DIFFVSGG: Diffusion-Driven Online Video Scene Graph Generation
Supplemental Material

The appendix is **structured** as follows:

- §1 supplements additional implementation details.
- §2 analyzes additional quantitative results.
- §3 provides pseudo codes of major components of DIFFVSGG.
- §4 shows an additional diagram of the diffusion-based reasoning process.
- §5 presents more visualization results.
- §6 demonstrates failure case analysis.
- §7 boardly discusses the limitations and social impact.

1. More Implementation Details

Conventional denoising diffusion models typically follow a multi-step image-to-noise process, wherein noise is progressively added and subsequently reversed to generate refined outputs. However, these models are computationally intensive. To mitigate this,DIFFVSGG adopts an efficient forward diffusion strategy that decouples the traditional multi-step image-to-noise process into two sub-processes: image-to-zero and zero-to-noise. The projector heads are implemented as two 1×1 convolutional layers with ReLU as the activation function.

2. Additional Quantitative Analysis

Additional Backbone. To validate the generalization capability of DIFFVSGG, we also conducted experiments using **ResNet-50 + DETR** as an alternative backbone. The results, shown in Table 2, demonstrate that DIFFVSGG achieves state-of-the-art performance in SGDET, SGCLS and PredCLS, particularly in PredCLS, where it surpasses OED [14] with a notable improvement of 1.8%, 1.4%, and 1.4% in **R@10**, **R@20**, and **R@50** under **with constraint** setting. In addition, under **without constraint** setting, DIFFVSGG still achieves competitive performance in terms of both mR and R metrics. In SGDET, the performance improvement of **R@10**, **R@20**, and **R@50** are 1.9%, 1.1%, and 1.3%, respectively. These improvements verify the significant efficacy of DIFFVSGG in leveraging its diffusion-based reasoning mechanism to capture object relationships and temporal dynamics across frames.

Additional Dataset. To further assess the adaptability of DIFFVSGG, we conducted experiments on the **ImageNet-VidVRD** dataset as an additional benchmark. Compared to AG [15] which mostly contains human-related interaction at indoor scenes, **ImageNet-VidVRD** [17] focuses on a wider range of relations not limited to human-centric interactions, where the average number of relations and objects is 9.7 and 2.5 in each frame respectively. Videos in VidVRD are selected with the criteria of whether they have clear visual relations, containing 35 object categories and 132 relation categories, respectively. Following the standard evaluation task settings and metrics [18–20], we utilize relation tagging (RelTag) and relation detection (RelDet) to evaluate the performance of DIFFVSGG:

- **RelTag** is the task to find all object categories and existing relations. It needs to determine whether the top-K classification results of triplets occur within the videos without considering the localization of objects and relations. We employ top-K precision (P@k , where k = 1, 5, 10) as the evaluation metric.

- **RelDet** is a more comprehensive task for both evaluating object categories, trajectories and existing relations. Same as AG, Recalls and mAP of relation detection are used to evaluate our model. The threshold for viewing a predicted box as a hit is 0.5.

The results, presented in Table 1, compare DIFFVSGG with previous methods on the **ImageNet-VidVRD** dataset. Compared to traditional graph-based methods such as GCN [21] and STGC [22], DIFFVSGG achieves over a +10% improvement, demonstrating its superiority in capturing contextualized information. Recent efforts such as BIG [18] and HCM [23] highlight the importance of temporal reasoning in video relation detection. Comparisons with these methods further validate the effectiveness of DIFFVSGG in spatial-temporal reasoning modeling via denoising diffusion process. Specifically, in **RelDet** task, compared to the previous best method, HCM [23], DIFFVSGG improves mAP by +0.47%, R@50 by +0.13%, and R@100 by +0.06%. These improvements highlight that DIFFVSGG is more effective in detecting and tracking relations. Similar trend is observed in the relation classification setting, **RelTag**, where DIFFVSGG

Method	Relation Detection			Relation Tagging		
	mAP	R@50	R@100	P@1	P@5	P@10
VidVRD [17]	8.58	5.54	6.37	43.00	28.90	20.80
GSTEG [24]	9.52	7.05	8.67	51.50	39.50	28.23
MHRA [25]	13.27	6.82	7.39	41.00	28.70	20.95
GCN [21]	14.27	7.43	8.75	59.50	40.50	27.58
STGC [22]	18.38	11.21	13.69	60.00	43.10	32.24
Fabric [20]	19.23	12.74	16.19	57.50	43.40	31.90
VidVRD-II [19]	23.85	9.74	10.86	73.00	53.20	39.75
BIG [18]	26.08	14.10	16.25	73.00	55.10	40.00
HCM [23]	29.68	17.97	21.45	78.50	57.40	43.55
Ours	**30.15**	**18.10**	**21.51**	**79.95**	**58.80**	**43.71**

Table 1. Comparison of state-of-the-art VRD methods on ImageNet-VidVRD [17].

surpasses HCM by +1.45% in P@1, +1.40% in P@5, and +0.16% in P@10.

3. Pseudo Code

We provide pytorch-style pseudo code of the proposed Graph Construction strategy in Algorithm 1 and Conditional Temporal Reasoning in Algorithm 2.

4. Additional Diagram

We provide an additional diagram in Fig. 1 to illustrate the core concept of iterative reasoning in DIFFVSGG. This diagram highlights the step-by-step reasoning process for scene graph generation, where the graph is progressively updated and refined through a series of iterative steps that integrate spatiotemporal cues.

5. Further Qualitative Results

In this section, we provide more qualitative comparison with existing method DSG-DETR [13] on Action Genome `test` [15]. It could be observed in Fig. 2/ Fig. 4 that DIFFVSGG performs better in distinguish difference between hard relationship such as spatial relation: "in front of" *vs* "behind", and contact relation:"lying on" *vs* "sitting on".

6. Failure Case Analysis

Due to the long-tail distribution of visual relationships in Action Genome [15], the model struggles to accurately capture tail classes in the text, leading to biased scene graph generation. We summarize the most representative failure cases in Fig. 3. As observed, the contact category "writing" is misclassified as "touching."

7. Discussion

Limitations. Although DIFFVSGG has demonstrated remarkable performance, it has some limitations, particularly

Algorithm 1 Pseudo-code of Graph Construction in a PyTorch-like style.

```
"""
I: input video sequence with frames I^1, I^2, ...,
    I^t.
F_det: pretrained object detector.
N_max: Max number of objects in a frame
"""

# Detect
# B_t: set of bounding boxes detected in frame t.
# b_i : bounding box for object i.
# b_j : bounding box for object j
# O_t: set of class predictions for objects in
    frame t.
# F_t: feature map extracted from frame t.
F_t, b_i, b_j, O_t = F_det((I^t))

# Perform graph construction for each video frame
def construct_adjacency_matrix(F_t, b_i, b_j, O_t):

  N_max = 100
  D_embedding = 128
  # extract instance-level feature for object i
  F_o_i = roi_align(F_t, b_i)
  # extract union feature from union box of i and
      j
  F_union = roi_align(F_t, torch.cat([b_i, b_j],
      dim=1))
  # F_b_i box-to-feature mapping for bounding box
      of object j
  F_b_i = box_to_map(b_j)

  return torch.cat([F_o_i, F_union, F_b_j], dim
      =-1)

# Initialize adjacency matrix
N_t = B_t.shape[0]
A_t = torch.zeros(N_max, N_max, D_embedding)

for i in range(N_t):
   for j in range(N_t):
      if i != j:
         At[i, j] = construct_adjacency_matrix(F_t,
             b_i, b_j, O_t):

# Pad to fixed size
if N_t < N_max:
  padding = torch.randn(N_max-N_t, N_max,
      D_embedding)

  A_t[N_t:, :, :] = padding
  A_t[:, N_t:, :] = torch.transpose(padding, 0, 1)

  return A_t
```

its reliance on a multi-step denoising process. Achieving high-quality outputs requires iteratively refining the predictions over numerous steps, which can be time-consuming. Drawing inspiration from recent advancements in step-reduction techniques [26, 27], a potential future improvement would be to incorporate these methods to reduce the number of required steps and accelerate inference. Another limitation of DIFFVSGG is its dependence on the quality and distribution of training , which is also a common challenge for other VSGG methods. Biased predicate sample distributions within the dataset can lead to spurious correlations between input object pairs and predicate labels, negatively impacting the model's accuracy, especially for long-tail cate-

Method	PredCLS						SGCLS						SGDET					
	R@10	R@20	R@50	mR@10	mR@20	mR@50	R@10	R@20	R@50	mR@10	mR@20	mR@50	R@10	R@20	R@50	mR@10	mR@20	mR@50
ResNet-101+Faster-RCNN																		
RelDN [1]	20.3	20.3	20.3	6.2	6.2	6.2	11.0	11.0	11.0	3.4	3.4	3.4	9.1	9.1	9.1	3.3	3.3	3.3
TRACE [2]	27.5	27.5	27.5	15.2	15.2	15.2	14.8	14.8	14.8	8.9	8.9	8.9	13.9	14.5	14.5	8.2	8.2	8.2
VRD [3]	51.7	54.7	54.7	-	-	-	32.4	33.3	33.3	-	-	-	19.2	24.5	26.0	-	-	-
Motif Freq [4]	62.4	65.1	65.1	-	-	-	40.8	41.9	41.9	-	-	-	23.7	31.4	33.3	-	-	-
MSDN [5]	65.5	68.5	68.5	-	-	-	43.9	45.1	45.1	-	-	-	24.1	32.4	34.5	-	-	-
VCTREE [6]	66.0	69.3	69.3	-	-	-	44.1	45.3	45.3	-	-	-	24.4	32.6	34.7	-	-	-
GPS-Net [7]	66.8	69.9	69.9	-	-	-	45.3	46.5	46.5	-	-	-	24.7	33.1	35.1	-	-	-
STTran [8]	68.6	71.8	71.8	37.8	40.1	40.2	46.4	47.5	47.5	27.2	28.0	28.0	25.2	34.1	37.0	16.6	20.8	22.2
APT [9]	69.4	73.8	73.8	-	-	-	47.2	48.9	48.9	-	-	-	26.3	36.1	38.3	-	-	-
STTran-TPI [10]	69.7	72.6	72.6	37.3	40.6	40.6	47.2	48.3	48.3	28.3	29.3	29.3	26.2	34.6	37.4	15.6	20.2	21.8
TR2 [11]	70.9	73.8	73.8	-	-	-	47.7	48.7	48.7	-	-	-	26.8	35.5	38.3	-	-	-
TEMPURA [12]	68.8	71.5	71.5	42.9	46.3	46.3	47.2	48.3	48.3	34.0	35.2	35.2	28.1	33.4	34.9	18.5	22.6	23.7
DSG-DETR [13]	-	-	-	-	-	-	50.8	52.0	52.0	-	-	-	30.3	34.8	36.1	-	-	-
DIFFVSGG	71.9	74.5	74.5	48.1	50.2	50.2	52.5	53.7	53.7	37.3	38.4	38.4	32.8	39.9	45.5	20.9	23.6	26.2
ResNet-50+DETR																		
OED [14]	73.0	76.1	76.1	-	-	-	-	-	-	-	-	-	33.5	40.9	48.9	-	-	-
DIFFVSGG	**74.8**	**77.5**	**77.5**	**53.3**	**56.1**	**56.1**	**54.0**	**54.9**	**54.9**	**40.7**	**42.5**	**42.5**	**34.7**	**41.9**	**47.3**	**24.7**	**27.3**	**28.4**

Table 2. Comparison of state-of-the-art VSGG methods on Action Genome `test` [15] under the ***w* constraint** setting.

Method	PredCLS						SGCLS						SGDET					
	R@10	R@20	R@50	mR@10	mR@20	mR@50	R@10	R@20	R@50	mR@10	mR@20	mR@50	R@10	R@20	R@50	mR@10	mR@20	mR@50
ResNet-101+Faster-RCNN																		
RelDN [1]	44.2	75.4	89.2	31.2	63.1	75.5	25.0	41.9	47.9	18.6	36.9	42.6	13.6	23.0	36.6	7.5	18.8	33.7
VRD [3]	51.7	54.7	54.7	-	-	-	32.4	33.3	33.3	-	-	-	19.2	24.5	26.0	-	-	-
Motif Freq [4]	62.4	65.1	65.1	-	-	-	40.8	41.9	41.9	-	-	-	23.7	31.4	33.3	-	-	-
MSDN [5]	65.5	68.5	68.5	-	-	-	43.9	45.1	45.1	-	-	-	24.1	32.4	34.5	-	-	-
VCTREE [6]	66.0	69.3	69.3	-	-	-	44.1	45.3	45.3	-	-	-	24.4	32.6	34.7	-	-	-
TRACE [2]	72.6	91.6	96.4	50.9	73.6	82.7	37.1	46.7	50.5	31.9	42.7	46.3	26.5	35.6	45.3	22.8	31.3	41.8
GPS-Net [7]	76.0	93.6	99.5	-	-	-	-	-	-	-	-	-	24.5	35.7	47.3	-	-	-
STTran [8]	77.9	94.2	99.1	51.4	67.7	82.7	54.0	63.7	66.4	40.7	50.1	58.5	24.6	36.2	48.8	20.9	29.7	39.2
APT [9]	78.5	95.1	99.2	-	-	-	55.1	65.1	68.7	-	-	-	25.7	37.9	50.1	-	-	-
TR2 [11]	83.1	96.6	**99.9**	-	-	-	57.2	64.4	66.2	-	-	-	27.8	39.2	50.0	-	-	-
TEMPURA [12]	80.4	94.2	99.4	61.5	85.1	98.0	56.3	64.7	67.9	48.3	61.1	66.4	29.8	38.1	46.4	24.7	33.9	43.7
DSG-DETR [13]	-	-	-	-	-	-	59.2	69.1	72.4	-	-	-	32.1	40.9	48.3	-	-	-
DIFFVSGG	83.1	94.5	99.1	66.3	90.5	98.4	60.5	70.5	74.4	51.0	64.2	68.8	35.4	42.5	51.0	27.2	37.0	45.6
ResNet-50+DETR																		
TPT [16]	**85.6**	**97.4**	**99.9**	-	-	-	-	-	-	-	-	-	32.0	39.6	51.5	-	-	-
OED [14]	83.3	95.3	99.2	-	-	-	-	-	-	-	-	-	35.3	44.0	51.8	-	-	-
DIFFVSGG	84.5	95.9	99.5	**67.9**	**91.6**	**98.9**	**62.3**	**71.8**	**75.9**	**52.7**	**65.1**	**69.5**	**37.4**	**45.1**	**53.1**	**29.7**	**37.9**	**46.4**

Table 3. Comparison of state-of-the-art VSGG methods on Action Genome `test` [15] under the ***w/o* constraint** setting.

gories. Fig. 3 illustrates several failure cases where biased scene graphs are generated due to the long-tailed distribution of predicates in the Action Genome dataset [15]. We aim to address this limitation by introducing additional training strategy to debias the predicate learning in our future work

Broader Impact. Currently, we have only demonstrated the effectiveness of the denoising diffusion model in the VSGG task. However, the temporal reasoning capabilities across frames via diffusion offer valuable insights for designing task-specific condition prompting in related vision tasks. Integrating more informative cues from preceding frames could be an effective starting point for improving current frame predictions. The proposed continuous temporal reasoning approach could potentially be extended to tasks such as Action Recognition (AR), Video Event Detection (VED), Video-based Human-Object Interaction (V-HOI), and Multi-Object Tracking (MOT).

On the negative side, it is important to acknowledge the risks associated with DIFFVSGG regarding the generation of false content and data bias. The generative nature of the model during training poses the risk of creating false information about individuals, potentially damaging their reputation and privacy, and even leading to legal and ethical challenges. Furthermore, if the dataset used for training contains biases or imbalances, such as underrepresentation of certain races, genders, or social groups, the model's video analysis could exacerbate existing prejudices and injustices, resulting in biased and unfair decisions in real-world applications. For example, in security surveillance or crowd analysis, this bias could lead to certain groups being disproportionately monitored or wrongly accused, while others remain under-identified, ultimately affecting social equity and public trust in the technology.

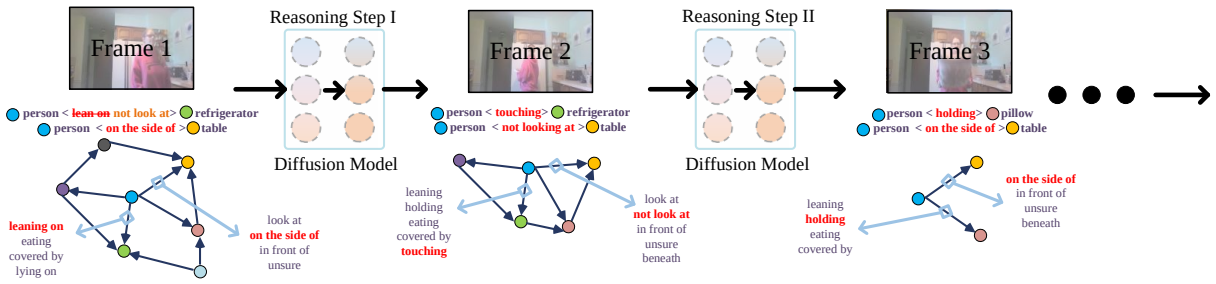

Figure 1. A diagram illustrating step-by-step reasoning process of DIFFVSGG.

Algorithm 2 Pseudo-code of Temporal Reasoning in a PyTorch-like style.

```
"""
A_t_k: Noisy adjacency matrix at step k for frame
    t.
A_t_prev_0: Denoised adjacency matrix from the
    previous frame t-1.
A_t_k_minus_1: adjacency matrix at the end of the
    k step in the denoising process for frame t
beta_t: Noise scheduling parameter for step k.
alpha_t: Alpha value for step k.
epsilon_theta: Denoising model.
epsilon_pred: The noise predicted by the denoising
     model.
"""

# Perform temporal reasoning for each denoising
    step
def temporal_reasoning_step(A_t_k, A_t_prev_0,
    beta_t, alpha_t, epsilon_theta, k):

   # Predict noise with temporal conditioning
   epsilon_pred = epsilon_theta(A_t_k, k,
       A_t_prev_0)

   # Compute reverse denoising step
   A_t_k_minus_1 = (1 / torch.sqrt(alpha_t)) * (
       A_t_k - (beta_t / torch.sqrt(1 - alpha_t))
       * epsilon_pred)

   return A_t_k_minus_1

# Iterate through denoising steps with temporal
    conditioning
def temporal_reasoning_process(A_t_k, A_t_prev_0,
    beta_schedule, alpha_schedule, epsilon_theta,
    K):

      # Iterate from step K to 1
   for k in range(K, 0, -1):
      A_t_k = temporal_reasoning_step(A_t_k,
          A_t_prev_0, beta_schedule[k-1],
          alpha_schedule[k-1], epsilon_theta, k)

   return A_t_k
```

Figure 2. More **visual comparison** with [13] in different time steps.

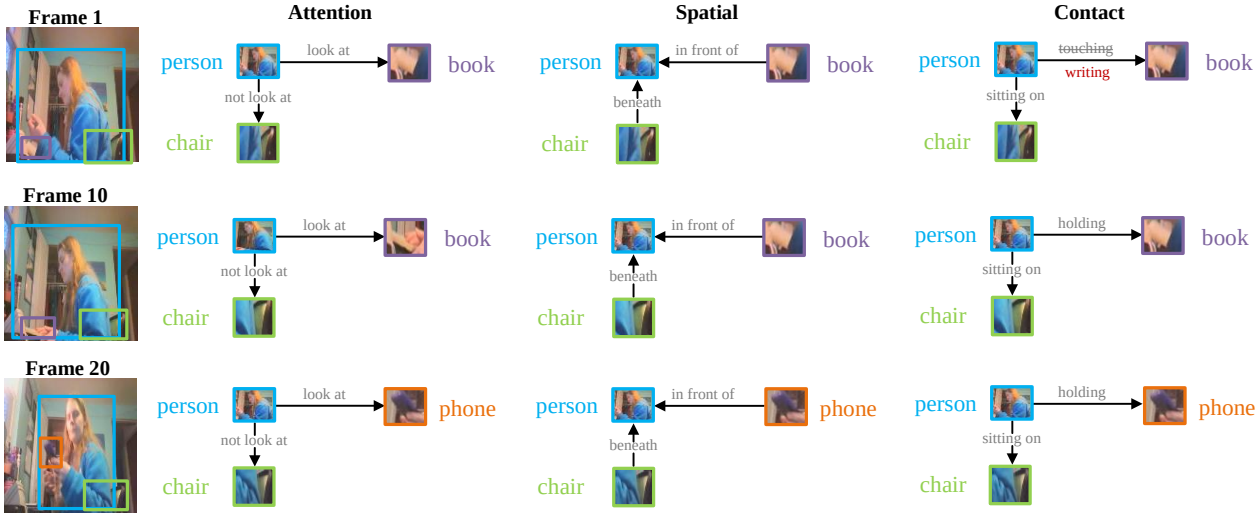

Figure 3. **Failure case** due to dataset bias issue.

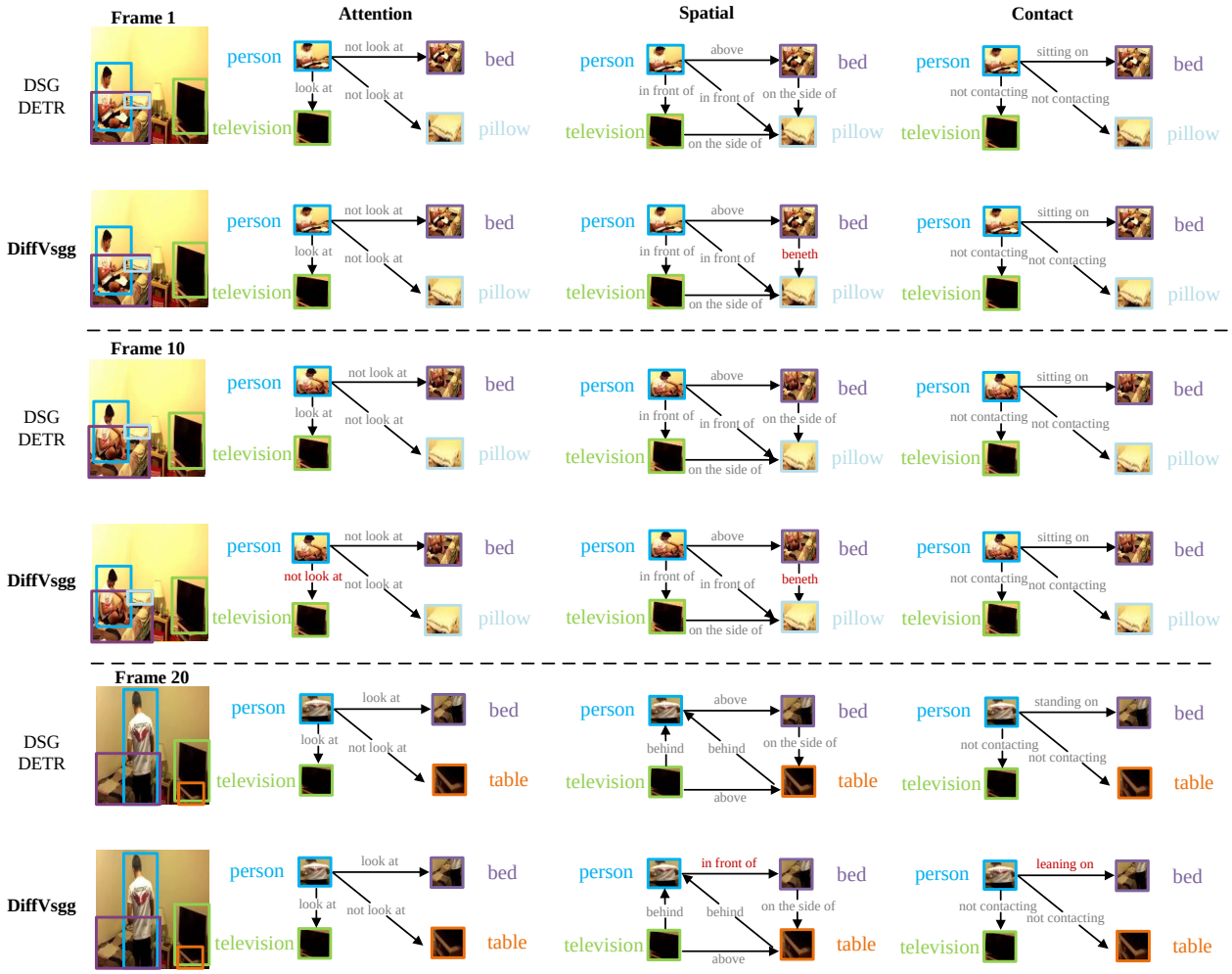

Figure 4. More **visual comparison** with [13] in different time steps.

References

[1] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019. 3

[2] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target adaptive context aggregation for video scene graph generation. In *ICCV*, 2021. 3

[3] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. 3

[4] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 3

[5] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *CVPR*, 2017. 3

[6] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019. 3

[7] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, 2020. 3

[8] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *ICCV*, 2021. 3

[9] Yiming Li, Xiaoshan Yang, and Changsheng Xu. Dynamic scene graph generation via anticipatory pre-training. In *CVPR*, 2022. 3

[10] Shuang Wang, Lianli Gao, Xinyu Lyu, Yuyu Guo, Pengpeng Zeng, and Jingkuan Song. Dynamic scene graph generation via temporal prior inference. In *ACM MM*, 2022. 3

[11] Jingyi Wang, Jinfa Huang, Can Zhang, and Zhidong Deng. Cross-modality time-variant relation learning for generating dynamic scene graphs. *arXiv preprint arXiv:2305.08522*, 2023. 3

[12] Sayak Nag, Kyle Min, Subarna Tripathi, and Amit K Roy-Chowdhury. Unbiased scene graph generation in videos. In *CVPR*, 2023. 3

[13] Shengyu Feng, Hesham Mostafa, Marcel Nassar, Somdeb Majumdar, and Subarna Tripathi. Exploiting long-term dependencies for generating dynamic scene graphs. In *WACV*, 2023. 2, 3, 5, 6

[14] Guan Wang, Zhimin Li, Qingchao Chen, and Yang Liu. Oed: Towards one-stage end-to-end dynamic scene graph generation. In *CVPR*, 2024. 1, 3

[15] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, 2020. 1, 2, 3

[16] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. End-to-end video scene graph generation with temporal propagation transformer. *IEEE Transactions on Multimedia*, 2023. 3

[17] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *ACM MM*, 2017. 1, 2

[18] Kaifeng Gao, Long Chen, Yulei Niu, Jian Shao, and Jun Xiao. Classification-then-grounding: Reformulating video scene graphs as temporal bipartite graphs. In *CVPR*, 2022. 1, 2

[19] Xindi Shang, Yicong Li, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Video visual relation detection via iterative inference. In *ACM MM*, 2021. 2

[20] Shuo Chen, Zenglin Shi, Pascal Mettes, and Cees GM Snoek. Social fabric: Tubelet compositions for video relation detection. In *ICCV*, 2021. 1, 2

[21] Xufeng Qian, Yueting Zhuang, Yimeng Li, Shaoning Xiao, Shiliang Pu, and Jun Xiao. Video relation detection with spatio-temporal graph. In *ACM MM*, 2019. 1, 2

[22] Chenchen Liu, Yang Jin, Kehan Xu, Guoqiang Gong, and Yadong Mu. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In *CVPR*, 2020. 1, 2

[23] Meng Wei, Long Chen, Wei Ji, Xiaoyu Yue, and Roger Zimmermann. In defense of clip-based video relation detection. *IEEE Transactions on Image Processing*, 2024. 1, 2

[24] Yao-Hung Hubert Tsai, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *CVPR*, 2019. 2

[25] Donglin Di, Xindi Shang, Weinan Zhang, Xun Yang, and Tat-Seng Chua. Multiple hypothesis video relation detection. In *BigMM*, 2019. 2

[26] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *CVPR*, 2024. 2

[27] Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024. 2