VGNMN: Video-grounded Neural Module Network to Video-Grounded Language Tasks

Hung Le[†§], **Nancy F. Chen**[§], **Steven C.H. Hoi**[†‡]
[†] Singapore Management University
hungle.2018@smu.edu.sg
[§]Institute for Infocomm Research, A*STAR
nfychen@i2r.a-star.edu.sg
[‡] Salesforce Research Asia
shoi@salesforce.com

Abstract

Neural module networks (NMN) have achieved success in image-grounded tasks such as Visual Question Answering (VQA) on synthetic images. However, very limited work on NMN has been studied in the video-grounded language tasks. These tasks extend the complexity of traditional visual tasks with the additional visual temporal variance. Motivated by recent NMN approaches on image-grounded tasks, we introduce Video-grounded Neural Module Network (VGNMN) to model the information retrieval process in video-grounded language tasks as a pipeline of neural modules. VGNMN first decomposes all language components to explicitly resolve any entity references and detect corresponding action-based inputs from the question. The detected entities and actions are used as parameters to instantiate neural module networks and extract visual cues from the video. Our experiments show that VGNMN can achieve promising performance on two video-grounded language tasks: video QA and video-grounded dialogues.

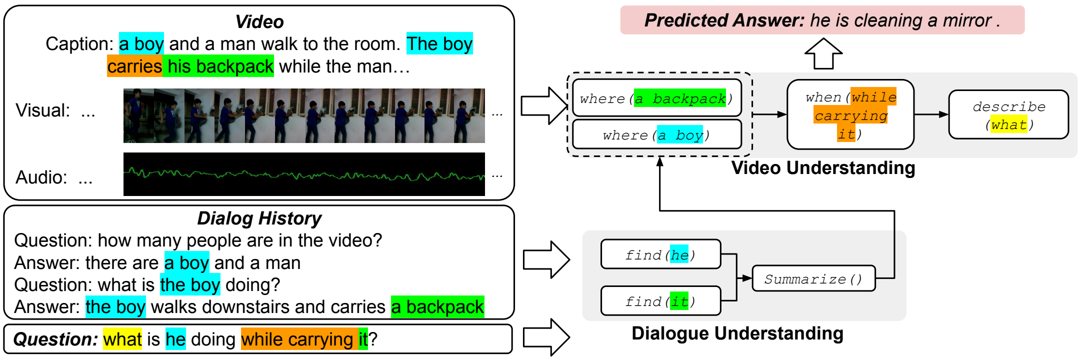
1 Introduction

Vision-language tasks have been studied to build intelligent systems that can perceive information from multiple modalities, such as images, videos, and text. Extended from imaged-grounded tasks, e.g. (Antol et al., 2015), recently Jang et al. (2017); Lei et al. (2018) propose to use video as the grounding features. This modification poses a significant challenge to previous image-based models with the additional temporal variance through video frames. Recently Alamri et al. (2019) further develop video-grounded language research into the dialogue domain. In the proposed task, *video-grounded dialogues*, the dialogue agent is required to answer questions about a video over multiple dialogue turns. Using Figure 1 as an example, to answer questions correctly, a dialogue agent has to resolve references in dialogue context, e.g. "he" and "it", and identify the original entity, e.g. "a boy" and "a backpack". Besides, the dialogue agent also needs to identify the actions of these entities, e.g. "carrying a backpack" to retrieve information from the video.

Current state-of-the-art approaches to video-grounded language tasks, e.g. Le et al. (2019b); Fan et al. (2019) have achieved remarkable performance through the use of deep neural networks to retrieve grounding video signals based on language inputs. However, these approaches often assume the reasoning structure, including resolving references of entities and detecting the corresponding actions to retrieve visual cues, is implicitly learned. An explicit reasoning structure becomes more beneficial as the tasks complicate in two scenarios: video with complex spatial and temporal dynamics, and language inputs with sophisticated semantic dependencies, e.g. questions positioned in a dialogue context. These scenarios often challenge researchers to interpret model hidden outputs, identify errors, and assess model reasoning capability.

Similar challenges have been observed in image-grounded tasks in which deep neural networks often exhibit shallow understanding capability as they simply exploit superficial visual cues (Agrawal et al., 2016; Goyal et al., 2017; Feng et al., 2018; Serrano and Smith, 2019). Andreas et al. (2016b) propose neural model networks (NMNs) by decomposing a question into sub-sequences called *program* and assembling a network of neural operations. Motivated by this line of research, we propose a new approach, VGNMN, to video-grounded language tasks. Our approach benefits from integrating neural networks with a compositional reasoning structure to exploit low-level information signals in video. An example of the reasoning structure can be seen on the right side of Figure 1.

Video-grounded Neural Module Network (VGNMN) tackles video understanding through



Caption: a boy and a man walk to the room. The boy carries his backpack while the man…

Question: how many people are in the video?
Answer: there are a boy and a man
Question: what is the boy doing?
Answer: the boy walks downstairs and carries a backpack

Question: what is he doing while carrying it?

Predicted Answer: he is cleaning a mirror .

Figure 1: A sample video-grounded dialogue with a demonstration of a reasoning process

action and entity-level NMNs to retrieve video features. We first decompose question into a set of entities and extract video features related to these entities. VGNMN then extracts the temporal steps by focusing on relevant actions that are associated with these entities. VGNMN is analogous to how human performs reasoning by gradually retrieving linguistic and visual information based on a set of high-level information, such as entities and actions.

To tackle dialogue understanding, VGNMN is trained to resolve any co-reference in language inputs, e.g. questions in a dialogue context, to identify the unique entities in each dialogue. Previous approaches to video-grounded dialogues often obtain question global representations in relation to dialogue context. These approaches might be suitable to represent general semantics in open-domain dialogues (Serban et al., 2016) but they are not ideal to detect fine-grained entity-based information as the dialogue context evolves over several turns. However, in a video-grounded dialogue, it is empirical to dissect the dialogue context and detect the specific entity dependencies between questions and past dialogue turns.

In summary, we introduce a neural module network approach to video-grounded language tasks through a reasoning pipeline with entity and action representations applied to the spatio-temporal dynamics of video. In our evaluation, we achieve competitive performance on the large-scale benchmark Audio-visual Scene-aware Dialogues (AVSD) (Alamri et al., 2019). Moreover, our experiments and ablation analysis indicate better model interpretability and robustness against the complexity in videos and dialogues. Finally, We adapt VGNMN for video QA and obtain significant performance on the TGIF-QA benchmark (Jang et al., 2017) across all tasks.

2 Related Work

Our work is related to three research areas: video-language understanding, video-grounded dialogues, and neural module networks.

2.1 Video-Language Understanding

Video QA has been a proxy for evaluating a model's understanding capability of language and video and the task is treated as a visual information retrieval task. Jang et al. (2017); Gao et al. (2018); Jiang et al. (2020) propose to learn attention guided by question global representation to retrieve spatial-level and temporal-level visual features. Li et al. (2019); Fan et al. (2019); Jiang and Han (2020) model interaction between all pairs of question token-level representations and temporal-level features of the input video through similarity matrix, memory networks, and graph networks respectively. Gao et al. (2019); Le et al. (2019c, 2020b); Lei et al. (2020); Huang et al. (2020) extends the previous approach by dividing a video into equal segments, sub-sampling video frames, or considering object-level representations of input video. We propose to replace token-level and global question representations with compositional question representations of specific entities and actions. Visual information is retrieved as a sequential reasoning program over these entities and actions for more transparent and accurate information extraction.

2.2 Video-grounded Dialogues

Extended from video QA, video-grounded dialogue is an emerging task that combines dialogue response generation and video-language understanding research. As an orthogonal extension, this task entails a novel requirement for models to learn dialogue semantics and decode entity co-references in questions. Nguyen et al. (2018); Hori et al. (2019); Hori et al. (2019); Sanabria et al. (2019); Le et al.

(2019a,b) extend traditional QA models by adding dialogue history neural encoders. Kumar et al. (2019) enhances dialogue features with topic-level representations to express the general topic in each dialogue. Schwartz et al. (2019) treats each dialogue turn as an independent sequence and allows interaction between questions and each dialogue turn. Le et al. (2019b) encodes dialogue history as a sequence with embedding and positional representations. Sanabria et al. (2019) considers the task as a video summary task and concatenates question and dialogue history into a single sequence and proposes to transfer parameter weights from a large-scale video summary model. Different from prior work, we dissect the question sequence and explicitly detect and decode any entities and their references. Our models also benefit from the insights on how models rely on the bias of component linguistic inputs to extract visual information.

2.3 Neural Module Network

Extending from research on neural semantic parsing (Jia and Liang, 2016; Liang et al., 2017), Andreas et al. (2016b,a) introduce NMNs to address visual QA by decomposing questions into linguistic sub-structures, known as programs, to instantiate a network of neural modules. NMN models have achieved significant success in synthetic image domains where a complex reasoning process is required (Johnson et al., 2017b; Hu et al., 2018; Han et al., 2019). Yi et al. (2018); Han et al. (2019); Mao et al. (2019) improve previous work by decoupling visual-language understanding and visual concept learning. Our work is related to the recent work that extends NMN models to real data domains. For instance, Kottur et al. (2018); Jiang and Bansal (2019); Gupta et al. (2020) extend NMNs to visual dialogues and reading comprehension tasks. In this paper, we introduce a new approach that exploits NMN to learn dependencies between the composition in language inputs and the spatio-temporal dynamics in videos. This is not present in prior NMN models which are designed to apply on a two-dimensional image input without temporal variance. We propose to construct a reasoning structure from text, from which detected entities are used to extract visual information in the spatial space and detected actions are used to find visual information in the temporal space.

3 Method

3.1 Task Definition

The input to the model consists of a dialogue $\mathcal{D}$ which is grounded on a video $\mathcal{V}$. The input components include the question of current dialogue turn $\mathcal{Q}$, dialogue history $\mathcal{H}$, and the features of the input video, including visual and audio input. The output is a dialogue response, denoted as $\mathcal{R}$. Each text input component is a sequence of words $w_1, ..., w_m \in \mathbb{V}^{in}$, the input vocabulary. Similarly, the output response $\mathcal{R}$ is a sequence of tokens $w_1, ..., w_n \in \mathbb{V}^{out}$, the output vocabulary. The objective of the task is the generation objective that output answers of the current dialogue turn t:

$$\begin{aligned}\hat{\mathcal{R}}_t &= \arg\max_{\mathcal{R}_t} P(\mathcal{R}_t|\mathcal{V}, \mathcal{H}_t, \mathcal{Q}_t; \theta) \\ &= \arg\max_{\mathcal{R}_t} \prod_{n=1}^{L_\mathcal{R}} P_m(w_n|\mathcal{R}_{t,1:n-1}, \mathcal{V}, \mathcal{H}_t, \mathcal{Q}_t; \theta)\end{aligned}$$

In a Video-QA task, the dialogue history $\mathcal{H}$ is simply absent and the output response is typically collapsed to a single-token response.

3.2 Encoders

Text Encoder. A text encoder is shared to encode text inputs, including dialogue history, questions, and captions. The text encoder converts each text sequence $\mathcal{X} = w_1, ..., w_m$ into a sequence of embeddings $X \in \mathbb{R}^{m \times d}$. We use a trainable embedding matrix to map token indices to vector representations of d dimensions through a mapping function ϕ. These vectors are then integrated with ordering information of tokens through a positional encoding function with layer normalization (Ba et al., 2016; Vaswani et al., 2017). The embedding and positional representations are combined through element-wise summation. The encoded dialogue history and question of the current turn are defined as $H = \text{Norm}(\phi(\mathcal{H}) + \text{PE}(\mathcal{H})) \in \mathbb{R}^{L_\text{H} \times d}$ and $Q = \text{Norm}(\phi(\mathcal{Q}) + \text{PE}(\mathcal{Q})) \in \mathbb{R}^{L_\text{Q} \times d}$.

To decode program and response sequences autorepressively, a special token "*_sos*" is concatenated as the first token w_0. The decoded token w_1 is then appended to w_0 as input to decode w_2 and so on. Similarly to input source sequences, at decoding time step j, the input target sequence is encoded to obtain representations for dialogue understanding program $P_\text{dial}|_0^{j-1}$, video understanding program $P_\text{vid}|_0^{j-1}$, and system response $R|_0^{j-1}$. We combine

Module	Input	Output	Description
find	$\mathtt{P},\mathtt{H}$	$\mathtt{H}_{\text{ent}}$	For related entities in question, select the relevant tokens from dialogue history
summarize	$\mathtt{H}_{\text{ent}},\mathtt{Q}$	$\mathtt{Q}_{\text{ctx}}$	Based on contextual entity representations, summarise the question semantics
where	$\mathtt{P},\mathtt{V}$	$\mathtt{V}_{\text{ent}}$	Select the relevant spatial position corresponding to original (resolved) entities
when	$\mathtt{P},\mathtt{V}_{\text{ent}}$	$\mathtt{V}_{\text{ent+act}}$	Select the relevant entity-aware temporal steps corresponding to the action parameter
describe	$\mathtt{P},\mathtt{V}_{\text{ent+act}}$	$\mathtt{V}_{\text{ctx}}$	Select visual entity-action features based on non-binary question types
exist	$\mathtt{Q},\mathtt{V}_{\text{ent+act}}$	$\mathtt{V}_{\text{ctx}}$	Select visual entity-action features based on binary (yes/no) question types

Table 1: Description of the modules and their functionalities. We denote P as the parameter to instantiate each module, H as the dialogue history, Q as the question of the current dialogue turn, and V as video input.

vocabulary of input and output sequences and share the embedding matrix $E \in \mathbb{R}^{|\mathbb{V}|\times d}$ where $\mathbb{V} = \mathbb{V}^{in} \cap \mathbb{V}^{out}$.

Video Encoder. To encode video, we use pretrained models to extract visual and audio features. We denote F as the sampled video frames or video clips. For object-level visual features, we denote O as the maximum number of objects considered in each frame. The resulting output from a pretrained object detection model is $Z_{\text{obj}} \in \mathbb{R}^{F\times O\times d_{\text{vis}}}$. We concatenate each object representation with the corresponding coordinates projected to d_{vis} dimensions. We also make use of a CNN-based pretrained model to obtain features of temporal dimension $Z_{\text{cnn}} \in \mathbb{R}^{F\times d_{\text{vis}}}$. The audio feature is obtained through a pretrained audio model, $Z_{\text{aud}} \in \mathbb{R}^{F\times d_{\text{aud}}}$. We passed all video features through a linear transformation layer with ReLU activation to the same embedding dimension d.

3.3 Neural Modules

We introduce neural modules that are used to assemble an executable program constructed by the generated sequence from question parsers. We provide an overview of neural modules in Table 1 and demonstrate dialogue understanding and video understanding modules in Figure 2 and 3 respectively. Each module parameter, e.g. "a backpack", is extracted from the parsed program (See Section 3.4). For each parameter, we denote $P \in \mathbb{R}^d$ as the average pooling of component token embeddings.

find(P,H)$\rightarrow\mathbf{H}_{\text{ent}}$. This module handles entity tracing by obtaining a distribution over tokens in the dialogue history. We use an entity-to-dialogue-history attention mechanism applied from an entity P_i to all tokens in the dialogue history. Any neural network that learn to generate attention between two tensors is applicable .e.g. (Bahdanau et al., 2015; Vaswani et al., 2017). The attention matrix normalized by softmax, $A_{\text{find,i}} \in \mathbb{R}^{L_{\text{H}}}$, is used to compute the weighted sum of dialogue history token representations. The output is combined with entity embedding P_i to obtain contextual entity representation $H_{\text{ent,i}} \in \mathbb{R}^d$.

summarize(H$_{\text{ent}}$**,Q)**$\rightarrow\mathbf{Q}_{\text{ctx}}$. For each contextual entity representation $H_{\text{ent,i}}$, $i = 1, ..., N_{\text{ent}}$, it is projected to L_{Q} dimensions and is combined with question token embeddings through elementwise summation to obtain entity-aware question representation $Q_{\text{ent,i}} \in \mathbb{R}^{L_{\text{Q}}\times d}$. It is fed to a one-dimensional CNN with max-pooling layer (Kim, 2014) to obtain a contextual entity-aware question representation. We denote the final output as $Q_{\text{ctx}} \in \mathbb{R}^{N_{\text{ent}}\times d}$.

While previous models usually focus on global or token-level dependencies (Hori et al., 2019; Le et al., 2019b) to encode question features, our modules compress fine-grained question representations at the entity level. Specifically, find and summarize modules can generate entity-dependent local and global representations of question semantics. We show that our modularized approach can achieve better performance and transparency than traditional approaches to encode dialogue context (Serban et al., 2016; Vaswani et al., 2017) (Section 4).

where(P,V)$\rightarrow\mathbf{V}_{\text{ent}}$. Similar to the find module, this module handles entity-based attention to the video input. However, the entity representation P, in this case, is parameterized by the original entity in dialogue rather than in question (See Section 3.4 for more description). Each entity P_i is stacked to match the number of sampled video frames/clips F. An attention network is used to obtain entity-to-object attention matrix $A_{\text{where,i}} \in \mathbb{R}^{F\times O}$. The attended feature are compressed through weighted sum pooling along the spatial dimension, resulting in $V_{\text{ent,i}} \in \mathbb{R}^{F\times d}$, $i = 1, ..., N_{\text{ent}}$.

when(P,V$_{\text{ent}}$**)**$\rightarrow\mathbf{V}_{\text{ent+act}}$. This module follows a similar architecture as the where module. However, the action parameter P_i is stacked to match N_{ent} dimensions. The attention matrix $A_{\text{when,i}} \in \mathbb{R}^F$ is then used to compute the visual entity-action representations through weighted sum along the temporal dimension. We denote the out-

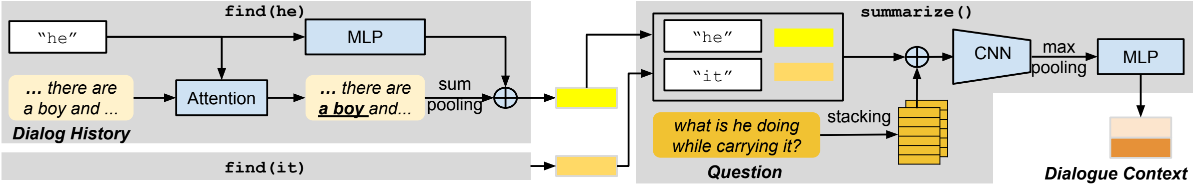


Figure 2: `find` and `summarize` neural modules for dialogue understanding

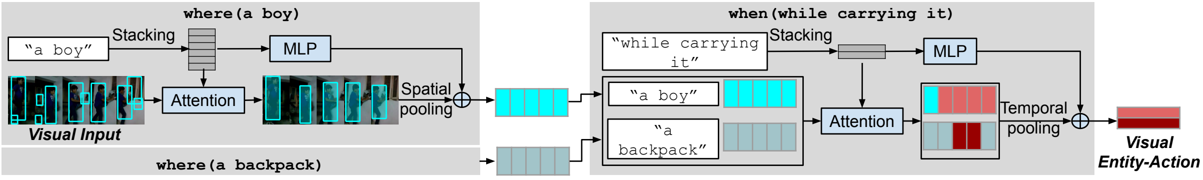


Figure 3: `where` and `when` neural modules for video understanding

put for all actions P_i as $V_{\text{ent+act}} \in \mathbb{R}^{N_{\text{ent}}\times N_{\text{act}}\times d}$

describe(P,V$_{\text{ent+act}}$)$\rightarrow$**V$_{\text{ctx}}$**. This module is a linear transformation to compute $V_{\text{ctx}} = {W_{\text{desc}}}^T[V_{\text{ent+act}}; P_{\text{stack}}] \in \mathbb{R}^{N_{\text{ent}}\times N_{\text{act}}\times d}$ where $W_{\text{desc}} \in \text{R}^{2d\times d}$, P_{stack} is the stacked representations of parameter embedding P to $N_{\text{ent}} \times N_{\text{act}}$ dimensions, and $[;]$ is the concatenation operation.

The `exist` module is a special case of `describe` module where the parameter P is the average pooled question embeddings. The above `where` module is applied to object-level features. For temporal-based features such as CNN-based and audio features, the same neural operation is applied along the temporal dimension. Each resulting entity-aware output is then incorporated to frame-level features through element-wise summation.

An advantage of our architecture is that it separates dialogue and video understanding. We adopt a transparent approach to solve linguistic entity references during the dialogue understanding phase. The resolved entities are fed to the video understanding phase to learn entity-action dynamics in the video. We show that our approach is robust when dialogue evolves to many turns and video extends over time (Please refer to Section 4).

3.4 Question Parsers

To learn compositional programs, we follow (Johnson et al., 2017a; Hu et al., 2017) and consider program generation as a sequence-to-sequence task. We adopt a simple template "$\langle\text{param}_1\rangle\langle\text{module}_1\rangle\langle\text{param}_2\rangle\langle\text{module}_2\rangle$..." as the target sequence. The resulting target sequences for dialogue and video understanding programs are sequences $\mathcal{P}_{\text{dial}}$ and $\mathcal{P}_{\text{vid}}$ respectively.

The parsers decompose questions into sub-sequences to construct compositional reasoning programs for dialogue and video understanding. Each parser is a vanilla Transformer decoder, including multi-head attention layers on questions and past dialogue turns. For more details, please refer to the Supplementary Material.

3.5 Response Decoder

System response is decoded by incorporating the dialogue context and video context outputs from the corresponding reasoning programs to target token representations. We follows a vanilla Transformer decoder architecture (Le et al., 2019b), which consists of 3 attention layers: self-attention to attend on existing tokens, attention to Q_{ctx} from dialogue understanding program execution, and attention to V_{ctx} from video understanding program execution.

$$A^{(1)}_{\text{res}} = \text{Attention}(R|_0^{j-1}, R|_0^{j-1}, R|_0^{j-1}) \in \mathbb{R}^{j\times d}$$
$$A^{(2)}_{\text{res}} = \text{Attention}(A^{(1)}_{\text{res}}, Q_{\text{ctx}}, Q_{\text{ctx}}) \in \mathbb{R}^{j\times d}$$
$$A^{(3)}_{\text{res}} = \text{Attention}(A^{(2)}_{\text{res}}, V_{\text{ctx}}, V_{\text{ctx}}) \in \mathbb{R}^{j\times d}$$

Multimodal Fusion. For video features come from multiple modalities, visual and audio, the contextual features, denoted V_{ctx}, is obtained through a weighted sum of component modalities, e.g. contextual visual features $V^{\text{vis}}_{\text{ctx}}$ and contextual audio features $V^{\text{aud}}_{\text{ctx}}$. The scores S_{fusion} to compute the weighted sum is defined as:

$$S_{\text{fusion}} = \text{Softmax}(W^T_{\text{fusion}}[Q_{\text{stack}}; V^{\text{vis}}_{\text{ctx}}; V^{\text{aud}}_{\text{ctx}}])$$

where Q_{stack} is the mean pooling output of question embeddings Q which is then stacked to $N_{\text{ent}} + N_{\text{act}}$ dimensions, and $W_{\text{fusion}} \in \mathbb{R}^{3d\times 2}$ are trainable model parameters. The resulting S_{fusion} has a dimension of $\in \mathbb{R}^{(N_{\text{ent}}+N_{\text{act}})\times 2}$.

3.6 Optimization

We optimize models by joint training to minimize the cross-entropy losses:

$$\begin{aligned}\mathcal{L} &= \alpha\mathcal{L}_{\text{dial}} + \beta\mathcal{L}_{\text{vid}} + \mathcal{L}_{\text{res}} \\ &= \alpha\sum_{j} -\log(\mathbf{P}_{\text{dial}}(\mathcal{P}_{\text{dial,j}})) \\ &+ \beta\sum_{l} -\log(\mathbf{P}_{\text{video}}(\mathcal{P}_{\text{video,l}})) \\ &+ \sum_{n} -\log(\mathbf{P}_{\text{res}}(\mathcal{R}_{\text{n}}))\end{aligned}$$

where $\mathbf{P}$ is the probability distribution of an output token. The probability is computed by passing output representations from the parsers and decoder to a linear layer $W \in \mathbb{R}^{d\times V}$ with softmax activation. We share the parameters between W and embedding matrix E.

4 Experiments

4.1 Experimental Setup

We use the AVSD benchmark from the 7^{th} Dialogue System Technology Challenge (DSTC7) (Hori et al., 2019). The benchmark consists of dialogues grounded on the Charades videos (Sigurdsson et al., 2016). Each dialogue contains up to 10 dialogue turns, each turn consists of a question and expected response about a given video. For visual features, we use the 3D CNN-based features from a pretrained I3D model (Carreira and Zisserman, 2017) and object-level features from a pretrained FasterRNN model (Ren et al., 2015b). The audio features are obtained from a pretrained VGGish model (Hershey et al., 2017). In the experiments with AVSD, we consider two settings: one with video summary and one without video summary as input. In the setting with video summary, the summary is concatenated to the dialogue history before the first dialogue turn. We also adapt VGNMN to the video QA benchmark TGIF-QA (Jang et al., 2017). For more details of data and training procedure, please refer to the Supplementary.

4.2 Experimental Results

AVSD Results. We evaluate model performance by the objective metrics, including BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), and CIDEr (Vedantam et al., 2015), between each generated response and 6 reference gold responses. As seen in Table 2, our models outperform most of existing approaches. In particular, the performance of our model in the setting without video summary input is comparable to the GPT-based RLM (Li et al., 2020) with a much smaller model size. The Student-Teacher baseline (Hori et al., 2019) specifically focuses on the performance gap between models with and without textual signals from video summary through a dual network of expert and student models. Instead, VGNMN reduces this performance gap by efficiently extracting relevant visual/audio information based on fine-grained entity and action signals. We also found that VGNMN applied to object-level features is competitive to the model applied to CNN-based features. The flexibility of VGNMN neural programs show when we treat the caption as an input equally to visual or audio inputs and execute entity-action level neural operations on the encoded caption sequence.

Robustness. To evaluate model robustness, we report BLEU4 and CIDEr of model variants in various experimental settings. Specifically, we compare against performance of output responses in the first dialogue turn position (i.e. 2^{nd}-10^{th} turn vs. the 1^{st} turn), or responses grounded on the shortest video length range (video ranges are intervals of 0-10^{th}, 10-20^{th} percentile and so on). We report results of the following model variants: (1) *w/o video NMN*: VGNMN without using video-based modules, e.g. `when` and `where`. Video features are retrieved through a token-level representation of questions (Le et al., 2019b). (2) *no NMN*: (1) + without dialogue-based modules, e.g. `find` and `summarize`. Dialogue history is encoded by a hierarchical LSTM encoder (Hori et al., 2019).

In Table 3, we compare VGNMN and model variant (1) to gain insights on model robustness to video complexity. We noted that the performance gap between VGNMN and (1) is quite distinct, with 7/10 cases of video ranges in which VGNMN outperforms. However, in lower ranges (i.e. 1-23 seconds) and higher ranges (37-75 seconds), VGNMN performs not as well as model (1). We observed that related factors might affect the discrepancy, such as the complexity of the questions for these short and long-range videos. Potentially, our question parser for the video understanding program needs a more sophisticated decoding method (e.g. for tree-based program) to retrieve information in these ranges. In Table 4, we gauge model robustness to the compelexity in dialogue. We observed

Model	PT	Visual	Audio	BLEU4	METEOR	ROUGE-L	CIDEr
Audio/Visual only (without Video Summary/Caption)							
Baseline (Hori et al., 2019)	-	I	-	0.305	0.217	0.481	0.733
Baseline (Hori et al., 2019)	-	I	V	0.309	0.215	0.487	0.746
Baseline+GRU+Attn. (Le et al., 2019a)	-	I	V	0.315	0.239	0.509	0.848
FGA (Schwartz et al., 2019)	-	I	V	-	-	-	0.806
JMAN (Chu et al., 2020)	-	I	-	0.309	0.240	0.520	0.890
Student-Teacher (Hori et al., 2019)	-	I	V	0.371	0.248	0.527	0.966
MTN (Le et al., 2019b)	-	I	-	0.343	0.247	0.520	0.936
MTN (Le et al., 2019b)	-	I	V	0.368	0.259	0.537	0.964
MSTN (Lee et al., 2020)	-	I	V	0.379	0.261	0.548	1.028
BiST (Le et al., 2020a)		RX	V	0.390	0.259	0.552	1.030
RLM-GPT2 (Li et al., 2020)	✓	I	V	**0.402**	0.254	0.544	1.052
VGNMN	-	I	-	0.397	0.262	**0.550**	**1.059**
VGNMN	-	FR	-	0.388	0.259	0.549	1.040
VGNMN	-	-	V	0.381	0.252	0.534	1.004
VGNMN	-	I	V	0.396	**0.263**	0.549	**1.059**
Audio/Visual only (with Video Summary/Caption)							
TopicEmb (Kumar et al., 2019)	-	I	A	0.329	0.223	0.488	0.762
Baseline+GRU+Attn. (Le et al., 2019a)	-	I	V	0.310	0.242	0.515	0.856
JMAN (Chu et al., 2020)	-	I	-	0.334	0.239	0.533	0.941
FA+HRED (Nguyen et al., 2018)	-	I	V	0.360	0.249	0.544	0.997
VideoSum (Sanabria et al., 2019)	-	RX	-	0.394	0.267	0.563	1.094
VideoSum+How2 (Sanabria et al., 2019)	✓	RX	-	0.387	0.266	0.564	1.087
MSTN (Lee et al., 2020)	-	I	V	0.377	0.275	0.566	1.115
Student-Teacher (Hori et al., 2019)	-	I	V	0.405	0.273	0.566	1.118
MTN (Le et al., 2019b)	-	I	-	0.392	0.269	0.559	1.066
MTN (Le et al., 2019b)	-	I	V	0.410	0.274	0.569	1.129
BiST (Le et al., 2020a)		RX	V	0.429	0.284	0.581	1.192
VGD-GPT2 (Le and Hoi, 2020)	✓	I	V	0.436	0.282	0.579	1.194
RLM-GPT2 (Li et al., 2020)	✓	I	V	**0.459**	**0.294**	**0.606**	**1.308**
VGNMN	-	I	-	0.421	0.277	0.574	1.171
VGNMN	-	FR	-	0.421	0.275	0.571	1.148
VGNMN	-	I	V	0.421	0.277	0.573	1.167
VGNMN	-	I+C	V	0.429	0.278	0.578	1.188

Table 2: AVSD test results: The visual features are: I (I3D), ResNeXt-101 (RX), Faster-RCNN (FR), C (caption as a video input). The audio features are: VGGish (V), AclNet (A). ✓on PT denotes models using pretrained weights and/or additional finetuning. The best and second best results are bold and underlined respectively.

	BLEU4		CIDEr	
video range (seconds)	VGNMN	Model (1)	VGNMN	Model (1)
1-23	0.432	**0.447**	1.298	**1.355**
23-28	**0.436**	0.433	**1.264**	1.165
28-30	**0.398**	0.376	**1.203**	1.164
30-30.6	**0.441**	0.418	**1.220**	1.202
30.6-31	**0.413**	0.411	**1.250**	1.166
31-31.6	0.439	**0.451**	1.249	**1.295**
31.6-32	**0.430**	0.419	**1.217**	1.192
32-33	**0.468**	0.445	**1.343**	1.237
33-37	**0.388**	0.381	**1.149**	1.124
37-75	0.356	**0.365**	0.910	**0.962**

Table 3: Performance by video length between VGNMN and model variant (1): VGNMN *w/o video NMN*.

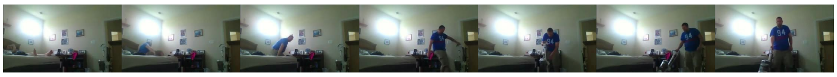
	BLEU4		CIDEr	
turn position	Model (1)	Model (2)	Model (1)	Model (2)
1	0.579	**0.587**	1.623	**1.650**
2	0.429	**0.430**	**1.155**	1.142
3	0.275	**0.289**	**0.867**	0.846
4	**0.309**	0.305	**0.859**	0.855
5	**0.355**	0.335	**1.088**	1.023
6	**0.357**	0.329	**1.044**	0.950
7	**0.342**	0.325	**0.896**	0.847
8	**0.361**	0.332	**1.025**	0.973
9	0.383	**0.431**	1.043	**1.182**
10	**0.395**	0.371	0.931	**0.977**

Table 4: Performance breakdown by dialogue turn between model variants (1) (VGNMN *w/o video NMN*) and (2) (*no NMN*)

that model (1) performs better than model (2) overall, especially in higher turn positions, i.e. from the 4^{th} turn to 8^{th} turn. Interestingly, we noted some mixed results in very low turn position, i.e. the 2^{nd} and 3^{rd} turn, and very high turn position, i.e. the 10^{th} turn. Potentially, with a large dialogue turn position, the neural-based approach such as hierarchical RNN can better capture the global dependencies within dialogue context than the entity-based compositional NMN method. For additional analysis, please refer to the Supplementary.

Intepretability. From Figure 4, we observe that in cases where predicted dialogue programs and video programs match or are close to the gold labels, the model can generate generally correct responses. For cases where some module parameters do not exactly match but are closed to the gold labels, the model can still generate responses with the correct visual information (e.g. the 4^{th} turn in example B). In cases of wrong predicted responses, we can further look at how the model understands the questions based on predicted programs. In the 3^{rd} turn of example A, the output response is miss-

Turn	Question	Dialogue NMN	Video NMN	Response
3	...what are they doing in the scene ? are they on a rooftop ?	**Predicted**: `find(they)`, `find(the scene)` `→summarize()` ✘ **Gold**: `find(they)` `→summarize()`	**Predicted**: `where(two men in the video),where(the scene)→ when(doing in the scene)→ describe(what)` ✘ **Gold**: `where(two men), where(rooftop)` `→when(doing in the scene)` `→describe(what)`	**Predicted**: one is washing a chair and the other is taking pictures ✘ **Gold**: yes , on a second floor roof deck , one man is washing a chair , another man is either filming or taking pictures
4	does he sit in *the chair* after washing it ?	**Predicted**: `find(he)`, `find(the chair)` `→summarize()`✓ **Gold**: `find(he)`, `find(the chair)→` `summarize()`	**Predicted**: `where(one man in the video), where(a chair)→ when(sit in the chair after washing it)→exist()` ✓ **Gold**: `where(one man), where(a chair)→` `when(sit in the chair after washing it)→exist()`	**Predicted**: no , he does not sit in the chair ✓ **Gold**: no he does not , there is a pipe with water running all over



Turn	Question	Dialogue Understanding Program	Video Understanding Program	Response
3	in the door way to the next room, there is an object. what is that?	**Predicted**: `find(the room), find(the door)→summarize()` ✘ **Gold**: `summarize()`	**Predicted**: `where(what room)` `→when(what is that)→exist()` ✘ **Gold**: `describe(what)`	**Predicted**: it looks like he is in a living room. ✘ **Gold**: he went to the doorway for a vacuum.
4	when he gets up, does he have anything in *his hand*?	**Predicted**: `find(he)`, `find(his)→` `summarize()` ✓ **Gold**:`find(he)`, `find(his)→` `summarize()`	**Predicted**: `where(one person in the video)→when(get up, have anything in his hands)→` `describe(when)` ✘ **Gold**: `where(one person in the video)→when(get up),` `when(have anything in his hands)→ describe(when)`	**Predicted**: he has a vacuum in his hands. ✓ **Gold**: he goes for the vacuum.

Figure 4: Intepretability of model outputs from a dialogue in the test split of the AVSD benchmark. We demonstrate Example A (Top) and Example B (Bottom).

ing a minor detail as compared to the label response because the video program fails to capture "rooftop" as a `where` parameter. These subtle yet important details can determine whether output responses can fully address user queries. In the 3^{rd} turn of example B, the model wrongly identifies "what room" as a `where` parameter and subsequently generates a response to indicate it is "a living room". For additional qualitative analysis, please refer to the Supplementary.

TGIF-QA Results. We report the result using the L2 loss in *Count* task and accuracy in other tasks. From Table 5, VGNMN outperforms majority of the baseline models in all tasks by a large margin. Compared to AVSD experiments, the TGIF-QA experiments emphasize the video understanding ability of the models, removing the requirement for dialogue understanding and natural language generation. Since TGIF-QA questions follow a very specific question type distribution (count, action, transition, and frameQA), the question structures are simpler and easier to learn than AVSD. Using exact-match accuracy of parsed programs vs. label programs as a metric, our question parser can achieve a performance 81% to 94% accuracy in TGIF-QA vs. 41-45% in AVSD. The higher accuracy in decoding a reasoning structure translates to better matches between training and test distributions, resulting in higher performance gains.

Model	Visual	Count (Loss)	Action (Acc)	Transition (Acc)	FrameQA (Acc)
VIS(avg) (Ren et al., 2015a)	R	4.80	0.488	0.348	0.350
MCB (aggr) (Fukui et al., 2016)	R	5.17	0.589	0.243	0.257
Yu et al. (Yu et al., 2017)	R	5.13	0.561	0.640	0.396
ST-VQA (t) (Gao et al., 2018)	R+F	4.32	0.629	0.694	0.495
Co-Mem (Gao et al., 2018)	R+F	4.10	0.682	0.743	0.515
PSAC (Li et al., 2019)	R	4.27	0.704	0.769	0.557
HME (Fan et al., 2019)	R+C	4.02	0.739	0.778	0.538
STA (Gao et al., 2019)	R	4.25	0.723	0.790	0.566
CRN+MAC (Le et al., 2019c)	R	4.23	0.713	0.787	0.592
MQL (Lei et al., 2020)	V	-	-	-	0.598
QueST (Jiang et al., 2020)	R	4.19	0.759	0.810	0.597
HGA (Jiang and Han, 2020)	R+C	4.09	0.754	0.810	0.551
GCN (Huang et al., 2020)	R+C	3.95	0.743	0.811	0.563
HCRN (Le et al., 2020b)	R+RX	3.82	0.750	0.814	0.559
BiST (Le et al., 2020a)	RX	**2.14**	**0.847**	0.819	0.648
VGNMN	R	2.65	0.845	**0.887**	**0.747**

Table 5: Experiment results on the TGIF-QA benchmark. The visual features are: ResNet-152 (R), C3D (C), Flow CNN from two-stream model (F), VGG (V), ResNeXt-101 (RX).

5 Conclusion

In this work, we introduce Video-grounded Neural Module Network (VGNMN). VGNMN consists of dialogue and video understanding neural modules, each of which performs entity and action-level operations on language and video components. Our comprehensive experiments on AVSD and TGIF-QA benchmarks show that our models can achieve competitive performance while promoting a compositional and interpretable learning approach.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Stefan Lee, Peter Anderson, Irfan Essa, Devi Parikh, Dhruv Batra, Anoop Cherian, Tim K. Marks, and Chiori Hori. 2019. Audio-visual scene-aware dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1545–1554, San Diego, California. Association for Computational Linguistics.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Joao Carreira and Andrew Zisserman. 2017. 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.

Yun-Wei Chu, Kuan-Yen Lin, Chao-Chun Hsu, and Lun-Wei Ku. 2020. Multi-step joint-modality attention network for scene-aware dialogue system. *DSTC Workshop @ AAAI*.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In *International Conference on Learning Representations*.

Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. 2019. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1999–2007.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, Austin, Texas. Association for Computational Linguistics.

Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. 2018. Motion-appearance co-memory networks for video question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6576–6585.

Lianli Gao, Pengpeng Zeng, Jingkuan Song, Yuan-Fang Li, Wu Liu, Tao Mei, and Heng Tao Shen. 2019. Structured two-stream attention network for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6391–6398.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, volume 1, page 3.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2020. Neural module networks for reasoning over text. In *International Conference on Learning Representations*.

Chi Han, Jiayuan Mao, Chuang Gan, Josh Tenenbaum, and Jiajun Wu. 2019. Visual concept-metaconcept learning. In *Advances in Neural Information Processing Systems*, pages 5002–5013.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE.

C. Hori, H. Alamri, J. Wang, G. Wichern, T. Hori, A. Cherian, T. K. Marks, V. Cartillier, R. G. Lopes, A. Das, I. Essa, D. Batra, and D. Parikh. 2019. End-to-end audio visual scene-aware dialog using multimodal attention-based video features. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2352–2356.

Chiori Hori, Anoop Cherian, Tim K Marks, and Takaaki Hori. 2019. Joint student-teacher learning for audio-visual scene-aware dialog. *Proc. Interspeech 2019*, pages 1886–1890.

Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2018. Explainable neural computation via stack neural module networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 53–69.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564.

Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. 2020. Location-aware graph convolutional networks for video question answering. In *The AAAI Conference on Artificial Intelligence (AAAI)*, volume 1.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2758–2766.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. 2020. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *The AAAI Conference on Artificial Intelligence (AAAI)*.

Pin Jiang and Yahong Han. 2020. Reasoning with heterogeneous graph alignment for video question answering. In *The AAAI Conference on Artificial Intelligence (AAAI)*.

Yichen Jiang and Mohit Bansal. 2019. Self-assembling modular networks for interpretable multi-hop reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4474–4484, Hong Kong, China. Association for Computational Linguistics.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017a. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017b. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2989–2998.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2018. Visual coreference resolution in visual dialog using neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–169.

Shachi H Kumar, Eda Okur, Saurav Sahay, Jonathan Huang, and Lama Nachman. 2019. Leveraging topics and audio features with multimodal attention for audio visual scene-aware dialog. *arXiv preprint arXiv:1912.10131*.

Hung Le, S Hoi, Doyen Sahoo, and N Chen. 2019a. End-to-end multimodal dialog systems with hierarchical multimodal attention on video features. In *DSTC7 at AAAI2019 workshop*.

Hung Le and Steven C.H. Hoi. 2020. Video-grounded dialogues with pretrained generation language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5842–5848, Online. Association for Computational Linguistics.

Hung Le, Doyen Sahoo, Nancy Chen, and Steven Hoi. 2019b. Multimodal transformer networks for end-to-end video-grounded dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5612–5623, Florence, Italy. Association for Computational Linguistics.

Hung Le, Doyen Sahoo, Nancy Chen, and Steven C.H. Hoi. 2020a. BiST: Bi-directional spatio-temporal reasoning for video-grounded dialogues. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1846–1859, Online. Association for Computational Linguistics.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. 2019c. Learning to reason with relational video representation for question answering. *arXiv preprint arXiv:1907.04553*.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. 2020b. Hierarchical conditional relation networks for video question answering. *arXiv preprint arXiv:2002.10698*.

Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2020. Dstc8-avsd: Multimodal semantic transformer network with retrieval style word generator. *DSTC Workshop @ AAAI 2020*.

Chenyi Lei, Lei Wu, Dong Liu, Zhao Li, Guoxin Wang, Haihong Tang, and Houqiang Li. 2020. Multi-question learning for visual question answering. In *The AAAI Conference on Artificial Intelligence (AAAI)*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. TVQA: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379, Brussels, Belgium. Association for Computational Linguistics.

Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. 2019. Beyond rnns: Positional self-attention with co-attention for video question answering. In *The 33rd AAAI Conference on Artificial Intelligence*, volume 8.

Zekang Li, Zongjia Li, Jinchao Zhang, Yang Feng, Cheng Niu, and Jie Zhou. 2020. Bridging text and video: A universal multimodal transformer for video-audio scene-aware dialog. *arXiv preprint arXiv:2002.00163*.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on Freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33, Vancouver, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.

Dat Tien Nguyen, Shikhar Sharma, Hannes Schulz, and Layla El Asri. 2018. From film to video: Multi-turn question answering with multi-modal context. In *AAAI 2019 Dialog System Technology Challenge (DSTC7) Workshop*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015a. Exploring models and data for image question answering. In *Advances in neural information processing systems*, pages 2953–2961.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015b. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Ramon Sanabria, Shruti Palaskar, and Florian Metze. 2019. Cmu sinbad's submission for the dstc7 avsd challenge. In *DSTC7 at AAAI2019 workshop*, volume 6.

Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G Schwing. 2019. Factor graph attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2039–2048.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. 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 30*, pages 5998–6008. Curran Associates, Inc.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1031–1042.

Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. 2017. End-to-end concept word detection for video captioning, retrieval, and question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3165–3173.

A VGNMN Model Overview

An overview of VGNMN can be seen in Figure 5. VGNMN includes 4 major components: (1) encoders that encode dialogue and video components into continuous vector representations; (2) question parsers that parse question of the current dialogue turn into compositional programs for dialogue and video understanding; (3) an inventory of neural modules that operate on dialogue and video input components; and (4) a response decoder that generates natural language sequence using dialogue-based and video-based execution outputs.

B Question Parsers

To learn compositional programs, we follow (Johnson et al., 2017a; Hu et al., 2017) and consider program generation as a sequence-to-sequence task. We adopt a simple template "$\langle\mathrm{param}_1\rangle\langle\mathrm{module}_1\rangle\langle\mathrm{param}_2\rangle\langle\mathrm{module}_2\rangle$..." as the target sequence. The resulting target sequences for dialogue and video understanding programs are sequences $\mathcal{P}_{\mathrm{dial}}$ and $\mathcal{P}_{\mathrm{vid}}$ respectively.

The parsers decompose questions into sub-sequences to construct compositional reasoning programs for dialogue and video understanding. Each parser is an attention-based Transformer decoder. The Transformer attention is a multi-head attention on query q, key k, and value v tensors, denoted as $\mathrm{Attention}(\mathrm{q}, \mathrm{k}, \mathrm{v})$. For each token in the q sequence , the distribution over tokens in the k sequence is used to obtain the weighted sum of the corresponding representations in the v sequence.

$$\mathrm{Attention}(q, k, v) = \mathrm{softmax}(\frac{qk^T}{\sqrt{d_k}})v \in \mathbb{R}^{L_{\mathrm{q}} \times d_{\mathrm{q}}}$$

Each attention is followed by a feed-forward network applied to each position identically. We exploit the multi-head and feed-forward architecture, which show good performance in NLP tasks such as NMT and QA (Vaswani et al., 2017; Dehghani et al., 2019), to efficiently incorporate contextual cues from dialogue components to parse question into reasoning programs. At decoding step 0, we simply use a special token *_sos* as the input to the parser. In each subsequent decoding step, we concatenate the prior input sequence with the generated token to decode in an auto-regressive manner. We share the vocabulary sets of input and output components and thus, use the same embedding matrix. Given the encoded question Q, to decode the program for dialogue understanding, the contextual signals are integrated through 2 attention layers: one attention on previously generated tokens, and the other on question tokens. At time step j, we

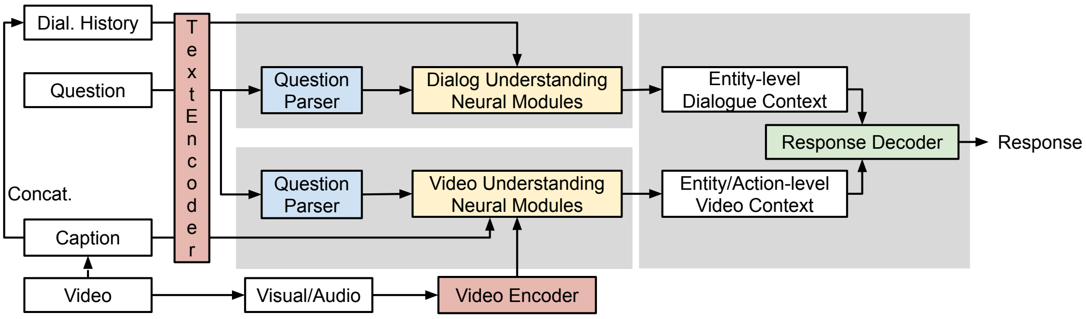

Dial. History
Question
Concat.
Caption
Video
TextEncoder
Question Parser
Dialog Understanding Neural Modules
Question Parser
Video Understanding Neural Modules
Visual/Audio
Video Encoder
Entity-level Dialogue Context
Entity/Action-level Video Context
Response Decoder
Response

Figure 5: An overview of the VGNMN approach

denote the output from an attention layer as $A_{\text{dial,j}}$.

$$A^{(1)}_{\text{dial}} = \text{Attention}(P_{\text{dial}}|^{j-1}_0, P_{\text{dial}}|^{j-1}_0, P_{\text{dial}}|^{j-1}_0)$$
$$A^{(2)}_{\text{dial}} = \text{Attention}(A^{(1)}_{\text{dial}}, Q, Q) \in \mathbb{R}^{j\times d}$$

To generate programs for video understanding, the contextual signals are learned and incorporated in a similar manner. However, to exploit dialogue contextual cues, the execution output of dialogue understanding neural modules Q_{ctx} is incorporated to each vector in P_{dial} through an additional attention layer. This layer integrates the resolved entity information to decode the original entities for video understanding. It is equivalent to a reasoning process that converts the question from its original multi-turn semantics to single-turn semantics.

$$A^{(1)}_{\text{vid}} = \text{Attention}(P_{\text{vid}}|^{j-1}_0, P_{\text{vid}}|^{j-1}_0, P_{\text{vid}}|^{j-1}_0)$$
$$A^{(2)}_{\text{vid}} = \text{Attention}(A^{(1)}_{\text{vid}}, Q, Q) \in \mathbb{R}^{j\times d}$$
$$A^{(3)}_{\text{vid}} = \text{Attention}(A^{(2)}_{\text{vid}}, Q_{\text{ctx}}, Q_{\text{ctx}}) \in \mathbb{R}^{j\times d}$$

C Additional Dataset Details

Different from AVSD, TGIF-QA contains a diverse set of QA tasks:

- *Count*: open-ended task which counts the number of repetitions of an action
- *Action*: multiple-choice (MC) task which asks about a certain action occurring for a fixed number of times
- *Transition*: MC task which emphasizes temporal transition in video
- *Frame*: open-ended task which can be answered from visual contents of one of the video frames

For the TGIF-QA benchmark, we use the extracted features from a pretrained ResNet model (He et al., 2016).

Table 6: Summary of DSTC7 AVSD and TGIF-QA benchmark

	#	Train	Val.	Test
AVSD	Dialogs	7,659	1,787	1,710
	Turns	153,180	35,740	13,490
	Words	1,450,754	339,006	110,252
TGIFQA	Count QA	24,159	2,684	3,554
	Action QA	18,428	2,047	2,274
	Trans. QA	47,434	5,270	6,232
	Frame QA	35,453	3,939	13,691

D Additional Training Procedure Details

We follow prior approaches (Hu et al., 2017, 2018; Kottur et al., 2018) by obtaining the annotations of the programs through a language parser (Hu et al., 2016) and a reference resolution model (Clark and Manning, 2016). During training, we directly use these soft labels of programs and the given ground-truth responses to train the models. The labels are augmented with label smoothing technique (Szegedy et al., 2016). During inference time, we generate all programs and responses from given dialogues and videos. We run beam search to enumerate programs for dialogue and video understanding and dialogue responses.

We use a training batch size of 32 and embedding dimension $d = 128$ in all experiments. Where Transformer attention is used, we fix the number of attention heads to 8 in all attention layers. In neural modules with MLP layers, the MLP network is fixed to 2 linear layers with a ReLU activation in between. In neural modules with CNN, we adopt a vanilla CNN architecture for text classification (without the last MLP layer) where the number of input channels is 1, the kernel sizes are $\{3, 4, 5\}$, and the number of output channels is d. We initialize models with uniform distribution (Glorot and Bengio, 2010). During training, we adopt the Adam optimizer (Kingma and Ba, 2015) and a decaying learning rate (Vaswani et al., 2017) where we fix the warm-up steps to 15K training steps. We employ dropout (Srivastava et al., 2014) of 0.2 at all networks except the last linear layers of question parsers and response decoder. We train models up to 50 epochs and select the best models based on the average loss per epoch in the validation set.

D.1 Intepretability

We extract the predicted programs and responses for some example dialogues in Figure 6, 7, 8, and 9 and report our observations:

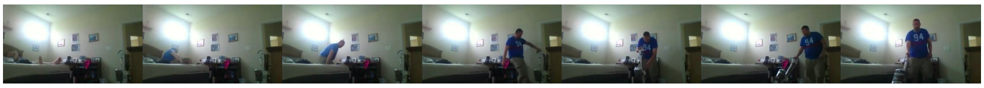
- We observe that when the predicted programs are correct, the output responses generally match the ground-truth (See the 1^{st} and 2^{nd} turn in Figure 6, and the 1^{st} and 4^{th} turn in Figure 8) or close to the ground-truth responses (1^{st} turn in Figure 7).

- When the output responses do not match the ground truth, we can understand the model mistakes by interpreting the predicted programs. For example, in the 3^{rd} turn in Figure 6, the output response describes a room because the predicted video program focuses on the entity "what room" instead of the entity "an object" in the question. Another example is the 3^{rd} turn in Figure 8 where the entity "rooftop" is missing in the video program. These mismatches can deviate the information retrieved from the video during video program execution, leading to wrong output responses with wrong visual contents.

- We also note that in some cases, one or both of the predicted programs are incorrect, but the predicted responses still match the ground-truth responses. This might be explained as the predicted module parameters are still close enough to the "gold" labels (e.g. 4^{th} turn in Figure 6). Sometimes, our model predicted programs that are more appropriate than the ground truth. For example, in the 2^{nd} turn in Figure 7, the program is added with a `where` module parameterized by the entity "the shopping bag" which was solved from the reference "them" mentioned in the question.

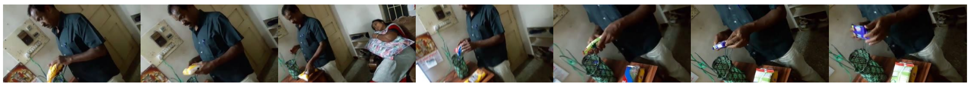
- We observe that for complex questions that involve more than one queries (e.g. the 3^{rd} turn in Figure 8), it becomes more challenging to decode an appropriate video understanding program and generate responses that can address all queries.

- In Figure 9, we demonstrate some output examples of VGNMN and compare with two baselines: Baseline (Hori et al., 2019) and MTN (Le et al., 2019b). We noted that VGNMN can include important entities relevant to the current dialogue turn to construct output responses while other models might miss some entity details, e.g. "them/dishes" in example A and "the magazine" in example B. These small yet important details can determine the correctness of dialogue responses.



Turn	Question	Dialogue Understanding Program	Video Understanding Program	Response
1	how many people are in the video ?	**Predicted**: `summarize()` ✓ **Gold**: `summarize()`	**Predicted**: `describe(how many people)` ✓ **Gold**: `describe(how many people)`	**Predicted**: there is only one person in the video. ✓ **Gold**: there is one person in the video.
2	does this person ever leave the field of view ?	**Predicted**: `find(this person)→summarize()` ✓ **Gold**: `find(this person)→ summarize ()`	**Predicted**: `where(one person in the video)→when(leave the field of view)→exist()` ✓ **Gold**: `where(one person in the video)→when(leave the field of view)→exist()`	**Predicted**: no, he does not leave the view. ✓ **Gold**: no, he is in the video the whole time.
3	in the door way to the next room, there is an object. what is that?	**Predicted**: `find(the room), find(the door)→summarize()` ✘ **Gold**: `summarize()`	**Predicted**: `where(what room) →when(what is that)→exist()` ✘ **Gold**: `describe(what)`	**Predicted**: it looks like he is in a living room. ✘ **Gold**: he went to the doorway for a vacuum.
4	when he gets up, does he have anything in *his* *hand*?	**Predicted**: `find(he), find(his)→ summarize()` ✓ **Gold**:`find(he), find(his)→ summarize()`	**Predicted**: `where(one person in the video)→when(get up, have anything in his hands) → describe(when)` ✘ **Gold**: `where(one person in the video)→when(get up), when(have anything in his hands)→ describe(when)`	**Predicted**: he has a vacuum in his hands. ✓ **Gold**: he goes for the vacuum.

Figure 6: Example output programs and responses from a dialogue in the test split of the AVSD benchmark.



Turn	Question	Dialogue Understanding Program	Video Understanding Program	Response
1	what is is holding in *his* hand in the beginning ?	**Predicted**: `find(his) →summarize()` ✓ **Gold**: `find(his) →summarize()`	**Predicted**: `where(a man) →when(holding in his hand in the beginning)→ describe (what)` ✓ **Gold**: `where(a man)→ when (holding in his hand in the beginning)→ describe(what)`	**Predicted**: he is holding a shopping bag in his hand . ✘ **Gold**: looks like a bag of beans
2	does he put *them* down ?	**Predicted**: `find(he), find(them) →summarize()` ✘ **Gold**: `find(he)→ summarize()`	**Predicted**: `where(a man), where (a shopping bag)→when(put them down)→exist()` ✘ **Gold**: `where(a man)→ when(put them down)→exist()`	**Predicted**: yes he puts them down on the table . ✓ **Gold**: yes he puts it down after looking at it

Figure 7: Example output programs and responses from a dialogue in the test split of the AVSD benchmark.

Turn	Question	Dialogue Understanding Program	Video Understanding Program	Response
1	hi how many people are in the video from start to finish ?	**Predicted**: `summarize()` ✓ **Gold**: `summarize()`	**Predicted**: `describe(how many people)` ✓ **Gold**: `describe(how many people)`	**Predicted**: there are two people in the video . ✓ **Gold**: hello , there are two men
2	are they carrying on a conversation between *themselves* ?	**Predicted**: `find(they) →summarize()` ✓ **Gold**: `find(they), find(themselves) →summarize()`	**Predicted**: `where(two men in the video)→when(carrying on a conversation between themselves)→exist()` ✓ **Gold**: `where(two men)→ when (carrying a conversation between themselves)→exist()`	**Predicted**: yes , they are carrying a conversation ✗ **Gold**: no , background noise is birdsong and shuffling noises
3	what are they doing in the scene ? are they on a rooftop ?	**Predicted**: `find(they), find(the scene) →summarize()` ✗ **Gold**: `find(they) →summarize()`	**Predicted**: `where(two men in the video),where(the scene)→ when(doing in the scene)→ describe(what)` ✗ **Gold**: `where(two men), where(rooftop)→when(doing in the scene) →describe(what)`	**Predicted**: one is washing a chair and the other is taking pictures ✗ **Gold**: yes , on a second floor roof deck , one man is washing a chair , another man is either filming or taking pic tures
4	does he sit in *the chair* after washing it ?	**Predicted**: `find(he), find(the chair) →summarize()` ✓ **Gold**: `find(he), find(the chair)→ summarize()`	**Predicted**: `where(one man in the video), where(a chair)→ when(sit in the chair after washing it)→exist()` ✓ **Gold**: `where(one man), where(a chair)→ when(sit in the chair after washing it) →exist()`	**Predicted**: no , he does not sit in the chair ✓ **Gold**: no he does not , there is a pipe with water running all over

Figure 8: Example output programs and responses from a dialogue in the test split of the AVSD benchmark.

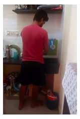
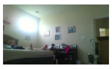
Video (Sample Frame)	Dialog History and Question	Predicted Dialogue & Video Understanding Program	Predicted and Ground-truth Responses
A 	**Dialog History**: ...what is going on when the video starts ? a man is looking out a doorway...what is he doing in there then ? he starts to wash dishes ... **Question**: does he wipe *them* with a towel?	**Dialogue Program (VilNMN)**: `find(he), find(them) →summarize()` **Video Program (VilNMN)**: `where(a man), where(the dishes) → when(wipe them with a towel) → exist()`	**Predicted (VilNMN)**: no he doesn't wipe them with a towel . ✓ **Predicted (Baseline)**: no he does not take a picture of the camera. ✗ **Predicted (MTN)**: no he does not do anything with with hands. ✗ **Gold**: looks like he just uses his hands .
B 	**Dialog History**: ...what is he reading? He is reading a magazine and he is laughing... **Question**: is he laughing because of *the magazine*?	**Dialogue Program (VilNMN)**: `find(he), find(the magazine) → summarize()` **Dialogue Program (VilNMN)**: `where(one young boy in the video), where(a magazine)→ when (laughing because of the magazine)→ exist()`	**Predicted (VilNMN)**: yes, he is laughing at the magazine . ✓ **Predicted (Baseline)**: yes he is talking to the camera . ✗ **Predicted (MTN)**: no he is not laughing at the end of the video . ✗ **Gold**: maybe , because then he throws the magazine aside .
C 	**Dialog History**: ...how many people are in the video? There is one person in the video... **Question**: in the door *way* to the *next* room, there is *an object*. What is that?	**Dialogue Program (VilNMN)**: `find(the room), find(the door)→ summarize()` **Dialogue Program (VilNMN)**: `where(what room)→ when (what is that)→ exist()`	**Predicted (VilNMN)**: it looks like he is in a living room . ✗ **Predicted (Baseline)**: i m not sure what it is . ✗ **Predicted (MTN)**: he walks into the room . ✗ **Gold**: he went to the doorway for a vacuum .

Figure 9: Intepretability of example outputs from VGNMN and baselines models (Hori et al., 2019; Le et al., 2019b)