Unbiased Scene Graph Generation in Videos

Sayak Nag[1], Kyle Min[2], Subarna Tripathi[2], Amit K. Roy-Chowdhury[1]
[1]University of California, Riverside, USA, [2]Intel Corporation, USA
snag005@ucr.edu, kyle.min@intel.com, subarna.tripathi@intel.com, amitrc@ece.ucr.edu

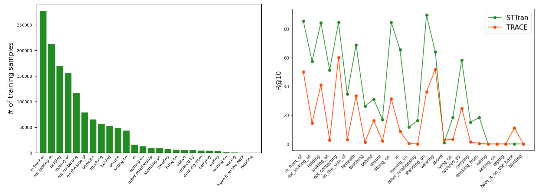
Abstract

The task of dynamic scene graph generation (SGG) from videos is complicated and challenging due to the inherent dynamics of a scene, temporal fluctuation of model predictions, and the long-tailed distribution of the visual relationships in addition to the already existing challenges in image-based SGG. Existing methods for dynamic SGG have primarily focused on capturing spatio-temporal context using complex architectures without addressing the challenges mentioned above, especially the long-tailed distribution of relationships. This often leads to the generation of biased scene graphs. To address these challenges, we introduce a new framework called TEMPURA: TEmporal consistency and Memory Prototype guided UnceRtainty Attenuation for unbiased dynamic SGG. TEMPURA employs object-level temporal consistencies via transformer-based sequence modeling, learns to synthesize unbiased relationship representations using memory-guided training, and attenuates the predictive uncertainty of visual relations using a Gaussian Mixture Model (GMM). Extensive experiments demonstrate that our method achieves significant (up to 10% in some cases) performance gain over existing methods highlighting its superiority in generating more unbiased scene graphs. Code: https://github.com/sayaknag/unbiasedSGG.git

1. Introduction

Scene graphs provide a holistic scene understanding that can bridge the gap between vision and language [25, 31]. This has made image scene graphs very popular for high-level reasoning tasks such as captioning [14, 37], image retrieval [49, 60], human-object interaction (HOI) [40], and visual question answering (VQA) [23]. Although significant strides have been made in scene graph generation (SGG) from static images [12, 31, 33, 37, 39, 42, 56, 62–64], research on dynamic SGG is still in its nascent stage.

Dynamic SGG involves grounding visual relationships jointly in space and time. It is aimed at obtaining a structured representation of a scene in each video frame along with learning the temporal evolution of the relationships between each pair of objects. Such a detailed and structured form of video understanding is akin to how humans perceive real-world activities [4, 25, 43] and with the exponential growth of video data, it is necessary to make similar strides in dynamic SGG.



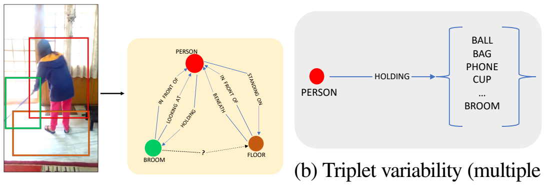
(a) (b)

Figure 1. (a) Long-tailed distribution of the predicate classes in Action Genome [25]. (b) Visual relationship or predicate classification performance of two state-of-the-art dynamic SGG methods, namely STTran [10] and TRACE [57], falls off significantly for the tail classes.

In recent years, a handful of works have attempted to address dynamic SGG [10, 16, 24, 38, 57], with a majority of them leveraging the superior sequence processing capability of transformers [1, 5, 19, 26, 44, 53, 58]. These methods simply focused on designing more complex models to aggregate spatio-temporal contextual information in a video but fail to address the data imbalance of the relationship/predicate classes, and although their performance is encouraging under the Recall@k metric, this metric is biased toward the highly populated classes. An alternative metric was proposed in [7, 56] called mean-Recall@k which quantifies how SGG models perform over all the classes and not just the high-frequent ones.

Fig 1a shows the long-tailed distribution of predicate classes in the benchmark Action Genome [25] dataset and Fig 1b highlights the failure of some existing state-of-the-art methods is classifying the relationships/predicates in the tail of the distribution. The high recall values in prior works suggest that they may have a tendency to overfit on popular predicate classes (e.g. *in front of / not looking at*), without considering how the performances on rare classes (e.g. *eating/wiping*) are getting impacted [12]. Predicates lying in the tails often provide more informative depictions of underlying actions and activities in the video. Thus, it is important to be able to measure a model's long-term performance not only on the frequently occurring relationships but also on the infrequently occurring ones.

Data imbalance is, however, not the only challenge in dynamic SGG. As shown in Fig. 2 and Fig. 3, several other factors, including noisy annotations, motion blur, temporal fluctuations of predictions, and a need to focus on only



(a) Incomplete annotations and multiple correct predicates.

(b) Triplet variability (multiple possible object pairs for the same relationship)

Figure 2. Noisy scene graph annotations in Action Genome [25] increase the uncertainty of predicted scene graphs.

active objects that are involved in an action contribute to the bias in training dynamic SGG models [3]. As a result, the visual relationship predictions have high uncertainty, thereby increasing the challenge of dynamic SGG manyfold.

In this paper, we address these sources of bias in dynamic SGG and propose methods to compensate for them. We identify missing annotations, multi-label mapping, and triplet ($<subject-predicate-object>$) variability (Fig 2) as labeling noise, which coupled with the inherent temporal fluctuations in a video can be attributed as data noise that can be modeled as the *aleatoric uncertainty* [11]. Another form of uncertainty called the *epistemic uncertainty*, relates to misleading model predictions due to a lack of sufficient observations [28] and is more prevalent for long-tailed data [22]. To address the bias in training SGG models [3] and generate more unbiased scene graphs, it is necessary to model and attenuate the predictive uncertainty of an SGG model. While multi-model deep ensembles [13, 32] can be effective, they are computationally expensive for large-scale video understanding. Therefore, we employ the concepts of single model uncertainty based on Mixture Density Networks (MDN) [9, 28, 54] and design the predicate classification head as a Gaussian Mixture Model (GMM) [8, 9]. The GMM-based predicate classification loss penalizes the model if the predictive uncertainty of a sample is high, thereby, attenuating the effects of noisy SGG annotations.

Due to the long-tailed bias of SGG datasets, the predicate embeddings learned by existing dynamic SGG frameworks significantly underfit to the data-poor classes. Since each object pair can have multiple correct predicates (Fig 2), many relationship classes share similar visual characteristics. Exploiting this factor, we propose a memory-guided training strategy to debias the predicate embeddings by facilitating knowledge transfer from the data-rich to the data-poor classes sharing similar characteristics. This approach is inspired by recent advances in meta-learning and memory-guided training for low-shot, and long-tail image recognition [17, 45, 48, 51, 65], whereby a memory bank, composed of a set of prototypical abstractions [51] each compressing information about a predicate class, is designed. We propose a progressive memory computation approach and an attention-based information diffusion strategy [58]. Backpropagating while using this approach, teaches the model to *learn how to generate* more balanced predicate representations generalizable to all the classes.

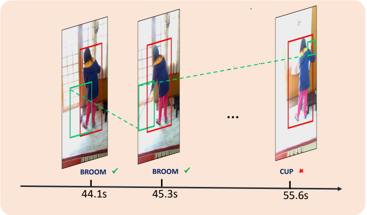


Figure 3. Occlusion and motion blur caused by moving objects in videos renders off-the-self object detectors such as FasterRCNN [47] ineffective in producing consistent object classification.

Finally, to ensure the correctness of a generated graph, accurate classification of both nodes (objects) and edges (predicates) is crucial. While existing dynamic SGG methods focus on innovative visual relation classification [10, 24, 38], object classification is typically based on proposals from off-the-shelf object detectors [47]. These detectors may fail to compensate for dynamic nuances in videos such as motion blur, abrupt background changes, occlusion, etc. leading to inconsistent object classification. While some works use bulky tracking algorithms to address this issue [57], we propose a simpler yet highly effective learning-based approach combining the superior sequence processing capability of transformers [58], with the discriminative power of contrastive learning [18] to ensure more temporally consistent object classification. Therefore, combining the principles of temporally consistent object detection, uncertainty-aware learning, and memory-guided training, we design our framework called **TEMPURA**: **TE**mporal consistency and **M**emory **P**rototype guided **U**nce**R**tainty **A**tentuation for unbiased dynamic SGG. To the best of our knowledge,

2. Related Work

Image Scene Graph Generation. SGG from images aims to obtain a graph-structured summarization of a scene where the nodes are objects, and the edges describe their interaction or relationships (formally called predicates). Since the introduction of the image SGG benchmark Visual Genome (VG) [31], research on SGG from *single images* has evolved significantly, with earlier works addressing image SGG utilizing several ways to aggregate spatial context [37, 39, 42, 62, 64] and latest ones [12, 33–36, 55, 56, 61] addressing fundamental problems such as preventing biased scene graphs caused by long-tailed predicate distribution and noisy annotations in image SGG dataset [31].

Dynamic Scene Graph Generation. Dynamic Scene Graph Generation aims at learning the *spatio-temporal dependencies* of visual relationships between different object pairs over all the frames in a video [25]. Similar to SGG from images, long-tailed bias and noisy annotations pose a significant challenge to dynamic SGG, further compounded by the temporal fluctuations of predictions. In recent years, a handful of works have attempted to address dynamic SGG [10, 24, 29, 38, 57, 59] and benchmarked their methods on Action Genome (AG) [25] dataset. While methods like TRACE [57] introduced temporal context from pretrained 3D models [6], the majority resorted to using the superior sequence processing ability of transformers [1,44,53,58] for spatio-temporal reasoning of visual relations. However, despite their success, the performance gains of these methods are mostly realized for the high-frequency relationships and they fail to address the long-tail bias – the focus of this paper.

Mixture Density Networks. Mixture density networks have been successful in modeling predictive uncertainty and attenuation of noise in many deep-learning tasks. They have been used in many tasks that involve noisy data such as reinforcement learning [9], active learning [8], semantic segmentation [28] and even in compensating for data imbalance in image recognition, [22]. This work is the first to apply such concepts to dynamic SGG.

Memory guided low shot and long-tailed learning. Memory-guided training strategies [48, 52] have become successful in addressing learning with data scarcity such as few-shot learning [15, 27, 51] and long tail recognition [45, 65]. They enable the learning of generalizable representations by transferring knowledge from data-rich to data-poor classes. We exploit these principles in this paper for learning more unbiased representations of visual relationships in videos.

3. Method

3.1. Problem Statement

The goal of dynamic SGG is to describe a structured representation $G_t = \{S_t, R_t, O_t\}$ of each frame I_t in a video $\mathcal{V} = \{I_1, I_2, ..., I_T\}$. Here $S_t = \{s^t_1, s^t_2, ..., s^t_{N(t)}\}$ and $O_t = \{o^t_1, o^t_2, ..., o^t_{N(t)}\}$ map to the same set of $N(t)$ detected objects in the t^{th} frame. They are combinatorially arranged as subject-object pairs (s^t_j, r^t_k, o^t_i) with $R_t = \{r^t_1, r^t_2, ..., r^t_{K(t)}\}$ being the set of $K(t)$ predicates describing the visual relationships between all subject-object pairs in the t^{th} frame. Formally each $<subject-predicate-object>$ or (s^t_j, r^t_k, o^t_i) is called a triplet. The set of object and predicate classes are referred to as $\mathcal{Y}_o = \{y_{o_1}, y_{o_2}, ..., y_{o_{\mathcal{C}_o}}\}$ and $\mathcal{Y}_r = \{y_{r_1}, y_{r_2}, ..., y_{r_{\mathcal{C}_r}}\}$ respectively.

3.2. Overview

To generate more unbiased scene graphs from videos, it is necessary to address the challenges highlighted in Fig 1, 2 and 3. To this end, we propose **TEMPURA** for unbiased dynamic SGG. As shown in Fig 4, TEMPURA works with a predicate embedding generator (PEG) that can be obtained from any existing dynamic SGG model [10, 57]. Since transformer-based models have shown to be better learners of spatio-temporal dynamics, we model our PEG as the spatio-temporal transformer of [10] which is built on top of the vanilla transformer architecture of [58]. The object sequence processing unit (OSPU) facilitates temporally consistent object classification. The memory diffusion unit (MDU) and the Gaussian Mixture Model (GMM) head address the long-tail bias and overall noise in video SGG data, respectively. In the subsequent sections, we describe these units in more detail, along with the training and testing mechanism of TEMPURA.

3.3. Object Detection and Temporal Consistency

We first describe how we enforce more consistent object classification across the entire video. Using an off-the-self object detector, we obtain the set of objects $O_t = \{o^t_i\}^{N(t)}_{i=1}$ in each frame, where $o^t_i = \{b^t_i, \boldsymbol{v}^t_i, c^t_{o_i}\}$ with $b^t_i \in \mathbb{R}^4$ being the bounding box, $\boldsymbol{v}^t_i \in \mathbb{R}^{2048}$ the RoIAligned [20] proposal feature of o^t_i and $c^t_{o_i}$ is its predicted class. Existing methods [10, 24, 38, 59] either directly use $c^t_{o_i}$ as the object classification or pass $\boldsymbol{v}^t_i$ through a single/multi-layered feed-forward network (FFN) to classify o_i. However, object detectors trained on static images fail to compensate for dynamic nuances and temporal fluctuations in videos, making them prone to misclassify the same object in different frames. Some works address this by incorporating object tracking algorithms [57], but we incorporate a simple but effective learning-based strategy.

We introduce an Object Sequence Processing Unit (OSPU) which utilizes a transformer encoder [58] referred to as sequence encoder or $SeqEnc$ (Fig 4), to process a set of sequences, $\mathcal{T}_\mathcal{V}$, which is constructed as follows,

$$\mathcal{T}_\mathcal{V} = \{\mathcal{T}^1_{t_1k_1}, \mathcal{T}^2_{t_2k_2}, ..., \mathcal{T}^{\hat{\mathcal{C}}_o}_{t_{\hat{\mathcal{C}}_o}k_{\hat{\mathcal{C}}_o}}\};\ \mathcal{T}^j_{t_jk_j} = \{\boldsymbol{v}^t_i, \boldsymbol{v}^{t+1}_i, ..., \boldsymbol{v}^{t+k}_i\}, \quad (1)$$

where each entry of $\mathcal{T}^j_{t_jk_j}$ has the same detected class c_{o_j}, $1 \leq t_j, k_j \leq T$ and $\hat{\mathcal{C}}_o \leq \mathcal{C}_o$ refers to all detected object classes in the video $\mathcal{V}$. Zero-padding is used to turn $\mathcal{T}_\mathcal{V}$ into a functioning tensor. $SeqEnc$ utilizes the multi-head self-attention to learn the long-term temporal dependencies in each $\mathcal{T}^j_{t_jk_j}$. For any input $\boldsymbol{X}$, a single attention head, $\mathbb{A}$, is defined as follows:

$$\mathbb{A}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D_k}})\boldsymbol{V}, \quad (2)$$

where D_k is the dimension of $\boldsymbol{K}$, and $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are the query, key and value vectors which for self-attention is $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{X}$. The multi-head attention, $\mathbb{MA}$, is shown below,

$$\begin{aligned} &\mathbb{MA}(\boldsymbol{X}) = Concat(a_1, a_2, ..a_H)W_H, \\ &a_i = \mathbb{A}(\boldsymbol{X}W_{Q_i}, \boldsymbol{X}W_{K_i}, \boldsymbol{X}W_{V_i}), \end{aligned} \quad (3)$$

where $W_{Q_i} \in \mathbb{R}^{D \times D_{Q_i}}$, $W_{K_i} \in \mathbb{R}^{D \times D_{K_i}}$, $W_{V_i} \in \mathbb{R}^{D \times D_{V_i}}$ and $W_H \in \mathbb{R}^{HD_v \times D}$ are learnable weight matrices. As shown in Fig 4, we follow the classical design of [58] for $SeqEnc$, whereby a residual connection is used to add $\boldsymbol{V}$ with $\mathbb{MA}(\boldsymbol{X})$ followed by normalization [2], and subsequent passing through an FFN. The output of an n layered sequence encoder is as follows,

$$\boldsymbol{X}^{(n)}_{out} = SeqEnc(\boldsymbol{X}^{(n-1)}_{out});\ \boldsymbol{X}^{(0)}_{out} = \hat{\mathcal{T}}_\mathcal{V}, \quad (4)$$

where $\hat{\mathcal{T}}_\mathcal{V} = \mathcal{T}_\mathcal{V} + \boldsymbol{E}^T_o$ with $\boldsymbol{E}^T_o$ being fixed positional encodings [58] for injecting each object's temporal position. The final object logits, $\hat{\mathcal{Y}}_o = \{\hat{y}_{o_i}\}^{\mathcal{C}_o}_{i=1}$, are obtained by passing $\boldsymbol{X}^{(n)}_{out}$ through a 2-layer FFN. The corresponding object classification loss, $\mathcal{L}_o$, is modeled as the cross-entropy between $\hat{\mathcal{Y}}_o$ and $\mathcal{Y}_o$.

To enhance the $SeqEnc$'s capability of enforcing temporal consistency, we add a supervised contrastive loss [18] over its output embeddings, as shown below,

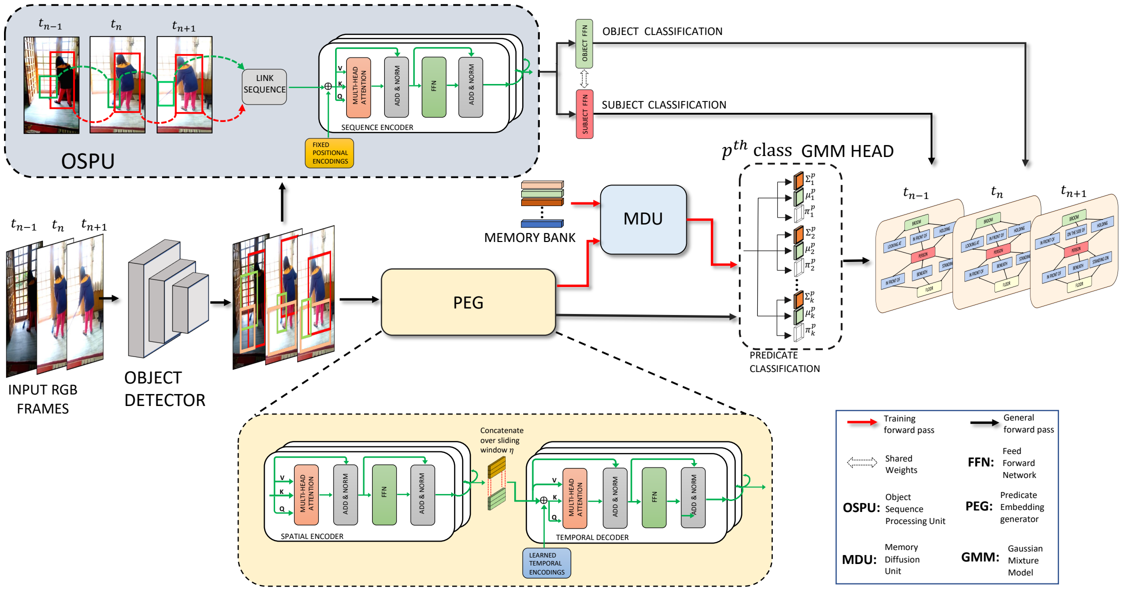


Figure 4. **Framework of TEMPURA.** The object detector generates initial object proposals for each RGB frame in a video. The proposals are then passed to the OSPU, where they are first linked into sequences based on the object detector's confidence scores. These sequences are processed with a transformer encoder to generate temporally consistent object embeddings for improved object classification. The proposals and semantic information of each subject-object pair are passed to the PEG to generate a spatio-temporal representation of their relationships. Modeled as a spatio-temporal transformer [10], the PEG's encoder learns the spatial context of the relationships and its decoder learns their temporal dependencies. Due to the long-tail nature of the relationship/predicate classes, a Memory Bank in conjunction with the MDU is used during training to debias the PEG, enabling the production of more generalizable predicate embeddings. Finally, a $\mathcal{K}$-component GMM head classifies the PEG embeddings and models the uncertainty associated with each predicate class for a given subject-object pair.

$$\mathcal{L}_{intra}=\sum_i\sum_j||\hat{\boldsymbol{x}}_{o_i}-\hat{\boldsymbol{x}}^+_{o_j}||^2_2+\sum_k max(0,1-||\hat{\boldsymbol{x}}_{o_i}-\hat{\boldsymbol{x}}^-_{o_k}||^2_2), \quad (5)$$

where $\hat{\boldsymbol{x}}_{o_i}\in\boldsymbol{X}^{(n)}_{out}$. $\mathcal{L}_{intra}$ enforces intra-video temporal consistency by pulling closer the embeddings of positive pairs sharing the same ground-truth class and pushing apart the embeddings of negative pairs with different ground-truth class.

3.4. Predicate Embedding Generator

A predicate embedding generator (PEG) assimilates the information of each subject-object pair to generate an embedding that summarizes the relationship(s) between them. For dynamic SGG, the PEG must learn the temporal as well as the spatial context of the relationship between each pair. In our setup, we model the PEG as the Spatio-Temporal transformer of [10]. For each pair (i,j), we construct the input to the PEG as shown below,

$$\boldsymbol{r}^t_k=Concat(f_v(\boldsymbol{v}^t_i),f_v(\boldsymbol{v}^t_j),f_u(\boldsymbol{u}^t_{ij}+f_{box}(b^t_i,b^t_j)),\boldsymbol{s}^t_i,\boldsymbol{s}^t_j), \quad (6)$$

where $\boldsymbol{v}^t_i$ and $\boldsymbol{v}^t_j$ are the subject and object proposal features, $\boldsymbol{u}^t_{ij}\in\mathbb{R}^{256\times7\times7}$ is the feature map of the union box computed by RoIAlign [20], $\boldsymbol{s}^t_i,\boldsymbol{s}^t_j\in\mathbb{R}^{200}$ are the semantic glove embeddings [46] of the subject and object class determined from $\hat{\mathcal{Y}}_o$, f_v and f_u are FFN based non-linear projections, f_{box} is the bounding box to feature map projection of [62]. The set of t^{th} frame input representations are $\boldsymbol{R}_t=\{\boldsymbol{r}^j_t\}^{K(t)}_{j=1}\in\mathbb{R}^{K(t)\times1936}$. As shown in Fig 4 the PEG consists of a spatial encoder, $SpaEnc$, and a temporal decoder, $TempDec$, where the former learns the spatial context of the visual relations and the latter learns their temporal dependencies. Therefore for an n layered spatial encoder, its output $\boldsymbol{R}^t_{spa}$ is computed as follows,

$$\boldsymbol{Z}^{(n)}_{spa,t}=SpaEnc(\boldsymbol{Z}^{(n-1)}_{spa,t});\ \boldsymbol{Z}^{(0)}_{spa,t}=\boldsymbol{R}_t\ , \quad (7)$$

where $\boldsymbol{R}^t_{spa}=\boldsymbol{Z}^{(n)}_{spa,t}$. The formulation of $SpaEnc$ is the same as $SeqEnc$ (Eq 4). To learn the temporal dependencies of the relationships, the decoder input is constructed as a sequence over a non-overlapping sliding window whereby,

$$\boldsymbol{Z}_{tem}=\{\boldsymbol{R}^t_{spa},...,\boldsymbol{R}^{t+\eta-1}_{spa}\},\ t\in[1,T-\eta+1], \quad (8)$$

where $\eta\leq T$ is the sliding window and T is the length of the video. As shown in Fig 4, the inputs to $TempDec$'s $\mathbb{MA}$ are, $\boldsymbol{Q}=\boldsymbol{K}=\boldsymbol{Z}_{tem}+\boldsymbol{E}^\eta_r$ and $\boldsymbol{V}=\boldsymbol{Z}_{tem}$ where $\boldsymbol{E}^\eta_r=\{\boldsymbol{e}^1_r,\boldsymbol{e}^2_r,...,\boldsymbol{e}^\eta_r\}$ are learnable temporal encodings [10] injecting the temporal position of each predicate. The final output $\mathcal{R}_{tem}$ of an n layered temporal decoder is,

$$\boldsymbol{Z}^{(n)}_{tem}=TempDec(\boldsymbol{Z}^{(n-1)}_{tem});\ \boldsymbol{Z}^{(0)}_{tem}=\boldsymbol{Z}_{tem}\ , \quad (9)$$

Therefore, the final set of predicate embeddings generated by the PEG is $\mathcal{R}_{tem}=\boldsymbol{Z}^{(n)}_{tem}=\{\boldsymbol{R}^t_{tem}\}^{T-\eta+1}_{t=1}$ with $\boldsymbol{R}^t_{tem}=\{\boldsymbol{r}^j_{tem}\}^{K(t)}_{j=1}\in\mathbb{R}^{K(t)\times1936}$.

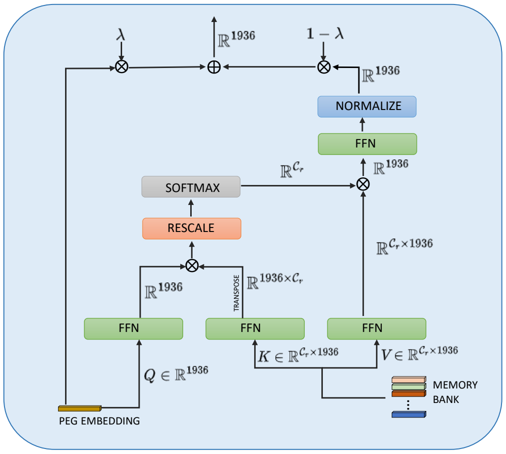

λ
$\mathbb{R}^{1936}$
$1-\lambda$
$\mathbb{R}^{1936}$
NORMALIZE
FFN
$\mathbb{R}^{\mathcal{C}_r}$
$\mathbb{R}^{1936}$
SOFTMAX
RESCALE
$\mathbb{R}^{\mathcal{C}_r\times 1936}$
$\mathbb{R}^{1936}$
TRANSPOSE
$\mathbb{R}^{1936\times\mathcal{C}_r}$
FFN
FFN
FFN
$K\in\mathbb{R}^{\mathcal{C}_r\times 1936}$
$V\in\mathbb{R}^{\mathcal{C}_r\times 1936}$
$Q\in\mathbb{R}^{1936}$
MEMORY BANK
PEG EMBEDDING

Figure 5. **Illustration of the Memory Diffusion Unit (MDU).** $\otimes$ and $\oplus$ are matrix multiplication and element-wise addition respectively.

3.5. Memory guided Debiasing

Due to the long-tailed bias in SGG datasets, the direct PEG embeddings, $\mathcal{R}_{tem}$, are biased against the rare predicate classes, necessitating the need to debias them. We accomplish this via a memory-guided training strategy, whereby for any given relationship embedding, $\boldsymbol{r}^j_{tem} \in \mathcal{R}_{tem}$, a Memory Diffusion Unit (MDU) first retrieves relevant information from a predicate class centric memory bank $\boldsymbol{\Omega}_R$ and uses it to enrich $\boldsymbol{r}^j_{tem}$ which results in a more balanced embedding $\hat{\boldsymbol{r}}^j_{tem}$. The memory bank $\boldsymbol{\Omega}_R = \{\boldsymbol{\omega}_p\}_{p=1}^{\mathcal{C}_r}$ is composed of a set of memory prototypes each of which is an abstraction of a predicate class and is computed as a function of their corresponding PEG embeddings. In our setup, the prototype is defined as a class-specific centroid, whereby, $\boldsymbol{\omega}_p = \frac{1}{N_{y_{rp}}}\sum_{j=1}^{N_{y_{rp}}} \boldsymbol{r}^j_{tem}\ \forall\, p \in \mathcal{Y}_r$, with $N_{y_{rp}}$ being the total number of subject-object pairs mapped to the predicate class y_{r_p}, in the *entire training set*.

Progressive Memory Computation. $\boldsymbol{\Omega}_R$ is computed in a progressive manner whereby the model's last state is used to compute memory for the current state, i.e., the memory of epoch α is computed using the model weights of epoch $\alpha-1$. This enables $\boldsymbol{\Omega}_R$ to become more refined with every epoch. Since no memory is available for the first epoch, the MDU remains inactive for this state, and $\hat{\boldsymbol{r}}^j_{tem} = \boldsymbol{r}^j_{tem}$.

Memory Diffusion Unit. As shown in Fig 5 for a given query the MDU uses the attention operator [58] to retrieve relevant information from $\boldsymbol{\Omega}_R$ as a diffused memory feature $\boldsymbol{r}^j_{mem}$ i.e.,

$$\boldsymbol{r}^j_{mem} = \mathbb{A}(\boldsymbol{Q}W_Q^{mem}, \boldsymbol{K}W_K^{mem}, \boldsymbol{V}W_V^{mem}), \qquad (10)$$

where, $\boldsymbol{Q} = \boldsymbol{r}^j_{tem}$ and $\boldsymbol{K} = \boldsymbol{V} = \boldsymbol{\Omega_r}$ and $W_Q^{mem}, W_K^{mem}\ \& W_V^{mem} \in \mathbb{R}^{1936\times 1936}$ are learnable weight matrices. Since each subject-object pair has multiple predicates mapped to it, many visual relations share similar characteristics, which means their corresponding memory prototypes $\boldsymbol{\omega}_p$ share multiple predicate embeddings. Therefore the attention operation of Eq 10 facilitates knowledge transfer from data-rich to data-poor classes utilizing the memory bank, whereby $\boldsymbol{r}^j_{mem}$ hallucinates compensatory information about the data-poor classes otherwise missing in $\boldsymbol{r}^j_{tem}$. This information is diffused back to $\boldsymbol{r}^j_{tem}$ to obtain the balanced embedding $\hat{\boldsymbol{r}}^j_{tem}$ as shown below,

$$\hat{\boldsymbol{r}}^j_{tem} = \lambda \boldsymbol{r}^j_{tem} + (1-\lambda)\boldsymbol{r}^j_{mem}\ , \qquad (11)$$

where $0 < \lambda \leq 1$. As shown in Fig 4, the MDU is used during the training phase only since it does not function as a network module to forward pass through but rather as a meta-learning inspired [45, 51, 65] structural meta-regularizer. Since $\boldsymbol{\Omega}_R$ is computed directly from the PEG embeddings, backpropagating over the MDU refines the computed memory prototypes, in turn enabling better information diffusion and inherently teaching the PEG how to generate more balanced embeddings that do not underfit to the data-poor relationships. λ over here acts as a gradient scaling factor, which during backpropagation asymmetrically scales the gradients associated with $\boldsymbol{r}^j_{tem}$ and $\boldsymbol{r}^j_{mem}$ in the residual operation of Eq 11. Since the initial PEG embeddings are heavily biased towards the data-rich classes, if λ is too high, the compensating effect of the diffused memory feature is drastically reduced. On the other hand, if λ is too low, excessive knowledge gets transferred from the data-rich to the data-poor classes resulting in poor performance on the former.

3.6. Uncertainty Attenuated Predicate Classification

To address the noisy annotations in SGG data, we model the predicate classification head as a $\mathcal{K}$ component Gaussian Mixture Model (GMM) [28]. Given a sample embedding $\mathbf{z}_i$ the mean, variance and mixture weights for the p^{th} predicate class are estimated as follows:

$$\mu^k_{i,p} = f^k_\mu(\mathbf{z}_i),\ \Sigma^k_{i,p} = \sigma(f^k_\Sigma(\mathbf{z}_i)),\ \pi^k_{i,p} = \frac{e^{f^k_\pi(\mathbf{z}_i)}}{\sum\limits_{k=1}^{\mathcal{K}} e^{f^k_\pi(\mathbf{z}_i)}}, \qquad (12)$$

where $f^k_\mu, f^k_\Sigma, f^k_\pi$ are FFN based projection functions and σ is the sigmoid non-linearity which ensures $\Sigma^k_{i,p} \geq 0$. The class-specific aleatoric and epistemic uncertainty, for the sample $\mathbf{z}_i$ are computed as follows:

$$U^p_{al}(\mathbf{z}_i) = \sum_{k=1}^{\mathcal{K}} \pi^k_{i,p}\Sigma^k_{i,p}\ \ ;\ \ U^p_{ep}(\mathbf{z}_i) = \sum_{k=1}^{\mathcal{K}} \pi^k_{i,p}||\mu^k_{i,p} - \sum_{j=1}^{\mathcal{K}} \pi^j_{i,p}\mu^j_{i,p}||^2_2\ , \qquad (13)$$

Therefore, by using a GMM head, we are modeling the inherent uncertainty associated with the data from a Bayesian perspective [8, 9, 28]. During training $\mathbf{z}_i = \hat{\boldsymbol{r}}^i_{tem}$ and the probability distribution for the p^{th} predicate is given as,

$$\hat{y}^i_{r_p} = \sum_{k=1}^{\mathcal{K}} \pi^k_{i,p}\mathcal{N}(\mu^k_{i,p}, \Sigma^k_{i,p}), \qquad (14)$$

where $\mathcal{N}$ is the Gaussian distribution. Since the sampling $\mathcal{N}(\mu^k_p, \Sigma^k_p)$ is non-differentiable we use the re-parameterization trick of [30] to compute $\hat{y}^i_{r_p}$ as shown below:

$$\hat{y}^i_{r_p} = \sum_{k=1}^{\mathcal{K}} \pi^k_{i,p}\sigma(\hat{c}^i_{p,k});\ \ \hat{c}^i_{p,k} = \mu^k_{i,p} + \varepsilon\sqrt{\Sigma^k_{i,p}}\ , \qquad (15)$$

where $\varepsilon \sim \mathcal{N}(0,1)$ and is of the same size as Σ^k_p. The overall set of predicate logits is $\hat{\mathcal{Y}}_r = \{\hat{y}_{r_p}\}^{\mathcal{C}_r}_{p=1}$. The predicate classification loss $\mathcal{L}_p$ is modeled as the GMM sigmoidal cross entropy [8] as shown below,

$$\mathcal{L}_p = -\sum_{i=1}^{N_{r,p}}\sum_{p=1}^{\mathcal{C}_r} y^i_{r_p} \log \sum_{k=1}^{\mathcal{K}} \pi^k_p \sigma(\hat{c}^i_{p,k}), \tag{16}$$

where $y^i_{r_p}$ is the ground-truth predicate class mapped to $\boldsymbol{z}_i$. By incorporating the modeled aleatoric uncertainty of $\mathbf{z}_i$ ($\Sigma^k_{i,p}$) in $\mathcal{L}_p$, we essentially utilize it as an *attenuation* factor, which penalizes the model if $\Sigma^k_{i,p}$ is large. This principle is called *learned loss attenuation* [9, 28], and it discourages the model from predicting high uncertainty thereby attenuating the effects of uncertain samples due to inherent annotation noise in the data.

3.7. Training and Testing

Training. As explained in section 3.5, memory computation and utilization of MDU is activated from the second epoch, and so for the first epoch, $\hat{\boldsymbol{r}}^i_{tem} = \boldsymbol{r}^i_{tem}$. The OSPU and the GMM head obviously start firing from the first epoch itself. The entire framework is trained end-to-end by minimizing the following loss,

$$\mathcal{L}_{total} = \mathcal{L}_p + \mathcal{L}_o + \mathcal{L}_{intra} \; , \tag{17}$$

Testing. The forward pass during testing is highlighted in Fig 4. After training, the MDU has served its purpose of teaching the PEG to generate more unbiased embeddings, and therefore during inference, $\boldsymbol{r}^i_{tem}$ is directly passed to the GMM head to obtain the predicate confidence scores, $\hat{y}^i_{r_p}$, which during testing are computed as follows,

$$\hat{y}^i_{r_p} = \sum_{k=1}^{\mathcal{K}} \pi^k_{i,p} \sigma(\mu^k_{i,p}), \tag{18}$$

.

4. Experiments

4.1. Dataset and Implementation

Dataset. We perform experiments on the Action Genome (AG) [25] dataset, which is the largest benchmark dataset for video SGG. It is built on top of Charades [50] and has 234,253 annotated frames with 476,229 bounding boxes for 35 object classes (without person), with a total of 1,715,568 annotated predicate instances for 26 relationship classes.

Metrics and Evaluation Setup. We evaluate the performance of TEMPURA with standard metrics Recall@K (R@K) and mean-Recall@K (mR@K), for $K = [10,20,50]$. As discussed before, R@K tends to be biased towards the most frequent predicate classes [56] whereas mR@K is a more balanced metric enabling evaluation of SGG performance on all the relationship classes [56]. As per standard practice [10, 25, 31, 57], three SGG tasks are chosen, namely: (1) Predicate classification (*PREDCLS*): Prediction of predicate labels of object pairs, given ground truth labels and bounding boxes of objects; (2) Scene graph classification (*SGCLS*): Joint classification of predicate labels and the ground truth bounding boxes; (3) Scene graph detection (*SGDET*): End-to-end detection of the objects and predicate classification of object pairs. Evaluation is conducted under two setups: **With Constraint** and **No constraints**. In the former the generated graphs are restricted to at most one edge, i.e., each subject-object pair is allowed only one predicate and in the latter, the graphs can have multiple edges. We note that mean Recall is averaged over all predicate classes, thus reflective of an SGG model's long-tailed performance as opposed to Recall, which might be biased towards head classes.

Implementation details. Following prior work, [10, 38], we choose FasterRCNN [47] with ResNet-101 [21] as the object detector. For the predicate embedding generator, we choose the Spatio-temporal transformer architecture of [10], with the same number of encoder-decoder layers and attention heads. The gradient scaling factor λ is set to 0.5 for *PREDCLS* and *SGDET* and 0.3 for *SGCLS*. The number of GMM components $\mathcal{K}$ is set to 4 for *SGCLS* and *SGDET* and 6 for PREDCLS. The framework is trained end to end for 10 epochs using the AdamW optimizer [41] and a batch size of 1. The initial learning rate is set to 10^{-5}.

4.2. Comparison to state-of-the-art

We compare our method with existing dynamic SGG methods such as STTran [10], TRACE [57], STTran-TPI [59], APT [38], ISGG [29]. We also compare with ReLDN [63] which is a static SGG method. Table 1 shows the comparative results for *SGDET* in terms of both mR@K and R@K. Tables 2 and 3 show the comparative results for *PREDCLS* + *SGCLS* in terms of mR@K and R@K respectively. We utilized the official code for several state-of-the-art dynamic SGG methods to obtain respective mR@K values for all three SGG tasks under both **With Constraint** and **No Constraints** setup. We also relied on email communications with the authors of several papers on the mR values where the source code are not publicly available. From Tables 1 and 2, we observe that TEMPURA significantly outperforms the other methods in mean Recall. Specifically, in comparison to the best baseline, we observe improvements of **5.1**% on *PREDCLS*-mR@10, **5.7**% on *SGCLS*-mR@10 and **1.9**% on *SGDET*-mR@10 under the **With Constraint** setup. For the **No Constraints** setup the improvements are even more significant with **10.1**% on *PREDCLS*-mR@10, **7.6**% on *SGCLS*-mR@10 and **3.8**% on *SGDET*-mR@10. This clearly shows that TEMPURA can generate more unbiased scene graphs by better detecting both data-rich and data-poor classes. This is further verified from Fig 6 where we compare mR@10 values for the *HEAD*, *BODY* and *TAIL* classes of AG with that of STTran and TRACE. TEMPURA significantly improves performance on the *TAIL* classes without compromising performance on the *HEAD* and *BODY* classes. Similar charts for the **No Constraints** setup are provided in the supplementary. The comparative per-class performance in Fig 7 further shows that TEMPURA outperforms

Table 1. Comparative results for SGDET task, on AG [25], in terms of mean-Recall@K and Recall@K. Best results are shown in bold.

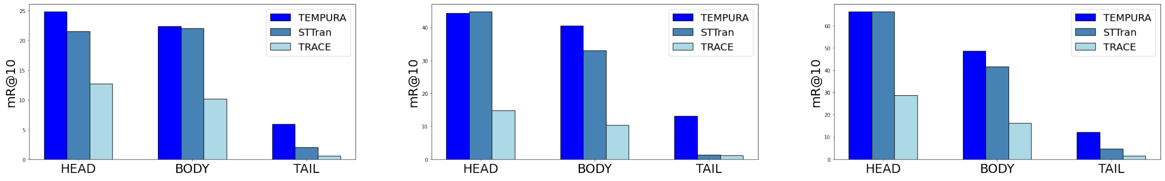
Method	With Constraint						No Constraints					
	mR@10	mR@20	mR@50	R@10	R@20	R@50	mR@10	mR@20	mR@50	R@10	R@20	R@50
RelDN [64]	3.3	3.3	3.3	9.1	9.1	9.1	7.5	18.8	33.7	13.6	23.0	36.6
HCRD supervised [16]	-	8.3	9.1	-	27.9	30.4	-	-	-	-	-	-
TRACE [57]	8.2	8.2	8.2	13.9	14.5	14.5	22.8	31.3	41.8	26.5	35.6	45.3
ISGG [29]	-	19.7	22.9	-	29.2	35.3	-	-	-	-	-	-
STTran [10]	16.6	20.8	22.2	25.2	34.1	37.0	20.9	29.7	39.2	24.6	36.2	48.8
STTran-TPI [59]	15.6	20.2	21.8	26.2	34.6	37.4	-	-	-	-	-	-
APT [38]	-	-	-	26.3	**36.1**	**38.3**	-	-	-	25.7	37.9	**50.1**
TEMPURA	**18.5**	**22.6**	**23.7**	**28.1**	33.4	34.9	**24.7**	**33.9**	**43.7**	**29.8**	**38.1**	46.4

Table 2. Comparative results for SGG tasks: PREDCLS and SGCLS, on AG [25], in terms of mean-Recall@K. Best results are shown in bold.

Method	With Constraint						No Constraints					
	PredCLS			SGCLS			PredCLS			SGCLS		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
RelDN [64]	6.2	6.2	6.2	3.4	3.4	3.4	31.2	63.1	75.5	18.6	36.9	42.6
TRACE [57]	15.2	15.2	15.2	8.9	8.9	8.9	50.9	73.6	82.7	31.9	42.7	46.3
STTran [10]	37.8	40.1	40.2	27.2	28.0	28.0	51.4	67.7	82.7	40.7	50.1	58.8
STTran-TPI [59]	37.3	40.6	40.6	28.3	29.3	29.3	-	-	-	-	-	-
TEMPURA	**42.9**	**46.3**	**46.3**	**34.0**	**35.2**	**35.2**	**61.5**	**85.1**	**98.0**	**48.3**	**61.1**	**66.4**

Table 3. Comparative results for SGG tasks: PREDCLS and SGCLS, on AG [25], in terms of Recall@K. Best results are shown in bold.

Method	With Constraint						No Constraints					
	PredCLS			SGCLS			PredCLS			SGCLS		
	R@10	R@20	R@50	R@10	R@20	R@50	R@10	R@20	R@50	R@10	R@20	R@50
RelDN [64]	20.3	20.3	20.3	11.0	11.0	11.0	44.2	75.4	89.2	25.0	41.9	47.9
TRACE [57]	27.5	27.5	27.5	14.8	14.8	14.8	72.6	91.6	96.4	37.1	46.7	50.5
STTran [10]	68.6	71.8	71.8	46.4	47.5	47.5	77.9	94.2	99.1	54.0	63.7	66.4
STTran-TPI [59]	**69.7**	72.6	72.6	**47.2**	48.3	48.3	-	-	-	-	-	-
APT [38]	69.4	**73.8**	**73.8**	**47.2**	**48.9**	**48.9**	78.5	**95.1**	99.2	55.1	**65.1**	**68.7**
TEMPURA	68.8	71.5	71.5	**47.2**	48.3	48.3	**80.4**	94.2	**99.4**	**56.3**	64.7	67.9



(a) SGDET

(b) SGCLS

(c) PREDCLS

Figure 6. Comparison of mR@10 for the HEAD, BODY and TAIL classes in Action Genome [25] under the ”with constraint” setup.

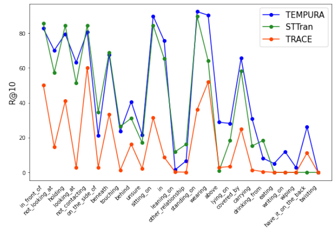


Figure 7. Comparative per class performance for PREDCLS task. Results are in terms of R@10 under "with constraint".

both STTran and TRACE for most predicate classes. Tables 1 and 3 show that TEMPURA does not compromise Recall values and achieves comparable or better performance than the existing methods, which made deliberate efforts to achieve high Recall values without considering their long-tailed performances. Qualitative visualizations are shown in Fig 8.

4.3. Ablation Studies

We conduct ablation experiments on *SGCLS* and *SGDET* tasks to study the impact of the OSPU, MDU, and GMM head, the combination of which enables TEMPURA to generate more unbiased scene graphs. When all these components are removed, TEMPURA essentially boils down to the baseline STTran architecture [10], where the object proposals and PEG embeddings are mapped to a few layers of FFN for respective classification.

Uncertainty Attenuation and Memory guided Training. We first study the impact of uncertainty-aware learning and memory-guided debiasing. For the first case, we remove the MDU during training and use only the GMM head. For the second case, we substitute the GMM head with a simple FFN head as the classifier, with the predicate loss $\mathcal{L}_p$ converted to a simple multi-label binary cross entropy. The results of these respective

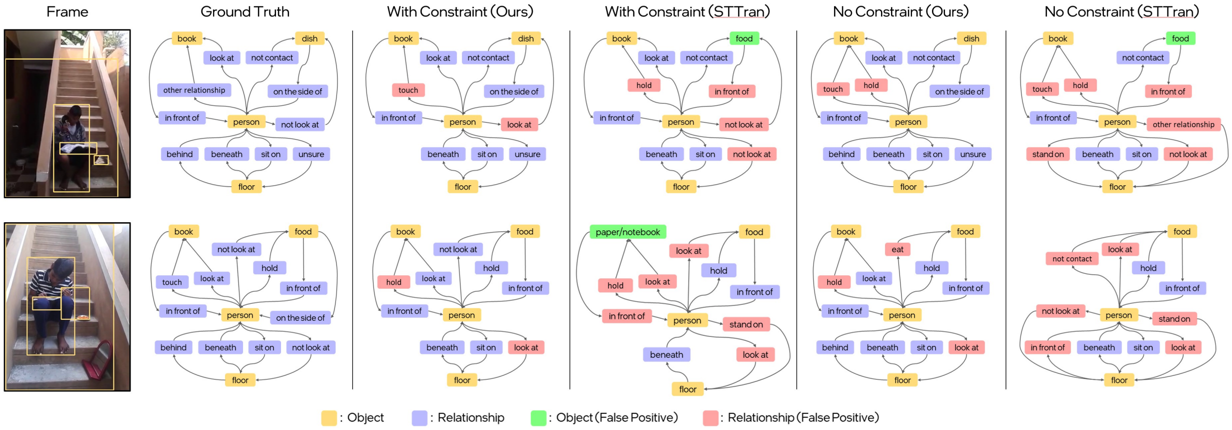


Figure 8. **Comparative qualitative results**. From left to right: input video frames, ground truth scene graphs, scene graphs generated by TEMPURA, and the scene graphs generated by the baseline STTran [10]. Incorrect object and predicate predictions are shown in green and pink, respectively.

Table 4. Importance of uncertainty attenuation, memory guided debiasing, and temporally consistent object classification for SGCLS and SGDET.

Uncertainty Attenuation	Memory guided Debiasing	Temporal Consistency	With Constraint				No Constraints			
			SGCls		SGDet		SGCls		SGDet	
			mR@10	mR@20	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20
-	-	-	27.2	28.0	16.5	20.8	40.7	50.1	20.9	29.7
✓	-	✓	30.6	31.9	16.7	21.1	43.5	58.9	20.9	30.5
-	✓	✓	31.8	33.2	16.8	20.9	45.7	59.7	21.7	30.7
✓	✓	-	30.9	32.1	17.0	21.4	45.7	59.3	21.6	30.1
✓	✓	✓	**34.0**	**35.2**	**18.5**	**22.6**	**48.3**	**61.1**	**24.7**	**33.9**

Table 5. Performance of TEMPURA for varying numbers of GMM components, $\mathcal{K}$. Results are in terms of mR@10 for the **With Constraint** setup, with the best results shown in bold.

$\mathcal{K}$ Task	1	2	4	6	8
PREDCLS	40.1	40.8	42.6	**42.9**	42.1
SGCLS	31.0	33.1	**34.0**	32.7	32.6
SGDET	16.7	17.0	**18.5**	18.2	17.6

cases are shown in rows 1 & 2 of Table 4. It can be observed that the resulting models improve mR@K performance over the baseline. This indicates two things: 1) Modeling and attenuation of the predictive uncertainty of an SGG model can effectively address the noise associated with the TAIL classes, preventing it from under-fitting to them [22]. 2) MDU-guided training enables the PEG to generate embeddings that are more robust and generalizable to all the predicate classes, which performs slightly better than just using uncertainty-aware learning for all three SGG tasks. Combining both these principles gives the best performance, as seen in the final row of both tables.

Temporally Consistent Object Classification. By comparing rows 3 and 4 of Table 4, we can see that without the OSPU-based enforcement of temporal consistency on object classification, the performance drops significantly, highlighting the fact that object misclassification due to temporal nuances in videos is also a major source of noise in existing SGG frameworks. For the *PREDCLS* task the ground-truth bounding boxes and labels are already provided, so the OSPU has no role, and its weights are frozen during training.

Number of Gaussian components $\mathcal{K}$. The performance of TEMPURA for different values of $\mathcal{K}$ is shown in Table 5. Keeping $\mathcal{K}$ b/w 4 and 6 gives the best performance, beyond which the model incurs a heavy memory footprint with diminishing returns. More ablation experiments are provided in the supplementary.

5. Conclusions

The difficulty in generating dynamic scene graphs from videos can be attributed to several factors ranging from imbalanced predicate class distribution, video dynamics, temporal fluctuation of predictions, etc. Existing methods on dynamic SGG have mostly focused only on achieving high recall values, which are known to be biased towards head classes. In this work, we identify and address these sources of bias and propose a method, namely **TEMPURA**: **TE**mporal consistency and **M**emory **P**rototype guided **U**nce**R**tainty **A**ttentuation for dynamic SGG that can compensate for those biases. We show that TEMPURA significantly outperforms existing methods in terms of mean recall metric, showing its efficacy in long-term unbiased visual relationship learning from videos.

Acknowledgments. SN, KM, and ST were supported by Intel Corporation. SN and ARC were partially supported by ONR grant N000141912264.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 1, 3

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[3] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13137–13146, 2021. 2

[4] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 105–121, 2018. 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2

[7] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1

[8] Jiwoong Choi, Ismail Elezi, Hyuk-Jae Lee, Clement Farabet, and Jose M Alvarez. Active learning for deep object detection via probabilistic modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10264–10273, 2021. 2, 3, 5, 6

[9] Sungjoon Choi, Kyungjae Lee, Sungbin Lim, and Songhwai Oh. Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6915–6922. IEEE, 2018. 2, 3, 5, 6

[10] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-Temporal Transformer for Dynamic Scene Graph Generation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, October 2021. 1, 2, 3, 4, 6, 7, 8

[11] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009. 2

[12] Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15404–15413, 2021. 1, 2

[13] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017. 2

[14] Lizhao Gao, Bo Wang, and Wenmin Wang. Image captioning with scene-graph based semantic concepts. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, ICMLC 2018, page 225–229, New York, NY, USA, 2018. Association for Computing Machinery. 1

[15] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018. 3

[16] Raghav Goyal12 and Leonid Sigal123. A simple baseline for weakly-supervised human-centric relation detection. 2021. 1, 7

[17] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 2

[18] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2, 3

[19] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2022. 1

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3, 4

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[22] Yingsong Huang, Bing Bai, Shengwei Zhao, Kun Bai, and Fei Wang. Uncertainty-aware learning against label noise on imbalanced datasets. 2022. 2, 3, 8

[23] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[24] Jingwei Ji, Rishi Desai, and Juan Carlos Niebles. Detecting human-object relationships in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8106–8116, 2021. 1, 2, 3

[25] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action Genome: Actions as Composition of Spatio-temporal Scene Graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 6, 7

[26] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022. 1

[27] Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *arXiv preprint arXiv:1703.03129*, 2017. 3

[28] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc. 2, 3, 5, 6

[29] Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. 2022. 2, 6, 7

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

[31] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis,

Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, May 2017. 1, 2, 6

[32] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 2

[33] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19486–19496, 2022. 1, 2

[34] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11109–11119, 2021. 2

[35] Wei Li, Haiwei Zhang, Qijie Bai, Guoqing Zhao, Ning Jiang, and Xiaojie Yuan. Ppdl: Predicate probability distribution based loss for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2022. 2

[36] Xingchen Li, Long Chen, Jian Shao, Shaoning Xiao, Songyang Zhang, and Jun Xiao. Rethinking the evaluation of unbiased scene graph generation. *arXiv preprint arXiv:2208.01909*, 2022. 2

[37] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1270–1279, 2017. 1, 2

[38] Yiming Li, Xiaoshan Yang, and Changsheng Xu. Dynamic scene graph generation via anticipatory pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13874–13883, June 2022. 1, 2, 3, 6, 7

[39] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3746–3753, 2020. 1, 2

[40] Miao Liu, Siyu Tang, Yin Li, and James M Rehg. Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In *European Conference on Computer Vision*, pages 704–721. Springer, 2020. 1

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[42] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016. 1, 2

[43] Albert Michotte. *The perception of causality*. Routledge, 2017. 1

[44] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*, 2021. 1, 3

[45] Sarah Parisot, Pedro M Esperança, Steven McDonagh, Tamas J Madarasz, Yongxin Yang, and Zhenguo Li. Long-tail recognition via compositional knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6939–6948, 2022. 2, 3, 5

[46] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 4

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA, 2015. MIT Press. 2, 6

[48] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016. 2, 3

[49] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. 1

[50] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 6

[51] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 2, 3, 5

[52] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015. 3

[53] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 3

[54] Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[55] 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*, pages 3716–3725, 2020. 2

[56] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 6

[57] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target adaptive context aggregation for video scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13688–13697, 2021. 1, 2, 3, 6, 7

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 2, 3, 5

[59] Shuang Wang, Lianli Gao, Xinyu Lyu, Yuyu Guo, Pengpeng Zeng, and Jingkuan Song. Dynamic scene graph generation via temporal prior inference. In *ACM International Conference on Multimedia (MM '22)*, 2022. 2, 3, 6, 7

[60] Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1508–1517, 2020. 1

[61] Bin Wen, Jie Luo, Xianglong Liu, and Lei Huang. Unbiased scene graph generation via rich and fair semantic extraction. *arXiv preprint arXiv:2002.00176*, 2020. 2

[62] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. *CoRR*, abs/1711.06640, 2017. 1, 2, 4

[63] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019. 1, 6

[64] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019. 1, 2, 7

[65] Linchao Zhu and Yi Yang. Inflated episodic memory with region self-attention for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4344–4353, 2020. 2, 3, 5