Unbiased Video Scene Graph Generation via Visual and Semantic Dual Debiasing

Supplementary Material

A. Extended Related Work

A.1. Scene Graph Generation

Scene Graph Generation (SGG) approaches focus on static image-based SGG (ImgSGG) and video-based SGG (VidSGG). For ImgSGG, sequential encoders like LSTM [23] and attention mechanisms [19] capture global context. Recent works [3, 7, 15] improve entity and predicate proposals, with SGTR [7] advancing this through a transformer-based architecture for more accurate scene graph generation.

VidSGG extends Scene Graph Generation (SGG) to dynamic contexts by incorporating intra-frame relationships, crucial for capturing temporal dependencies [5]. Recent advancements enhance temporal coherence through improved modeling of temporal dependencies [9], while efficiency is boosted by adaptive structures [18]. Additionally, social context modeling [1] refines the understanding of interactions between multiple agents, further improving scene graph accuracy and robustness.

Despite their success, existing methods ignore unbiasedness. This results in biased SGG with inaccurate and misleading relationships in scenes. Consequently, such biases make models less reliable and less generalizable to real-world applications.

A.2. Unbiased Scene Graph Generation

Unlike conventional SGG, unbiased SGG focuses on reducing biases towards frequent objects and relationships. Key approaches in this area include knowledge distillation [6] and dual-branch architectures [24], which tackle different aspects of bias. For example, LS-KD [6] uses knowledge distillation to address multi-predicate challenges by leveraging a teacher-student framework to improve relationship diversity. Building on this, DHL [24] employs a dual-branch architecture to ensure balanced attention between head and tail classes, preventing the dominance of common predicates. These methods complement each other by addressing bias from different angles, contributing to more accurate and balanced scene graph generation.

For unbiased VidSGG, Transformer-based models [2] and Gaussian Mixture Models [12] have been explored to capture video dynamics better and reduce bias in visual relations. Iterative approaches [14] use conditional variables to improve video relation detection but lack a hierarchical strategy that integrates both visual and semantic representations, as in our method.

Unlike existing approaches that merely apply traditional debiasing strategies to VidSGG, our framework leverages the inherent visual-semantic nature of scene graphs. This enables a solution that is both experimentally validated and theoretically robust, effectively addressing the fundamental challenges in achieving unbiased VidSGG.

B. Additional Details on MGSM

In this section, we derive the variance of the memory representation $\mathbf{M}_i^t$ and establish the upper and lower bounds for the update parameter λ. In our proposed Memory Guided Sequence Modeling (MGSM) module, the memory representation $\mathbf{M}_i^t$ is updated using the following equation:

$$\mathbf{M}_i^{t+1} = (1-\lambda)\mathbf{M}_i^t + \lambda \mathbf{v}_i^t, \tag{1}$$

where λ is the update rate, and $\mathbf{v}_i^t$ represents the feature vector at time step t for object i. The feature vector $\mathbf{v}_i^t$ is modeled as:

$$\mathbf{v}_i^t = \mathbf{v}_i + \boldsymbol{\epsilon}_i^t, \tag{2}$$

with $\boldsymbol{\epsilon}_i^t$ being zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}$, i.e.,

$$\mathbb{E}[\boldsymbol{\epsilon}_i^t] = \mathbf{0}, \quad \mathrm{Cov}[\boldsymbol{\epsilon}_i^t] = \boldsymbol{\Sigma}. \tag{3}$$

We assume that the noise terms are independent across different time steps and objects. To derive the variance of the memory representation $\mathbf{M}_i^t$, we proceed as follows.

Expectation of Memory Representation Taking the expectation of both sides of the update equation (1):

$$\mathbb{E}[\mathbf{M}_i^{t+1}] = (1-\lambda)\mathbb{E}[\mathbf{M}_i^t] + \lambda \mathbb{E}[\mathbf{v}_i^t]. \tag{4}$$

Since $\mathbb{E}[\boldsymbol{\epsilon}_i^t] = \mathbf{0}$ from equation (3), we have:

$$\mathbb{E}[\mathbf{v}_i^t] = \mathbf{v}_i. \tag{5}$$

Assuming steady-state where $\mathbb{E}[\mathbf{M}_i^{t+1}] = \mathbb{E}[\mathbf{M}_i^t] = \mathbf{M}$, equation (4) simplifies to:

$$\mathbf{M} = (1-\lambda)\mathbf{M} + \lambda \mathbf{v}_i \quad \Rightarrow \quad \mathbf{M} = \mathbf{v}_i. \tag{6}$$

Variance of Memory Representation Next, we compute the variance $\mathrm{Var}[\mathbf{M}_i^t]$. Taking the variance of both sides of the update equation (1):

$$\mathrm{Var}[\mathbf{M}_i^{t+1}] = \mathrm{Var}[(1-\lambda)\mathbf{M}_i^t + \lambda \mathbf{v}_i^t]. \tag{7}$$

Since $\mathbf{M}_i^t$ and $\mathbf{v}_i^t$ are independent, the variance propagates as:

$$\mathrm{Var}[\mathbf{M}_i^{t+1}] = (1-\lambda)^2 \,\mathrm{Var}[\mathbf{M}_i^t] + \lambda^2 \,\mathrm{Var}[\mathbf{v}_i^t]. \tag{8}$$

Given that $\mathrm{Var}[\mathbf{v}_i^t] = \mathbf{\Sigma}$ from equation (3), we substitute into equation (8):

$$\mathrm{Var}[\mathbf{M}_i^{t+1}] = (1-\lambda)^2 \,\mathrm{Var}[\mathbf{M}_i^t] + \lambda^2 \mathbf{\Sigma}. \tag{9}$$

Assuming steady-state where $\mathrm{Var}[\mathbf{M}_i^{t+1}] = \mathrm{Var}[\mathbf{M}_i^t] = \mathbf{V}$, we get:

$$\mathbf{V} = (1-\lambda)^2 \mathbf{V} + \lambda^2 \mathbf{\Sigma}. \tag{10}$$

Solving for $\mathbf{V}$:

$$\mathbf{V}\left[1-(1-\lambda)^2\right] = \lambda^2 \mathbf{\Sigma}, \tag{11}$$

$$1-(1-\lambda)^2 = 2\lambda - \lambda^2, \tag{12}$$

$$\mathbf{V}(2\lambda - \lambda^2) = \lambda^2 \mathbf{\Sigma}, \tag{13}$$

$$\mathbf{V} = \frac{\lambda^2 \mathbf{\Sigma}}{2\lambda - \lambda^2} = \frac{\lambda \mathbf{\Sigma}}{2-\lambda}. \tag{14}$$

For small λ, the expression simplifies to:

$$\mathrm{Var}[\mathbf{M}_i^t] = \frac{\lambda \mathbf{\Sigma}}{2-\lambda} \approx \frac{\lambda \mathbf{\Sigma}}{2}. \tag{15}$$

While minimizing the variance of the memory representation is desirable for enhancing stability, the update parameter λ must be carefully selected to balance variance reduction and the model's ability to adapt to new information. Specifically, λ cannot be too small, as excessively small values slow the adaptation to new information and may introduce significant bias.

Bias-Variance Trade-off The total error in the memory representation can be decomposed into bias and variance:

$$\text{Total Error} = \text{Bias}^2 + \text{Variance}. \tag{16}$$

From the variance derivation in equation (15), we have:

$$\mathrm{Var}[\mathbf{M}_i^t] \approx \frac{\lambda \mathbf{\Sigma}}{2}. \tag{17}$$

Next, we analyze the bias introduced by the update mechanism. Assume that the feature vector evolves over time as:

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \boldsymbol{\delta}, \tag{18}$$

where $\boldsymbol{\delta}$ is a constant change vector representing the feature change rate. Substituting equation (18) into the memory update equation (1), we get:

$$\mathbf{M}_i^{t+1} = (1-\lambda)\mathbf{M}_i^t + \lambda(\mathbf{v}_i^t + \boldsymbol{\delta}). \tag{19}$$

Assuming no noise for bias analysis ($\boldsymbol{\epsilon}_i^t = \mathbf{0}$) and iteratively applying the update equation starting from $t = 0$, we can derive the bias over time.

Bias Derivation At $t = 0$, the initial memory is set to the initial feature:

$$\mathbf{M}_i^0 = \mathbf{v}_i^0 = \mathbf{v}_i. \tag{20}$$

For $t \geq 0$, the update equation without noise becomes:

$$\mathbf{M}_i^{t+1} = (1-\lambda)\mathbf{M}_i^t + \lambda(\mathbf{v}_i^t + \boldsymbol{\delta}). \tag{21}$$

Substituting the feature evolution from equation (18):

$$\mathbf{v}_i^t = \mathbf{v}_i^{t-1} + \boldsymbol{\delta} = \mathbf{v}_i + t\boldsymbol{\delta}. \tag{22}$$

Thus, the update equation becomes:

$$\begin{aligned}\mathbf{M}_i^{t+1} &= (1-\lambda)\mathbf{M}_i^t + \lambda(\mathbf{v}_i + t\boldsymbol{\delta} + \boldsymbol{\delta}) \\ &= (1-\lambda)\mathbf{M}_i^t + \lambda\mathbf{v}_i + \lambda(t+1)\boldsymbol{\delta}.\end{aligned} \tag{23}$$

We can unroll this recurrence relation to find the general expression for $\mathbf{M}_i^t$:

$$\begin{aligned}\mathbf{M}_i^t = (1-\lambda)^t \mathbf{M}_i^0 &+ \lambda \sum_{k=0}^{t-1} (1-\lambda)^k \mathbf{v}_i^{t-k-1} \\ &+ \lambda \sum_{k=0}^{t-1} (1-\lambda)^k \boldsymbol{\delta}.\end{aligned} \tag{24}$$

Since $\mathbf{M}_i^0 = \mathbf{v}_i$, and $\mathbf{v}_i^t = \mathbf{v}_i + t\boldsymbol{\delta}$, the expression simplifies over multiple iterations.

Steady-State Bias As $t \to \infty$, the influence of the initial memory and transient terms diminishes, leading to a steady-state bias. From equation (19), in steady-state, we have:

$$\mathbf{M} = (1-\lambda)\mathbf{M} + \lambda\mathbf{v}_i + \lambda\boldsymbol{\delta}. \tag{25}$$

Solving for $\mathbf{M}$:

$$\lambda\boldsymbol{\delta} = \lambda(\mathbf{v}_i - \mathbf{M}) \quad \Rightarrow \quad \mathbf{M} = \mathbf{v}_i - \boldsymbol{\delta}. \tag{26}$$

Thus, the bias in the memory representation at steady-state is:

$$\text{Bias} = \mathbf{M} - \mathbf{v}_i = -\boldsymbol{\delta}. \tag{27}$$

However, this simplistic analysis overlooks the dynamic nature of $\mathbf{M}_i^t$. A more rigorous approach considers the cumulative effect of λ over time, leading to a residual bias that depends inversely on λ.

Alternative Bias Derivation Assuming that at each time step, the feature vector increases by $\boldsymbol{\delta}$, the memory update equation becomes:

$$\begin{aligned}\mathbf{M}_i^{t+1} &= (1-\lambda)\mathbf{M}_i^t + \lambda(\mathbf{v}_i + t\boldsymbol{\delta} + \boldsymbol{\delta}) \\ &= (1-\lambda)\mathbf{M}_i^t + \lambda\mathbf{v}_i + \lambda(t+1)\boldsymbol{\delta}.\end{aligned} \tag{28}$$

Unfolding this recursion, we find that the bias accumulates over time and converges to:

$$\text{Bias}_\infty = \lim_{t\to\infty} (\mathbf{M}_i^t - \mathbf{v}_i^t) = -\frac{\boldsymbol{\delta}}{\lambda}. \tag{29}$$

This shows that the steady-state bias is inversely proportional to λ.

Lower Bound for λ From equation (29), we observe that:

$$\text{Bias}_\infty = -\frac{\boldsymbol{\delta}}{\lambda}. \tag{30}$$

As λ decreases, the magnitude of Bias_∞ increases. To ensure that the bias remains within acceptable limits, λ must be bounded below by a positive value. Specifically, to maintain $\|\text{Bias}_\infty\| \leq \epsilon$, where ϵ is the maximum tolerable bias, we derive:

$$\left\| -\frac{\boldsymbol{\delta}}{\lambda} \right\| \leq \epsilon \quad \Rightarrow \quad \lambda \geq \frac{\|\boldsymbol{\delta}\|}{\epsilon}. \tag{31}$$

Therefore, the lower bound for λ is:

$$\lambda \geq \frac{\|\boldsymbol{\delta}\|}{\epsilon}. \tag{32}$$

This implies that λ cannot be arbitrarily small, as doing so would result in an unbounded increase in bias, thereby compromising the accuracy and reliability of the memory representation.

Optimal λ To minimize the total error, which comprises both bias and variance, we balance the two components. From equations (30) and (17), the total error is:

$$\text{Total Error} = \|\text{Bias}_\infty\|^2 + \text{Var}[\mathbf{M}_i^t] = \frac{\|\boldsymbol{\delta}\|^2}{\lambda^2} + \frac{\lambda \boldsymbol{\Sigma}}{2}. \tag{33}$$

To find the optimal λ, we take the derivative of the total error with respect to λ and set it to zero:

$$\frac{d}{d\lambda}\left(\frac{\|\boldsymbol{\delta}\|^2}{\lambda^2} + \frac{\lambda \boldsymbol{\Sigma}}{2}\right) = -2\frac{\|\boldsymbol{\delta}\|^2}{\lambda^3} + \frac{\boldsymbol{\Sigma}}{2} = 0. \tag{34}$$

Solving for λ:

$$-2\frac{\|\boldsymbol{\delta}\|^2}{\lambda^3} + \frac{\boldsymbol{\Sigma}}{2} = 0, \tag{35}$$

$$2\frac{\|\boldsymbol{\delta}\|^2}{\lambda^3} = \frac{\boldsymbol{\Sigma}}{2}, \tag{36}$$

$$\lambda^3 = \frac{4\|\boldsymbol{\delta}\|^2}{\boldsymbol{\Sigma}}, \tag{37}$$

$$\lambda = \left(\frac{4\|\boldsymbol{\delta}\|^2}{\boldsymbol{\Sigma}}\right)^{\frac{1}{3}}. \tag{38}$$

Thus, the optimal λ that minimizes the total error is:

$$\lambda_{\text{opt}} = \left(\frac{4\|\boldsymbol{\delta}\|^2}{\boldsymbol{\Sigma}}\right)^{\frac{1}{3}}. \tag{39}$$

B.1. Empirical Validation

In practice, the optimal λ is determined based on the specific values of $\boldsymbol{\delta}$ and $\boldsymbol{\Sigma}$ derived from the data. From the previous calculation, we get the approximate λ value as following:

$$\lambda \approx 0.04, \tag{40}$$

it suggests that:

$$\lambda = \left(\frac{4\|\boldsymbol{\delta}\|^2}{\boldsymbol{\Sigma}}\right)^{\frac{1}{3}} \approx 0.04. \tag{41}$$

This optimal value balances the trade-off between minimizing bias and controlling variance, ensuring robust feature estimation in the MGSM module.

Through the derivation, we establish that the variance of the memory representation decreases with a smaller λ, enhancing stability, while the bias increases inversely with λ, reducing adaptability. The optimal λ of approximately 0.04 in our experiments effectively balances this trade-off, providing both robust and adaptable feature representations essential for mitigating visual bias in video scene graph generation.

C. Additional Metrics and Evaluation Setup

We evaluate our approach using the standard metric for unbiased VidSGG, mean Recall@K (mR@K) with $K \in \{10, 20, 50\}$. TEMPURA [12] serves as our baseline method. Following established protocols [2, 5, 12], we conduct three Scene Graph Generation (SGG) tasks:

Predicate Classification (*PREDCLS*) which delivers object localization and classes, necessitating the model to discern predicate classes. **Scene Graph Classification** (*SGCLS*) furnishes precise localization, expecting the model to identify both object and predicate classes. **Scene Graph Detection** (*SGDET*) requires the model to initially detect bounding boxes before classifying objects and predicate classes. Evaluation is conducted across three distinct settings: **With Constraint**, **Semi Constraint**, and **No Constraint**. Under the With Constraint setting, the generated scene graphs are limited to at most one predicate per subject-object pair. The Semi Constraint setting allows for multiple predicates, yet only those surpassing a specified confidence threshold (=0.9) are considered. Scene graphs can contain multiple predicates between pairs without any restrictions under the No Constraint. It is important to emphasize that another standard metric R@K is susceptible to a bias favoring predominant predicate classes [12, 16], whereas mR@K is averaged across all predicate classes, thus providing an indicator of a model's performance on the unbiased VidSGG. Consequently, for the task of generating unbiased VidSGG, we will afford greater scrutiny to the mR@K metric, as it offers a more balanced assessment of model performance.

D. Additional Implementation Details

Following prior work [2, 8, 12], we adopted Faster R-CNN [13] with ResNet-101 [4] as the object detector, initially trained on the AG dataset. To ensure a fair comparison, we utilized the official implementations of these methods. For our MGSM module, we set the λ parameter to

Table 1. Ablation study of MGSM and IRG under With Constraint.

	With Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	**46.9**	**52.0**	**52.0**	**40.8**	**42.5**	**42.6**	**27.3**	**27.3**	**30.7**
w/o MGSM	46.9	52.0	52.0	35.7	38.2	38.2	18.8	24.6	26.5
w/o MGSM & IRG	42.9	46.3	46.3	34.0	35.2	35.2	18.5	22.6	23.1

Table 2. Ablation study of MGSM and IRG under Semi Constraint.

	Semi Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	**51.3**	**56.3**	**56.4**	**47.8**	**52.6**	**52.6**	**31.7**	**31.7**	**33.2**
w/o MGSM	51.3	56.3	56.4	44.6	48.4	48.6	23.1	27.5	28.5
w/o MGSM & IRG	40.7	44.5	44.6	36.9	39.5	39.5	18.5	21.8	22.5

Table 3. Ablation study of MGSM and IRG under No Constraint.

	No Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	**65.8**	**89.1**	**99.8**	**52.0**	**66.3**	**71.4**	**30.7**	**36.7**	**50.4**
w/o MGSM	65.8	89.1	99.8	49.0	62.0	67.2	27.9	34.7	47.2
w/o MGSM & IRG	61.5	85.1	95.9	48.3	61.1	66.0	24.7	33.9	45.9

Table 4. Ablation study of HSE under With Constraint.

	With Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	**46.9**	**52.0**	**52.0**	**40.8**	**42.5**	**42.6**	**27.3**	**27.3**	**30.7**
w/o HSE	44.5	49.2	49.2	38.4	40.2	40.3	25.8	25.6	28.3

0.04 for the *SGCLS* task and 0.06 for the *SGDET* task. In the IRG module, we implemented a dual-procedure setup, enabling iterative relational inference with the number of iterations N set to 1. The framework was trained end to end for 15 epochs using the AdamW optimizer [10] and a batch size of 1. The initial learning rate was 10^{-5}. We reduced the initial learning rate by 0.5 whenever the performance plateaus. All code ran on a single RTX 4090.

E. Supplementary Experimental Results

E.1. Complete Ablation study on VISA modules

Due to page constraints, the mR@50 results for the ablation study on VISA modules were omitted from the main text. Here, we present the complete ablation study in Tables 1, 2, and 3, demonstrating the effects of the visual and semantic debiasing modules. Consistent with the ablation study in the main body, we first removed the MGSM module. Focusing on the mR@50 results, this removal led to a minimal decrease of **-3.2%** in mR@50 for SGDET under the No Constraint setting, underscoring MGSM's strong visual debiasing capability. Similarly, excluding the IRG module resulted in a minimal decrease of **-1.3%** in mR@50 for SGDET under the No Constraint setting, highlighting IRG's effectiveness in mitigating semantic bias. These findings validate our approach of partitioning scene graph biases into visual and semantic components, demonstrating that our VISA framework effectively mitigates both biases in VidSGG. Notably, MGSM was not applied to the PREDCLS task, as this task relies on ground-truth visual input, rendering the inclusion of MGSM neutral to the results.

Table 5. Ablation study of HSE under Semi Constraint.

	Semi Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	**51.3**	**56.3**	**56.4**	**47.8**	**52.6**	**52.6**	**31.7**	**31.7**	**33.2**
w/o HSE	49.1	54.2	54.6	45.4	50.4	50.4	29.1	29.1	31.5

Table 6. Ablation study of HSE under No Constraint.

	No Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	**65.8**	**89.1**	**99.8**	**52.0**	**66.3**	**71.4**	**30.7**	**36.7**	**50.4**
w/o HSE	63.3	87.0	98.4	50.0	64.0	69.2	27.9	34.7	48.5

Table 7. Influence of increasing iteration count *N* under With Constraint.

	With Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	46.9	52.0	52.0	40.8	42.5	42.6	27.3	27.3	30.7
VISA$_{N=2}$	47.9	53.0	53.0	41.7	43.4	43.4	28.7	28.7	31.9
VISA$_{N=3}$	48.1	53.3	53.3	42.2	44.0	44.0	28.9	28.9	32.5
VISA$_{N=4}$	48.9	53.8	53.8	42.8	**44.5**	**44.5**	29.5	29.5	**33.1**
VISA$_{N=5}$	**50.1**	**53.9**	**53.9**	**43.0**	44.3	44.3	**29.6**	**29.6**	33.0

E.2. Ablation Study of HSE

In this section, we evaluated the effectiveness of the Hierarchical Semantics Extractor (HSE) by replacing it with a simple concatenation method. Specifically, the composite object feature $\mathbf{p}_{j,i}^{t}$ was concatenated with the integrated triplet embeddings $\mathbf{C}_{pre,(j,i)}^{t}$ and fed into the Spatial Encoder. The results of this ablation study were presented in Tables 4, 5, and 6. The results demonstrate that using the concatenation approach led to a decrease of at least **-1.4%** in mR@50 for the PREDCLS task under the No Constraint setting. This reduction was attributed to the simplistic visual-semantic fusion strategy, which failed to effectively integrate fine-grained visual and semantic features. The hierarchical structure of HSE, in contrast, facilitated a more sophisticated fusion process, capturing intricate relationships between visual and semantic information. This enhanced integration was crucial for mitigating biases and improving the accuracy of scene graph generation. The observed performance decline underscored the importance of maintaining hierarchical semantics extraction within the VISA framework to ensure unbiased VidSGG.

E.3. Extended Study on the Influence of Iteration Count *N*

In this section, we investigated how varying the iteration number *N* affects our framework's performance and determine the point at which computing costs outweigh performance gains. We incrementally increased the iteration count until this phenomenon occurs. The results of this analysis were detailed in Tables 7, 8, and 9. Examining the effect of iteration number N on the performance of unbiased VidSGG, measured by mR@K, we observed that higher values of N generally yield improved results. Notably, performance gains plateau at $N = 4$, and by $N = 5$, the computational costs begin to outweigh the benefits, even resulting in a decline in unbiased generation capabilities. We attributed this phenomenon to the limited training datasets, which caused the model's self-correction capabilities to reach a bottleneck. Consequently, future work may explore incorporating large language models (LLMs) to enhance the framework's adaptability and performance further. LLMs' inherent self-correction and language generation capabilities naturally complement this unbiased task. Expanding the training dataset could help overcome the current limitations, allowing for higher iteration counts without incurring prohibitive computational costs.

Table 8. Influence of increasing iteration count *N* under Semi Constraint.

	Semi Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	51.3	56.3	56.4	47.8	52.6	52.6	31.7	31.7	33.2
VISA$_{N=2}$	52.5	56.9	57.0	48.2	53.6	53.6	31.9	31.9	32.7
VISA$_{N=3}$	52.8	57.1	57.2	48.5	53.9	53.9	32.0	32.0	32.9
VISA$_{N=4}$	53.0	57.3	57.3	**48.6**	**54.0**	**54.0**	32.1	32.1	33.0
VISA$_{N=5}$	**53.1**	**57.4**	**57.4**	48.4	54.0	54.0	**32.2**	**32.2**	**33.1**

Table 9. Influence of increasing iteration count *N* under No Constraint.

	No Constraint								
	PREDCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
VISA	65.8	89.1	99.8	52.0	66.3	71.4	30.7	36.7	50.4
VISA$_{N=2}$	66.0	90.8	99.8	53.1	67.0	72.0	31.8	37.2	51.4
VISA$_{N=3}$	66.2	91.2	99.9	53.5	67.2	72.5	32.0	37.5	51.9
VISA$_{N=4}$	66.3	91.3	99.9	**53.7**	**67.5**	**72.7**	32.2	37.7	33.2
VISA$_{N=5}$	**66.3**	**91.4**	**99.6**	53.5	67.3	72.5	**32.3**	**37.8**	**33.3**

F. Failure Cases

To elucidate the limitations inherent in our model, we meticulously analyze the results and identify the most prevalent types of failure cases. We identify and illustrate several typical scenarios where VISA faces challenges, as depicted in Fig. 1. (1) *Undetected Small Objects.* Small objects like cups may be too diminutive for detection by the object detector, leading to the omission of related triplets in VISA's output. (2) *Noisy Annotations and Challenging Scenes in AG.* This includes incorrect annotations, low-resolution videos, and extreme scene conditions. For instance, scenes that are too dim to discern events accurately. (3) *Ambiguity in Object Recognition.* Certain objects are indistinguishable even to human observers, such as differentiating between a person holding a book and food. (4) *Ambiguity in Relationship Interpretation.* Some relationships are also challenging to discern, like determining whether a person is looking at a cup or not.

We speculate that Failure (1), the adoption of a more advanced object detector could potentially offer a solution. Addressing Failure (2) may involve comprehensive data cleansing efforts. As for Failures (3) and (4), which we attribute to the intrinsic constraints of human-labeled annotations, an unsupervised learning approach might present a viable resolution.

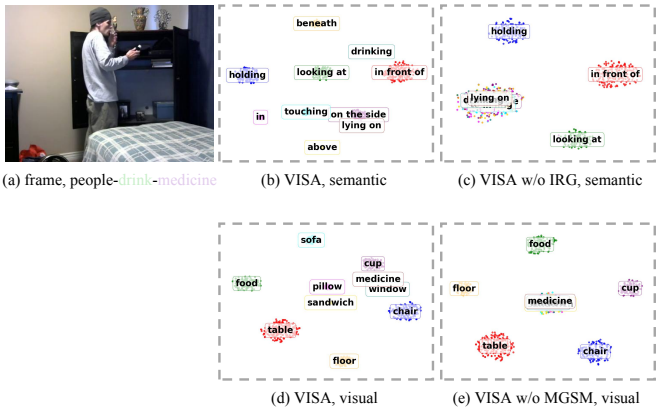
G. Supplementary visualization results

Figure 2 shows our t-SNE results for semantic (b,c) and visual (d,e) features, effectively separating high- and low-frequency classes (e.g., drink, medicine) and mitigating both visual and semantic biases. This figure also offers more results of complex scenes, featuring low-frequency predicates (e.g., *drink*) and nouns (e.g., *medicine*) that are smaller and harder to recognize.

Figure 1. **Prevalent Failure Cases.** (1) Undetected Small Objects (2) Noisy Annotations and Challenging Scenes (3) Ambiguity in Object Recognition (4) Ambiguity in Relationship Interpretation

H. Recall@K explanation

Recall@K(R@K) was used as a standard metric for previous VidSGG method [3, 12]. However, we excluded R@K in the main paper due to reporting bias identified by leading unbiased image-based SGG methods [11, 17, 20]. R@K favors high-frequency representations, a bias overlooked in previous VidSGG methods. The R@K results were obtained under the same experimental settings as in the main



(a) frame, people-drink-medicine (b) VISA, semantic (c) VISA w/o IRG, semantic

(d) VISA, visual (e) VISA w/o MGSM, visual

Figure 2. **Supplementary visualizations.**

paper (Table 10) and are included for completeness, though they are not the primary focus in VidSGG.

Table 10. Quantitative R@K results.

Constraint	Method	PREDCLS			SGCLS			SGDET		
		R@10	R@20	R@50	R@10	R@20	R@50	R@10	R@20	R@50
With	TEMPURA	68.8	71.5	71.5	47.2	48.3	48.3	28.1	33.4	34.9
	FloCoDe	70.1	74.2	74.2	48.4	51.2	51.2	31.5	38.4	42.4
	VISA (Ours)	**70.2**	**74.9**	**75.3**	**49.1**	**51.9**	**52.3**	**32.2**	**39.3**	**43.8**
Semi	TEMPURA	66.9	69.6	69.7	48.3	50.0	50.0	28.1	33.3	34.8
	VISA (Ours)	**70.8**	**76.6**	**76.7**	**56.8**	**61.2**	**61.2**	**35.9**	**40.9**	**42.0**
No	TEMPURA	80.4	94.2	99.4	56.3	64.7	67.9	29.8	38.1	46.4
	FloCoDe	82.8	97.2	**99.9**	57.4	66.2	68.8	32.6	43.9	51.6
	VISA (Ours)	**83.5**	**98.5**	**99.9**	**58.0**	**67.2**	**70.1**	**33.2**	**44.7**	**52.4**

As shown in Table 10, VISA surpasses all previous methods across all R@K metrics in unbiased VidSGG.

I. More details on PVSG and 4DPVSG

We keep the baseline unchanged from the original paper [21, 22]. For PVSG, we adopt a Mask2Former-based method for image panoptic segmentation, then use UniTrack for visual representations, and finally apply a Transformer encoder for relationship prediction. For 4DPVSG, we process 3D video clips with Mask2Former for frame-level panoptic segmentation, link instance embeddings across frames via UniTrack, and employ a Spatial-Temporal Transformer to incorporate temporal context and inter-object interactions.

References

[1] Shizhe Chen, Zhiyuan Shi, Pascal Mettes, and Cees G. M. Snoek. Social fabric: Tubelet compositions for video relation detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021. S1

[2] Yunhan Cong, Wentao Liao, Heiko Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. S1, S3

[3] Yunhan Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023. S1, S6

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. S3

[5] Jiading Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. S1, S3

[6] Lifeng Li, Jing Xiao, Huan Shi, Wei Wang, Jing Shao, Alan Liu, Yueting Yang, and Lin Chen. Label semantic knowledge distillation for unbiased scene graph generation. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2023. S1

[7] Ruofei Li, Shuhui Zhang, and Xiaochun He. Sgtr: End-to-end scene graph generation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. S1

[8] Yifei Li, Xinyi Yang, and Chenglin Xu. Dynamic scene graph generation via anticipatory pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. S3

[9] Chang Liu, Yining Jin, Kaixuan Xu, Guohao Gong, and Yuesheng Mu. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. S1

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. S4

[11] Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. S6

[12] Sourya Nag, Kyusong Min, Swami Tripathi, and Amit K. Roy-Chowdhury. Unbiased scene graph generation in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. S1, S3, S6

[13] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. S3

[14] Xun Shang, Yicheng Li, Jing Xiao, Weiqing Ji, and Tat-Seng Chua. Video visual relation detection via iterative inference. In *Proceedings of the 29th ACM International Conference on Multimedia (ACMMM)*, 2021. S1

[15] Shubham Shit, Raphael Koner, Benedikt Wittmann, Johannes Paetzold, Ivan Ezhov, Haoyang Li, Jindong Pan, Sina Sharifzadeh, Georgios Kaissis, and Volker Tresp. Relationformer: A unified framework for image-to-graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. S1

[16] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wei Luo, and Wei Liu. Learning to compose dynamic tree structures for

visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. S3

[17] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. S6

[18] Yudong Teng, Liang Wang, Zhenyu Li, and Gang Wu. Target adaptive context aggregation for video scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021. S1

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. S1

[20] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *Proceedings of the 28th ACM international conference on multimedia (ACM MM)*, 2020. S6

[21] Jingkang Yang, Jun Cen, Wenxuan Peng, Fangzhou Liu, Shuai amd Hong, Xiangtai Li, Kaiyang Zhou, Qifeng Chen, and Ziwei Liu. 4d panoptic scene graph generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. S7

[22] Jingkang Yang, Wenxuan Peng, Xiangtai Li, Zujin Guo, Liangyu Chen, Bo Li, Zheng Ma, Kaiyang Zhou, Wayne Zhang, Chen Change Loy, et al. Panoptic video scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. S7

[23] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. S1

[24] Cheng Zheng, Linfeng Gao, Xinyi Lyu, Ping Zeng, Abdulmotaleb El Saddik, and Heng Tao Shen. Dual-branch hybrid learning network for unbiased scene graph generation. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2023. S1