Towards Scene Graph Anticipation

Anonymous ECCV 2024 Submission

Paper ID #12130

Abstract. Spatio-temporal scene graphs represent interactions in a video by decomposing scenes into individual objects and their pair-wise temporal relationships. Long-term anticipation of the fine-grained pair-wise relationships between objects is a challenging problem. To this end, we introduce the task of Scene Graph Anticipation (SGA). We adapt state-of-the-art scene graph generation methods as baselines to anticipate future pair-wise relationships between objects and propose a novel approach SceneSayer. In SceneSayer, we leverage object-centric representations of relationships to reason about the observed video frames and model the evolution of relationships between objects. We take a continuous time perspective and model the latent dynamics of the evolution of object interactions using concepts of NeuralODE and NeuralSDE, respectively. We infer representations of future relationships by solving an Ordinary Differential Equation and a Stochastic Differential Equation, respectively. Extensive experimentation on the Action Genome dataset validates the efficacy of the proposed methods.

Keywords: Scene Graphs · Scene Understanding · Differential Equations

1 Introduction

We focus on spatio-temporal scene graphs [15], which is a widely used framework for representing the evolving spatial and temporal relationships among objects. These graphs contain information about the objects present in a video, including their categories, positions, sizes, and spatial dependencies. Simultaneously, they illustrate how these relationships evolve over time, revealing objects' movement, interactions, and configuration changes across consecutive frames in a video sequence. They facilitate our understanding of dynamic scenes and serve as a valuable tool for addressing downstream tasks in applications such as action recognition and video analysis, where the temporal dynamics of object interactions play a crucial role.

This paper introduces a novel task known as *Scene Graph Anticipation (SGA)*, which, given a video stream, aims to forecast future interactions between objects, as shown in Figure 1. The Scene Graph Anticipation (SGA) task holds significance across diverse domains due to its potential applications and relevance to several downstream tasks. For instance, it contributes to *enhanced video understanding* by predicting spatiotemporal relationships within video scenes, facilitating improved video analysis and interpreting complex object interactions over time. SGA plays a crucial role in *activity recognition*, enabling systems to predict future object interactions for more accurate classification and advanced surveillance. Anticipation aids in *anomaly detection* by identifying deviations from expected object relationships, thereby enhancing the detection of abnormal events in video sequences. *Intelligent surveillance systems* can also

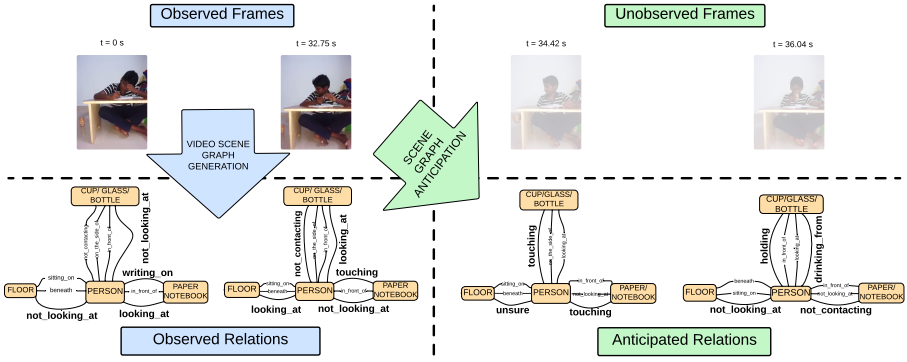

Fig. 1: Task Description. We contrast the task of Video Scene Graph Generation (VidSGG) on the left with the proposed task of Scene Graph Anticipation (SGA) on the right. VidSGG entails the identification of relationships from the observed data, such as (*Person*, **looking_at**, *Floor*) and (*Person*, **not_contacting**, *Cup*). SGA aims to anticipate the evolution of these relationships to (*Person*, **touching**, *Cup*), and eventually, (*Person*, **drinking_from**, *Cup*).[1]

benefit from SGA, allowing systems to predict and respond to security threats by understanding evolving object relationships in monitored environments. Finally, SGA is essential for predicting object movements and interactions for applications in *robotics and autonomous systems*, contributing to safer and more efficient navigation and decision-making processes.

To tackle the challenges of SGA, we introduce two novel approaches that extend a state-of-the-art scene graph generation method [7]. These approaches utilize object-centric representations of relationships, allowing us to analyze observed video frames and effectively model the dynamic evolution of interactions between objects. This detailed understanding of temporal dynamics is the basis for our proposed methods.

In a departure from traditional sequential modelling, our two approaches adopt a continuous-time perspective. Drawing inspiration from Neural Ordinary Differential Equations (NeuralODE) [3] and Neural Stochastic Differential Equations (NeuralSDE) [17], respectively, we develop methods that capture the latent dynamics governing the evolution of object interactions. By formulating the anticipation problem as solving Ordinary Differential Equations (ODE) and Stochastic Differential Equations (SDE) and thus using a continuous representation of the anticipated relationships, we hope to significantly expand the fidelity of our predictions. We rigorously validated our proposed methods and a strong generation-based baseline on the Action Genome dataset [15], a benchmark for spatio-temporal scene understanding. Our experimental results demonstrate the superior performance of our approaches in accurately anticipating fine-grained pair-wise relationships between objects.

[1] The relationships presented are taken from the ground truth annotations of the Action Genome

2 Related Work

2.1 Structured Representation

Spatial Graphs. Learning to represent visual content in static data such as 2D/3D images as a spatial graph where the objects act as nodes and edges to describe the visual relation between the objects is called *Image Scene Graph Generation* (ImgSGG). There has been extensive research in 2D ImgSGG direction following the seminal work of Visual Genome [20]. Following this, the task has been extended to 3D by the introduction of [19]. Recently, a surge of ImgSGG methods explored the role of foundation models in open-vocabulary ImgSGG [5], weakly supervised ImgSGG [18], panoptic ImSGG [42] and zero-shot ImSGG [41].

Spatio-Temporal Graphs. Dynamic visual content, such as videos, provides a more natural set of contextual features describing dynamic interaction between objects. Encoding such content into a structured spatiotemporal graphical representation for frames where nodes describe objects and edges describe the temporal relations is called Video Scene Graph Generation (VidSGG). Early approaches on VidSGG extended previously proposed ImSGG-based methods to the temporal domain. We refer to [43] for a comprehensive survey on earlier work. Recent work explored learning better representations using architectures like Transformers [7, 33] and unbiased representations [16, 27] owing to the long-tailed nature of the benchmark VidSGG datasets Action Genome [15], VidVRD [32].

Applications. Structured representation of visual content has been used in various downstream tasks, including task planning [1], image manipulation [8], visual question answering [6] using scene graphs, video synthesis using action graphs [2] and using scene graph as a knowledge base [21]. To the best of our knowledge, we are the first to formally introduce the task of Scene Graph Anticipation (SGA), propose baseline methodologies, and establish evaluation metrics. [2]

2.2 Dynamics Models

Video Prediction. Early video prediction methods treat dynamics modelling as a sequential prediction problem in pixel space using image-level features [22]. Later approaches proposed include using external knowledge as priors [10, 37], better architectural design to model contextual information [30, 40], and focusing on object-centric representations [39]. Recently, there has been a surge in methods proposed that use diffusion models to estimate the distribution of a short future video clip [14, 36]. Contrary to the conventional dense pixel-based generation often seen in video prediction techniques, SGA emphasizes the importance of learning representations that facilitate the prediction of long-term interactions and their nuanced relationships. Our experimental

[2] SGA is related to [38], wherein they use relation forecasting as an intermediate step for action prediction. However, we aim for precise anticipation of future relationships between interacting objects. SGA is also closely related to [26], which constructed a dataset from Action Genome by sampling frames and truncating the videos. However, we differentiate our method by utilizing the complete dataset to train our models. Furthermore, we acknowledge that the approach in [26] resembles the baseline approach proposed in this paper.

results demonstrate that this approach can effectively estimate these relationship representations for periods extending beyond 30 seconds into the future.

Neural Differential Equations. After the introduction of NeuralODE [3], several methods were proposed to explore latent dynamics models using the framework of NeuralODE. These include methods on trajectory prediction [24], traffic forecasting [25], video generation [28], and multi-agent dynamical systems [12, 13, 29].

3 Background

3.1 Ordinary Differential Equations (ODEs)

The initial value problem (IVP) for an ODE is given by:

$$\frac{\mathbf{dz}(t)}{\mathbf{d}t} = \mathbf{f}(\mathbf{z}(t), t), \quad \mathbf{z}(t_0) = z_0 \tag{1}$$

Here, $\mathbf{f} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ represents a time-varying smooth vector field, and $\mathbf{z}(t)$ is the solution to the IVP. Chen et al. [3] introduced the framework of NeuralODEs wherein they relaxed the time-variance of the vector field $\mathbf{f}$ and parameterized it through neural networks, thus enabling efficient learning of dynamical systems from data. This approach has paved the way for modelling the dynamics of latent states through LatentODEs. Analytical solutions for complex ODEs are typically infeasible; hence, we resort to numerical methods that discretize the time domain into finite intervals, approximating the solution at each step. Thus, they trade off precision with computation time. On the faster, less accurate side, we have single-step methods, such as Euler, while on the slower, more accurate side, we have multistep methods, such as Adams-Bashforth.

3.2 Stochastic Differential Equations (SDEs)

An initial value problem for an SDE is formulated as:

$$\mathbf{dz}(t) = \mu(\mathbf{z}(t), t)\mathbf{d}t + \sigma(\mathbf{z}(t), t)\mathbf{dW}(t), \quad \mathbf{z}(t_0) = z_0 \tag{2}$$

Here, $\mu(\mathbf{z}(t), t) : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, $\sigma(\mathbf{z}(t), t) : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^{(d \times m)}$ represent drift and diffusion terms, respectively and W_t denotes m-dimensional Wiener process. NeuralSDEs employ neural networks to parameterize drift and diffusion terms, enabling the learning of stochastic processes from data [17, 23]. Solving SDEs analytically is often challenging. Numerical solutions depend on the choice of interpretation of the SDE, which is often connected to the conceptual model underlying the equation. There are many interpretations of an SDE, of which Ito and Stratonovich are popular amongst them. Stratonovich's interpretation of an SDE is commonly used to model physical systems subjected to noise. SDE solvers use discretization methods to approximate the continuous dynamics of the system over small time intervals. The most common numerical methods for solving SDEs include Euler-Maruyama, Milstein, and Runge-Kutta schemes, each varying in accuracy and computational complexity.

4 Notation & Problem Description

Notation. Given an input video segment V_1^T, we represent it using a set of frames $V_1^T = \{I^t\}_{t=1}^T$ defined on discretized time steps $t = \{1, 2, \cdots, T\}$, where the total number of observed frames T varies across video segments. A scene graph is a symbolic representation of the objects present in the frame and their pair-wise relationships. In each frame I^t, we represent the set of objects observed in it using $O^t = \{o_k^t\}_{k=1}^{N(t)}$, where $N(t)$ the total number of objects observed in a frame and varies across frames. Let $\mathcal{C}$ be the set comprising all object categories, then each instance of an object o_k^t is defined by its bounding box information $\mathbf{b}_k^t$ and object category $\mathbf{c}_k^t$, where $\mathbf{b}_k^t \in [0,1]^4$ and $\mathbf{c}_k^t \in \mathcal{C}$. Let $\mathcal{P}$ be the set comprising all predicate classes that spatio-temporally describe pair-wise relationships between two objects; then each pair of objects $\left(o_i^t, o_j^t\right)$ may exhibit multiple relationships, defined through predicates $\{p_{ijk}^t\}_k$ where $p_{ijk}^t \in \mathcal{P}$.

We define a relationship instance r_{ijk}^t as a triplet $\left(o_i^t, p_{ijk}^t, o_j^t\right)$ that combines two distinct objects $\left(o_i^t, o_j^t\right)$ and a predicate p_{ijk}^t. Thus, the scene graph $\mathcal{G}^t$ is the set of all relationship triplets $\mathcal{G}^t = \{r_{ijk}^t\}_{ijk}$. Additionally, for each observed object o_i^t and a pair of objects $\left(o_i^t, o_j^t\right)$, we use $\hat{\mathbf{c}}_i^t$ and $\hat{\mathbf{p}}_{ij}^t$ to represent the distributions over object categories and predicate classes respectively. Here, $\hat{\mathbf{c}}_i^t \in [0,1]^{|\mathcal{C}|}$ and $\hat{\mathbf{p}}_{ij}^t \in [0,1]^{|\mathcal{P}|}$.

Problem Description. We formally define and contrast the tasks of Video Scene Graph Generation (VidSGG) and Scene Graph Anticipation (SGA) as follows:

1. The goal of **VidSGG** is to build scene graphs $\{\mathcal{G}^t\}_{t=1}^T$ for the observed video segment $V_1^T = \{I^t\}_{t=1}^T$. It entails the detection of objects $\{o_k^t\}_{k=0}^{N(t)}$ in each frame and the prediction of all pair-wise relationships $\{r_{ijk}^t\}_{ijk}$ between detected objects.
2. The goal of **SGA** is to build scene graphs $\{\mathcal{G}^t\}_{t=T+1}^{T+H}$ for future frames $V_{T+1}^{T+H} = \{I^t\}_{t=T+1}^{T+H}$ of the video based on the observed segment of the video V_1^T, here H represents the anticipation horizon. Thus, it entails anticipating objects and their pair-wise relationships in future scenes. We note that anticipating the emergence of new objects in future frames is a significantly harder problem. So, we presuppose the continuity of observed objects in future frames (i,e $\{o_i^t\}_{i=1}^{N(t)} = \{o_i^T\}_{i=1}^{N(T)}, \forall t > T$) and predict the evolution of relationships $\left\{\{r_{ijk}^t\}_{t=T+1}^{T+H}\right\}_{ijk}$ in future scenes.

Graph Building Strategies. To build scene graphs for future frames, we employ two strategies that are widely established within VidSGG literature: (a) **With Constraint Strategy**: This approach enforces a unique interaction constraint between any pair of objects in a scene. Specifically, for a pair of objects $\left(o_i^t, o_j^t\right)$, there exists a single predicate p_{ij}^t that describes the relationship between them, and we incorporate relationship triplets $\{r_{ij}^t\}_{ij}$ into the scene graph $\mathcal{G}^t$. (b) **No Constraint Strategy**: In contrast, this method embraces a more complex and detailed graph structure by permitting multiple edges between any pair of objects $\left(o_i^t, o_j^t\right)$. Specifically, we incorporate all predicted relationship triplets $\{r_{ijk}^t\}_{ijk}$ into the scene graph $\mathcal{G}^t$.

5 Technical Approach

We propose SceneSayer for Scene Graph Anticipation. As illustrated in Fig 2, SceneSayer incorporates an object representation processing unit (ORPU) that captures representations of the objects detected in a frame, a spatial context processing unit (SCPU)[3] that builds spatial context-aware relationship representations and a latent dynamics processing unit (LDPU), which is designed to understand spatio-temporal dynamics and predict the evolution of relationships between interacting objects. Distinctively, SceneSayer employs a *continuous-time framework* to model the latent dynamics of the evolution of relationships. In the following sections, we provide a detailed description of these units and the methodologies employed for training and testing SceneSayer.

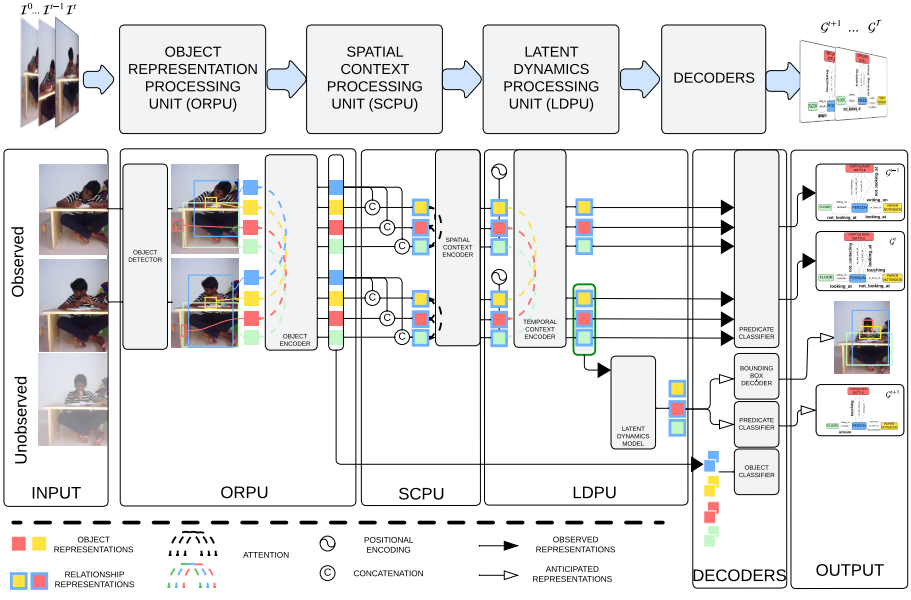

Fig. 2: Overview of SceneSayer. The forward pass of SceneSayer begins with ORPU, where initial object proposals are generated for each frame. These proposals are then fed to a temporal encoder to ensure that the object representations remain consistent over time. Next, in SCPU, we construct initial relationship representations by concatenating the representations of interacting objects. These representations are further refined using a spatial encoder, embedding the scene's spatial context into these relationship representations. Then, the representations undergo further enhancement in LDPU, where another temporal encoder fine-tunes them, imbuing the data with comprehensive spatio-temporal scene knowledge. These refined relationship representations from the final observed frame are then input to a Latent Dynamics Model (LDM), where a generative model, either a NeuralODE or a NeuralSDE, generates relationship representations of interacting objects in future frames by solving the corresponding differential equations. Finally, these future relationship representations are decoded to construct anticipated scene graphs.

[3] Both ORPU and SCPU can be adapted from any existing VidSGG models.

5.1 Object Representation Processing Unit

To achieve temporally consistent object representations across observed frames, our approach first extracts visual features using a pre-trained object detector and then further refines these features by processing them through an encoder. Specifically, We utilize a pre-trained Faster R-CNN [31] for extracting visual features ($\{\mathbf{v}_i^t\}_{i=1}^{N(t)}$), bounding boxes ($\{\mathbf{b}_i^t\}_{i=1}^{N(t)}$), and object category distributions ($\{\hat{\mathbf{c}}_i^t\}_{i=1}^{N(t)}$), for object proposals $\{o_i^t\}_{i=1}^{N(t)}$ in the observed frames. We then compute the matrix $\mathbf{V}_i$ by stacking visual features $\{\mathbf{v}_i^t\}_{t=1}^{T}$. We employ a transformer encoder [35] to aggregate temporal information across visual features from all frames. The input for our encoder consists of sequences $\mathbf{V}_i$, which serve simultaneously as queries $\mathbb{Q}$, keys $\mathbb{K}$ and values $\mathbb{V}$. Formally, the transformation process at the n-th encoder layer is given by

$$\mathbf{V}_i^{(n)} = \text{ObjectEncoder}\left(\mathbb{Q} = \mathbb{K} = \mathbb{V} = \mathbf{V}_i^{(n-1)}\right) \tag{3}$$

5.2 Spatial Context Processing Unit

To learn spatial context-aware relationship representations. We first construct relationship representations between interacting objects in a scene and further process them by passing through an encoder. Specifically, following [7], we construct relationship representation $\mathbf{z}_{ij}^t$ by concatenating visual and semantic features of the objects as follows:

$$\mathbf{z}_{ij}^t = \text{Concat}(\mathbf{W}_1\mathbf{v}_i^t, \mathbf{W}_2\mathbf{v}_j^t, \mathbf{W}_3\mathbf{U}_{ij}^t, \mathbf{S}_i^t, \mathbf{S}_j^t) \tag{4}$$

Here, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ represent the learnable weights of linear layers. $\mathbf{U}_{ij}^t$ represent processed feature maps of the union box computed by RoIAlign [31]. $\mathbf{S}_i^t$ and $\mathbf{S}_j^t$ correspond to the semantic embedding vectors of the object categories. Subsequently, we employ a transformer encoder to integrate spatial contextual information. Here, for each observed frame $\mathcal{I}^t$, we construct $\mathbf{Z}^t$ by stacking all relationship features $\{\mathbf{z}_{ij}^t\}_{ij}$ corresponding to objects observed in a frame. We then feed it as input for a transformer encoder which operates on queries $\mathbb{Q}$, keys $\mathbb{K}$, and values $\mathbb{V}$ that are derived from the same source, the preceding layer's output or the initial relationship features. Formally, the transformation process at the n-th encoder layer is given by:

$$\mathbf{Z}_{(n)}^t = \text{SpatialEncoder}\left(\mathbb{Q} = \mathbb{K} = \mathbb{V} = \mathbf{Z}_{(n-1)}^t\right) \tag{5}$$

5.3 Latent Dynamics Processing Unit

To understand spatio-temporal dynamics of the evolution of relationships between interacting objects, departing from traditional approaches that architectural variants of transformers [11, 34], we take a continuous time approach and learn a governing differential equation in latent space of relationship representations. Our LDPU contains two components: (1) Temporal Context Encoder and (2) Latent Dynamics Model.

Temporal Context Encoder. To learn spatio-temporal context-aware relationship representations, we further process the output of SCPU by passing it through a transformer encoder that fine-tunes the representation. Specifically, we compute the matrix $\mathbf{Z}_{ij}$ by vertically stacking the relationship representations $\{\mathbf{z}_{ij}^t\}_{t=1}^T$. As described in Eq. 6, we feed $\mathbf{Z}_{ij}$ through an encoder that aggregates temporal information across all observed relationship representations. Here, the input $\mathbf{Z}_{ij}$, serves simultaneously as queries $\mathbb{Q}$, keys $\mathbb{K}$, and values $\mathbb{V}$.

$$\mathbf{Z}_{ij}^{(n)} = \text{TemporalEncoder}\left(\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{Z}_{ij}^{(n)}\right) \tag{6}$$

Latent Dynamics Models. We begin by abstracting the complexities introduced by external factors that add uncertainty to the evolution of relationships between interacting objects and construct a model based on the premise that the core dynamics driving the changes in these relationships are governed by a non-linear deterministic process. Specifically, we leverage the expressive nature of NeuralODEs [3] in learning time-invariant vector fields from data that approximate the underlying non-linear deterministic process and propose **SceneSayerODE**. We cast the problem of anticipating relationship representations of interacting objects as an instance of the ODE-IVP, where the initial condition is given by the relationship representation of the last observed interaction of the object pair $\left(o_i^t, o_j^t\right)$. The evolution of this relationship over time is then mathematically described as:

$$\mathbf{z}_{ij}^{T+H} = \mathbf{z}_{ij}^{T} + \int_{T}^{T+H} \mathbf{f}_\theta(\mathbf{z}_{ij}^t)\, \mathbf{d}t \tag{7}$$

In video data that is captured from a single viewpoint, we lose information due to issues such as blurry imagery and occlusions, leading to uncertain interpretations of the scenes. Therefore, it is crucial to integrate uncertainty into the modelling frameworks to represent these stochastic dynamics accurately. We assume the presence of a non-linear stochastic differential equation that governs this evolution and propose **SceneSayerSDE** which uses NeuralSDEs [17] to learn it from data. Specifically, we formulate the problem of anticipating future interactions as an SDE-IVP. Here, the initial conditions are set using the representations of the last observed representation object pair, thus facilitating a data-driven learning process that accounts for the inherent uncertainties. The evolution of this relationship over time is then described as:

$$\mathbf{z}_{ij}^{T+H} = \mathbf{z}_{ij}^{T} + \int_{T}^{T+H} \mu_\theta(\mathbf{z}_{ij}^t)\mathbf{d}t + \int_{T}^{T+H} \sigma_\phi(\mathbf{z}_{ij}^t)\mathbf{d}\mathbf{W}(t) \tag{8}$$

5.4 Decoders

Predicate Classification Head. We employ two predicate classification heads in our approach; we use one for classifying relationship representations between objects of the observed scenes and the other for classifying anticipated relationship representations of the future scenes. Our predicate classification head is implemented using a two layer neural network. **Bounding Box Regression Head.** Our regression head takes the

anticipated relationship representations as input and outputs bounding box information corresponding to the interacting objects. It is implemented using two layers of neural network. **Object Classification Head.** We integrate processed object representations obtained from the object encoder with object category distributions from the object detection process and pass them through a two-layer neural network to output refined object category distributions.

5.5 Loss Function.

Object Classification Loss. The classification effectiveness is evaluated using the standard cross-entropy loss, denoted as $\mathcal{L}^t_{(o)}$

$$\mathcal{L}^t_{(o)} = -\sum_{c=1}^{|\mathcal{C}|} y^t_{o,c} \log(p^t_{o,c}) \tag{9}$$

Predicate Classification Loss. The model employs a dual-component loss system to understand and predict object relationships over time given as follows:

$$\mathcal{L}_{\text{gen}} = \sum_{t=1}^{T} \sum_{ij} \mathcal{L}_{p^t_{ij}}, \quad \mathcal{L}_{\text{ant}} = \sum_{t=T+1}^{T+H} \sum_{ij} \mathcal{L}_{p^t_{ij}}, \tag{10}$$

The first component, the Generation Loss focuses on classifying the relationships between pairs of objects $\left(o^t_i, o^t_j\right)$ across all observed frames ($t \in [1, T]$). The second component, the Anticipation Loss $\mathcal{L}_{\text{ant}}$ is designed for future frames ($t \in [T+1, T+H]$) and applied on extrapolated relationship representations. Here $\mathcal{L}_{p^t_{ij}}$ is given by

$$\mathcal{L}_{p^t_{ij}}(\hat{\mathbf{p}}^t_{ij}) = \sum_{i \in \mathcal{P}^+} \sum_{j \in \mathcal{P}^-} \max(0, 1 - \hat{\mathbf{p}}^t_{ij}[j] + \hat{\mathbf{p}}^t_{ij}[i]) \tag{11}$$

Bounding Box Regression Loss. In our model, each anticipated relationship representation is input into a dedicated linear layer to calculate the predicted bounding boxes for objects. The deviation between these predicted bounding boxes and the ground truth bounding boxes is quantified using the Smoothed L1 Loss, denoted as $\mathcal{L}_{\text{boxes}}$

$$\mathcal{L}^t_{\text{boxes}} = \sum_{i \in \text{boxes}} \text{L}_{smooth}(y^t_i - \hat{y}^t_i) \tag{12}$$

Reconstruction Loss. This loss function ensures that the reconstructed relationship representations closely mirror the outputs of temporal encoder in LDPU.

$$\mathcal{L}^t_{recon} = \frac{1}{N(t) \times N(t)} \sum_{ij}^{(N(t) \times N(t))} \text{L}_{smooth}(\mathbf{z}^t_{ij} - \hat{\mathbf{z}}^t_{ij}) \tag{13}$$

Thus, the total objective for training the proposed method can be written as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{gen}} + \lambda_2 \mathcal{L}_{\text{ant}} + \lambda_3 \sum_{t=T+1}^{T+H} \mathcal{L}^t_{\text{boxes}} + \lambda_4 \sum_{t=1}^{T} \mathcal{L}^t_{(o)} + \lambda_5 \sum_{t=T+1}^{T+H} \mathcal{L}^t_{recon} \tag{14}$$

6 Experiments

Dataset. We apply the proposed method to anticipate future interactions on the Action Genome [15]. We pre-process the data and filter out videos with less than 3 annotated frames. Thus, we obtained 11.4K videos in total; we adhered to the train and test split provided by the dataset. The dataset encompasses 35 object classes and 25 relationship classes. These 25 relationship classes are grouped into three categories, namely: (1) **Attention Relations** comprise relationship classes which primarily describe attention of the subject towards the object, (2) **Spatial Relations** comprise relationship classes that describe the spatial relationship between two objects, and (3) **Contacting Relations** comprises relationship classes that indicate different ways the object is contacted.

Remark. This dataset primarily includes videos featuring a single actor engaging with objects[4] in various real-world environments. In the context of the action genome, a subject-object pair can demonstrate multiple spatial and contacting relationships.

Evaluation Metric. We evaluate our models using the standard *Recall@K* and *meanRecall@K* metrics, where K is set to values within the set $\{10, 20, 50\}$. Recall@K metric assists in assessing the ability of our model to anticipate the relationships between observed objects in future frames. The long-tailed distribution of relationships in the training set [27] can generate biased scene graphs. While the performance on more common relationships can dominate the Recall@K metrics, the mean recall metric introduced in [4] is a more balanced metric that scores the model's generalisation to all predicate classes. In the following sections, we will assess the performance of models by modifying the initial fraction of the video, denoted as $\mathcal{F}$ provided as input. We set $\mathcal{F}$ to 0.3, 0.5, 0.7, and 0.9 to facilitate a comprehensive understanding of the model's proficiency in short-term and long-term relationship anticipation.

Settings. We introduce three settings for evaluating models designed for the SGA task. Each setting is defined by the varying degrees of scene information made available to the model as input: (a) **Action Genome Scenes (AGS):** Here, the model's input is limited to frames of the video. With a basic level of information, the model's ability to interpret the scenes with minimal context is challenged. (b) **Partially Grounded Action Genome Scenes (PGAGS):** In this intermediate setting, the model receives additional information beyond the basic video frames. Specifically, it is supplied with precise bounding box details for active interacting objects within the scene. (c) **Grounded Action Genome Scenes (GAGS):** This setting provides the most comprehensive level of scene information. The model is given precise bounding box information and the categories of the active objects involved in the interaction. In video data, frames frequently encounter challenges such as object occlusions and blurry imagery, which complicate object detection and interpretation of interaction. In a setting where complete information is available, noise induced by these challenges significantly decreases. Consequently, we aim to evaluate the model's ability to understand spatio-temporal dynamics.

[4] Adopting an actor-centric approach emphasises the actor's importance in the scene and also aligns with the literature on VidSGG. While SceneSayer can anticipate relationship representations between any two objects in a scene, we specifically tailor it to address the dynamics between the actor (subject) and the objects with which they interact.

6.1 Baselines

We select two methods from VidSGG literature, STTran [7] and DSGDetr [9], as our strong baselines[5] for adaptations. In our adaptation of selected methods to anticipate the relationships, we retain the object representation processing unit (ORPU) and spatial context processing unit (SCPU) as proposed originally and introduce two variants of latent dynamics processing unit (LDPU) (see Fig. 3). The proposed two variants differ in (1) architecture and (2) loss functions.

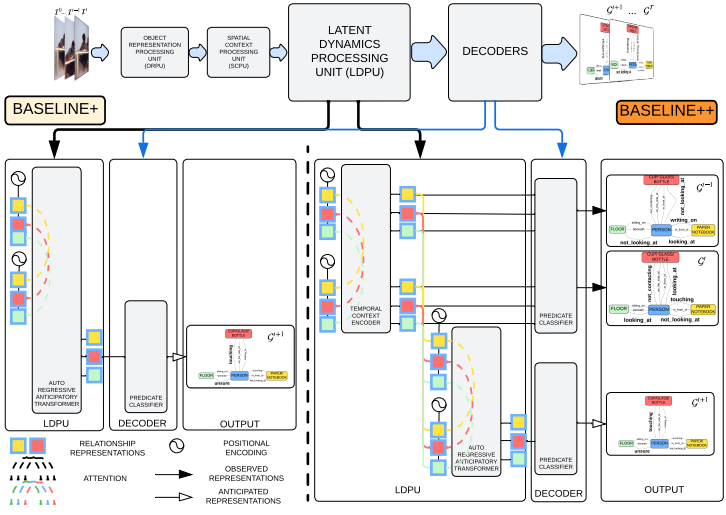

Fig. 3: Overview of Baselines. In our proposed *Variant 1*(shown to the left), we input relationship representations to an anticipatory transformer to generate relationship representations for future frames auto-regressively. A predicate classification network is then employed to decode these anticipated relationship representations. Meanwhile, in *Variant 2*(shown to the right), we enhance relationship representations by passing them through a temporal encoder. These representations are fed to an auto-regressive anticipatory transformer to anticipate future relationship representations. Here, we employ two predicate classification heads: one decodes the observed relationship representations, and the other decodes the anticipated relationship representations. In both variants, the auto-regressive anticipatory transformer acts as the generative model, predicting the evolution of relationships between the objects.

In terms of architectural differences, in our *baseline*+ variants, we employ an anticipatory transformer built using the vanilla transformer architecture [35] to generate future relationship representations by processing the relationship representations in the observed temporal context. Meanwhile, in our *baseline*++ variants, we introduce an additional temporal encoder to process the representations before we feed them through

[5] Although VidSGG literature witnessed many approaches recently, most use the representation processing pipeline proposed by the two selected transformer-based VidSGG methods.

the anticipatory transformer. This component further refines the representations by enhancing the model's understanding of spatio-temporal dynamics. In the context of loss functions, the approach followed by our *baseline+* variants focuses solely on decoding the anticipated relationship representations. However, in contrast, our *baseline++* variants adopt a more comprehensive strategy. These not only decode the anticipated representations but also simultaneously decode the observed representations. This dual-decoding approach allows for a more nuanced understanding of relationships.

Table 1: Results for **SGA of AGS**, when trained using anticipatory horizon of 3 future frames.

	SGA of AGS	Recall (R)						Mean Recall (mR)					
		With Constraint			No Constraint			With Constraint			No Constraint		
$\mathcal{F}$	Method	10	20	50	10	20	50	10	20	50	10	20	50
0.3	STTran+ [7]	12.5	19.6	20.7	13.9	21.6	40.8	3.4	5.3	5.7	3.5	7.3	20.3
	DSGDetr+ [9]	12.8	19.6	20.4	14.3	21.8	41.3	3.5	5.3	5.6	3.6	7.6	21.2
	STTran++ [7]	18.5	27.9	29.5	15.4	27.2	48.6	5.9	10.4	11.3	6.2	14.1	31.2
	DSGDetr++ [9]	19.5	28.3	29.4	16.8	29.0	**48.9**	6.0	10.3	11.0	8.4	16.7	32.3
	SceneSayerODE (Ours)	23.1	29.2	31.4	23.3	32.5	45.1	10.6	13.8	15.0	13.3	20.1	33.0
	SceneSayerSDE (Ours)	**25.0**	**31.7**	**34.3**	**25.9**	**35.0**	47.4	**11.4**	**15.3**	**16.9**	**15.6**	**23.1**	**37.1**
0.5	STTran+ [7]	13.1	21.1	22.2	14.9	22.6	42.9	3.6	5.8	6.2	3.7	7.6	21.4
	DSGDetr+ [9]	13.6	20.9	21.9	15.2	23.1	43.3	3.8	5.8	6.1	3.9	8.0	22.2
	STTran++ [7]	19.7	30.2	31.8	16.6	29.1	51.5	6.3	11.3	12.3	6.6	14.7	33.4
	DSGDetr++ [9]	20.7	30.3	31.6	17.4	30.5	**51.9**	6.4	11.0	11.7	8.4	17.0	33.9
	SceneSayerODE (Ours)	25.9	32.6	34.8	26.4	36.6	49.8	11.6	15.2	16.4	14.3	21.4	36.0
	SceneSayerSDE (Ours)	**27.3**	**34.8**	**37.0**	**28.4**	**38.6**	51.4	**12.4**	**16.6**	**18.0**	**16.3**	**25.1**	**39.9**
0.7	STTran+ [7]	14.9	23.4	24.7	16.6	25.1	47.2	4.1	6.5	7.0	4.2	8.5	24.0
	DSGDetr+ [9]	15.5	23.4	24.3	16.8	25.3	47.4	4.3	6.5	6.9	4.3	8.8	24.7
	STTran++ [7]	22.1	33.6	35.2	19.0	32.8	**56.8**	7.0	12.6	13.6	7.7	17.1	36.8
	DSGDetr++ [9]	22.9	33.6	34.9	19.8	34.1	56.7	7.1	12.6	13.3	9.5	19.2	37.2
	SceneSayerODE (Ours)	30.3	36.6	38.9	32.1	42.8	55.6	12.8	16.4	17.8	16.5	24.4	39.6
	SceneSayerSDE (Ours)	**31.4**	**38.0**	**40.5**	**33.3**	**44.0**	56.4	**13.8**	**17.7**	**19.3**	**18.1**	**27.3**	**44.4**
0.9	STTran+ [7]	15.2	24.1	25.4	17.5	26.8	49.6	4.5	7.0	7.5	4.6	9.2	24.3
	DSGDetr+ [9]	16.2	24.8	25.9	17.9	27.7	51.4	4.8	7.2	7.6	4.7	9.7	25.9
	STTran++ [7]	23.6	35.5	37.4	20.2	35.0	60.2	7.4	13.4	14.6	8.9	18.4	38.8
	DSGDetr++ [9]	24.4	36.1	37.6	22.2	37.1	61.0	7.4	13.8	14.8	11.4	21.0	39.5
	SceneSayerODE (Ours)	33.9	40.4	42.6	36.6	48.3	61.3	14.0	18.1	19.3	17.8	27.4	43.4
	SceneSayerSDE (Ours)	**34.8**	**41.9**	**44.1**	**37.3**	**48.6**	**61.6**	**15.1**	**19.4**	**21.0**	**20.8**	**30.9**	**46.8**

6.2 Results

Action Genome Scenes. We present the results in Table. 1. In this setting, we observe that both STTran++ and DSGDetr++ consistently outperform their basic counterparts. Furthermore, it can be clearly observed that the SceneSayerSDE model consistently outperforms the SceneSayerODE model, which, in turn, performs better than the baseline variants. SceneSayerODE/SDE models only fall short on the R@50 metric in the No Constraint graph generation strategy. Nevertheless, the R@K metric holds greater importance for lower values of K and, in R@10 and R@20 metrics, consistently outperforms other models; we observe that SceneSayerSDE is up to $\sim 70\%$ better on the R@10 metric than the best baseline variant. Additionally, the SceneSayerODE/SDE models exhibit reduced prediction bias, as evidenced by their consistently better performance across the mean recall metrics.

Partially Grounded Action Genome Scenes. We present the results in Table. 2. We observed that the STTran adaptation demonstrates superior performance compared

Table 2: Results for **SGA of PGAGS**, when trained using anticipatory horizon of 3 future frames.

	SGA of PGAGS	Recall (R)						Mean Recall (mR)					
		With Constraint			No Constraint			With Constraint			No Constraint		
$\mathcal{F}$	Method	10	20	50	10	20	50	10	20	50	10	20	50
0.3	STTran+ [7]	22.3	22.9	22.9	25.5	38.3	45.6	8.6	9.1	9.1	13.1	24.8	42.3
	DSGDetr+ [9]	13.6	14.1	14.1	22.2	33.9	46.0	4.8	5.0	5.0	7.9	16.9	40.9
	STTran++ [7]	22.1	22.8	22.8	28.1	39.0	45.2	9.2	9.8	9.8	17.7	30.6	42.0
	DSGDetr++ [9]	18.2	18.8	18.8	27.7	39.2	**47.3**	8.9	9.4	9.4	15.3	26.6	44.0
	SceneSayerODE (Ours)	27.0	27.9	27.9	33.0	40.9	46.5	12.9	13.4	13.4	19.4	27.9	46.9
	SceneSayerSDE (Ours)	**28.8**	**29.9**	**29.9**	**34.6**	**42.0**	46.2	**14.2**	**14.7**	**14.7**	**21.5**	**31.7**	**48.2**
0.5	STTran+ [7]	24.2	24.9	24.9	27.5	41.6	49.9	9.3	9.9	9.9	13.8	26.7	43.1
	DSGDetr+ [9]	15.8	16.2	16.2	24.7	38.2	52.4	5.5	5.7	5.7	8.7	18.5	41.5
	STTran++ [7]	24.5	25.2	25.2	30.6	43.2	50.2	10.1	10.7	10.7	18.4	29.5	43.1
	DSGDetr++ [9]	20.7	21.4	21.4	30.4	44.0	**52.7**	10.2	10.8	10.8	16.5	30.8	45.1
	SceneSayerODE (Ours)	30.5	31.5	31.5	36.8	45.9	51.8	14.9	15.4	15.5	21.6	30.8	48.0
	SceneSayerSDE (Ours)	**32.2**	**33.3**	**33.3**	**38.4**	**46.9**	51.8	**15.8**	**16.6**	**16.6**	**23.5**	**35.0**	**49.6**
0.7	STTran+ [7]	28.9	29.4	29.4	33.6	49.9	58.7	10.7	11.3	11.3	16.3	32.2	49.9
	DSGDetr+ [9]	18.8	19.1	19.1	29.9	44.7	60.2	6.5	6.8	6.8	10.4	22.4	47.7
	STTran++ [7]	29.1	29.7	29.7	36.8	51.6	58.7	11.5	12.1	12.1	21.2	34.6	49.0
	DSGDetr++ [9]	24.6	25.2	25.2	36.7	51.8	**60.6**	12.0	12.6	12.6	19.7	36.4	50.6
	SceneSayerODE (Ours)	36.5	37.3	37.3	44.6	54.4	60.3	16.9	17.3	17.3	25.2	36.2	53.1
	SceneSayerSDE (Ours)	**37.6**	**38.5**	**38.5**	**45.6**	**54.6**	59.3	**18.4**	**19.1**	**19.1**	**28.3**	**40.9**	**54.9**
0.9	STTran+ [7]	30.9	31.3	31.3	39.9	56.6	63.8	11.5	11.9	11.9	20.3	37.7	54.3
	DSGDetr+ [9]	21.3	21.6	21.6	38.4	54.9	**68.7**	7.5	7.7	7.7	13.7	29.0	55.4
	STTran++ [7]	31.1	31.6	31.6	43.5	57.6	63.9	12.4	12.8	12.8	25.3	39.6	54.0
	DSGDetr++ [9]	27.6	28.1	28.1	45.8	61.5	68.5	13.2	13.7	13.7	25.8	42.9	58.3
	SceneSayerODE (Ours)	41.6	42.2	42.2	52.7	61.8	66.5	19.0	19.4	19.4	29.4	42.2	59.2
	SceneSayerSDE (Ours)	**42.5**	**43.1**	**43.1**	**53.8**	**62.4**	66.2	**20.6**	**21.1**	**21.1**	**32.9**	**46.0**	**59.8**

to the DSGDetr adaptation across both model variants. Notably, the SceneSayer models significantly outperform all proposed baseline models. In particular, the SceneSayerSDE model achieves up to $\sim 30\%$ improvement on the R@10 metric compared to the best-performing baseline model. Additionally, both SceneSayerODE/SDE models outperform all proposed baseline models in mean recall metrics, suggesting their proficiency in predicting relationships with a reduced bias.

Table 3: Results for **SGA of GAGS**, when trained using anticipatory horizon of 3 future frames.

	SGA of GAGS	Recall (R)						Mean Recall (mR)					
		With Constraint			No Constraint			With Constraint			No Constraint		
$\mathcal{F}$	Method	10	20	50	10	20	50	10	20	50	10	20	50
0.3	STTran+ [7]	30.8	32.8	32.8	30.6	47.3	62.8	7.1	7.8	7.8	9.5	19.6	45.7
	DSGDetr+ [9]	27.0	28.9	28.9	30.5	45.1	62.8	6.7	7.4	7.4	9.5	19.4	45.2
	STTran++ [7]	30.7	33.1	33.1	35.9	51.7	64.1	11.8	13.3	13.3	16.5	29.3	50.2
	DSGDetr++ [9]	25.7	28.2	28.2	36.1	50.7	64.0	11.1	12.8	12.8	19.7	32.0	51.1
	SceneSayerODE (Ours)	34.9	37.3	37.3	40.5	54.1	63.9	15.1	16.6	16.6	19.6	31.6	55.8
	SceneSayerSDE (Ours)	**39.7**	**42.2**	**42.3**	**46.9**	**59.1**	**65.2**	**18.4**	**20.5**	**20.5**	**24.6**	**37.8**	**59.0**
0.5	STTran+ [7]	35.0	37.1	37.1	34.4	53.4	70.8	8.0	8.7	8.8	10.5	21.5	48.8
	DSGDetr+ [9]	31.2	33.3	33.3	34.3	51.0	70.8	7.8	8.6	8.6	10.5	21.4	48.4
	STTran++ [7]	35.6	38.1	38.1	40.3	58.4	72.2	13.4	15.2	15.2	17.8	32.5	53.7
	DSGDetr++ [9]	29.3	31.9	32.0	40.3	56.9	72.0	12.2	13.8	13.9	20.6	34.3	54.0
	SceneSayerODE (Ours)	40.7	43.4	43.4	47.0	62.2	72.4	17.4	19.2	19.3	22.8	35.2	60.2
	SceneSayerSDE (Ours)	**45.0**	**47.7**	**47.7**	**52.5**	**66.4**	**73.5**	**20.7**	**23.0**	**23.1**	**26.6**	**40.8**	**63.8**
0.7	STTran+ [7]	40.0	41.8	41.8	41.0	62.5	80.4	9.1	9.8	9.8	12.6	26.3	57.5
	DSGDetr+ [9]	35.5	37.3	37.3	41.0	59.8	80.7	8.9	9.6	9.6	12.6	26.2	57.4
	STTran++ [7]	41.3	43.6	43.6	48.2	68.8	82.0	16.3	18.2	18.2	22.3	39.5	63.1
	DSGDetr++ [9]	33.9	36.3	36.3	48.0	66.7	81.9	14.2	15.9	15.9	24.5	41.1	63.4
	SceneSayerODE (Ours)	49.1	51.6	51.6	58.0	74.0	82.8	21.0	22.9	22.9	27.3	43.2	70.5
	SceneSayerSDE (Ours)	**52.0**	**54.5**	**54.5**	**61.8**	**76.7**	**83.4**	**24.1**	**26.5**	**26.5**	**31.9**	**48.0**	**74.2**
0.9	STTran+ [7]	44.7	45.9	45.9	50.9	74.8	90.9	10.3	10.8	10.8	16.3	33.7	71.4
	DSGDetr+ [9]	38.8	40.0	40.0	51.0	71.7	91.2	10.2	10.7	10.7	16.3	33.7	71.6
	STTran++ [7]	46.0	47.7	47.7	60.2	81.5	92.3	19.6	21.4	21.4	29.6	49.1	76.4
	DSGDetr++ [9]	38.1	39.8	39.8	58.8	78.8	92.2	16.3	17.7	17.7	30.7	50.3	77.2
	SceneSayerODE (Ours)	58.1	59.8	59.8	72.6	86.7	93.2	25.0	26.4	26.4	35.0	51.7	80.2
	SceneSayerSDE (Ours)	**60.3**	**61.9**	**61.9**	**74.8**	**88.0**	**93.5**	**28.5**	**29.8**	**29.8**	**40.0**	**57.7**	**87.2**

Grounded Action Genome Scenes. We present the results in Table. 3. We observe that in our proposed baseline models, the adaptation of STTran surpasses the adaptation of DSGDetr for both model variations. SceneSayerODE/SDE consistently outperforms all baselines across all metrics. The SceneSayerSDE model, in particular, demonstrated remarkable efficiency, marking approximately a 26% improvement in the Recall@10 (R@10) metric compared to the previously best-performing baseline. Furthermore, both variations of the SceneSayer models, ODE and SDE, demonstrated superior performance over the conventional baseline models on mean recall metrics.

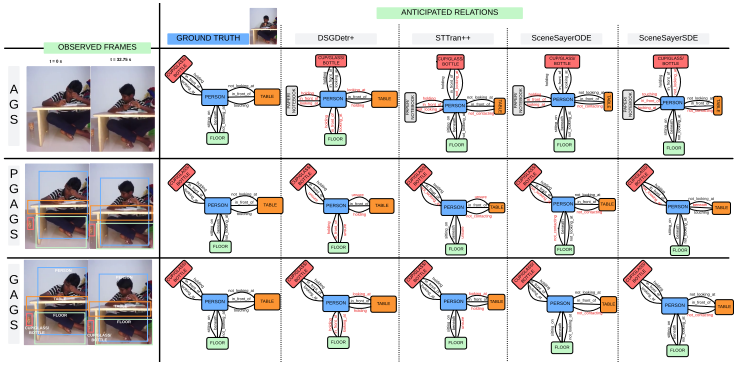

Fig. 4: Qualitative Results To the left, we show a sampled subset of the frames observed by the models. The second column provides a ground truth scene graph corresponding to a future frame. In the subsequent columns, we contrast the performance of baseline variants with the proposed SceneSayer models. In each graph above, correct anticipations of relationships are denoted with text in black and incorrect anticipation of the relationships are highlighted with text in red.

Qualitative Results. In Fig. 4, we provide qualitative SGA results on AG dataset under different settings. These results are the output of baseline variants and SceneSayer on 70% observation. The comparative analysis demonstrates SceneSayer's superior performance relative to the baseline variants. Specifically, under the AGS setting, SceneSayerODE outshines all with the fewest incorrect anticipations, closely followed by SceneSayerSDE. In the PGAGS and GAGS, SceneSayerSDE surpasses the performance of all other models.

7 Conclusion

We introduced a novel task, Scene Graph Anticipation (SGA) and proposed approaches that model the latent dynamics of the evolution of relationship representations between interacting objects. Our work opens up several avenues for future work. First, an exciting direction is to include localization into the framework. Second, SGA as a tool can be used to develop methods for Surveillance Systems, Robotics, etc.

References

1. Agia, C., Jatavallabhula, K.M., Khodeir, M., Miksik, O., Vineet, V., Mukadam, M., Paull, L., Shkurti, F.: Taskography: Evaluating robot task planning over large 3d scene graphs (2022) 3
2. Bar, A., Herzig, R., Wang, X., Rohrbach, A., Chechik, G., Darrell, T., Globerson, A.: Compositional video synthesis with action graphs. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 662–673. PMLR (2021), http://proceedings.mlr.press/v139/bar21a.html 3
3. Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.: Neural ordinary differential equations. Neural Information Processing Systems (2018). https://doi.org/null 2, 4, 8
4. Chen, T., Yu, W., Chen, R., Lin, L.: Knowledge-embedded routing network for scene graph generation (2019) 10
5. Chen, Z., Wu, J., Lei, Z., Zhang, Z., Chen, C.: Expanding scene graph boundaries: Fully open-vocabulary scene graph generation via visual-concept alignment and retention (2023) 3
6. Cherian, A., Hori, C., Marks, T.K., Roux, J.L.: (2.5+1)d spatio-temporal scene graphs for video question answering (2022) 3
7. Cong, Y., Liao, W., Ackermann, H., Yang, M., Rosenhahn, B.: Spatial-temporal transformer for dynamic scene graph generation. IEEE International Conference on Computer Vision (2021). https://doi.org/10.1109/iccv48922.2021.01606 2, 3, 7, 11, 12, 13
8. Dhamo, H., Farshad, A., Laina, I., Navab, N., Hager, G.D., Tombari, F., Rupprecht, C.: Semantic image manipulation using scene graphs. In: CVPR (2020) 3
9. Feng, S., Mostafa, H., Nassar, M., Majumdar, S., Tripathi, S.: Exploiting long-term dependencies for generating dynamic scene graphs. IEEE Workshop/Winter Conference on Applications of Computer Vision (2021). https://doi.org/10.1109/wacv56688.2023.00510 11, 12, 13
10. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction (2016) 3
11. Girdhar, R., Grauman, K.: Anticipative Video Transformer. In: ICCV (2021) 7
12. Huang, Z., Sun, Y., Wang, W.: Learning continuous system dynamics from irregularly-sampled partial observations (2020) 4
13. Huang, Z., Sun, Y., Wang, W.: Coupled graph ode for learning interacting system dynamics. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. p. 705–715. KDD '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467385, https://doi-org.libproxy.utdallas.edu/10.1145/3447548.3467385 4
14. Höppe, T., Mehrjou, A., Bauer, S., Nielsen, D., Dittadi, A.: Diffusion models for video prediction and infilling (2022) 3
15. Ji, J., Krishna, R., Fei-Fei, L., Li, F.F., Niebles, J.C.: Action genome: Actions as composition of spatio-temporal scene graphs. arXiv: Computer Vision and Pattern Recognition (2019). https://doi.org/10.1109/cvpr42600.2020.01025 1, 2, 3, 10
16. Khandelwal, A.: Correlation debiasing for unbiased scene graph generation in videos (2023) 3
17. Kidger, P., Foster, J., Li, X., Lyons, T.: Efficient and accurate gradients for neural sdes (2021) 2, 4, 8
18. Kim, K., Yoon, K., Jeon, J., In, Y., Moon, J., Kim, D., Park, C.: Llm4sgg: Large language model for weakly supervised scene graph generation (2023) 3

19. Kim, U.H., Park, J.M., jin Song, T., Kim, J.H.: 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. IEEE Transactions on Cybernetics **50**, 4921–4933 (2019), https://api.semanticscholar.org/CorpusID:199577350 3
20. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. **123**(1), 32–73 (2017). https://doi.org/10.1007/S11263-016-0981-7, https://doi.org/10.1007/s11263-016-0981-7 3
21. Kurenkov, A., Lingelbach, M., Agarwal, T., Jin, E., Li, C., Zhang, R., Fei-Fei, L., Wu, J., Savarese, S., Martín-Martín, R.: Modeling dynamic environments with scene graph memory (2023) 3
22. Lee, A.X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic adversarial video prediction (2018) 3
23. Li, R., Zhang, S., He, X.: Sgtr: End-to-end scene graph generation with transformer. Computer Vision and Pattern Recognition (2021). https://doi.org/10.1109/cvpr52688.2022.01888 4
24. Liang, Y., Ouyang, K., Yan, H., Wang, Y., Tong, Z., Zimmermann, R.: Modeling trajectories with neural ordinary differential equations. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 1498–1504. International Joint Conferences on Artificial Intelligence Organization (8 2021). https://doi.org/10.24963/ijcai.2021/207, https://doi.org/10.24963/ijcai.2021/207, main Track 4
25. Liu, Z., Shojaee, P., Reddy, C.K.: Graph-based multi-ode neural networks for spatio-temporal traffic forecasting (2023) 4
26. Mi, L., Ou, Y., Chen, Z.: Visual relationship forecasting in videos. arXiv.org (2021). https://doi.org/null 3
27. Nag, S., Min, K., Tripathi, S., Roy-Chowdhury, A.K.: Unbiased scene graph generation in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22803–22813 (2023) 3, 10
28. Park, S., Kim, K., Lee, J., Choo, J., Lee, J., Kim, S., Choi, E.: Vid-ode: Continuous-time video generation with neural ordinary differential equation (2021) 4
29. Poli, M., Massaroli, S., Park, J., Yamashita, A., Asama, H., Park, J.: Graph neural ordinary differential equations (2021) 4
30. Qi, H., Wang, X., Pathak, D., Ma, Y., Malik, J.: Learning long-term visual dynamics with region proposal interaction networks (2021) 3
31. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks (2016) 7
32. Shang, X., Ren, T., Guo, J., Zhang, H., Chua, T.S.: Video visual relation detection. In: Proceedings of the 25th ACM International Conference on Multimedia. p. 1300–1308. MM '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3123266.3123380, https://doi-org.libproxy.utdallas.edu/10.1145/3123266.3123380 3
33. Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., Li, H., Pan, J., Sharifzadeh, S., Kaissis, G., Tresp, V., Menze, B.: Relationformer: A unified framework for image-to-graph generation (2022) 3
34. Thickstun, J., Hall, D., Donahue, C., Liang, P.: Anticipatory music transformer (2023) 7
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances

in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf 7, 11
36. Voleti, V., Jolicoeur-Martineau, A., Pal, C.: Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation (2022) 3
37. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: Forecasting from static images using variational autoencoders (2016) 3
38. Wu, X., Zhao, J., Wang, R.: Anticipating future relations via graph growing for action prediction. Proceedings of the AAAI Conference on Artificial Intelligence **35**(4), 2952–2960 (May 2021). https://doi.org/10.1609/aaai.v35i4.16402, https://ojs.aaai.org/index.php/AAAI/article/view/16402 3
39. Wu, Z., Dvornik, N., Greff, K., Kipf, T., Garg, A.: Slotformer: Unsupervised visual dynamics simulation with object-centric models (2023) 3
40. Yu, W., Chen, W., Yin, S., Easterbrook, S., Garg, A.: Modular action concept grounding in semantic video prediction (2022) 3
41. Zhao, S., Xu, H.: Less is more: Toward zero-shot local scene graph generation via foundation models (2023) 3
42. Zhou, Z., Shi, M., Caesar, H.: Vlprompt: Vision-language prompting for panoptic scene graph generation (2023) 3
43. Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Feng, M., Zhao, X., Miao, Q., Shah, S.A.A., Bennamoun, M.: Scene graph generation: A comprehensive survey (2022) 3