graphs with a variable number of objects and relations.

2.3. Motivation

In summary, scene graph generation methods provide robust mechanisms for extracting semantic structure, but do not address prediction. On the other hand, forecasting models lack the flexibility to handle dynamic graphs or the ability to condition predictions on human activity and object motion.

Our work builds upon these ideas by proposing an autoregressive prediction framework that incrementally generates scene graphs. It incorporates temporal context using transformer-based encoders, handles variable graph sizes, and enables future extensions such as motion-aware features and activity-driven relational reasoning.

3. Method

In this section, we present our method for autoregressive scene graph prediction that incorporates temporal context, dynamic memory, user activity vectors, and object-level motion cues. The core idea is to forecast the evolution of scene structure by generating the next scene graph G_{t+1} based on a sequence of previous graphs $\{G_{t-k}, \ldots, G_t\}$.

3.1. Problem Formulation

Let $G_t = (V_t, E_t)$ denote a scene graph at time t, where V_t is the set of objects (nodes) and E_t is the set of relations

(edges). Given a sequence of k past graphs, our goal is to predict the future graph G_{t+1}, including the number of nodes, their features, edge existence, and edge attributes.

3.2. Model Overview

Our model consists of the following components:

- **Temporal Encoder:** A Transformer-based encoder processes the temporal sequence of graphs and produces context-aware node embeddings.
- **Autoregressive Node Decoder:** Nodes of the future graph are predicted one-by-one using a GRU-based decoder, conditioned on the temporal context and previously generated nodes.
- **Edge Predictor:** For each newly generated node, we predict its connections to previously generated nodes using a learned edge scoring function and produce edge attributes.
- **Stop Token Predictor:** A special module determines when to terminate node generation based on a learned threshold.
- **Activity and Motion Features:** The model is optionally conditioned on user activity vectors and motion cues (e.g., relative velocities, movement direction) to better capture spatio-temporal relations like *"approaching"* or *"following"*.
- **Memory Module (future work):** A memory cell h_t is maintained and updated at each time step to store global information about the scene dynamics.

3.3. Autoregressive Node Generation

Rather than fixing the number of predicted nodes, we generate them sequentially in an autoregressive manner. At each decoding step i, the model outputs a new node feature $\hat{v}_i$ and its edges to the existing partial graph $\hat{G}_i = (\hat{V}_{<i}, \hat{E}_{<i})$. This allows the model to naturally handle dynamic changes in the number of objects.

Formally, the decoder updates its hidden state as:

$$h_i = \text{GRUCell}(z_i, h_{i-1}), \tag{2}$$

where z_i is the input at step i (either ground truth node embedding for teacher forcing, or the last predicted node). The new node is produced as:

$$\hat{v}_i = \text{MLP}(h_i), \tag{3}$$

and its connections to previous nodes $\{v_j\}_{j<i}$ are predicted via:

$$p_{ij} = \sigma(f_{\text{edge}}([h_i, h_j])), \tag{4}$$

where f_{edge} is a learnable scoring function.

3.4. Loss Function

We optimize the model using a composite loss:

$$\mathcal{L} = \lambda_v \mathcal{L}_{\text{nodes}} + \lambda_e \mathcal{L}_{\text{edges}} + \lambda_a \mathcal{L}_{\text{edge_attr}} + \lambda_s \mathcal{L}_{\text{stop}}, \tag{5}$$

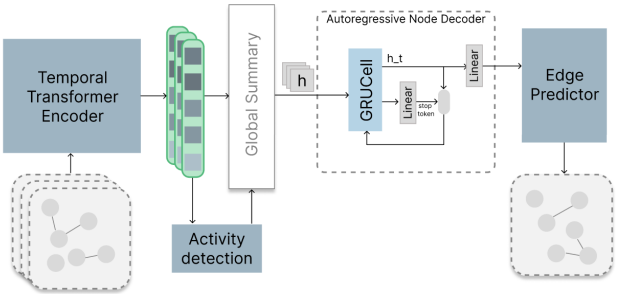


Figure 1. Architecture of the proposed method. The model encodes a sequence of scene graphs using a Temporal Transformer Encoder. A global context vector, optionally enriched by an activity detection module, is passed to an autoregressive node decoder based on a GRUCell. At each step, the decoder generates a new node embedding, predicts a stop signal, and estimates edge connections to previously generated nodes. The final output is a predicted future scene graph.

where:

- $\mathcal{L}_{\text{nodes}}$ is the MSE loss on predicted node features;
- $\mathcal{L}_{\text{edges}}$ is the binary cross-entropy on edge existence;
- $\mathcal{L}_{\text{edge_attr}}$ is the MSE on edge attributes;
- $\mathcal{L}_{\text{stop}}$ is a binary cross-entropy loss for the stop token.

3.5. Motivating Human-Centered Dynamics

Unlike existing methods that treat scene graphs as static structures, we explicitly model human-centric context through an *activity vector* and integrate *object motion* representations (e.g., velocity, direction). These inputs are concatenated to the node features before temporal encoding, enhancing the model's ability to predict future interactions and actions.

4. Experiments

4.1. Experimental Setup

A series of experiments were conducted to evaluate the effectiveness of different model architectures for the task of scene graph prediction. All models were trained and validated on a subset of the Action Genome dataset, where dynamic scene graphs represent temporal interactions between objects and agents.

4.2. Baseline Architecture

Initial baseline model consisted of a temporal transformer encoder that processed sequences of scene graphs, followed by a multilayer perceptron (MLP) decoder.

While this architecture performed reasonably well in cases with a fixed number of objects, it struggled with scenarios where the number of objects changed dynamically over time (e.g., objects appearing or disappearing from the scene). This limitation forced us to specify a maximum

number of output nodes in advance, which proved to be a significant constraint in real-world scenarios.

4.3. Autoregressive Node Decoder

To overcome the limitations of the fixed-size decoder, we introduced an autoregressive decoder that generates one object at a time. This approach incorporates a GRUCell and a learned stop token mechanism to determine when the decoder should stop generating new nodes. At each decoding step, the model also predicts the edges between the newly generated node and all previously generated nodes, along with corresponding edge attributes.

This autoregressive formulation significantly improved flexibility and performance in cases with varying numbers of objects. It allowed the model to generate variable-sized scene graphs and naturally capture the evolving structure of the scene.

4.4. Relation Decoder

While our model currently predicts edge presence and attributes using concatenated embeddings and MLPs, we observed that this approach could be enhanced by incorporating a pretrained *relation decoder* module. Such a module would take object features and scene context as input and predict the most plausible relations, possibly conditioned on memory or temporal cues from past interactions.

4.5. Additional Enhancements

We explored the idea of incorporating human activity recognition into the model pipeline, under the hypothesis that human intention and actions drive many dynamic changes in scene graphs. While this feature has not yet been implemented, we plan to extend the architecture with an activity embedding branch based on action classifiers or pose analysis.

Furthermore, the node features in our current implementation are obtained using CLIP-based embeddings, allowing for semantic generalization across object categories and paving the way toward an open-vocabulary scene graph forecasting framework.

4.6. Quantitative Trends

We trained the autoregressive model using a composite loss function that combines node prediction (MSE), edge presence (BCE), and edge attribute reconstruction (MSE). Figure 2 shows the training curves of each component loss.

4.7. Training Dynamics

We observe that the training curves (Figure 2) show stable and monotonic convergence across all loss components. This consistent behavior suggests that the model is able to effectively learn the dynamics present in the dataset.

Given the relatively simple structure and limited variability of the training scenes, the model quickly adapts to the dominant patterns. The total loss decreases steadily, and both the edge presence and edge attribute losses converge within a few epochs. Node prediction loss shows slightly slower convergence, which aligns with the higher variance in node-level features.

References

[1] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16372–16382, 2021. 2

[2] Shengyu Feng, Hesham Mostafa, Marcel Nassar, Somdeb Majumdar, and Subarna Tripathi. Exploiting long-term dependencies for generating dynamic scene graphs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5130–5139, 2023. 2

[3] Nicolas Gorlo, Lukas Schmid, and Luca Carlone. Long-term human trajectory prediction using 3d dynamic scene graphs. *IEEE Robotics and Automation Letters*, 9(12):10978–10985, 2024. 2

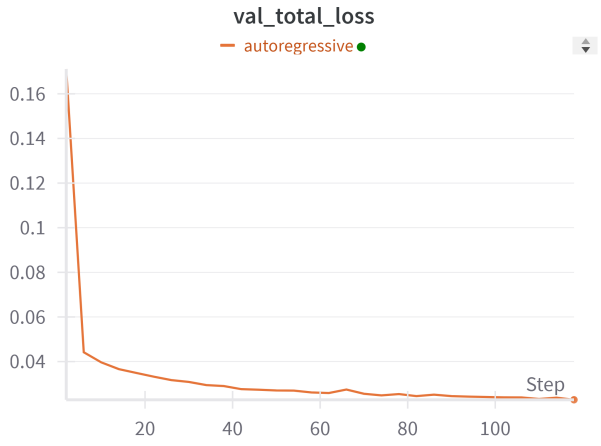
[4] Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun Wu, Silvio Savarese, and Roberto Martın-Martın. Modeling dynamic environments with scene graph memory. In *International Conference on Machine Learning*, pages 17976–17993. PMLR, 2023. 1, 2

[5] Maximilian Listl. Spatio-temporal graph neural networks for multiple object tracking. 2022. 2

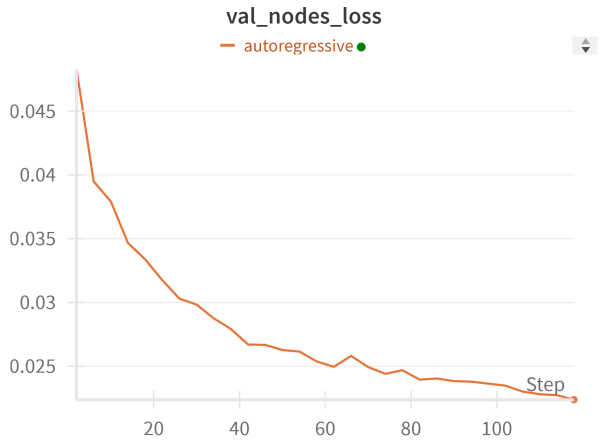
[6] Samuel Looper, Javier Rodriguez-Puigvert, Roland Siegwart, Cesar Cadena, and Lukas Schmid. 3d vsg: Long-term semantic scene change prediction through 3d variable scene graphs. IEEE International Conference on Robotics and Automation (ICRA), 2023. 2

[7] Sayak Nag, Kyle Min, Subarna Tripathi, and Amit K Roy-Chowdhury. Unbiased scene graph generation in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22803–22813, 2023. 2

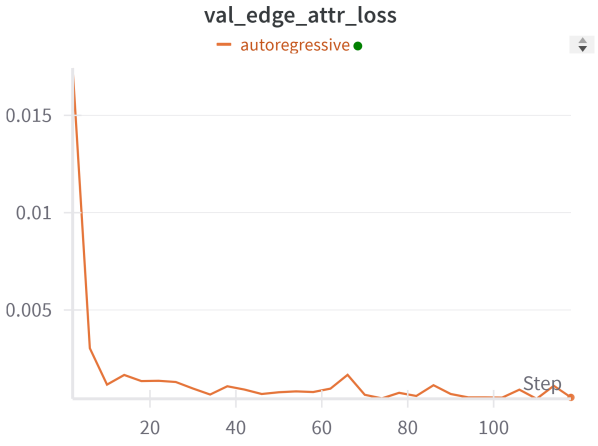
[8] Maithili Patel, Aswin Prakash, and Sonia Chernova. Predicting routine object usage for proactive robot assistance. *arXiv preprint arXiv:2309.06252*, 2023. 2



(a) Total loss over training.



(b) Node prediction loss (MSE).



(c) Edge attribute loss (MSE).

Figure 2. Loss components over training steps.