FOREcasting human activities via latent SCENE graphs diffusion

Antonio Alliegro[1] Francesca Pistilli[1] Tatiana Tommasi[1] Giuseppe Averta[1]
[1] Politecnico di Torino
firstname.lastname@polito.it

Abstract

Forecasting human-object interactions in daily activities is challenging because of the high variability of human behavior. Although training models to solve this task from plain videos is feasible, directly operating on raw frames is often limited by visual noise and confounding factors unrelated to the task. Scene graphs offer a promising alternative by providing structured representations of actor and objects actively participating in the action, and their relationships, potentially evolving over time. However, existing approaches to Scene Graph Anticipation (SGA) often rely on unrealistic assumptions, such as fixed objects over time, which limit their applicability to dynamic, real-world scenarios. In this paper, we propose FORESCENE, a novel framework for SGA that jointly predicts the temporal evolution of both objects and their interactions, based on a graph auto-encoder and a conditional latent diffusion model. We evaluated FORESCENE on the Action Genome dataset, showing that providing full graph prediction improves the model capabilities in human activity forecasting and outperforms prior SGA methods.

1. Introduction

Effective human-robot collaboration in shared environments, such as homes, hospitals, or industrial settings, requires robots to understand and forecast human activities. This goes far beyond recognizing static actions; rather, the model needs to reason about how activities evolve over time, how humans interact with surrounding objects and environment, and how actions are composed and interleaved to form long-horizon activities.

Imagine a human and a robot assembling a chair together. The human uses a screwdriver to fasten the seat, then sets it aside to reach for a chair leg. To collaborate effectively, the robot must anticipate this shift, recognizing when objects like the screwdriver become inactive and new ones become relevant, and prepare accordingly by handing over parts. This requires learning representations of human activities that capture actors, active objects, and their evolving relationships over time. Thanks to the growing availability

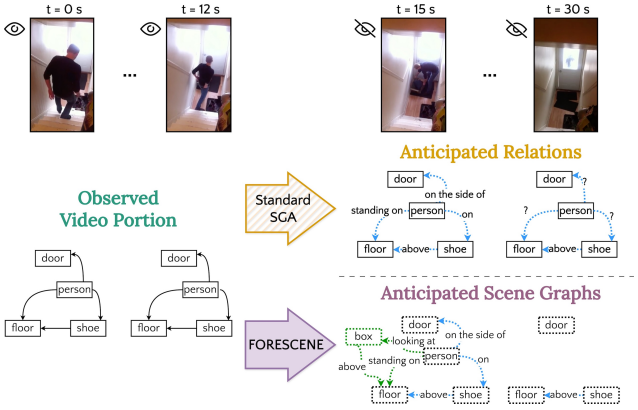

Figure 1. Scene graphs for human-environment interactions. Dotted lines indicate predictions, while solid lines fixed elements.

of large-scale video datasets capturing diverse human activities, it is possible to learn this reasoning paradigm directly from observation [4]. One promising direction is to leverage directly visual input, which provides rich contextual information but requires disentangling relevant information from irrelevant details [9]. Alternatively, textual descriptions offer compact, high-level representations but often face ambiguity and grounding challenges [9].

We argue that Scene Graphs (SGs) offer a more effective representation, combining the grounded detail of visual data with the structured semantics and compactness of textual descriptions. In SGs, humans and active objects are represented as nodes, while their semantic and spatial relationships are modeled as edges [7]. This structured representation enables models to reason about dynamic human-environment interactions by focusing on behaviorally relevant elements, without sacrificing physical grounding. To anticipate human behavior effectively, the robot must predict how these graph structures evolve over time.

In this paper, we address the underexplored task of video-based Scene Graph Anticipation (SGA). Prior work [10] focuses on forecasting only edge attributes (*i.e.*, relationships) while keeping the set of nodes (*i.e.*, objects) fixed to those in the last observed frame. While this is sufficient for short-term activities where active objects remain unchanged, it fails in longer activities where the ob-

ject set evolves dynamically (see Fig. 1). We introduce FORESCENE, a novel model for SGA that predicts both objects and their relationships over time. Our approach operates in two stages: (i) a Graph Auto-Encoder maps observed video segments into a latent representation; and (ii) a Latent Diffusion Model forecasts the temporal evolution of SGs conditioned on the observed portion. We evaluate on the Action Genome dataset [7], measuring performance in terms of the ability to predict objects (Object Discovery), and relationships (Triplet Recall) over time. Our experiments demonstrate the importance of predicting the evolution of complete SGs, outperforming existing methods while solving a more complex task.

2. Method

Our goal is to design a model able to predict future instances of SGs from a partial video observation. We tackle this by first encoding SGs extracted from observed RGB frames into fixed-size latent vectors using a tailored Graph Auto-Encoder (GAE), and then using a Latent Diffusion Model (LDM) to generate future instances of SGs conditioned on the observed ones. We train the LDM on temporally ordered sequences of graph latents to capture the temporal evolution of SGs and patterns of human activity in everyday scenarios. At inference, the LDM produces temporally coherent latent codes for future SGs, which the GAE decodes into complete SGs (see Fig. 2), effectively forecasting the future evolution of human-environment interactions based on partial observations. **Notation and Task Definition** We define a SG as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ represents the set of nodes, each corresponding to an object, and $\mathcal{E}$ denotes the set of edges, capturing the relationships between nodes. Each node $v_i \in \mathcal{V}$ is associated with an object category label $v_i^c \in \mathcal{C}$ and bounding box coordinates $v_i^b \in [0,1]^4$. An edge $e_{ij} \in \mathcal{E}$ serves to model the relationship between nodes v_i and v_j, with v_i acting as the *subject* and v_j as the *object* of the relation. The relationship category is represented by $p_{ij}^c \in \mathcal{P}$. **SGA** predicts SGs $\mathcal{G}_{F_{s+1}}, \ldots, \mathcal{G}_{F_{last}}$ for future unseen frames $\{I\}_{F_{s+1}}^{F_{last}}$, based on a partial video observation $\{I\}_0^{F_s}$.

2.1. Graph Auto-Encoder

The first stage of our method is a GAE that encodes observed frames into fixed-size latent graph representations and decodes them into SGs. We use a GCN-based encoder and a transformer decoder [14].

Graph Encoder ($\mathcal{G} \xrightarrow{E_\xi} \mathbf{z}$) The encoder learns to map a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ to a latent code $\mathbf{z}$. Input graphs are populated with node features $\{\phi_{v,i}\}_{i=1}^{|\mathcal{V}|}$ and edge features $\{\phi_{e,ij}\}_{e=1}^{|\mathcal{E}|}$. Following [10], *node features* $\phi_{v,i}$ are initialized using projected object features v_i^f extracted from a frozen object detector [12] and bounding box coordinates v_i^b. For *edge features* $\phi_{e,ij}$, we follow [2], and concatenate visual and semantic features (*i.e.*, the object category) of object pairs. The node and edge features are then structured as triplets $t_{ij} = \langle \phi_{v,i}, \phi_{e,ij}, \phi_{v,j} \rangle$ and fed into the GCN architecture proposed in [8]. The node and edge features of the last GCN layer are max-pooled, concatenated, and linearly projected to obtain the latent graph representation $\mathbf{z} \in \mathbb{R}^C$. To encourage the encoder to learn semantic features for nodes and edges and store them in the latent $\mathbf{z}$, we add an encoder loss to predict node and edge categories.

Graph Decoder ($\mathbf{z} \xrightarrow{D_\gamma} \hat{\mathcal{G}}$) Inspired by [1], our decoder D_γ processes the graph latent representation $\mathbf{z}$ alongside N object queries to generate a feature representation of each node in the SG. The architecture is composed of L stacked blocks, each of which sequentially performs cross-attention, self-attention, and feed-forward operations. The final output is fed to $\text{MLP}_{\text{objects}}$ and $\text{MLP}_{\text{boxes}}$, which respectively output the predicted object category distribution over the $\mathcal{C}$ object classes, $\hat{v}^c \in \mathbb{R}^{N \times \mathcal{C}}$, and bounding boxes $\hat{v}^b \in \mathbb{R}^{N \times 4}$ for each object query. For *edge prediction*, we follow [6] to obtain the refined predicted relational features. These features $\hat{\phi}_e$ are processed by $\text{MLP}_{\text{edges}}$ and MLP_{con}, both followed by a sigmoid function to obtain the predicted relation matrix and the predicted connectivity matrix. The decoder loss is based on object detection and categorical relations, and connectivity prediction.

GAE Training Objective The GAE is trained minimizing the following loss function: $\mathcal{L}_{\text{GAE}} = \mathcal{L}_{\text{enco}} + \mathcal{L}_{\text{deco}} + \mathcal{L}_{\text{reg}}$, where $\mathcal{L}_{\text{enco}}$ ensures that the latent code captures node and edge semantics, $\mathcal{L}_{\text{deco}}$ enforces accurate SG reconstruction from latent representations, and $\mathcal{L}_{\text{reg}}$ regularizes the latent space to support subsequent Latent Diffusion modeling [3].

2.2. Latent Diffusion Model

Diffusion models [5, 13] are probabilistic models that learn a data distribution $p(x)$ by iteratively denoising a variable initially sampled from a normal distribution. The working principle can be interpreted as reversing a predefined Markov chain of length T through a sequence of denoising autoencoders $\epsilon_\theta(x_t, t)$, $t = 1 \ldots T$, each trained to predict a cleaner version of its input x_t , with x_t being a noisy version of the original data x.

Scene Graph Anticipation through LDM We train the LDM on temporally ordered sequences of ground-truth SG latent vectors $\mathbf{Z} = \{\mathbf{z}_f\}_{f=0}^{F_{last}}$ obtained from our Graph Encoder E_ξ. Each $\mathbf{z}_f \in \mathbb{R}^C$ represents the graph latent for the f-th frame. F_{last} is the total number of frames in the video. For each video, the diffusion process targets an ordered sequence of latent codes. The LDM is trained using fixed-size sliding windows of width S across the entire sequence $\mathbf{Z}$. At each window, we randomly select half of the tokens (each representing a SG latent) and add Gaussian noise to these, using the remaining tokens as conditioning to guide the de-

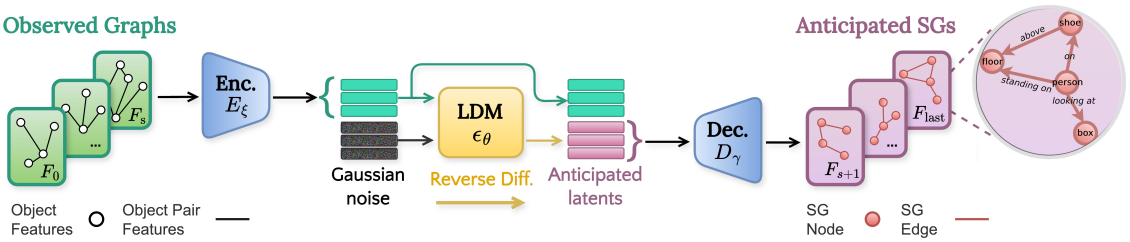

Figure 2. Overview of the proposed method at inference time to solve the task of Scene Graph Anticipation (SGA).

noising process. Each token in the window is positionally encoded to maintain temporal coherence. The denoising model $\epsilon_\theta(\cdot, t)$ is implemented as a DiT [11] transformer. We augment the denoising model input with the diffusion-timestep embedding and token-specific embeddings to indicate whether each token $\mathbf{z}_f^t$ at diffusion-timestep t is a conditioning or noised token. In the forward diffusion, each latent $\mathbf{z}_f^t$ is obtained by adding t-scheduled Gaussian noise to $\mathbf{z}_f$ from the Graph Encoder E_ξ. During reverse diffusion, samples from the prior distribution $p(\mathbf{z})$ can be decoded into SGs in a single pass through our Graph Decoder D_γ.

2.3. Scene Graph Anticipation

At inference, we use the observed video portion $\{I\}_0^{F_s}$ to forecast complete SGs for the future, unseen frames $\{I\}_{F_{s+1}}^{F_{last}}$. Following [10], the observed portion is represented as a sequence of graphs based solely on the visual features and category information of objects (see Sec. 2.1). The Graph Encoder E_ξ maps the observed graphs into latent representations. Let $\mathbf{Z}_{\text{seen}} = \mathbf{z}_{f=0}^{F_s}$ denote the sequence of observed graph latents, which serve as conditioning for forecasting future latents via the LDM. Future unknown latents are initialized with Gaussian noise and refined through iterative reverse diffusion, leveraging conditioning from the observed temporal context. Finally, the Graph Decoder D_γ maps the predicted latents to complete (objs + rels) SGs.

3. Experimental Setup

Dataset. We evaluate our approach on the Action Genome (AG, [7]) dataset, which provides dense SG annotations for human-object interactions in videos. It includes 35 object and 25 relationship categories. Relationships are grouped into three types: *attention* (*e.g.*, *looking at*), *spatial* (*e.g.*, *in front of*), and *contacting* (*e.g.*, *sitting on*). Multiple relationships may coexist between two entities.

Evaluation metrics. Evaluating performance for SGA is non-trivial. We propose a $Recall@K$ metric for Object Discovery, named **Object Recall**, with $K \in \{5, 10, 20\}$ based on the average number of objects per scene. This metric quantifies the fraction of ground-truth objects in the top-K predictions. Object Recall measures the ability to predict relevant objects but does not penalize the prediction of object categories absent from the ground truth scene graph. To address this, we adopt the **Jaccard index** defined as $J_{sim} = \frac{1}{|F_{last}-F_s|} \sum_{f=F_{s+1}}^{F_{last}} \frac{|\hat{O}_f \cap O_f|}{|\hat{O}_f \cup O_f|}$, which quantifies the similarity between the predicted set of objects $\hat{O}_f$ and the corresponding ground truth ones O_f. To evaluate relationships prediction, we follow [10] and employ the $Recall@K$ metric to evaluate predicted triplets (**Triplets Recall**) - with $K \in \{10, 20, 50\}$. This metric measures the proportion of ground truth relationship triplets present in the top-K predictions, providing a robust assessment of our model's accuracy in forecasting relationships. We report Triplet Recall under two scenarios: (i) *With Constraint*, allowing only one relation per node pair; and (ii) *No Constraint*, allowing multiple relations per pair. **Evaluation Settings.** We follow the GAGS and PGAGS protocols from [10], evaluating performance across different observation fractions $\mathcal{F} \in 0.3, 0.5, 0.7, 0.9$. The two settings differ in the supervision available for the observed portion: GAGS uses ground-truth object categories and bounding boxes, while PGAGS uses object categories from a pre-trained Faster R-CNN [12]. **SGA Comparison.** We compare FORESCENE with baselines introduced in SceneSayer [10], which adapted STTran and DSGDetr to the SGA setting to predict the future relationships between a fixed set of objects. In addition, we compare with SceneSayer itself [10], which is explicitly designed for scene graph anticipation and models the temporal evolution of relationships between objects fixed to the ones from the last observed frame by menas of differential equations (Ordinary or Stochastic Differential Equations). Thanks to its generative nature, FORESCENE can produce diverse future activity predictions from the same video observation by varying the diffusion seed. We report results for single-sample ($r = 1$) and multi-sample settings ($r = 5$, $r = 10$), selecting the best predictions using the R@10 Triplets (No Constraint) metric.

4. Results

Scene Graph Anticipation The results for the GAGS setting are presented in Tab. 1. In terms of Object Discovery, the assumption of object continuity across frames appears to be a reasonable fit for this setting of the AG dataset. Notably, FORESCENE is able to *learn* when to preserve object nodes, maintaining entities when changes are unlikely, as reflected by its high Object Recall when a large portion

Table 1. SGA results in GAGS setting. Values in gray assume object continuity. Top results are bold, second-best are underlined.

		Objects Discovery				Triplets With Constraint			Triplets No Constraint		
$\mathcal{F}$	Method	J_{sim}	R@5	R@10	R@20	R@10	R@20	R@50	R@10	R@20	R@50
0.3	STTran++	0.69	-	-	-	30.7	33.1	33.1	35.9	51.7	64.1
	DSGDetr++	0.69	-	-	-	25.7	28.2	28.2	36.1	50.7	64.0
	SceneSayerODE	0.69	-	-	-	34.9	37.3	37.3	40.5	54.1	63.9
	SceneSayerSDE	0.69	-	-	-	39.7	42.2	42.3	46.9	59.1	65.2
	FORESCENE (r=1)	0.61	73.0	76.3	79.7	36.4	40.1	43.5	42.6	52.5	58.6
	FORESCENE (r=5)	0.67	77.6	80.5	83.2	42.7	46.4	49.7	50.6	60.4	66.2
	FORESCENE (r=10)	0.68	**78.7**	**81.4**	**83.8**	**44.3**	**47.9**	**51.2**	**52.6**	**62.2**	**67.8**
0.5	STTran++	0.73	-	-	-	35.6	38.1	38.1	40.3	58.4	72.2
	DSGDetr++	0.73	-	-	-	29.3	31.9	32.0	40.3	56.9	72.0
	SceneSayerODE	0.73	-	-	-	40.7	43.4	43.4	47.0	62.2	72.4
	SceneSayerSDE	0.73	-	-	-	45.0	47.7	47.7	52.5	66.4	73.5
	FORESCENE (r=1)	0.65	77.6	81.0	83.7	40.3	44.4	47.9	46.8	58.2	65.3
	FORESCENE (r=5)	0.70	82.5	85.2	87.3	47.3	51.2	54.6	55.6	66.6	73.3
	FORESCENE (r=10)	0.72	**83.5**	**86.1**	**88.0**	**49.1**	**52.9**	**56.2**	**58.0**	**68.5**	**74.9**
0.7	STTran++	0.81	-	-	-	41.3	43.6	43.6	48.2	68.8	82.0
	DSGDetr++	0.81	-	-	-	33.9	36.3	36.3	48.0	66.7	81.9
	SceneSayerODE	0.81	-	-	-	49.1	51.6	51.6	58.0	74.0	82.8
	SceneSayerSDE	0.81	-	-	-	52.0	54.5	54.5	61.8	76.7	83.4
	FORESCENE (r=1)	0.72	84.1	86.9	89.0	47.9	52.0	55.5	55.5	68.0	75.2
	FORESCENE (r=5)	0.78	88.5	90.5	92.0	55.0	58.7	62.2	65.2	76.3	82.4
	FORESCENE (r=10)	0.79	**89.4**	**91.4**	**92.7**	**56.7**	**60.4**	**63.8**	**67.7**	**78.1**	**84.0**
0.9	STTran++	0.89	-	-	-	46.0	47.7	47.7	60.2	81.5	92.3
	DSGDetr++	0.89	-	-	-	38.1	39.8	39.8	58.8	78.8	92.2
	SceneSayerODE	0.89	-	-	-	58.1	59.8	59.8	72.6	86.7	93.2
	SceneSayerSDE	0.89	-	-	-	60.3	61.9	61.9	74.8	88.0	**93.5**
	FORESCENE (r=1)	0.82	91.3	93.2	94.3	55.7	59.6	63.8	67.4	80.1	86.5
	FORESCENE (r=5)	0.87	94.7	96.0	96.8	63.4	66.9	70.7	77.5	87.0	92.2
	FORESCENE (r=10)	0.87	**95.1**	**96.5**	**97.1**	**64.5**	**68.0**	**71.7**	**79.7**	**88.1**	92.9

Table 2. SGA results in PGAGS setting. Values in gray assume object continuity. Top results are bold, second-best are underlined.

		Objects Discovery				Triplets With Constraint			Triplets No Constraint		
$\mathcal{F}$	Method	J_{sim}	R@5	R@10	R@20	R@10	R@20	R@50	R@10	R@20	R@50
0.3	STTran++	0.55	-	-	-	22.1	22.8	22.8	28.1	39.0	45.2
	DSGDetr++	0.55	-	-	-	18.2	18.8	18.8	27.7	39.2	47.3
	SceneSayerODE	0.55	-	-	-	27.0	27.9	27.9	33.0	40.9	46.5
	SceneSayerSDE	0.55	-	-	-	28.8	29.9	29.9	34.6	42.0	46.2
	FORESCENE (r=1)	0.48	62.1	66.8	71.7	24.7	27.6	31.6	29.9	34.8	39.1
	FORESCENE (r=5)	0.55	68.0	71.9	76.0	31.6	34.5	38.5	39.4	44.7	48.7
	FORESCENE (r=10)	**0.57**	**69.8**	**73.5**	**77.4**	**33.9**	**36.8**	**40.8**	**42.4**	**47.8**	**51.8**
0.5	STTran++	0.58	-	-	-	24.5	25.2	25.2	30.6	43.2	50.2
	DSGDetr++	0.58	-	-	-	20.7	21.4	21.4	30.4	44.0	52.7
	SceneSayerODE	0.58	-	-	-	30.5	31.5	31.5	36.8	45.9	51.8
	SceneSayerSDE	0.58	-	-	-	32.2	33.3	33.3	38.4	46.9	51.8
	FORESCENE (r=1)	0.51	65.6	69.5	74.1	28.0	30.9	34.6	33.6	39.6	44.2
	FORESCENE (r=5)	0.57	71.2	74.9	78.4	34.9	37.9	41.6	42.9	48.9	53.6
	FORESCENE (r=10)	**0.60**	**72.9**	**76.3**	**79.8**	**37.0**	**40.0**	**43.9**	**46.1**	**52.2**	**56.5**
0.7	STTran++	0.63	-	-	-	29.1	29.7	29.7	36.8	51.6	58.7
	DSGDetr++	0.63	-	-	-	24.6	25.2	25.2	36.7	51.8	60.6
	SceneSayerODE	0.63	-	-	-	36.5	37.3	37.3	44.6	54.4	60.3
	SceneSayerSDE	0.63	-	-	-	37.6	38.5	38.5	45.6	54.6	59.3
	FORESCENE (r=1)	0.55	69.5	73.7	77.6	32.1	35.1	39.3	38.5	45.2	50.0
	FORESCENE (r=5)	0.62	75.3	78.7	81.9	39.7	42.7	46.7	49.4	55.4	60.1
	FORESCENE (r=10)	**0.64**	**77.1**	**80.4**	**83.2**	**41.9**	**45.0**	**49.1**	**52.9**	**58.8**	**63.0**
0.9	STTran++	0.70	-	-	-	31.1	31.6	31.6	43.5	57.6	63.9
	DSGDetr++	0.70	-	-	-	27.6	28.1	28.1	45.8	61.5	68.5
	SceneSayerODE	0.70	-	-	-	41.6	42.2	42.2	52.7	61.8	66.5
	SceneSayerSDE	0.70	-	-	-	42.5	43.1	43.1	53.8	62.4	66.2
	FORESCENE (r=1)	0.62	75.2	78.8	82.1	38.2	41.5	46.0	46.6	52.8	58.2
	FORESCENE (r=5)	0.68	80.4	83.4	86.1	46.1	49.2	53.7	58.2	63.3	67.6
	FORESCENE (r=10)	**0.70**	**81.9**	**84.6**	**87.2**	**48.4**	**51.4**	**55.9**	**61.4**	**66.2**	**70.2**

of the video is observed ($\mathcal{F} = 0.9$). In terms of Triplet Recall our method is more accurate, outperforming existing solutions. Tab. 2 presents results for the more challenging PGAGS setting, which relaxes the dependence on object annotations for the observed frames. Here, the object continuity assumption proves less effective, as object categories are less reliable coming from model predictions rather than ground truth annotations. Consequently, the ability to forecast objects helps to correct potential recognition errors according to the activity context and improves overall performance, leading our method to outperform all baselines.

5. Conclusion

We present FORESCENE, an approach to tackle Scene Graph Anticipation in realistic scenarios by explicitly modeling the joint temporal evolution of objects and relations over time. Preliminary results highlight the importance of such jointly modeling. As shown in our experiments, FORESCENE consistently outperforms existing baselines, demonstrating its effectiveness in predicting realistic human-object interactions.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

References

[1] Nicolas Carion et al. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[2] Yuren Cong et al. Spatial-temporal transformer for dynamic scene graph generation. In *ICCV*, 2021. 2

[3] Partha Ghosh et al. From variational to deterministic autoencoders. In *ICLR*, 2020. 2

[4] Kristen Grauman et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2

[6] Jinbae Im et al. EGTR: Extracting graph from transformer for scene graph generation. In *CVPR*, 2024. 2

[7] Jingwei Ji et al. Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, 2020. 1, 2, 3

[8] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 2

[9] Bolin Lai et al. Lego: Learning egocentric action frame generation via visual instruction tuning. In *ECCV*, 2024. 1

[10] Rohith Peddi et al. Towards scene graph anticipation. In *ECCV*, 2024. 1, 2, 3

[11] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 3

[12] S. Ren et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 3

[13] Jascha Sohl-Dickstein et al. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2

[14] A. Vaswani et al. Attention is all you need. In *NeurIPS*, 2017. 2