Dynamic Graph Representation Learning for Video Dialog via Multi-Modal Shuffled Transformers

Shijie Geng[1], Peng Gao[2], Moitreya Chatterjee[3], Chiori Hori[4], Jonathan Le Roux[4], Yongfeng Zhang[1], Hongsheng Li[2], Anoop Cherian[4]

[1]Rutgers University, Piscataway, NJ, USA
[2]The Chinese University of Hong Kong, Hong Kong, China
[3]University of Illinois at Urbana Champaign, Urbana, IL, USA
[4]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

Abstract

Given an input video, its associated audio, and a brief caption, the audio-visual scene aware dialog (AVSD) task requires an agent to indulge in a question-answer dialog with a human about the audio-visual content. This task thus poses a challenging multi-modal representation learning and reasoning scenario, advancements into which could influence several human-machine interaction applications. To solve this task, we introduce a *semantics-controlled multi-modal shuffled Transformer reasoning* framework, consisting of a sequence of Transformer modules, each taking a modality as input and producing representations conditioned on the input question. Our proposed Transformer variant uses a shuffling scheme on their multi-head outputs, demonstrating better regularization. To encode fine-grained visual information, we present a novel dynamic scene graph representation learning pipeline that consists of an *intra-frame reasoning* layer producing spatio-semantic graph representations for every frame, and an *inter-frame aggregation* module capturing temporal cues. Our entire pipeline is trained end-to-end. We present experiments on the benchmark AVSD dataset, both on answer generation and selection tasks. Our results demonstrate state-of-the-art performances on all evaluation metrics.

Introduction

The success of deep learning in producing effective solutions to several fundamental problems in computer vision, natural language processing, and speech/audio understanding has provided an impetus to explore more complex multi-modal problems at the intersections of these domains, attracting wide interest recently (Zhu et al. 2020). A few notable ones include: (i) visual question answering (VQA) (Antol et al. 2015; Yang et al. 2003), the goal of which is to build an agent that can generate correct answers to free-form questions about visual content, (ii) audio/visual captioning (Hori et al. 2017; Venugopalan et al. 2015; Xu et al. 2015; Drossos, Lipping, and Virtanen 2019), in which the agent needs to generate a sentence in natural language describing the audio/visual content, (iii) visual dialog (Das et al. 2017), in which the agent needs to engage in a natural conversation with a human about a static image, and (iv) audio-visual scene-aware dialog (AVSD) (Alamri et al. 2019; Hori et al. 2019) – that generalizes (i), (ii), and (iii) – in which the agent needs to produce a natural answer to a question about a given audio-visual clip, in a conversation setting or select the correct answer from a set of candidates. The AVSD task[1] emulates a real-world human-machine conversation setting that is potentially useful in a variety of practical applications, such as building virtual assistants (Deruyttere et al. 2019) or in human-robot interactions (Thomason et al. 2019). See Figure 1 for an illustration of this task.

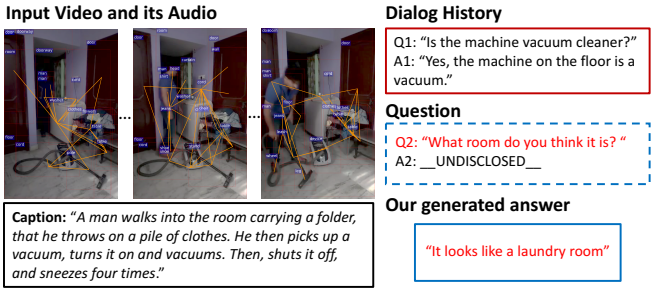

Figure 1: A result from our proposed model for the AVSD task. Given a video clip, its caption, dialog history, and a question, the AVSD generation task aims to generate the answer in natural language form.

The generality of the AVSD task, however, poses a challenging multi-modal representation learning and reasoning problem. Specifically, some of the input modalities to this task may offer complementary information (such as video and audio), while a few others may be independent (audio and captions), or even conflict with each other, e.g., the provided text (captions/dialogs) may include details from human experience that are absent in the video (e.g., "I think..."), or may include abstract responses ("happy", "bored", etc.) that may be subjective. Thus, the main quest in this task is to represent these modalities such that inference on them is efficient and effective. Previous approaches to this problem used holistic video features produced by a generic 3D convolutional neural network (Carreira and Zisserman 2017), and either focused on extend-

[1]https://video-dialog.com/

ing attention models on these features to include additional modalities (Alamri et al. 2019; Hori et al. 2019; Schwartz, Schwing, and Hazan 2019), or use vanilla Transformer networks (Vaswani et al. 2017) to produce effective multi-modal embeddings (Le et al. 2019). These off-the-shelf visual representations or Transformer architectures are not attuned to the task, potentially leading to sub-optimal performance.

In this paper, we present a neural inference algorithm that hierarchically reduces the complexity of the AVSD task using the machinery of graph neural networks and sequential multi-head Transformers. Specifically, we first present a spatio-temoral scene graph representation (STSGR) for encoding the video compactly while capturing its semantics. Specifically, our scheme builds on visual scene graphs (Johnson et al. 2015) towards video representation learning by introducing two novel modules: (i) an intra-frame reasoning module that combines graph-attention (Veličković et al. 2018) and edge-convolutions (Wang et al. 2019) to produce a semantic visual representation for every frame, (ii) subsequently, an inter-frame aggregation module uses these representations and updates them using information from temporal-neighborhoods, thereby producing compact spatio-temporal visual memories. We then couple these memories with temporally aligned audio features. Next, multi-head Transformers (Vaswani et al. 2017), encodes each of the other data modalities (dialog history, captions, and the pertinent question) separately alongside these audio-visual memories and fuses them sequentially using Transformer decoders. These fused features are then used to select or generate the *answers* auto-regressively. We also present a novel extension of the standard multi-head Transformer network in which the outputs of the heads are mixed. We call this variant a *shuffled Transformer*. Such random shuffling avoids overfitting of the heads to its inputs, thereby regularizing them, leading to better generalization.

To empirically evaluate our architecture, we present experiments on two variants of the AVSD dataset available as part of the 7th and 8th Dialog System Technology Challenges (DSTC). We provide experiments on both the answer generation and the answer selection tasks – the former requiring the algorithm to produce free-form sentences as answers, while the latter selects an answer from 100 choices for each question. Our results reveal that using the proposed STSGR and our shuffled Transformer lead to significant improvements on both tasks against state-of-the-art methods on all metrics. The key contributions of this paper are:

- We propose to represent videos as spatio-temporal scene graphs capturing key audio-visual cues and semantic structure. To the best of our knowledge, the combination of our intra/inter-frame reasoning modules is novel.
- We introduce a sequential Transformer architecture that uses shuffled multi-head attention, yielding question-aware representations of each modality while generating answers (or their embeddings) auto-regressively.
- Extensive experiments on the AVSD answer generation and selection tasks demonstrate the superiority of our approach over several challenging recent baselines.

Related Work

Our proposed framework has similarities with prior works along three different axes, viz. (i) graph-based reasoning, (ii) multi-modal attention, and (iii) visual dialog methods.

Scene Graphs: (Johnson et al. 2015) combine objects detected in static images, their attributes, and object-object relationships (Lu et al. 2016) to form a directed graph that not only provides an explicit and interpretable representation of the image, but is also seen to be beneficial for higher-order reasoning tasks such as image captioning (Li and Jiang 2019; Yang et al. 2019), and visual question answering (Ghosh et al. 2019; Norcliffe-Brown, Vafeias, and Parisot 2018; Geng et al. 2019, 2020). There have been efforts (Wang et al. 2018; Girdhar et al. 2019; Jain et al. 2016; Herzig et al. 2019; Jang et al. 2017; Tsai et al. 2019) at capturing spatio-temporal evolution of localized visual cues. In (Wang and Gupta 2018), a space-time graph reasoning framework is proposed for action recognition. Similarly, the efficacy of manually-labeled video scene graphs is explored in (Ji et al. 2020). Similar to ours, they use object detections per video frame, and construct a spatio-temporal graph based on the affinities of the features from the detected objects. Spatio-temporal graphs using knowledge distillation is explored for video captioning in (Pan et al. 2020). In contrast, our task involves several diverse modalities, demanding richer architectural choices. Specifically, we present a hierarchically organized intra/inter-frame reasoning pipeline for generating visual memories, trained via neural message passing, offering a powerful inference engine. Our ablation studies demonstrate the usefulness of these modules.

Multi-modal Fusion/Attention: has been explored in several prior works (Hori et al. 2017, 2018, 2019; Shi et al. 2020a), however does not use the power of Transformers. Self-attention and feature embeddings using Transformers is attempted in multi-modal settings (Gao et al. 2019a,b; Shi et al. 2020b), however only on static images. Bilinear fusion methods (Ben-Younes et al. 2019; Fukui et al. 2016) have been explored towards inter-modality semantic alignment, however they often result in high-dimensional interaction tensors that are computationally expensive during inference. In contrast, our pipeline is the first to leverage the power of Transformers in a hierarchical graph reasoning setup for video dialogs and is cheap to compute.

Multi-modal Dialogs: have been explored in various ways before. Free-form human-like answers were first considered in (Das et al. 2017), which also proposed the VisDial dataset, however on static images. A difficulty in designing algorithms on multi-modal data is in deriving effective attention mechanisms that can divulge information from disparate cues. To tackle this challenge, (Wu et al. 2018) proposed a sequential co-attention scheme in which the neural embeddings of various modalities are co-attended with visual embeddings in a specific order. (Schwartz et al. 2019) generalized the co-attention problem by treating the modalities as nodes of a graph, aggregating them as *factors*, using neural message passing. We use a combination of these two

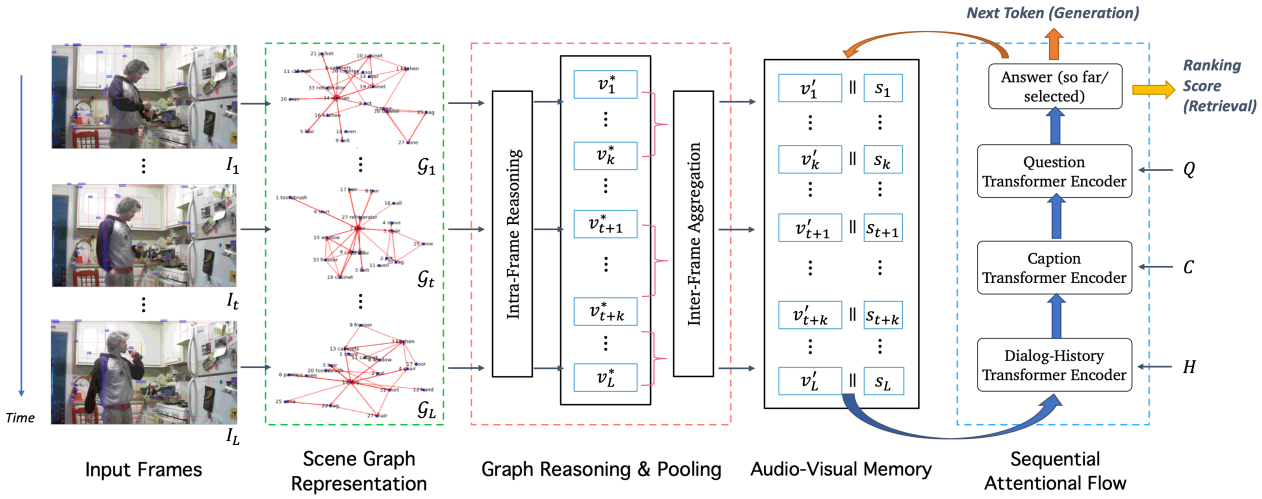

Figure 2: A schematic illustration of our overall pipeline for dialog response generation/retrieval.

approaches; specifically we use Transformer encoders for embedding each modality, and attend on these multi-modal embeddings sequentially to generate the answer. Further, in contrast to (Schwartz et al. 2019; Wu et al. 2018), that tackle solely the answer generation problem, we consider the answer selection task on AVSD as well. (Yeh et al. 2019) also proposed using Transformers (Vaswani et al. 2017) for fusing audio-visual features on the AVSD task. A multi-step reasoning scheme is proposed in (Gan et al. 2019) using joint attention via an RNN for generating a multi-modal representation. The Simple baseline (Schwartz, Schwing, and Hazan 2019) extends factor graphs (Schwartz et al. 2019) for the AVSD problem demonstrating promising results. A multi-modal Transformer for embedding various modalities and a query-aware attention is introduced in (Le et al. 2019). (Le and Hoi 2020) fine-tunes pretrained GPT-2 to obtain improved performance. However, these works neither consider richer visual representations using scene graphs, nor variations of the standard Transformers, like the shuffling scheme we present.

Proposed Method

In this section, we will first present our spatio-temporal scene graph representation (STSGR) for encoding the video sequences, following which we elaborate on our multi-modal shuffled Transformer architecture.

Overview of Spatio-Temporal Scene Graphs

Given a video sequence V, let C denote the associated human-generated video caption, and let (Q_i, A_i) represent the tuple of the text-based i-th question and its answer in the given human dialog about V (see Fig. 1). We will succinctly represent the dialog history by $H = \langle (Q_1, A_1), \ldots, (Q_{l-1}, A_{l-1}) \rangle$. Further, let Q_l represent the question under consideration. The audio-visual scene-aware dialog (AVSD) task requires the generation (or selection) of the answer denoted by A_l, corresponding to the question Q_l.

Our proposed pipeline to solve this task is schematically illustrated in Fig. 2. It consists of four components: (1) a *scene graph construction module*, which extracts objects and relation proposals from the video using pretrained neural network models, building a scene graph for every (temporally-sampled) video frame, (2) an *intra-frame reasoning module*, which conducts node-level and edge-level graph reasoning, producing compact feature representations for each scene graph, (3) an *inter-frame information aggregation module*, that aggregates these features within a temporal sliding window to produce a *visual memory* for each frame's scene graph (at the center frame in that window), and (4) a *semantics-controlled Transformer reasoning module*, which performs multi-modal reasoning and language modelling based on a semantic controller. In this module, we also use a newly-proposed shuffle-augmented co-attention to enable head interactions in order to boost performance. Below, we describe in detail each of these modules.

Scene Graph Representation of Video

Our approach to generate scene graphs for the video frames is loosely similar to the ones adopted in recent works such as (Pan et al. 2020; Herzig et al. 2019; Wang and Gupta 2018), and has three components: (a) object detection, (b) visual-relation detection, and (c) region of interest (ROI) re-crop on union bounding boxes. For (a), we train a Faster R-CNN model (Ren et al. 2015) on the Visual Genome dataset (Krishna et al. 2017) using the MMDetection repository (Chen et al. 2019). For a video frame I, this Faster-RCNN model produces: $\mathcal{F}_I, \mathcal{B}_I, \mathcal{S}_I = \text{RCNN}(I)$, where $\mathcal{F}_I \in \mathbb{R}^{N_o \times d_o}$ denotes the d_o-dimensional object features, $\mathcal{B}_I \in \mathbb{R}^{N_o \times 4}$ are the object bounding boxes, and $\mathcal{S}_I$ is a list of semantic labels associated with each bounding box. The pair $(\mathcal{F}_I, \mathcal{S}_I)$ forms the nodes of our scene graph. Next, to find the graph edges, we train a relation model on the VG200 dataset (Krishna et al. 2017), which contains 150 objects and 50 predicates, and apply this learned model on the frames from the given video. The output of this model is a set of

$\langle S, P, O\rangle$ triplets per frame, where S, P, and O represent the *subject*, *predicate*, and *object*, respectively. We keep the $\langle S, O\rangle$ pairs as relation proposals and discard the original predicate semantics, as the relation predicates of the model trained on VG200 are limited and fixed. Instead, we let our reasoning model learn implicit relation semantics during our end-to-end training. For the detected $\langle S, O\rangle$ pairs, we regard the union box of the bounding boxes for S and O as the predicate region of interest. Next, we apply the *ROI-align* operator (Ren et al. 2015) on the last layer of the backbone network using this union box and make the resultant feature an extra node in the scene graph.

Intra-Frame Reasoning

Representing videos directly as sequences of scene graphs leads to a complex graph reasoning problem that can be computationally challenging. To avoid this issue, we propose to hierarchically reduce this complexity by embedding these graphs into learned representation spaces. Specifically, we propose an intra-frame reasoning scheme that bifurcates a scene graph into two streams: (i) a *visual scene graph* that generates a representation summarizing the visual cues captured in the graph nodes, and (ii) a *semantic scene graph* that summarizes the graph edges. Formally, let us define a scene graph as $\mathcal{G} = \{(x_i, e_{ij}, x_j) \mid x_i, x_j \in \mathcal{V}, e_{ij} \in \mathcal{E}\}$, where $\mathcal{V}$ denotes the set of nodes consisting of single objects and $\mathcal{E}$ is the set of edges consisting of relations linking two objects. The triplet (x_i, e_{ij}, x_j) indicates that the subject node x_i and the object node x_j are connected by the directed relation edge e_{ij}. We denote by $\mathcal{G}_v$ and $\mathcal{G}_s$ the visual scene graph and the semantic scene graph respectively: the former is a graph attention network (Veličković et al. 2018) which computes an attention coefficient for each edge and updates node features based on these coefficients; the latter is based on EdgeConv (Wang et al. 2019), which computes extra edge features based on node features and then updates the node features by aggregating the edge features linked to a given node. Both networks are explained in detail next. We combine these two complementary graph neural networks in a cascade to conduct intra-frame reasoning.

Visual Scene Graph Reasoning: For M node features $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ in a scene graph, multi-head self-attention (Vaswani et al. 2017) is first performed for each pair of linked nodes. In each head k, for two linked nodes $\mathbf{x}_i$ and $\mathbf{x}_j$, the attention coefficient α_{ij}^k indicating the importance of node j to node i is computed by

$$\alpha_{ij}^k = \frac{\exp\left(\sigma\left(\mathbf{\Theta}_k^\top [\mathbf{W}_1^k \mathbf{x}_i \parallel \mathbf{W}_1^k \mathbf{x}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\sigma\left(\mathbf{\Theta}_k^\top [\mathbf{W}_1^k \mathbf{x}_i \parallel \mathbf{W}_1^k \mathbf{x}_k]\right)\right)}, \quad (1)$$

where $\parallel$ denotes feature concatenation, σ is a nonlinearity (Leaky ReLU), $\mathcal{N}_i$ indicates the neighboring graph nodes of object i (including i), $\mathbf{W}_1^k \in \mathbb{R}^{d_h \times d_{\text{in}}}$ is a (learned) weight matrix transforming the original features to a shared latent space, and $\mathbf{\Theta}_k \in \mathbb{R}^{2d_h}$ is the (learned) attention weight vector. Using the attention weights α^k and a set of learned weight matrices $\mathbf{W}_2^k \in \mathbb{R}^{d_h/K \times d_{\text{in}}}$, we update the node features as:

$$\mathbf{x}_i' = \Big\|_{k=1}^{K} \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}_2^k \mathbf{x}_j\Big). \quad (2)$$

Outputs of the K heads are concatenated to produce $\mathbf{x}_i' \in \mathbb{R}^{d_h}$, which is used as input to the semantic graph network.

Semantic Scene Graph Reasoning: This sub-module captures higher-order semantics between nodes in the scene graph. To this end, EdgeConv (Wang et al. 2019), which is a multi-layer fully-connected network $h_{\mathbf{\Lambda}}$, is employed to generate edge features $\mathbf{e}_{ij}$ from its two connected node features $(\mathbf{x}_i', \mathbf{x}_j')$: $\mathbf{e}_{ij} = h_{\mathbf{\Lambda}}(\mathbf{x}_i', \mathbf{x}_j')$, where $h_{\mathbf{\Lambda}} : \mathbb{R}^{d_h} \times \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_h}$ is a nonlinear transformation with learnable parameters $\mathbf{\Lambda}$. We then obtain the output node features $\mathbf{x}_i^\star$ by aggregating features from the edges that are directed to the object node i, i.e.,

$$\mathbf{x}_i^\star = \max_{j:(j,i) \in \mathcal{E}_i} \mathbf{e}_{ji}, \quad (3)$$

where $\mathcal{E}_i$ denotes the set of edges directed to node i. All object features inside the scene graph are updated by the above intra-frame feature aggregation.

Memory Generation with Graph Pooling: After conducting intra-frame reasoning to obtain higher-level features for each node, we pool the updated graph into a memory for further temporal aggregation. Since different frame-level scene graphs have different numbers of nodes and edges, we adopt graph average pooling (GAP) and graph max pooling (GMP) (Lee, Lee, and Kang 2019) to generate two graph memories and concatenate them to produce $V^\star$:

$$V^\star = \mathrm{GAP}(\mathbf{X}^\star, \mathcal{E}) \parallel \mathrm{GMP}(\mathbf{X}^\star, \mathcal{E}), \quad (4)$$

where $\mathcal{E}$ denotes the scene graph connection structure, and $\mathbf{X}^\star$ the M node features $\{\mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_M^\star\}$ from (3).

Inter-Frame Information Aggregation

Apart from the spatial graph representations described above, there is a temporal continuity of visual cues in the video frames that needs to be captured as well. To this end, we propose an inter-frame aggregation scheme that operates on the spatial graph embeddings. Specifically, for a sequence of scene graph memories $\langle v_1^\star, v_2^\star, \ldots, v_L^\star\rangle$ of length L produced using (4) on a sequence of L frames, we use temporal sliding windows of size τ to update the graph memory of the center frame in each window by aggregating the graph memories of its neighboring frames in that window, both in the past and in the future. Let $F \in \mathbb{R}^{2d_h \times \tau}$ denotes a matrix of graph embeddings within this window of length τ, then we perform window-level summarization over all frame memories within F as: $\beta = \mathrm{softmax}(\Gamma^\top \tanh(\mathbf{W}_\tau F))$, where $\mathbf{W}_\tau \in \mathbb{R}^{2d_h \times 2d_h}$ is a learned weight matrix, $\Gamma \in \mathbb{R}^{2d_h}$ is a weight vector, and β denotes the attention weights. We then use β to update the memory v_c of the center frame (in this window) by aggregating information across this window, as: $v_c' = F\beta^\top$. Repeating this step for all sliding windows, we get the final visual graph memory $V' = \langle v_1', v_2', \ldots, v_L'\rangle$ aggregating both spatial and temporal information. We also augment these visual features with their temporally-aligned audio embeddings $\langle s_1, s_2, \cdots, s_L\rangle$ produced using an AudioSet VGGish network (Hershey et al. 2017).

Semantics-Controlled Transformer Reasoning

The above modules encode a video into a sequence of graph memories via reasoning on visual and semantic scene

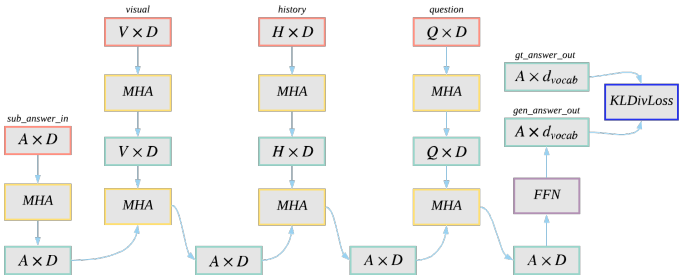

Figure 3: depicts the sequential attention flow in our semantics-controlled Transformer. MHA stands for multi-head attention. FFN is short for feed-forward networks. The acronyms A, V, H, and Q stand for the answers, visual memory, caption/dialog history, and the question, respectively.

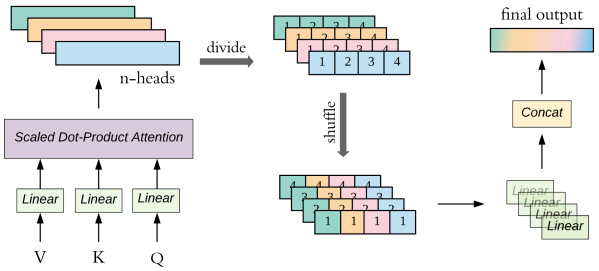

Figure 4: An illustration of our multi-head shuffled Transformer, where we shuffle the output of each head before passing it on to the FFN module.

graphs. Besides encoding audio-visual information, we also need to encode the text information available in the AVSD task. For the sentence generation task, we propose to generate the answer autoregressively (Anderson et al. 2018; Hori et al. 2018), i.e., predict the next word in the answer from the vocabulary based on source sequences including the visual memory, query Q_l, caption C, the dialog history $H = \langle (Q_1, A_1), \ldots, (Q_{l-1}, A_{l-1}) \rangle$, and the partially generated answer so far, denoted A_l^{in} (see Fig. 2 and Fig. 3). This sub-answer A_l^{in} forms the semantics that control the attention on the various modalities to generate the next word. As shown in Fig. 3, our semantics-controlled Transformer module consists of a graph encoder, a text encoder, and a multimodal decoder. It takes in source sequences and outputs the probability distribution of the next token for all tokens in the vocabulary. We detail the steps in this module next.

Encoder: We first use Transformer to embed all text sources $(H, C, Q_l, A_l^{\text{in}})$ using token and positional embeddings, generating feature matrices e_h, e_c, e_q, and e_a, each of the same feature dimensionality d_h. We also use a single-layer fully-connected network to transfer the audio-augmented visual memories in V' to d_h-dimensional features e_v that match the dimension of the text sources. Next, for the answer generation task, the input sub-answer (generated so far) e_a is encoded with a Transformer consisting of multi-head self-attention to get hidden representations h_{enc}^a:

$$h_{\text{enc}}^a = \text{FFN}^a(\text{Attention}(\mathbf{W}_{\mathbf{Q}}^a e_a, \mathbf{W}_{\mathbf{K}}^a e_a, \mathbf{W}_{\mathbf{V}}^a e_a)), \quad (5)$$

where $\mathbf{W}_{\mathbf{Q}}^a$, $\mathbf{W}_{\mathbf{K}}^a$, $\mathbf{W}_{\mathbf{V}}^a$ are weight matrices for query, key, and value respectively (Vaswani et al. 2017), FFN^a is a feed-forward module consisting of two fully-connected layers with ReLU in between. The Attention function is defined as in (Vaswani et al. 2017):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}})\mathbf{V}, \quad (6)$$

with a scaling factor $\sqrt{d_h}$ that maintains the order of magnitude in features, and $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ represent the query, key, and value triplets as described in (Vaswani et al. 2017). After encoding the input sub-answer, we conduct co-attention in turn for each of the other text and visual embeddings e_j, where $j \in \{v, c, h, q\}$, with a similar Transformer architecture. That is, the encoding h_{enc}^j for a given embedding type e_j is obtained by using the encoding $h_{\text{enc}}^{j'}$ for the previous embedding type $e_{j'}$ as query (Fig. 3):

$$h_{\text{enc}}^j = \text{FFN}^j(\text{Attention}(\mathbf{W}_{\mathbf{Q}}^j h_{\text{enc}}^{j'}, \mathbf{W}_{\mathbf{K}}^j e_j, \mathbf{W}_{\mathbf{V}}^j e_j)). \quad (7)$$

In our implementation, the embeddings for history and caption are concatenated as $e_{c+h} = e_c || e_h$. Processing occurs in the following order: starting from h_{enc}^a, we compute h_{enc}^v, then h_{enc}^{c+h}, and later h_{enc}^q. Finally, we get a feature vector $h_{\text{enc}}^\star$ that fuses all the information from the text and audio-visual sources by concatenating these multi-modal features.

Multi-head Shuffled Transformer: In this paper, we also propose to utilize head shuffling to further improve the performance of the Transformer structure as shown in Fig. 4. In the original Transformer (Vaswani et al. 2017), the feature vectors of all heads are directly concatenated before being fed into the last fully-connected layer. Thus, there is no interaction between those heads from the start to the end. To enable the interactions across heads, we propose to divide each head and shuffle all head vectors before passing them on to separate fully-connected layers. The outputs are finally concatenated in a late fusion style. This scheme is similar to ShuffleNet (Zhang et al. 2018), with the key difference that here we conduct shuffling between different heads within the multi-head attention, while in ShuffleNet the shuffling is across channels. Our empirical results show that our shuffling operation results in better generalization of the model.

Decoder: For the generation setting, with the final encoded feature $h_{\text{enc}}^\star$, we use a feed-forward network with softmax to predict the next token probability distribution P over all tokens in the vocabulary $\mathcal{V}$; i.e., $P = \text{softmax}(\text{FFN}(h_{\text{enc}}^\star))$. In the testing stage, we conduct beam search with b beams to generate an answer sentence.

Loss Function: Let $\mathcal{P}$ denote the collection of all next-token probability distributions $P_j \in \mathbb{R}^{|\mathcal{V}|}$, $j = 1, \ldots, N$ for batch size N, and let $\mathcal{G}$ be the collection of respective distributions G_j for the ground truth answer tokens. For the generation setting, we apply label smoothing (Müller, Kornblith, and Hinton 2019) to account for the sparsity of the token distributions, leading to $\tilde{\mathcal{G}}$. We use the cross-entropy (CE) loss between the predicted and the smoothed ground truth

distributions to train our model end-to-end:

$$\mathcal{L} = \mathrm{CE}(\mathcal{P}|\tilde{\mathcal{G}}) = -\frac{1}{N}\sum_{j=1}^{N}\sum_{u\in\mathcal{V}}\tilde{G}_j(u)\log P_j(u). \quad (8)$$

For the retrieval setting, we first concatenate the feature embeddings of the query and the various input modalities obtained from the Encoder module of our network (e_h, e_c, e_q, e_v). Next, the candidate answers are embedded into this joint space using LSTMs, and a dot product is taken between the concatenated inputs and embeddings of each of the answer candidates. We then train this model with the binary cross-entropy loss.

Experiments

In this section, we detail our experimental setup, datasets, and evaluation protocols, before furnishing our results.

Dataset and Evaluation: We use the audio-visual scene-aware dialog (AVSD) dataset (Alamri et al. 2019) for our experiments, which is the benchmark dataset for this task. This dataset emulates a real-world human-human natural conversation scenario about an audio-visual clip. See (Alamri et al. 2019) for details of this task and the dataset. We evaluate on two variants of this dataset corresponding to annotations available for the DSTC-7 and DSTC-8 challenges,[2] consisting of 7,659, 1,787, 1,710, and 1,710 dialogs for training, validation, DSTC-7 testing, and DSTC-8 testing, respectively for the answer generation task. The quality of the generated answers is evaluated using the standard MS COCO evaluation metrics (Chen et al. 2015), such as BLEU, METEOR, ROUGE-L, and CIDEr. Apart from the answer generation task (Hori et al. 2018), we also report experiments on the answer selection task, described in (Alamri et al. 2019) using their annotations and ground truth answers. This task requires selecting the answer to a question from a set of 100 answers. Specifically, in this task, an algorithm is to present a ranking over a set of 100 provided answers, with ideally the correct answer ranked as the first. The evaluation is then based on the mean retrieval rank over the test set.

Data Processing: We follow (Le et al. 2019) to perform text preprocessing which include lowercasing, tokenization, and building a vocabulary by only selecting tokens that occur at least five times. Thus, we use a vocabulary with 3,254 words, both for the generation and retrieval tasks.

Feature Extraction: Motivated by (Anderson et al. 2018), we train a detector on Visual Genome with 1601 classes and 401 attributes, which incorporates a "background" label and a "no-attribute" label. We use ResNext-101 as the neural backbone with a multiscale feature pyramid network. We further use fine-grained ROI-alignment instead of ROI-pooling for better feature representation. We extract the 1024-D features for the 36 highest scoring regions, their class labels, and attributes. After extracting the region features, we apply a pretrained relationship detector (Zhang et al. 2019) to find visually-related regions. We calculate the minimal bounding box which covers two visually-related regions and perform ROI-alignment to get compact representations for relationship regions. In order to incorporate audio into the STSGR framework, we extract AudioSet VGG-ish features (Hershey et al. 2017) from the audio stream for every video. These are 128-D features obtained from the AudioSet VGG-ish CNN, pretrained on 0.96s Mel Spectrogram patches on the AudioSet data (Gemmeke et al. 2017).

[2]https://sites.google.com/dstc.community/dstc8/home

Method	B4	MET	ROUGE	CIDEr
STSGR full model	**0.133**	**0.165**	**0.361**	**1.265**
w/o head shuffling	0.127	0.161	0.354	1.208
w/o GAT	0.118	0.160	0.347	1.125
w/o EdgeConv	0.131	0.162	0.356	1.244
w/o union box	0.124	0.163	0.352	1.175
w/o visual features	0.127	0.160	0.356	1.203
w/o temporal	0.125	0.164	0.357	1.212
STSGR + audio	**0.133**	**0.165**	**0.362**	**1.272**

Table 1: Ablation study using AVSD@DSTC7 dataset.

Model Training: We set our Transformer hyperparameters following (Vaswani et al. 2017). The feature dimension is 512, while the inner-layer dimension of the feed-forward network is set to 2048. For multi-head attention, we maintain $h = 8$ parallel attention heads and apply shuffling to boost performance. For the semantic labels, we build a 300-D embedding layer for the 1651 words in the vocabulary (which is available with the dataset), and initialize the embeddings using GloVe word vectors (Pennington, Socher, and Manning 2014). For semantic labels consisting of more than one word, we use the average word embedding as the label embedding. Our model is trained on one Nvidia Titan XP GPU with Adam optimizer (Kingma and Ba 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The batch size is set to 16 and we adopt the warm-up strategy as suggested in (Vaswani et al. 2017) for learning rate adjustment with about 10,000 steps.

Baselines: We consider the following four baselines on the generation task: (i) *Baseline* (Hori et al. 2019), (ii) *Multimodal Attention* (Hori et al. 2019), that uses attention over concatenated features, (iii) *Simple* (Schwartz et al. 2019) that uses factor-graph attention on the modalities, and (iv) *MTN* (Le et al. 2019) that applies self-attention and co-attention to aggregate multi-modal information. For the retrieval task, we compare our method against the state-of-the-art method of (Alamri et al. 2019) on the DSTC-7 split.

Ablation Study: To understand the importance of each component in our model, Table 1 details an ablation study. We analyze several key components: (i) shuffling in the Transformer structure, (ii) visual and semantic graph, (iii) ROI Recrop on the union bounding boxes, and (iv) temporal aggregation. From the table, we see that Graph Attention Network (GAT), which is used to produce the visual scene graph, is important to aggregate information from neighboring nodes (e.g., improving CIDEr from 1.125 to 1.265), while EdgeConv, used in the semantic graph, offers some improvement (e.g., CIDEr from 1.244 to 1.265). Moreover, the use of shuffling in the multi-head Transformer architecture boosts the performance significantly (from 1.208 to 1.265 for CIDEr). We can also conclude that union bounding boxes, semantic labels, and inter-frame aggregation contribute to stabilize the generation performance. Overall, by adopting all these key components, the full model outper-

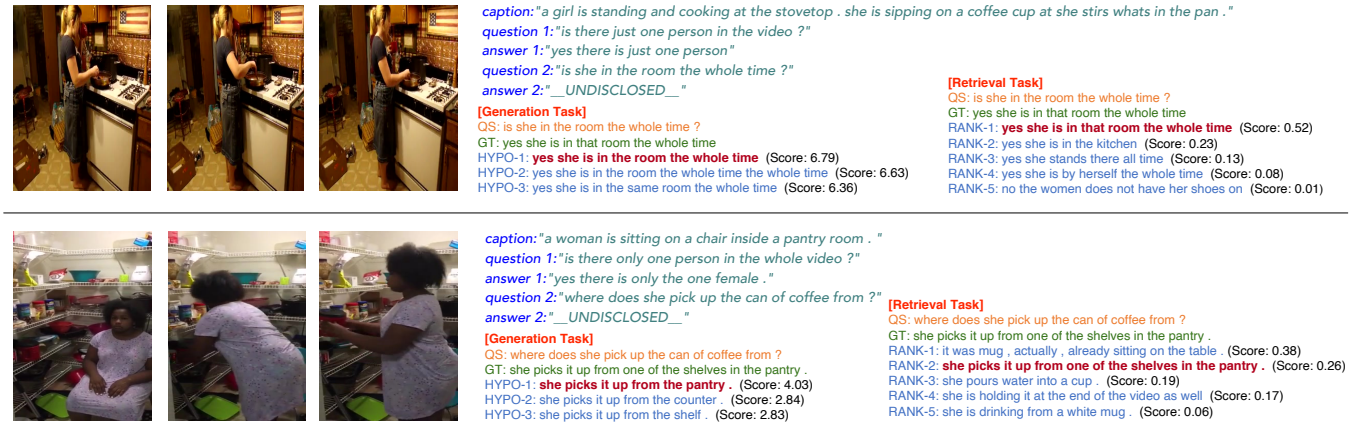

Figure 5: Qualitative results from our model on both generation and retrieval tasks of AVSD. Left: input video frames, Top-right: caption and dialog history, Bottom-middle: top-3 generated answers with confidence scores. Bottom-right: top-5 ranked candidate answers with confidence scores.

AVSD@DSTC7				
Method	B4	MET	ROUGE	CIDEr
Baseline	0.075	0.110	0.275	0.701
Multi-modal Attention	0.078	0.113	0.277	0.727
Simple	0.091	0.125	0.307	0.883
MTN	0.128	0.162	0.355	1.249
STSGR (Ours)	**0.133**	**0.165**	**0.362**	**1.272**
AVSD@DSTC8				
Baseline	0.289	0.210	0.480	0.651
Multi-modal Attention	0.293	0.212	0.483	0.679
Simple	0.311	0.224	0.502	0.766
MTN	0.352	0.263	0.547	0.978
STSGR (Ours)	**0.357**	**0.267**	**0.553**	**1.004**

Table 2: Comparisons of our method against the state of the art on the AVSD test splits for DSTC7 and DSTC8.

Method	Full model	w/o C	w/o C, H
Alamri et al. (2019)	5.88	N/A	7.41
Hori et al. (2019)	5.60	N/A	7.23
MTN w/o audio	4.51	4.90	6.85
MTN w/ audio	4.29	4.78	6.46
STSGR	4.33	4.67	6.54
STSGR w/ audio	**4.08**	**4.55**	**5.91**

Table 3: State-of-the-art comparisons on answer selection as measured by Mean Retrieval Rank (lower the better).

forms all the ablations. From Tables 1 and 3, we notice that incorporation of audio helps improve the performance of our model. For instance, on the retrieval setting we observe that incorporating audio lowers the Mean-Retrieval Rank noticeably down to 4.08 from 4.33 for the full model and to 5.91 from 6.54 when no language context is available.

Comparisons to the State of the Art: In Table 2, we compare STSGR against baseline methods on various quality metrics based on ground-truth answers. As is clear, our approach achieves better performance against all the baselines.

The performance on the answer selection task (mean retrieval rank) is provided in Table 3, demonstrating clearly state-of-the-art results against the baseline in (Alamri et al. 2019). We also show that including audio into the STSGR representation helps improve the mean retrieval rank.

Qualitative Results and Discussion: In Fig. 5, we provide two qualitative results from our STSGR model. For the first case, our model consistently detects the woman in the frames and finds that she maintains many connections with other objects inside the scene throughout the whole video, thus our model makes/selects the correct answer with high confidence. For the second case, the clutter background poses a challenge to our model. However, STSGR can still generate/rank the correct answer in top-2. In general, we find that STSGR can answer spatial and temporal questions very well. This is quantitatively evidenced by observing that while both STSGR and MTN (Le et al. 2019) use similar backends, they differ in the input representations (I3D in (Le et al. 2019) vs. scene graphs in ours), and our model outperforms MTN noticeably (1.272 vs 1.249 on CIDEr, Table 2), substantiating the importance of our STSGR representation.

Conclusions

We presented a novel hierarchical graph representation learning and Transformer reasoning framework for the problem of audio-visual scene-aware dialog. Specifically, our model generates object, frame, and video-level representations that are systematically integrated to produce visual memories, which are sequentially fused to the encodings of other modalities (dialog history, etc.) conditioned on the input question using a multi-head shuffled Transformer. Experiments demonstrate the benefits of our framework for both generation/selection tasks on the AVSD benchmark.

Acknowledgments

Shijie Geng, Peng Gao, and Moitreya Chatterjee worked on this project during MERL internships.

References

Alamri, H.; Cartillier, V.; Das, A.; Wang, J.; Cherian, A.; Essa, I.; Batra, D.; Marks, T. K.; Hori, C.; Anderson, P.; et al. 2019. Audio Visual Scene-Aware Dialog. In *CVPR*.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In *ICCV*.

Ben-Younes, H.; Cadene, R.; Thome, N.; and Cord, M. 2019. Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *AAAI*.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv:1906.07155* .

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325* .

Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M.; Parikh, D.; and Batra, D. 2017. Visual dialog. In *CVPR*.

Deruyttere, T.; Vandenhende, S.; Grujicic, D.; Van Gool, L.; and Moens, M.-F. 2019. Talk2Car: Taking Control of Your Self-Driving Car. In *EMNLP-IJCNLP*.

Drossos, K.; Lipping, S.; and Virtanen, T. 2019. Clotho: An Audio Captioning Dataset. *arXiv:1910.09387* .

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*.

Gan, Z.; Cheng, Y.; Kholy, A. E.; Li, L.; Liu, J.; and Gao, J. 2019. Multi-step reasoning via recurrent dual attention for visual dialog. In *ACL*.

Gao, P.; Jiang, Z.; You, H.; Lu, P.; Hoi, S. C. H.; Wang, X.; and Li, H. 2019a. Dynamic Fusion With Intra- and Inter-Modality Attention Flow for Visual Question Answering. In *CVPR*.

Gao, P.; You, H.; Zhang, Z.; Wang, X.; and Li, H. 2019b. Multi-Modality Latent Interaction Network for Visual Question Answering. In *ICCV*.

Gemmeke, J. F.; Ellis, D. P.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R. C.; Plakal, M.; and Ritter, M. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*.

Geng, S.; Zhang, J.; Fu, Z.; Gao, P.; Zhang, H.; and de Melo, G. 2020. Character Matters: Video Story Understanding with Character-Aware Relations. *arXiv:2005.08646* .

Geng, S.; Zhang, J.; Zhang, H.; Elgammal, A.; and Metaxas, D. N. 2019. 2nd Place Solution to the GQA Challenge 2019. *arXiv:1907.06794* .

Ghosh, S.; Burachas, G.; Ray, A.; and Ziskind, A. 2019. Generating natural language explanations for visual question answering using scene graphs and visual attention. *arXiv:1902.05715* .

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A. 2019. Video action transformer network. In *CVPR*.

Hershey, S.; Chaudhuri, S.; Ellis, D. P.; Gemmeke, J. F.; Jansen, A.; Moore, R. C.; Plakal, M.; Platt, D.; Saurous, R. A.; Seybold, B.; et al. 2017. CNN architectures for large-scale audio classification. In *ICASSP*.

Herzig, R.; Levi, E.; Xu, H.; Gao, H.; Brosh, E.; Wang, X.; Globerson, A.; and Darrell, T. 2019. Spatio-temporal action graph networks. In *ICCV Workshops*.

Hori, C.; Alamri, H.; Wang, J.; Wichern, G.; Hori, T.; Cherian, A.; Marks, T. K.; Cartillier, V.; Lopes, R. G.; Das, A.; et al. 2019. End-to-end audio visual scene-aware dialog using multimodal attention-based video features. In *ICASSP*.

Hori, C.; Hori, T.; Lee, T.-Y.; Zhang, Z.; Harsham, B.; Hershey, J. R.; Marks, T. K.; and Sumi, K. 2017. Attention-based multimodal fusion for video description. In *ICCV*.

Hori, C.; Hori, T.; Wichern, G.; Wang, J.; Lee, T.-y.; Cherian, A.; and Marks, T. K. 2018. Multimodal Attention for Fusion of Audio and Spatiotemporal Features for Video Description. In *CVPR Workshops*.

Jain, A.; Zamir, A. R.; Savarese, S.; and Saxena, A. 2016. Structural-rnn: Deep learning on spatio-temporal graphs. In *CVPR*.

Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In *CVPR*.

Ji, J.; Krishna, R.; Fei-Fei, L.; and Niebles, J. C. 2020. Action Genome: Actions as Composition of Spatio-temporal Scene Graphs. In *CVPR*.

Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Shamma, D.; Bernstein, M.; and Fei-Fei, L. 2015. Image retrieval using scene graphs. In *CVPR*.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV* .

Le, H.; and Hoi, S. C. 2020. Video-Grounded Dialogues with Pretrained Generation Language Models. In *ACL*.

Le, H.; Sahoo, D.; Chen, N. F.; and Hoi, S. C. 2019. Multimodal Transformer Networks for End-to-End Video-Grounded Dialogue Systems. In *ACL*.

Lee, J.; Lee, I.; and Kang, J. 2019. Self-Attention Graph Pooling. In *ICML*.

Li, X.; and Jiang, S. 2019. Know more say less: Image captioning based on scene graphs. *IEEE Transactions on Multimedia* .

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2016. Visual Relationship Detection with Language Priors. In *ECCV*.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? In *NeurIPS*.

Norcliffe-Brown, W.; Vafeias, S.; and Parisot, S. 2018. Learning conditioned graph structures for interpretable visual question answering. In *NeurIPS*.

Pan, B.; Cai, H.; Huang, D.-A.; Lee, K.-H.; Gaidon, A.; Adeli, E.; and Niebles, J. C. 2020. Spatio-Temporal Graph for Video Captioning with Knowledge Distillation. In *CVPR*.

Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global vectors for word representation. In *EMNLP*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*.

Schwartz, I.; Schwing, A. G.; and Hazan, T. 2019. A Simple Baseline for Audio-Visual Scene-Aware Dialog. In *CVPR*.

Schwartz, I.; Yu, S.; Hazan, T.; and Schwing, A. G. 2019. Factor Graph Attention. In *CVPR*.

Shi, L.; Geng, S.; Shuang, K.; Hori, C.; Liu, S.; Gao, P.; and Su, S. 2020a. Multi-Layer Content Interaction Through Quaternion Product For Visual Question Answering. In *ICASSP*.

Shi, L.; Shuang, K.; Geng, S.; Su, P.; Jiang, Z.; Gao, P.; Fu, Z.; de Melo, G.; and Su, S. 2020b. Contrastive Visual-Linguistic Pretraining. *arXiv:2007.13135* .

Thomason, J.; Padmakumar, A.; Sinapov, J.; Walker, N.; Jiang, Y.; Yedidsion, H.; Hart, J.; Stone, P.; and Mooney, R. J. 2019. Improving grounded natural language understanding through human-robot dialog. In *ICRA*.

Tsai, Y.-H. H.; Divvala, S.; Morency, L.-P.; Salakhutdinov, R.; and Farhadi, A. 2019. Video relationship reasoning using gated spatio-temporal energy graph. In *CVPR*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to sequence-video to text. In *ICCV*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *CVPR*.

Wang, X.; and Gupta, A. 2018. Videos as Space-Time Region Graphs. In *ECCV*.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)* .

Wu, Q.; Wang, P.; Shen, C.; Reid, I.; and Van Den Hengel, A. 2018. Are you talking to me? reasoned visual dialog generation through adversarial learning. In *CVPR*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Yang, H.; Chaisorn, L.; Zhao, Y.; Neo, S.-Y.; and Chua, T.-S. 2003. VideoQA: question answering on news video. In *ACM Multimedia*.

Yang, X.; Tang, K.; Zhang, H.; and Cai, J. 2019. Auto-encoding scene graphs for image captioning. In *CVPR*.

Yeh, Y.-T.; Lin, T.-C.; Cheng, H.-H.; Deng, Y.-H.; Su, S.-Y.; and Chen, Y.-N. 2019. Reactive multi-stage feature fusion for multimodal dialogue modeling. *arXiv:1908.05067* .

Zhang, J.; Shih, K. J.; Elgammal, A.; Tao, A.; and Catanzaro, B. 2019. Graphical Contrastive Losses for Scene Graph Generation. In *CVPR*.

Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*.

Zhu, H.; Luo, M.; Wang, R.; Zheng, A.; and He, R. 2020. Deep Audio-Visual Learning: A Survey. *arXiv:2001.04758* .