

# **Methods and Applications for Multimodal Conversational Models**

Von der Fakultät Informatik, Elektrotechnik und Informations-  
technik der Universität Stuttgart zur Erlangung der Würde  
eines Doktors der Naturwissenschaften (*Dr. rer. nat.*)  
genehmigte Abhandlung

*Vorgelegt von*

***Mohamed Adnen ABDESSAIED***

*aus Enfidha, Tunesien*

Hauptberichter: **Prof. Dr. Andreas BULLING**  
Mitberichter: **Prof. Dr. Marcus ROHRBACH**  
Mitberichter: **Prof. Dr. Mohamed ELHOSEINY**

Tag der mündlichen Prüfung: 29.04.2025

**Collaborative Artificial Intelligence (CAI)**  
**Institut für Visualisierung und Interaktive Systeme (VIS)**  
**Universität Stuttgart**  
**2025**



## PRÜFUNGSAUSSCHUSS

*Vorsitzender:* **Prof. Dr. Thang Vu, Universität Stuttgart**  
*Hauptberichter:* **Prof. Dr. Andreas BULLING, Universität Stuttgart**  
*Mitberichter:* **Prof. Dr. Marcus ROHRBACH, TU Darmstadt**  
*Mitberichter:* **Prof. Dr. Mohamed ELHOSEINY, KAUST**



*“If learning the truth is the scientist’s goal, then he must make himself the enemy of all that he reads.”*

*Ibn Al-Haytham – the father of modern optics*



# Acknowledgements

I want to begin by expressing my deepest gratitude to my advisor Andreas Bulling for offering me the opportunity to be part of his research lab and for his guidance, supervision, and unconditional support throughout the course of my doctorate studies. Furthermore, I want to thank Anna and Marcus Rohrbach for hosting me during my research visit to their lab in Darmstadt. I will always be grateful for this learning experience and for the fruitful collaboration it led to.

Many thanks and gratitude as well to Marcus Rohrbach and Mohamed Elhoseiny for accepting to be part of my PhD committee.

Of course, I would like to acknowledge the administrative assistance of Daniela and the IT support of Reiner over the past years. In particular, I would like to express my heartfelt thanks to all of my colleagues in Stuttgart (Matteo, Anna, Constantin, Lei, Dominike, Ekta, and Florian) and in Darmstadt (Marcel, Tobias, Jonas, and Mark) for providing such a pleasant working atmosphere and for the fun they brought to this journey. Special thanks and apologies to Anna, Dominike, and Marcel for supporting my noisy laptop.

To my close friends Mazen, Mohamed Ali, and Adam, many thanks for your support and faith during the years.

To my dearest friend Yassine, no words could ever truly capture my gratitude for your unwavering belief in me. Thank you for standing by my side during life's most challenging and rewarding moments.

To my father-in-law Ali, siblings-in-law Sarra, Ghada, Ahmed, and their lovely kids Adam and Lina, thank you for the unforgettable memories and for accepting me as part of the family. To my late mother-in-law Moufida who I never got the chance to know, may you rest in peace in heaven.

To my sister Sahar and my brother Mezen, you are the best siblings anyone could ever ask for. Thank you for always being by my side. May God protect you.

To my grandmothers Zohra and Njila, thank you for being my grandparents and for your warmth and love. To my late grandfathers Azouz and Omar, thank you for your protection and unforgettable memories and may you peacefully rest in heaven.

To my lovely wife Azza, you are a gift from God! Thank you for your unconditional love and support and for always being there for me when it mattered the most. Life is better with you.

Lastly, to my mom Souad and dad Azouz, I will always be in your debt. Thank you for all that you have done for us. Thank you for all the sacrifices you made and the countless sleepless nights you had. Thank you for an amazing childhood and for the great values you engraved in us. Whatever I do will not be enough to properly thank you.

*To my dear parents Souad & Azouz and to my lovely wife Azza.*



# Abstract

The domain of Vision-and-Language (VL) learning has experienced remarkable success across a multitude of tasks such as Visual Question Answering (VQA), Video Question Answering (VideoQA), image captioning, image-text retrieval, and video-text retrieval. Conversely, their considerably more challenging conversational counterparts, such as visual/image<sup>1</sup> and video dialog, received considerably less attention. Furthermore, methodologies addressing these conversational tasks have evolved in isolation from one another, despite their evident similarities.

This thesis aims to address these shortcomings by (1) introducing novel Dialog State Tracking (DST) approaches to further elevate the state-of-the-art performance across various multimodal conversational benchmarks, and (2) presenting efficient methodologies for unifying the widely utilized tasks of visual and video dialog.

We introduce the *first neuro-symbolic* method tailored for multimodal dialog. While this class of hybrid models has demonstrated substantial potential in single-round tasks such as question answering, its suitability for multimodal conversational tasks and DST remains insufficiently explored. Specifically, alongside the introduction of a novel Domain Specific Language (DSL) for CLEVR-Dialog, we propose a dynamic knowledge base that monitors the evolution/state of the dialog, coupled with two parsers that translate natural language questions into programs within said DSL. This methodology has proven effective for tasks characterized by predefined logic and a constrained vocabulary, such as CLEVR-Dialog and Minecraft-Dialog, achieving near-perfect performance. Nevertheless, due to its neuro-symbolic nature, this paradigm encounters limitations in

---

<sup>1</sup>We use "visual dialog" and "image dialog" interchangeably in this dissertation.

generalizing seamlessly across diverse datasets and tasks for which it is not feasible to construct a comprehensive DSL.

In response, the second part of this thesis focuses on enhancing conversational models through multiple *differentiable* DST mechanisms that can be trained end-to-end. Contrary to prior research which predominantly examined unimodal tracking mechanisms, we advocate for transitioning to a *multimodal* framework to adequately address the challenges presented by current multimodal conversational benchmarks. Initially, we propose to independently track the visual and language modalities using *continuous vectors* aggregated by attention-based mechanisms, which can be seamlessly integrated into pre-trained Large Language Models (LLMs).

Although this strategy yields promising outcomes on synthetic datasets such as DVD and SIMMC, it encounters scalability challenges with more complex real-world datasets that encompass a broader range of input modalities, such as VisDial and AVSD. To address this, we introduce more sophisticated *multimodal Graph Neural Network (GNN) based* state tracking modules aimed at augmenting the hidden state representations of LLMs to better handle such datasets.

Our preliminary efforts utilize *static graph topologies* constructed separately for each input modality. While this approach is feasible for datasets where ad hoc graphs can be computed, it fails when such graphs cannot be predetermined. In order to mitigate this limitation, we propose a more universal method whereby both the *graph topologies* and *their node embeddings* are *concurrently* learned. Notably, both approaches succeeded in achieving new state-of-the-art results across a wide range of benchmarks, including VisDial, VisDialPro, VisDialConv, and AVSD.

Finally, in the last part of the thesis, we present the first generative model designed to *unify* video and visual dialog. Our methodology leverages *multimodal experts* that facilitate the *joint learning* of spatial and temporal features in images and video input data. Moreover, we investigate the potential of *cross-learning benefits* between the prominent video and visual datasets of AVSD and VisDial. Our findings indicated significant potential in addressing both tasks collectively, surpassing recent strong baselines in both zero-shot and fine-tuning contexts.

# Zusammenfassung

Das Gebiet von Vision-and-Language-Learning hat beeindruckende Erfolge bei einer Vielzahl von multimodalen Aufgaben erzielt, wie zum Beispiel Bild/Video-basierte Fragebeantwortung, Bild-Text-Retrieval, und Bildbeschriftung. Im Gegensatz dazu erhielten die anspruchsvolleren dialogbasierten Aufgaben, wie Bild/Video-basierte Dialoge, deutlich weniger Aufmerksamkeit. Darüber hinaus haben sich die Methoden, die sich mit diesen dialogbasierten Aufgaben befassen, trotz ihrer offensichtlichen Gemeinsamkeiten unabhängig voneinander entwickelt.

Diese Dissertation zielt darauf ab, diese Schwächen zu mindern, indem (1) neuartige Ansätze vorgestellt werden, um die Leistung von modernen Modellen in verschiedenen multimodalen dialogbasierten Benchmarks weiterzusteigern, und (2) geeignete Lösungen eingeführt werden, um die weitverbreiteten Aufgaben der Bild und Video-basierten Dialoge zu vereinheitlichen.

Im ersten Teil dieser Dissertation, stellen wir *die erste neuro-symbolische* Methode für multimodale Dialoge vor. Insbesondere haben wir neben der Einführung einer neuartigen domänenspezifischen Sprache (DSL) für CLEVR-Dialog eine dynamische Wissensbasis vorgeschlagen, die den Verlauf/Zustand des Dialogs überwacht, sowie zwei Parser, die Sprachfragen in Programme der vorgeschlagenen DSL übersetzen. Diese Methodik hat sich bei Aufgaben mit vordefinierter Logik und begrenztem Vokabular, wie CLEVR-Dialog und Minecraft-Dialog, als effektiv erwiesen und erzielte somit nahezu perfekte Ergebnisse. Aufgrund ihrer neuro-symbolischen Natur stößt dieses Paradigma jedoch an Grenzen, wenn es auf reale Datensätze angewendet wird, bei denen eine vollständige DSL unmöglich zu konstruieren ist.

Als Antwort darauf konzentriert sich der zweite Teil dieser Dissertation auf die Verbesserung von dialogbasierten Modellen durch verschiedene *differenzierbare Dialogzustandsverfolgungsmechanismen*, die End-to-End trainiert werden können. Zunächst schlagen wir vor, die visuellen und sprachlichen Modalitäten unabhängig voneinander mithilfe *kontinuierlicher Vektoren* zu verfolgen, die durch Attention-basierende Mechanismen aggregiert werden und nahtlos in vortrainierte große Sprachmodelle (LLMs) integriert werden können. Diese Strategie zeigte vielversprechende Ergebnisse auf synthetischen Datensätzen wie DVD und SIMMC. Allerdings stieß sie bei komplexeren realen Datensätzen mit einer Vielzahl von Eingabemodalitäten, wie VisDial und AVSD, auf Skalierungsprobleme.

Um dieses Problem zu lösen, führten wir *multimodale GNN-basierte Zustandsverfolgungsmodule* ein mit dem Ziel, die versteckten Zustandsrepräsentationen von LLMs zu erweitern, um solche Datensätze besser zu handhaben. Unsere ersten Ansätze nutzten *statische Graphentopologien*, die für jede Eingabemodalität separat konstruiert wurden. Während dieser Ansatz für Datensätze mit leicht ad hoc berechenbaren Graphen praktikabel ist, versagt er, wenn solche Graphen nicht vorab bestimmt werden können. Um diese Einschränkung zu überwinden, schlugen wir eine universellere Methode vor, bei der sowohl die *Graphentopologien* als auch ihre *Knoteneinbettungen gleichzeitig* gelernt werden können. Beide Ansätze erzielten neue Bestleistungen in einer Vielzahl von Benchmarks, darunter VisDial, VisDialPro, VisDialConv und AVSD.

Schließlich präsentierten wir im letzten Teil der Dissertation das erste generative Modell, das Bild- und Video-basierte Dialoge *vereinheitlicht*. Unsere Methodik beruht auf *multimodalen Experten*, die das *gemeinsame Lernen* räumlicher und zeitlicher Merkmale von Bildern und Videos ermöglichen. Darüber hinaus untersuchten wir das *Potenzial von Cross-Learning* zwischen den prominenten Bild- und Video-Dialog Datensätzen von VisDial und AVSD. Unsere Ergebnisse zeigten ein erhebliches Potenzial, beide Aufgaben gemeinsam zu adressieren, und übertrafen bewährte Baselines sowohl im Zero-Shot als auch im Fine-Tuning-Kontext.

# Contents

<b>Acknowledgements</b>	<b>vii</b>
<b>Abstract</b>	<b>xi</b>
<b>Zusammenfassung</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Scope and Contributions . . . . .	2
1.1.1 Multimodal Dialog State Tracking . . . . .	2
1.1.2 Unification of Multimodal Conversational Tasks . . . . .	6
1.2 Outline . . . . .	7
<b>2 Related Work</b>	<b>11</b>
2.1 Visual & Video Dialog . . . . .	11
2.2 Dialog State Tracking (DST) . . . . .	14
2.3 Neuro-Symbolic Approaches & Reasoning . . . . .	15
2.4 Graph Structure Learning . . . . .	16
2.5 Multimodal Expert-based Training . . . . .	17
<b>3 Foundations</b>	<b>19</b>
3.1 Dataset Structure . . . . .	19
3.2 Training Paradigms . . . . .	20
3.2.1 Classification . . . . .	20
3.2.2 Ranking . . . . .	20
3.2.3 Generation . . . . .	21
3.2.4 Neuro-Symbolic . . . . .	21
3.3 Evaluation Metrics . . . . .	21

<b>4</b>	<b>Neuro-Symbolic DST</b>	<b>25</b>
4.1	Introduction & Contributions . . . . .	26
4.2	Method . . . . .	26
4.2.1	Scene Understanding . . . . .	28
4.2.2	DSL for CLEVR-Dialog . . . . .	28
4.2.3	Program Generation . . . . .	28
4.3	Experiments . . . . .	33
4.3.1	Performance Evaluation . . . . .	33
4.3.2	History during Evaluation . . . . .	33
4.3.3	Visual Dialog Performance . . . . .	34
4.3.4	History Length vs Coreference Distance . . . . .	36
4.3.5	Generalization to Unseen Scenes and Attributes . . . . .	38
4.3.6	Generalization to Longer Dialogs . . . . .	38
4.3.7	Generalization to Unseen Questions Types . . . . .	39
4.3.8	Generalization to Other Scene Domains . . . . .	40
4.4	Limitations & Closing Remarks . . . . .	42
<b>5</b>	<b>Attention-Based DST</b>	<b>43</b>
5.1	Introduction & Contributions . . . . .	44
5.2	Method . . . . .	44
5.2.1	Object Encoder . . . . .	46
5.2.2	Text Encoder . . . . .	46
5.2.3	Dialog State Tracking . . . . .	46
5.2.4	Encoder/Decoder . . . . .	49
5.3	Experiments . . . . .	49
5.3.1	Datasets & Metrics . . . . .	49
5.3.2	Combiner . . . . .	50
5.3.3	Number of Objects and History Turns . . . . .	50
5.3.4	Baselines . . . . .	51
5.3.5	Results on DVD . . . . .	51
5.3.6	Results on SIMMC 2.1 . . . . .	53
5.3.7	Ablation Study . . . . .	55
5.4	Limitations & Closing Remarks . . . . .	56
<b>6</b>	<b>GNN-Based DST with Static Graph Topologies</b>	<b>59</b>
6.1	Introduction & Contributions . . . . .	60
6.2	Method . . . . .	60
6.2.1	Graph Construction and Pruning . . . . .	62
6.2.2	VDGR Layer . . . . .	62
6.2.3	Loss Heads . . . . .	65

6.3	Experiments . . . . .	67
6.3.1	Datasets . . . . .	67
6.3.2	Quantitative Results . . . . .	67
6.3.3	Ablation Study . . . . .	72
6.3.4	Qualitative Results . . . . .	74
6.4	Limitations & Closing Remarks . . . . .	74
<b>7</b>	<b>GNN-Based DST with Latent Learnable Graphs</b>	<b>77</b>
7.1	Introduction & Contributions . . . . .	78
7.2	Method . . . . .	78
7.2.1	Input Representation Learning . . . . .	78
7.2.2	Latent Multimodal DST . . . . .	80
7.3	Experiments . . . . .	86
7.3.1	Datasets . . . . .	86
7.3.2	Metrics . . . . .	86
7.3.3	Main Results . . . . .	86
7.3.4	Ablation Study . . . . .	89
7.3.5	Qualitative Results . . . . .	94
7.4	Limitations & Closing Remarks . . . . .	96
<b>8</b>	<b>Unification of Multimodal Conversational Tasks</b>	<b>97</b>
8.1	Introduction & Contributions . . . . .	98
8.2	Method . . . . .	98
8.2.1	Joint Problem Formulation . . . . .	98
8.2.2	$V^2$ Dial . . . . .	100
8.2.3	Training . . . . .	102
8.3	Experiments . . . . .	105
8.3.1	Datasets . . . . .	105
8.3.2	Evaluation Metrics . . . . .	105
8.3.3	Zero-shot Evaluation . . . . .	107
8.3.4	Fine-tuning Evaluation . . . . .	109
8.3.5	Domain Shift Evaluation . . . . .	110
8.3.6	Expert Swapping Experiment . . . . .	113
8.3.7	Ablation Study . . . . .	113
8.4	Limitations & Closing Remarks . . . . .	114
<b>9</b>	<b>Conclusion</b>	<b>117</b>
9.1	Summary of Findings . . . . .	117
9.2	Future Work . . . . .	119

<b>A</b>	<b>Details &amp; Derivations</b>	<b>123</b>
A.1	The CLEVR-Dialog DSL . . . . .	123
A.2	The Derivation of the Normalized First Failure Round (NFFR) . . . . .	125
A.3	DSL Logic Validation . . . . .	127
A.4	Additional Details about OLViT . . . . .	127
A.4.1	Encoder Configuration . . . . .	127
A.4.2	State Tracker Configuration . . . . .	129
A.5	VDGR Graph Construction . . . . .	129
A.5.1	Image Modality . . . . .	129
A.5.2	Question Modality . . . . .	131
A.5.3	History Modality . . . . .	132
A.6	ELBO loss . . . . .	133
A.7	Training Objectives of $V^2$ Dial . . . . .	134
<b>B</b>	<b>Training Details</b>	<b>137</b>
B.1	OLViT . . . . .	137
B.2	VDGR . . . . .	137
B.3	MST <sub>MIXER</sub> . . . . .	138
B.4	$V^2$ Dial . . . . .	140
<b>C</b>	<b>Qualitative Samples</b>	<b>141</b>
C.1	NSVD . . . . .	141
C.2	VDGR . . . . .	145
C.3	MST <sub>MIXER</sub> . . . . .	145
C.4	$V^2$ Dial . . . . .	145
	<b>Bibliography</b>	<b>154</b>

# List of Figures

- 1.1 Detailed overview of the different work packages that constitute this thesis and how they are connected to each other. We also show the datasets used for each project and the venue where it is published. . . . . 10
- 4.1 Overview of NSVD . First, we learn a structured representation of the visual scene. Then, we induce a caption program that can be run by our executor to initialize its knowledge base. In each subsequent round, we use the question and history to induce a program that answers the question and updates the dynamic knowledge base. (SA = Self-Attention, AR = Attention Reduction, FC = Fully Connected). . . . . 27
- 4.2 **Top:** The concatenative encoder (“concat”) takes the question and the concatenated previous rounds as input and outputs a latent vector and a question representation to the decoder. **Bottom:** The stacking encoder (“stack”) takes the question and the previous rounds as input and attends to each round separately. Then, it outputs a latent vector and a question representation to the decoder. (SA = Self-Attention, AR = Attention Reduction, FC = Fully Connected). . . . . 29
- 4.3 Accuracy of NSVD when trained on limited amounts of data (20%, 40%, 60%, 80%, and 100% of the overall dataset). . . . . 34
- 4.4 Robustness for different coreference distance bins and varying number of rounds in the history. All models were trained with full histories. . . . . 35

4.5	<b>From left to right:</b> Samples of a training image, fine-tuning image with 10 objects, fine-tuning image with 15 objects, and fine-tuning image with 20 objects. . . .	37
4.6	Answer accuracy for different dialog lengths and scene complexities. Our models generalize better to longer dialogs <i>without</i> the need for fine-tuning. . . . .	39
4.7	Sample images from the Minecraft dataset. . . . .	41
5.1	We use MONet and DistilRoBERTa-Base to generate the object and text embeddings, respectively. Then, we add position encoding and append the special [CLS] token to the multimodal features. Finally, we combine the object and language latent state vectors of the current $t$ -th turn ( $s_o^{(t)}$ and $s_l^{(t)}$ ) with the rest of the input, which will be processed by the subsequent transformer layers. . .	45
5.2	Overview of the different variants of our combiner. . .	48
5.3	Performance comparison of $\text{OLViT}$ with different combiners and state tracker variants. . . . .	50
5.4	Performance comparison of $\text{OLViT}$ with different numbers of objects and history turns. . . . .	50
5.5	Qualitative samples of $\text{OLViT}$ . . . . .	54
6.1	$\text{VDGR}$ takes an image, a question, and a history as input and constructs their corresponding graph structures. Each of its layers is composed of cascaded multimodal graphs that learn latent DST representations that augment the hidden states of the subsequent LLM layer. . . . .	61
6.2	Only edges connecting two unmasked node features are eligible for masking with a 15% probability. . . . .	66
6.3	Performance comparison on different dialog rounds of the VisDial v1.0 <i>val</i> split. We only considered sparse metrics since the dense annotations used to compute the NDCG score are not defined on all 10 rounds of the validation dialogs. . . . .	70
6.4	Performance comparison with different number of GNN layers $K$ on the <i>val</i> split of VisDial v1.0. Results are shown for the <i>val</i> split of VisDial v1.0. . . . .	72

6.5	Qualitative samples from the VisDial v1.0 <i>val</i> split. The attention weights correspond to the K-th GNN of the first and last $\text{VDGR}$ layers. We first average them across all attention heads before normalizing them for each node using a $\text{softmax}$ function. For clarity, we only show a subset of the image graph nodes. . . . .	75
7.1	$\text{MST}_{\text{MIXER}}$ takes a video, a dialog history, and a question as output. It uses a BART backbone adapted to deal with multimodal input features and enhances it with our graph-based DST approach. . . . .	79
7.2	In Stage I, $\text{MST}_{\text{MIXER}}$ first gathers multimodal features $\{\mathbf{X}_i\}$ from the previous BART layer and computes their respective initial local structures $\{\tilde{\mathbf{A}}_i\}$ . Then, it simultaneously learns the local latent multimodal graphs and refines the features using a two-stream framework, i.e., $\{\mathbf{A}'_{i,j}, \mathbf{A}''_{i,j}\}$ and $\{\mathbf{Z}'_{i,j}, \mathbf{Z}''_{i,j}\}$ , respectively. Finally, it outputs the final multimodal latent graph $\mathbf{A}_i$ used to compute the local ELBO loss $\mathcal{L}_{\text{ELBO}}^{\text{local}} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{ELBO}}^{\text{local},i}$ . . . . .	81
7.3	Overview of mixing stage II. . . . .	84
7.4	<b>a)</b> Larger values of K make the learning of the global latent graphs more challenging. <b>b)</b> The local ELBO loss $\mathcal{L}_{\text{ELBO}}^{\text{local}}$ facilitates the learning of the global latent graphs. <b>c)</b> The global ELBO loss $\mathcal{L}_{\text{ELBO}}^{\text{global}}$ facilitates the learning of the local latent graphs. All models use SAM and audio features. . . . .	93
7.5	Qualitative comparison of different model ablations. on response generation and latent global graph inference of $q_\phi$ obtained from the last encoder layer. The diagonal blocks (from upper left to lower right) correspond to $\mathbf{V}_{\text{rgb}}, \mathbf{V}_{\text{flow}}, \mathbf{V}_{\text{sam}}, \mathbf{A}_{\text{vggish}}, \mathbf{T}_{\mathcal{H}}$ , and $\mathbf{T}_{\mathcal{Q}}$ , respectively. . . . .	95

8.1	Architectural overview of $\mathbb{V}^2\text{Dial}$ . We adopt a training strategy composed of three stages. <i>First</i> , we only train the multimodal expert layers using spatial-temporal and video/image text matching losses ( $\mathcal{L}_{\text{stm}}, \mathcal{L}_{\text{vtm}}$ ), spatial-temporal and video/image contrastive learning losses ( $\mathcal{L}_{\text{stc}}, \mathcal{L}_{\text{vtc}}$ ), and masked language modeling loss ( $\mathcal{L}_{\text{mlm}}$ ). <i>Second</i> , we couple the expert layers with a frozen pre-trained LLM and fine-tune them end-to-end using a generative loss $\mathcal{L}_{\text{gen}}$ to align their hidden representations. <i>Finally</i> , we additionally fine-tune the LLM weights on the downstream datasets. Each expert is a Feed Forward Network (FFN) composed of two fully connected layers.	99
8.2	<b>Overview of the training and evaluation pipeline of <math>\mathbb{V}^2\text{Dial}</math>.</b> We show the different datasets used to train our model at each stage. Evaluations are conducted on the most popular video and visual dialog datasets of AVSD and VisDial, respectively. (🎥 = video data, 📷 = image data, 🗣️ = closed / visual captioning data, 💬 = dialog data).	104
8.3	Instead of training a dedicated NSP head, we propose a ranking scheme based on the cosine similarity of the candidate answers' embeddings with the respect to those of the generated ones. We used RoBERTa <sub>large</sub> [127] and OpenAI Text Embedding-3 to generate these embeddings.	106
8.4	Zero-shot qualitative examples of $\mathbb{V}^2\text{Dial}$ before and after fine-tuning on VisDial and AVSD. The former teaches the model to answer question with brief responses whereas the latter teaches it to produce longer and more elaborate answers.	111

A.1	The program <code>extreme-center(cylinder, small)</code> induced from the caption “there is a small round thing sitting in the center of the view” does not lead to a unique initialization of our executor’s knowledge base as there are <i>two</i> small spheres in the center of the scene. By our logic, we consider it to be the cyan one. Incorrectly initializing the knowledge base leads to confusion when answering the subsequent questions. The blue and red colors indicate a match or a mismatch between the predicted answer and the ground truth, respectively. . . . .	126
A.2	The different spatial relationships used in constructing the image graph. The orange and green rectangles correspond to the bonding boxes of two objects within the scene. . . . .	130
A.3	<b>Right:</b> The distribution of the image graph relationship classes within the training split of VisDial v1.0. <b>Left:</b> The distribution of the question graph relationship classes within the training split of VisDial v1.0. . . . .	131
A.4	A sample of the dependency relationships between question word pairs. . . . .	132
A.5	A qualitative sample of the coreference relationships between different dialog rounds. The hub-node was not visualized for clarity. . . . .	132
C.1	Both of our models generated correct programs to predict the answers. The blue color indicates a match between the predicted program/answer and the ground truth. . . . .	142
C.2	While <i>NSVD-stack</i> answered all questions correctly, <i>NSVD-concat</i> failed at round 7. The blue and red colors indicate a match or a mismatch between the predicted program/answer and the ground truth, respectively. . . . .	143
C.3	Inference example of our models on a test instance of Minecraft-Dialog. The blue and red colors indicate a match or a mismatch between the predicted program/answer and the ground truth, respectively. . . . .	144

C.4	Qualitative samples from the <i>val</i> split of the VisDial v1.0 dataset. We compared the predictions of VDGR (orange) with the ground truth answers (blue) and the predictions of VD-PCR (pink). The answers of both models correspond to the top-1 predictions. . . . .	146
C.5	Qualitative samples from the <i>val</i> split of the VisDial v1.0 dataset. We compared the predictions of VDGR (orange) with the ground truth answers (blue) and the predictions of VD-PCR (pink). The answers of both models correspond to the top-1 predictions. . . . .	147
C.6	Qualitative samples of $MST_{MIXER}$ from the test split of AVSD-DSTC7. . . . .	148
C.7	Negative qualitative samples of $MST_{MIXER}$ from the test split of AVSD-DSTC7. . . . .	149
C.8	Zero-shot qualitative examples of $V^2Dial$ on AVSD. 🤖 = generated answers, 🏆 = golden ground-truth answers, ✅ / ❌ = success / failure cases). . . . .	150
C.9	Zero-shot qualitative examples of $V^2Dial$ on VisDial. 🏆 = top ranked candidate answers, . . . . .	151
C.10	Fine-tuning qualitative examples of $V^2Dial$ on AVSD. . . . .	152
C.11	Fine-tuning qualitative examples of $V^2Dial$ on VisDial. . . . .	153

# List of Tables

1.1	List of publications produced throughout the course of the doctoral studies. . . . .	9
3.1	Overview of the multimodal conversational datasets used in the course of this thesis. (📹 = video data, 📷 = image data, Syn = Synthetic, NS = Neuro-Symbolic, Cls = Classification, Rnk = Ranking, Gen = Generation, Acc = Accuracy). . . . .	22
4.1	Performance comparison on CLEVR-Dialog <i>test</i> . † represents $p < 0.01$ compared to the second best score in the respective column. Best and second best performances are in <b>bold</b> and <u>underlined</u> , respectively. . . . .	34
4.2	Results when training on simple scenes and testing on more complex ones. ‡, †, and * represent $p < 0.01$ , $p < 0.05$ and $p \geq 0.05$ compared to the second best score in each column, respectively. . . . .	37
4.3	Results when training on split AA and testing on split BB. ‡, †, and * represent $p < 0.01$ , $p < 0.05$ and $p \geq 0.05$ compared to the second best score in each column, respectively. . . . .	40
4.4	Performance comparison on Minecraft-Dialog. ‡ and * represents $p < 0.01$ and $p \geq 0.05$ compared to the second best score in the respective column, respectively. . . . .	41

5.1	Performance comparison on DVD. Best and second best performances are in <b>bold</b> and <u>underlined</u> , respectively. $Q$ , $C$ , and $V$ denote question, history/context, and visual input, respectively, and † denotes training with additional supervision. . . . .	52
5.2	Performance comparison on SIMMC 2.1. Best and second best performances are in <b>bold</b> and <u>underlined</u> , respectively. . . . .	53
5.3	Performance comparison of different ablations of OLViT on the DVD and SIMMC 2.1 <i>val</i> splits. † and ‡ represent significant improvement with $p < 0.05$ and $p < 0.01$ compared to the second best score. . . . .	55
5.4	The effect of pre-trained language models on the performance of our model. Performance is measured on the DVD <i>val</i> split. . . . .	56
6.1	Performance comparison on VisPro and VisDialConv. The best and second-best results are in <b>bold</b> and <u>underlined</u> , respectively. † indicates higher is better and ‡ indicates lower is better. . . . .	68
6.2	Performance comparison on the <i>val</i> split of VisDial v0.9 dataset. Normalized Discounted Cumulative Gain (NDCG) is not supported in this version of the dataset. . . . .	69
6.3	Performance comparison on the <i>val</i> split of VisDial v1.0 dataset. . . . .	69
6.4	Performance comparison on the <i>test-std</i> split of VisDial v1.0 dataset. ♣ indicates fine-tuning on dense annotations and ◇ denotes ensemble model. ‡ denotes the use of extra large datasets for training. . . . .	71
6.5	Performance comparison of our model ablations on the <i>val</i> split of VisDial v1.0. . . . .	73
7.1	Results on AVSD-DSTC7 and AVSD-DSTC8. Best and second best performances are in <b>bold</b> and <u>underlined</u> , respectively. ♣ = Two-stage training. . . . .	87
7.2	Results on AVSD-DSTC10. . . . .	88
7.3	Results on SIMMC. . . . .	89
7.4	Results on open-ended NExT-QA <sup>◇</sup> . . . . .	89
7.5	Ablation experiments of $MST_{MIXER}$ . . . . .	90

7.6	Comparison between different ablated versions of our model. All ablations use SAM and audio features. <b>TRN</b> means that the model replaces the global and local multimodal GNNs with vanilla transformer layers, and <b>RAND</b> denotes that it uses random latent graphs instead of learning them. Our full model is highlighted in blue. . . . .	92
7.7	Comparison between different ablated versions of our model. All ablations were trained with SAM and audio features and with the optimal hyperparameters as the full model. <b>IB</b> = Initialization Bias, <b>MMC</b> = Multimodal Conditioning. . . . .	93
8.1	Overview of the available features for each training stage and visual input type. . . . .	102
8.2	Zero-shot performance comparison on AVSD-DSTC10, AVSD-DSTC8 and AVSD-DSTC7. Best and second-best performances are in <b>bold</b> and <u>underlined</u> , respectively. <b>♦</b> indicates that we evaluated the model. ( <b>B-n</b> = BLEU-n, <b>M</b> = METEOR, <b>R</b> = ROUGE-L, <b>C</b> = CIDEr). . . . .	106
8.3	Zero-shot performance comparison in the generative setting on the VisDial v1.0 val split. OpenAI TE-3 = OpenAI Text Embedding-3. . . . .	107
8.4	Fine-tuning performance comparison on AVSD-DSTC10, AVSD-DSTC8 and AVSD-DSTC7. VideoLLAMA 2 [43] was trained on AVSD amongst other datasets. Additional model comparisons can be found in Table 7.1 and Table 7.2 of <b>Chapter 7</b> . . . . .	108
8.5	Fine-tuning performance comparison on the VisDial v1.0 val split. <b>♣</b> indicates that we trained and evaluated the model. . . . .	109
8.6	Domain shift evaluation between the respective <i>most prominent</i> video and visual dialog datasets of AVSD and VisDial. $\square \rightarrow \triangle$ means that the model was pre-trained on dataset $\square$ before fine-tuning on dataset $\triangle$ . RoBERTa <sub>large</sub> was used to generate the sentence embeddings for ranking. . . . .	112

8.7	Expert swapping results. $\mathcal{E}_\square \leftrightarrow \mathcal{E}_\Delta$ means that the $\square$ features are <i>inadequately</i> routed at test time through $\mathcal{E}_\Delta$ and vice versa. The other experts remain unchanged. RoBERTa <sub>large</sub> was used to generate the sentence embeddings for ranking. . . . .	113
8.8	Ablation results of $\mathbb{V}^2\text{Dial}$ . RoBERTa <sub>large</sub> was used to generate the sentence embeddings for ranking. . . . .	114
A.1	Our CLEVR-Dialog DSL. We present each function with its expected arguments types, output, and knowledge base operations. Given a set of $n$ arguments, the $\uplus$ operator selects $m \leq n$ possible ones. . . . .	124
A.2	Argument types of our DSL. . . . .	125
A.3	Quantitative analysis of our DSL. The high program accuracies demonstrate that our models follow the implemented logic to predict the correct answer, i.e. they do not execute false programs that by chance might lead to correct predictions. When tested with the ground truth scene annotations and programs, our executor reaches an answer-accuracy of 99.99% showcasing its highly reliable execution logic. . . . .	126
A.4	Encoder and state tracker configuration experiment. Performance is measured on the DVD <i>val</i> split. . . . .	128
B.1	Detailed hyperparameter setting for the different training stages of $\mathbb{V}\text{DGR}$ . $\text{Linear}_{\mathcal{I} \rightarrow \mathcal{H}}(\cdot)$ and $\text{Linear}_{\mathcal{Q} \rightarrow \mathcal{I}}(\cdot)$ denote the linear layers that produce the history and image hub-node features, respectively. . . . .	138
B.2	Detailed hyperparameter setting of the training and inference of our best $\text{MST}_{\text{MIXER}}$ model. . . . .	139
B.3	Detailed hyperparameter setting of $\mathbb{V}^2\text{Dial}$ . . . . .	140

# Acronyms

AI	Artificial Intelligence. 1, 4
AMT	Amazon Mechanical Turk. 20, 22
BLEU	Bilingual Evaluation Understudy. 23, 44, 50, 53, 55, 56, 86
CE	Cross-Entropy. 20, 21, 72, 103
CIDEr	Consensus-based Image Description Evaluation. 23, 86, 109
CNN	Convolutional Neural Network. 11
CV	Computer Vision. 1, 117
DSL	Domain Specific Language. xi, xii, xviii, xxviii, 3, 7, 21, 28, 31, 34, 42, 118–120, 123, 124, 126, 127
DST	Dialog State Tracking. xi, xii, xvii, xx, xxi, 2–5, 7, 13, 15–17, 26, 31, 42, 44, 53, 56, 57, 60–62, 65, 69, 70, 72–74, 78–80, 91, 96, 117–119
ELBO	Evidence Lower Bound. 85, 86, 91, 94, 133, 139
FFN	Feed Forward Network. xxii, 17, 99, 101, 121
FFR	First Failure Round. 33
GEM	Graph Edge Masking. 60, 65–67

GNN	Graph Neural Network. xii, xiv, xxi, 4, 5, 7, 12–16, 60, 62–64, 66, 69, 72–76, 78, 80, 82, 89–91, 96, 118, 138
IB	Initialization Bias. 85
LLM	Large Language Model. xii, xiv, xx, xxii, 2–7, 11–14, 16, 44, 56, 57, 60–62, 66, 74, 76, 99, 100, 103, 105, 118, 120, 121
LN	Layer Normalization. 47, 101
LST	Language State Tracker. 44, 47, 48, 51, 55, 56
LSTM	Long Short-Term Memory. 28, 30, 31
METEOR	Metric for Evaluation of Translation with Explicit Ordering. 23, 86
MLLM	Multimodal Large Language Model. 107, 121
MLM	Masked Language Modeling. 60, 65, 102
MLP	Multilayer Perceptron. 64
MMC	Multimodal Conditioning. 83, 85
MoEs	Mixture-of-Experts. 17, 121
MRM	Masked Region Modeling. 60, 65
MRR	Mean Reciprocal Rank. 21, 69, 70
MSA	Multi-Head Self-Attention. 47, 101
NDCG	Normalized Discounted Cumulative Gain. xxvi, 21, 69, 71, 72, 107, 110
NFFR	Normalized First Failure Round. xviii, 26, 33, 36, 40, 41, 117, 125
NLG	Natural Language Generation. 23
NLP	Natural Language Processing. 1, 117
NSP	Next Sentence Prediction. 13, 20, 60, 63, 65, 66, 98, 105
OOD	Out-of-Distribution. 26, 42, 117
OST	Object State Tracker. 44, 47, 51, 55, 56
R@k	Recall at k. 21

---

RL	Reinforcement Learning. 11
ROUGE	Recall-Oriented Understudy for Gisting Evaluation. 23, 86
STC	Spatial-Temporal Contrastive Learning. 102, 103, 134
STM	Spatial-Temporal Matching. 102, 103, 134
VAE	Variational Auto-Encoder. 11, 16
VCR	Visual Commonsense Reasoning. 12
VI	Variational Inference. 6, 7, 16, 83, 85, 133
VideoQA	Video Question Answering. xi, 2, 8, 12, 15, 26
VL	Vision-and-Language. xi, 1, 12, 17, 78, 96, 98, 118, 119, 121, 134
VLM	Vision-and-Language Model. 14, 17, 60, 78, 80, 96
VQA	Visual Question Answering. xi, 2, 12, 15, 26, 120
VTC	Vision-Text Contrastive Learning. 102, 134
VTM	Vision-Text Matching. 102



# 1

## Introduction

In the last decade, the field of Artificial Intelligence (AI) and deep learning witnessed unprecedented success in a wide range of domains, ranging from Computer Vision (CV) to Natural Language Processing (NLP). These breakthroughs were mainly driven by the abundance of training data and the advancement in model design. Eventually, interest in multimodal domains such as VL began to increase and many tasks (e.g. *visual question answering* [15], *video question answering* [190, 6], *image captioning* [37], *image-text retrieval* [28], *video-text retrieval* [215], etc.) gained more and more importance. In stark contrast, their significantly more challenging conversational counterparts, such as visual/image [48] and video dialog [9], received considerably less attention. Moreover, methods for these conversational tasks have advanced independently of each other despite the apparent similarities between them.

In this thesis, we aim to make progress along two major fronts in an attempt to address the previous limitations: (1) In the first part of the thesis, we propose novel architectural improvements primarily

geared towards multimodal conversational models to further push their capability and narrow the gap to other multimodal tasks. To this end, we propose not only neuro-symbolic but also end-to-end differentiable DST mechanisms that can be seamlessly integrated into pre-trained LLMs. (2) In the second part, we explore how to best unify the divergent works in video and visual dialog and study whether and to what extent these tasks can mutually benefit from one another.

## **1.1 Scope and Contributions**

Next, we give a more in-depth overview of the problems we address in this thesis. In particular, we touch on the main topics of the different chapters and highlight our contributions.

### **1.1.1 Multimodal Dialog State Tracking**

Modeled after human-human communication, multimodal dialog involves reasoning about a visual scene (i.e. an image or a video) through multiple rounds of questions and answers, making it inherently more challenging than its single-round counterparts (e.g. VQA and VideoQA). Specifically, dialog models need to consider the previous complex and potentially ambiguous conversation history in addition to the current question to be able to answer it efficiently.

Early works in this domain primarily focused on task-oriented dialog systems that are specific for particular domains, such as hotel or restaurant reservation [26]. One of the most successful approaches to address the specific challenges of conversational systems was the introduction of DST [79] that follows the user’s needs at each turn according to a conversation history (e.g. a person booking a hotel room). Previous works [145, 26] formulated DST as a slot filling task [192] where the slots and slot values are constrained by a knowledge domain (e.g. hotel domain) and database schema (e.g. tabular data). For example, some work [140] proposed detecting a pair of slot values that represent the user’s goal using a neural belief tracker (NBF). Other works [64, 187] suggested creating a dialog state that iteratively interacts with a user in natural language.

That said, the concept of DST has evolved over the years beyond merely tracking user’s needs and is now broadly used when a model keeps track of what it believes to be relevant to answer the question at hand. Although previous works have made considerable advances

in DST, they still suffer from some key limitations. Specifically, these approaches remained predominately unimodal even though some were applied within a multimodal context (e.g., video dialog [139, 104], visual dialog [146], image retrieval [64], recommender systems [187]) by only tracking one modality (mostly the visual input). However, the current landscape of the field requires transitioning to a multimodal DST paradigm in order to be able to deal with the challenge of multimodal conversational datasets. Some recent works [97, 138] have moved closer to achieving this objective. However, their contributions were limited to synthetic and automatically generated datasets that do not reflect the real challenges of real-world scenarios.

### **Neuro-Symbolic DST**

Our first attempt is motivated by the success of hybrid neuro-symbolic approaches [197, 136, 69, 165]. Despite their great potential in tackling single-round tasks [85, 196], previous works missed on their potential in modeling DST for multimodal conversational tasks.

To this end, we present `NSVD`, the first neuro-symbolic model specifically geared towards visual dialog. At its core, it employs a novel dynamic knowledge base to keep track of the dialog evolution and is updated after each turn by a program executor. This executor operates on top of two parsers that translate captions and questions in natural languages into programs in the target domain DSL. Through extensive experiments, we show that `NSVD` achieves near-perfect performance on CLEVR-Dialog [99] and Minecraft-Dialog while only requiring a fraction of the training data compared to fully-connectionist models<sup>1</sup>. Furthermore, we demonstrate strong out-of-distribution capabilities across many scenarios within CLEVR-Dialog.

However, despite all its advantages, `NSVD` faces crucial generalizability limitations, as it cannot be applied to datasets with unknown or too complex DSL. Thus, in the next part, we propose more flexible, differentiable, and end-to-end trainable DST approaches that can be seamlessly integrated into pre-trained LLMs.

### **End-to-End Differentiable DST**

*Attention-Based Approaches.* We first start by proposing `OLViT`, a novel model equipped with the first multimodal DST approach, which,

---

<sup>1</sup>Fully-connectionist models represent knowledge implicitly through connection weights, while neuro-symbolic models explicitly represent knowledge using symbols and rules.

in contrast to previous works, separately tracks the vision and language input modalities. At its core, it employs two attention-based blocks (i.e. one for the vision and the other for the language modality) to select the most important visual or language token embeddings based on the self-attention scores. Then, these selected features are further processed by transformer [177] layers to compute two dense vectors that form a latent representation of the dialog state. Finally, these state vectors are concatenated with the rest of the input tokens and are jointly processed by a pre-trained LLM.

In stark contrast to previous work [104], our novel idea of performing DST in a latent space makes it applicable across different tasks and datasets. Specifically,  $\text{OLViT}$  achieves new state-of-the-art performance not only on the discriminative video dialog dataset, i.e. DVD [108], where it has to predict an answer from a pool of candidate responses but also on SIMMC [138] where it needs to generate the answer tokens autoregressively. Although this initial effort to perform multimodal DST can theoretically be extended to multiple datasets, it mainly showed promising results on synthetic and automatically generated benchmarks. To be able to tackle the more challenging real-world benchmarks, we propose more sophisticated multimodal DST approaches that harness the advantages of two of the most promising neural network architectures todate, i.e. LLMs and GNNs.

*GNN-Based Approaches.* GNNs [25, 179, 93, 188] are gaining more importance in many AI fields due to their ability to generate low-dimensional node embeddings based on the individual features and the information from a local neighborhood. This advantage of learning locally informed features adequately complements the strengths of pre-trained LLMs that excel at learning the global underlying context of the input data. Eventually, early works on the intersection of these two learning paradigms [216, 113, 129] proposed a cascaded approach where the LLM and the GNN are deployed one after the other without any information exchange between the GNN nodes and the transformer [177] layers. More recent works [209, 193] proposed a more intertwined frameworks alternating between GNN and LLM layers where the features of the former augment those of the latter, which we adopt in this thesis.

Despite their success, these initial efforts primarily focused on unimodal tasks such as question answering or link prediction, and

neglected the more challenging multimodal counterparts. To this end, we pioneer the first multimodal GNN-enhanced conversational models in an attempt to close this research gap. Specifically, we propose a novel multimodal GNN-based DST approach that selects the most influential features of each input modality and processes them separately using dedicated GNNs. The resulting features are used to enhance the hidden states of the subsequent LLM layer and act as a latent representation of the dialog state. This approach differs from the previous attention-based approach of `OLViT` in the sense that the latent dialog state representation is not aggregated into a single dense vector per modality. Rather, every feature directly contributes to enhancing its representation in the upcoming layer.

Our first contribution to this line of research was the introduction of `VDGR`, which was to the best of our knowledge the first attempt to perform multimodal DST using GNNs. Specifically, `VDGR` constructs ad hoc graph topologies for each input modality. Then, it alternates between multimodal graph aggregation and LLM layer to augment their hidden state representations. In addition, we propose a novel feature propagation technique for multimodal GNNs which relies on hub-nodes that link to all other nodes of a different modality in order to alleviate their lack of intermodal context. When applied on visual dialog, `VDGR` achieves state-of-the-art results on four real-world datasets. In particular, it outperforms strong baselines on the popular and widely used discriminative `VisDial v1.0` [48] benchmark.

Our second contribution adopts the same idea of performing latent DST using multimodal graphs as in `VDGR` and remedies one of its main shortcomings. As mentioned above, one of the prerequisites of `VDGR` is being able to construct ad hoc graph topologies for each input modality. Although this might be feasible for some datasets such as `VisDial v1.0`, it poses great challenges for benchmarks where it is difficult or even impossible to construct such graphs. For instance, this becomes more apparent when transitioning from visual/image dialog to video dialog. Specifically, when addressing the most prominent video dialog dataset of `AVSD` [9], previous models used pre-extracted `I3D-flow`, `I3D-RGB`, and audio features for which there is no clear way to construct ad hoc graph topologies.

To this end, we present `MSTMIXER`, a novel video dialog model with the main advantage of simultaneously learning graph node embeddings and their topologies in the form of latent graphs. Specifically,

$\text{MST}_{\text{MIXER}}$  follows a divide-and-conquer paradigm. First, it tracks the most important constituents of each input modality in a similar way to  $\text{OLViT}$ . Then, it predicts the missing underlying structure of the selected constituents of each modality by learning local latent graphs using a novel multimodal graph structure learning method based on Variational Inference (VI). Subsequently, in the conquer stage, the learned local graphs and features are parsed together to form a global graph operating on the mix of all modalities, further refining its structure and node embeddings. Finally, the fine-grained graph node features are used similarly to  $\text{VDGR}$  to enhance the hidden states of the backbone LLM. Through extensive evaluations, we show that  $\text{MST}_{\text{MIXER}}$  manages to outperform state-of-the-art models on all three video dialog benchmarks of AVSD [9] and SIMMC [138]. Additionally, it achieves new state-of-the-art results on the recent video question dataset of NEXT-QA [189] underlining its generalizability potential.

### 1.1.2 Unification of Multimodal Conversational Tasks

In the last part of the thesis, we shift our attention to an equally important research gap in the field of multimodal conversational models. As discussed previously, research on visual/image dialog and video dialog evolved in isolation from one another despite their shared data structure similarities limiting their applicability potential. For instance, the respective most prominent visual and video dialog datasets of VisDial and AVSD come with similar data structures: a visual input in the form of an image or a video paired with a short caption/description and multiple rounds of question-answer pairs.

To this end, we propose  $\mathbb{V}^2\text{Dial}$ , a novel expert-based model specifically geared towards simultaneously handling image and video input data, thus unifying both video and visual dialog tasks. Our model jointly learns the spatial and temporal features of images and videos by routing them through dedicated experts, i.e. separate transformer [177] Feed-Forward Networks (FFNs), and aligns them using matching and contrastive learning techniques. Furthermore, we systemically study the domain shift between the two tasks by investigating whether and to which extent these seemingly related tasks can mutually benefit from their respective training data. Extensive evaluations on the widely used video and visual dialog datasets of AVSD and VisDial show that  $\mathbb{V}^2\text{Dial}$  achieves new state-of-the-art results across both datasets in zero-shot and fine-tuning settings.

## 1.2 Outline

This thesis is based on *five* scientific publications as shown in [Table 1.1](#). Furthermore, [Figure 1.1](#) illustrates how these publications are connected to each other, the datasets they used, and the venues where they were published. In the text below, we discuss the connections between the individual publications in more detail and outline the overall structure of the dissertation.

In [Chapter 2](#) and [Chapter 3](#), we discuss relevant related work to our research and cover the basic concepts of multimodal conversational tasks that will be encountered in later chapters of the thesis, respectively. Then, in [Chapter 4](#) we introduce `NSVD`, the first neuro-symbolic DST approach for visual dialog, alongside a dynamic knowledge base and a novel DSL for the target domain. It was primarily evaluated on symbolic and synthetic visual dialog datasets such as CLVER-Dialog [99] and Minecraft-Dialog [1].

Subsequently, we turn towards researching latent multimodal differentiable DST approaches. Specifically, in [Chapter 5](#) we propose `OLViT` that separately tracks the vision and language input modalities using attention-based blocks. It extends `NSVD` on two fronts: First, we transition to the video domain where the model needs to reason about a dynamic scene instead of a static images. Second, it can be trained end-to-end making it flexible to generalize to multiple tasks such as discriminative and generative video dialog which we assessed using the datasets DVD [108] and SIMMC [138], respectively.

Thereafter, in [Chapter 6](#) we transition to a more sophisticated GNN-based DST approach by developing `VDGR` that alternates between multimodal GNN and LLM layers. Unlike the previous two works, `VDGR` is able to deal with the complexities of real-world datasets and outperforms state-of-the-art models on multiple benchmarks such as VisDial [48], VisDialConv [7] and VisPro [203].

To overcome the limitation of `VDGR` that requires ad hoc construction of graph topologies, we propose `MSTMIXER` in [Chapter 7](#) that simultaneously learns these topologies alongside their node embeddings using VI. In addition, `MSTMIXER` extends `VDGR` to the video domain and is able to generate answers auto-regressively instead of merely ranking a given set of candidate responses. As can be seen in [Figure 1.1](#), `MSTMIXER` was benchmarked on a plethora of tasks and benchmarks such as video dialog (AVSD [9], SIMMC [97]) and

VideoQA (NExT-QA [189]).

Then, in **Chapter 8** we focus on unifying visual and video dialog and develop  $V^2Dial$  – an expert-based model capable of jointly learning from image and video input data. This model is motivated by the divergent research directions on visual and video dialog and aims to investigate the synergies and the cross-learning benefits of these two tasks. In contrast to  $VDGR$  and  $MST_{MIXER}$ ,  $V^2Dial$  is able to jointly learn on image and video data and thus can tackle visual and video dialog simultaneously. As shown in **Figure 1.1**, we assessed it on both AVSD [9] and VisDial [48] datasets.

Finally, in **Chapter 9** we conclude this thesis and provide an outlook on future research directions.

Chapter	Publication
<i>Thesis contributions</i>	
4	<b>Neuro-Symbolic Visual Dialog</b> [1] <u>Adnen Abdessaied</u> , Mihai Băce, Andreas Bulling In <i>Proceedings of the International Conference on Computational Linguistics (COLING)</i> , 2022 – [Oral]
5	<b>OLViT: Multi-Modal State Tracking via Attention-Based Embeddings for Video-Grounded Dialog</b> [2] <u>Adnen Abdessaied</u> , Manuel von Hochmeister, Andreas Bulling In <i>Proceedings of the International Conference on Computational Linguistics (COLING)</i> , 2024
6	<b>VD-GR: Boosting Visual Dialog with Cascaded Spatial-Temporal Multi-Modal GRaphs</b> [5] <u>Adnen Abdessaied</u> , Lei Shi, Andreas Bulling In <i>Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)</i> , 2024
7	<b>Multi-Modal Video Dialog State Tracking in the Wild</b> [4] <u>Adnen Abdessaied</u> , Lei Shi, Andreas Bulling In <i>Proceedings of the European Conference on Computer Vision (ECCV)</i> , 2024
8	<b>V<sup>2</sup>Dial: Unification of Video and Visual Dialog via Multimodal Experts</b> [3] <u>Adnen Abdessaied</u> , Anna Rohrbach, Marcus Rohrbach, Andreas Bulling In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)</i> , 2025
<i>Other contributions</i>	
n/a	<b>Limits of Theory of Mind Modelling in Dialogue-Based Collaborative Plan Acquisition</b> [23] Matteo Bortoletto, Constantin Ruhdorfer, <u>Adnen Abdessaied</u> , Lei Shi, Andreas Bulling In <i>Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)</i> , 2024
n/a	<b>Video Language Co-Attention with Multi-modal Fast-Learning Feature Fusion for VideoQA</b> [6] <u>Adnen Abdessaied</u> , Ekta Sood, Andreas Bulling In <i>Proceedings of the ACL Workshop on Representation Learning for NLP (ACL-W)</i> , 2022

TABLE 1.1: List of publications produced throughout the course of the doctoral studies.

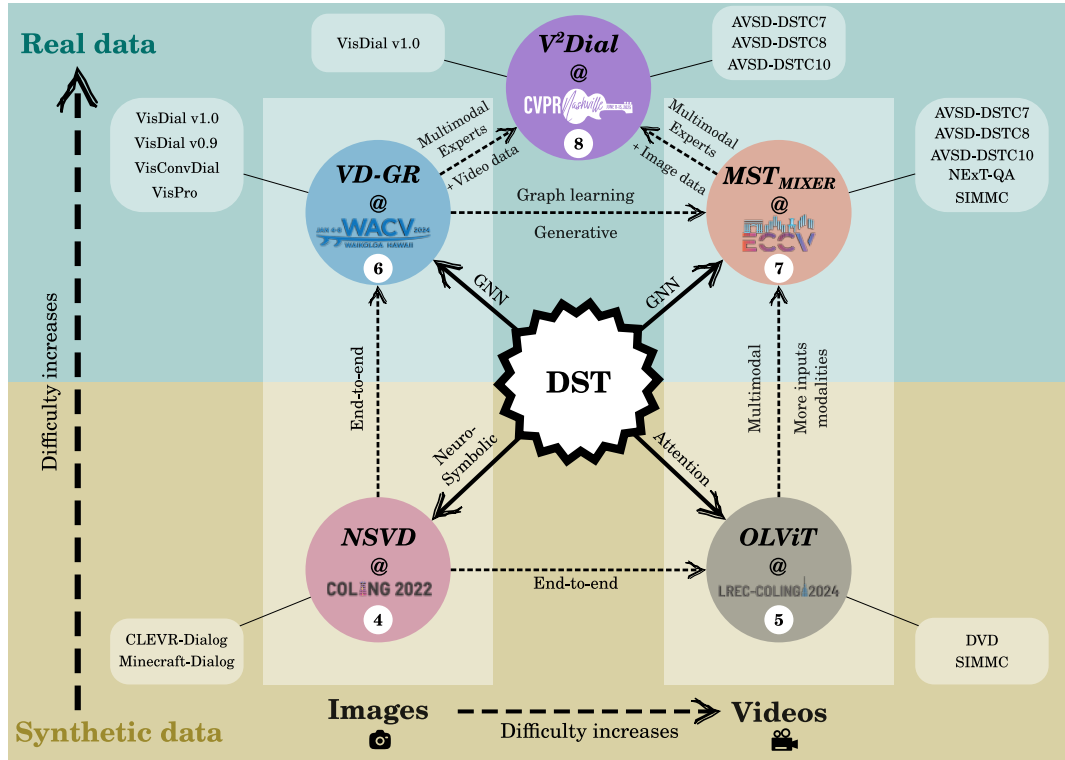


FIGURE 1.1: Detailed overview of the different work packages that constitute this thesis and how they are connected to each other. We also show the datasets used for each project and the venue where it is published.

# 2

## Related Work

### 2.1 Visual & Video Dialog

#### Visual Dialog

Das *et al.* [48] introduced early visual dialog datasets (i.e. VisDial v0.9 and v1.0) and models which extract feature representations from images, questions, and history rounds using Convolutional Neural Networks (CNNs) [110] and Long Short-Term Memory cells (LSTMs) [73]. These features are then fused together using attention and different other mechanisms ranging from multi-step attention [59], to incorporating scene graphs [82], over using Variational Auto-Encoders (VAEs) [149] to perform multimodal reasoning. Whereas some works [144] explored explicit reasoning based on the dialog structure, others proposed training models with Reinforcement Learning (RL) [49] techniques. That said, the performance of these early models remained quite insufficient and greatly benefited from the advances made by pre-trained LLMs.

ViLBERT [131] and XLMERT [176] were among the first attempts to leverage pre-trained LLMs for VL tasks (e.g. VQA [15], Visual Commonsense Reasoning (VCR) [207], and image retrieval [201]). In practice, they used a transformer encoder as a backbone and deployed a two-stream architecture to separately encode visual and textual inputs. VisDial-BERT [141] built on top of ViLBERT and was specifically adapted to the visual dialog task, achieving a new state-of-the-art performance on VisDial [48]. More recently, UTC [32] was introduced that built on VisDial-BERT and used two inter-task contrastive losses to improve training. Another class of models used a single stream to encode multimodal inputs such as B2T2 [11], VisualBERT [118], VL-BERT [170], and UNITER [39]. VD-BERT [181] adapted a single stream VQA model [214] to the visual dialog task and managed to achieve good performance without relying on external datasets. In parallel, other works leveraged an equally interesting class of models, i.e. GNNs, to improve the performance of visual dialog systems.

GNN-EM [212] was one of the earliest models to deploy GNNs for visual dialog. It relied on building graphically structured neural networks to approximate the learning and inference processes of graphical models [125, 172, 119, 55]. FGA [160] developed a general attention mechanism that borrowed from the core idea of GNN message passing, and was able to combine representations of many input modalities. CAG [63] proposed representing the image as a fully-connected graph of objects whose adjacency matrices could be dynamically updated using a question-based attention mechanism. Subsequently, GOG [33] proposed representing each modality as a graph before processing them by a light-weight fusion module [143] to rank the candidate answers. Similarly to CAG [63], DualVD [82] proposed representing the image as a graph consisting of connected objects. They argued that the visual view helped capture the appearance-level information, while the semantic view enabled the agent to understand high-level visual semantics from the entire image to the local regions [82].

## Video Dialog

Video dialog has emerged as a natural extension to VQA, VideoQA, and visual dialog. Almari *et al.*[9] proposed AVSD, one of the most popular video dialog datasets based on Charades [166], which has

become the default dataset for the task. In early work, Zhao *et al.* [210] proposed a hierarchical attention context learning approach and used a multi-stream network for joint video representation. Subsequently, Jin *et al.* [83] suggested progressively updating query information based on the dialog history and the video content while Le *et al.* [106] proposed the Multimodal Transformer Network (MTN), an approach that attended to inputs from different embedding spaces and fused the multimodal information into newly combined representations. Inspired by the success of graph neural networks, Le *et al.* [102] proposed a framework for discovering information flows among dialog turns through a semantic graph based on lexical components in each question and answer. Later works [105, 121] achieved great success by leveraging and fine-tuning pre-trained LLMs [112, 154] on such datasets. Others used GNNs to perform reasoning on the visual scene [90] in an attempt to improve performance. Pham *et al.* [151] proposed an object-centric model to track object-associated dialog states upon receiving new questions. Inspired by the success of neural module networks [13, 14], Le *et al.* [103] introduced VGNMN to model the information retrieval process in video-grounded language tasks as a pipeline of neural modules. More recently, Yoon *et al.* [199] introduced a dedicated regularization loss that helps models hallucinate less.

### **Joint Generative Formulation**

The best performing visual dialog models [182, 141, 202] leveraged pre-trained LLMs and were trained using a Next Sentence Prediction (NSP) head with negative sampling. Although some work [181, 32, 33] claimed to train generative visual dialog models, they did so by providing a generative mask where each token can only attend to its left tokens. However, most of them are trained using the NSP head like their discriminative counterparts. In contrast, works on video dialog follow a purely generative training paradigm and achieved great success building on top of powerful pre-trained LLMs [112, 154]. Although video dialog emerged as a natural extension to visual dialog with apparent data structure similarities, research on both tasks evolved separately missing out on the potential of a unifying model that learns from the available training data of both tasks.

This thesis differs from the aforementioned works in two major aspects: (1) We perform multimodal latent DST to actively track the

constituents of each input modalities separately. Specifically, we marry GNNs with pre-trained LLMs in a novel way to benefit from the advantages of both worlds. Contrarily to [194, 198] where simple integration approaches for plain monomodal tasks were introduced, we propose a sophisticated integration method for the complex and rich multimodal visual and video dialog tasks by exploiting the local structures of each modality through dedicated GNNs to enhance the hidden states of the LLM layers. (2) We propose to simultaneously learn the latent graph structures and their node embeddings to circumvent the problem of not being able to construct ad hoc graph topologies for certain datasets. (3) We posit that the discriminative training approach of visual dialog model is limiting and suboptimal for a joint Vision-and-Language Model (VLM) and advocate for a fully generative training paradigm of visual and video dialog. To this end, we propose a unifying model that can simultaneously learn both tasks without any architectural modifications and *for the first time*, systemically study the effect of domain shift between both tasks using the prominent datasets of AVSD [9] and VisDial [48].

## 2.2 Dialog State Tracking (DST)

Previous works [145] have formulated the problem of dialog state tracking as a slot filling task [192]. For example, [140] proposed the Neural Belief Tracker (NBF) to detect slot-value pairs representing the user’s goal by iterating over all possible pairs. Other works [64, 187] suggested creating a state based on the textual and visual data for visual dialog and proposed a dialog-based image retrieval model that iteratively interacted with a user in natural language. Pang *et al.* [146] proposed an attention-based tracking of visual features to better generate questions for the “GuessWhat?!” dataset [169]. Le *et al.* [104] extended the slot filling dialog state paradigm [192] to autoregressively predict the visual attributes of scene objects in plain text. However, this work still suffers from further limitations. First, it was specifically designed for the DVD dataset [108]. Therefore, it cannot generalize to other datasets and tasks, thus, significantly limiting its applicability. Second, its training paradigm heavily relies on extra supervision labels such as bounding box coordinates extracted from a fine-tuned Mask-RCNN [71] model. Finally, its state tracking approach did not lead to significant performance improvements on

the down-stream video dialog task. As a result, these approaches remained predominately unimodal even though they were applied within a multimodal context (e.g., video dialog [139], visual dialog [146], image retrieval [64], recommendation systems [187]).

We instead propose to transition to multimodal DST to be able to deal with challenges imposed by current multimodal conversational datasets. Some works made initial steps towards this goal. For example, SIMMC [138, 97] was introduced to develop agents capable of helping a human in a shopping scenario and, therefore, need to track the multimodal state of the dialog to fulfill its task efficiently. However, these efforts primarily focused on synthetic and automatically generated datasets, limiting their applicability potential.

In stark contrast, we propose to fully transition to a multimodal DST framework with equal consideration of all available input modalities. Despite the high multimodality of the current dialog tasks, all previous works missed out on the idea of performing explicit multimodal DST. Instead, they focused on general vanilla attention methods that particularly tracked *only* one modality (mostly the visual input) at the expense of the others. In this thesis, we propose to close this gap by performing latent multimodal DST on each input modality separately. Specifically, we research a variety of DST approaches ranging from neuro-symbolic formulations to latent attention- and GNN-based attempts capable of equally dealing with synthetic and real-world multimodal datasets.

## 2.3 Neuro-Symbolic Approaches & Reasoning

Previous works proposed to investigate models' reasoning capabilities by introducing controllable and bias free datasets for VQA [85], VideoQA [196], visual dialog [99], and video dialog [108]. Neuro-symbolic models [197, 136, 165, 69, 196] achieved strong performance on these datasets and often outperformed connectionist approaches. For instance, on CLEVR [85], these models exhibited higher flexibility and efficiency compared to their connectionist counterparts [86, 75, 150, 78]. Specifically, NS-VQA [197] was one of the first neuro-symbolic models to achieve a near-perfect test accuracy on CLEVR despite being trained on a fraction of the available data. Subsequently, NS-CL [136] achieved the same results while being trained

with significantly less supervision. Other works demonstrated the effectiveness of the neuro-symbolic approach on monomodal conversational tasks [183, 171, 16]. For instance, [12] introduced a method that represents dialog states as a dataflow graph to better deal with coreferences that constitute one of the most challenging problems for dialog systems in general. However, most of these works missed out on the potential of applying the neuro-symbolic paradigm to model multimodal DST. Although several works focused on coreference resolution in videos [155, 158] and 3D data [96], CLEVR-Dialog [99] was one of the first fully annotated diagnostic datasets for multi-round visual reasoning that is well suited for a neuro-symbolic DST framework. Subsequent work [163] on this benchmark introduced models that build on the Memory, Attention, and Composition (MAC) network [78]. Although these models achieved promising results, they remain computationally inefficient and require huge amounts of training data, thus suffering from the same limitations of the connectionist approach. However, with the recent advances made by the transformer architecture [177], several recent studies [101, 24, 67, 51] have suggested that LLMs can deal equally well with the challenges posed by such tasks. In this thesis, we further explore the dichotomy between these two paradigms in the context of DST by proposing novel approaches for both classes of models.

## 2.4 Graph Structure Learning

Early works on graph structure learning leveraged bi-level programming [46] to simultaneously learn GNN parameters and topology [56]. Yu *et al.* [205] proposed applying the linear structure equation model in conjunction with a VAE [157] to learn directed acyclic graphs. Subsequently, Elinas *et al.* [54] suggested using a stochastic VI model to jointly estimate the graph posterior and the GNN parameters. Chen *et al.* [40] proposed iteratively refining the graph topology in an end-to-end manner using graph similarity metric learning. Wu *et al.* [186] suggested an all-pair message passing method to efficiently propagate signals between arbitrary nodes. Our work differs from the aforementioned approaches in three distinct aspects: (1) We propose a novel multimodal graph structure learning method that relies on a two-stage divide-and-conquer procedure that first predicts local modality-specific latent graphs before tackling the global graph consisting of the mix of all available modalities. (2) We use our graph

learning approach as a latent DST scheme and utilize it to enhance the hidden states of a backbone VLM. (3) Instead of dealing with unimodal graph-based tasks (node, edge, or graph classification), we investigate the effect of our method on the multimodal, non-graph-related downstream VL tasks.

## 2.5 Multimodal Expert-based Training

Enhancing models with Mixture-of-Experts (MoEs) training has shown promising potential in boosting performance while maintaining computational efficiency [53, 217, 213, 80]. Some works [20, 180] explored using single modality specific experts within a multimodal transformer architecture. Specifically, they used *one* vision and *one* language specific FFN after a shared multi-head self-attention block. Other works [123, 142] explored using multiple sparse modality-agnostic experts and trained them using soft-routers. Our work is positioned at the middle ground of the previously mentioned research directions: We propose to use multiple hard-routed experts per modality to be able to capture more fine-grained features compared to a single expert or multiple modality agnostic experts. Furthermore, to the best of our knowledge, we are the *first* to propose learning disentangled spatial and temporal features using two dedicated experts to be able to jointly learn from image and video data and thus unify visual and video dialog.



# 3

## Foundations

We dedicate this chapter to introducing basic concepts that will often be encountered during the course of the thesis. Specifically, we include the major problem formulations and the prominent learning paradigms linked to multimodal conversational tasks, which we address in each of the following chapters.

### 3.1 Dataset Structure

The multimodal conversational datasets we studied in this thesis share a common structure: Given some visual grounding  $\mathcal{V}$  in the form of an image or a video, and a dialog history  $\mathcal{H}_t$  at round  $t$  composed of multiple rounds of question-answer pairs, i.e.

$$\mathcal{H}_t = \{(Q_1, \mathcal{A}_1), \dots, (Q_{t-1}, \mathcal{A}_{t-1})\}, \quad (3.1)$$

a multimodal conversational model is tasked to predict an appropriate answer  $\mathcal{A}_t$  to the final dialog question  $Q_t$ . Some datasets complement the dialog history with a brief description of the visual grounding in

the form of a caption  $\mathcal{C}$ , i.e.

$$\mathcal{H}_t = \{\mathcal{C}, (\mathcal{Q}_1, \mathcal{A}_1), \dots, (\mathcal{Q}_{t-1}, \mathcal{A}_{t-1})\}. \quad (3.2)$$

However, these datasets differ in the way in which they were generated. As shown in [Figure 1.1](#), we can divide them into two categories: Whereas some are based on synthetic, automatically generated data, others use real-world, human-generated data (both for the vision and language modalities), increasing their complexity. Typically, the synthetic datasets use pre-defined rules and automated scripts to generate data at scale. On the other hand, their real-world counterparts rely on Amazon Mechanical Turks (AMTs) and human workers to construct the dialog data.

## 3.2 Training Paradigms

Despite their common dataset structure, multimodal conversational benchmarks follow different training paradigms to achieve the main objective of correctly answering the final dialog question.

### 3.2.1 Classification

Some benchmarks formulate the problem of answering the last dialog question as a classification task in which the model is trained to predict one answer from a fixed pool of candidate responses  $\Omega_{\mathcal{A}}$ . Typically, these models are transformer-based and train a classification head on a special [CLS] token using Cross-Entropy (CE). During inference, the model predicts the answer  $\hat{\mathcal{A}}_t$  at round  $t$  with the highest likelihood  $P$ , i.e.

$$\hat{\mathcal{A}}_t = \arg \max_{\mathcal{A} \in \Omega_{\mathcal{A}}} [\log P(\mathcal{A} | \mathcal{V}, \mathcal{H}_t, \mathcal{Q}_t)]. \quad (3.3)$$

### 3.2.2 Ranking

Other benchmarks follow an alternative approach and formulate the problem as a ranking task where the model is trained to re-rank a set of  $N$  candidate answers  $\Omega_{\mathcal{A}}^N$  that change for every round of each dialog. Specifically, these models use negative sampling to train a NSP head that predicts whether the sampled answer is correct or not using binary CE. During inference, the model is run  $N$  times on each candidate answer and the softmax NSP scores corresponding to the

correct class are ranked to predict the final answer, i.e.

$$\hat{\mathcal{A}}_t = \arg \max_{\mathcal{A} \in \Omega_{\mathcal{A}}^N} [\text{softmax}(P(\mathcal{V}, \mathcal{H}_t, \mathcal{Q}_t, \tilde{\mathcal{A}}))]. \quad (3.4)$$

This results in a considerable computational overhead compared to Equation 3.3 for which the model is inferred only once.

### 3.2.3 Generation

In contrast to the previous two formulations, open-ended benchmarks treat the problem as a generation task and train models to autoregressively predict the answer tokens using teacher forcing [184] and CE. During inference, the model generates one token at a time until a special [EOS] is predicted or a maximum length is reached, i.e.

$$\hat{a}_t^j = \arg \max_{a \in \Omega_{\mathcal{V}}} [\log P(a | \mathcal{V}, \mathcal{H}_t, \mathcal{Q}_t, \hat{\mathcal{A}}_t^{j-1})], \quad (3.5)$$

where  $\hat{\mathcal{A}}_t^{j-1} = [\hat{a}_t^1, \dots, \hat{a}_t^{j-1}]$  and  $\Omega_{\mathcal{V}}$  are the set of previously predicted tokens and the vocabulary, respectively.

### 3.2.4 Neuro-Symbolic

Finally, the neuro-symbolic paradigm is a hybrid approach between classification and generation. Contrary to the previous approaches, it does not follow an end-to-end training scheme. Specifically, it learns to generate programs (i.e. a function and its arguments) in a DSL as discussed in Equation 3.5. Then, a deterministic executor  $\text{Exec}$  runs the generated program  $\mathcal{P}_t$  on top of a structured representation of the visual grounding  $\tilde{\mathcal{V}}$  to predict the final answer, i.e.

$$\mathcal{A}_t = \text{Exec}(\tilde{\mathcal{V}}, \mathcal{P}_t). \quad (3.6)$$

## 3.3 Evaluation Metrics

We use the established and official evaluation metrics for each dataset in order to fairly compare our models with previous work. Whereas we use vanilla accuracy for answer **classification** tasks, we report sparse and dense scores for **ranking** formulations. Specifically, we compute the typical retrieval metric scores of Recall at k (R@k) for  $k \in \{1, 5, 10\}$ , Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG).











Dataset	Vision Modality			Language Modality		Formulation	Metrics	Chap.
	Medium	Base	Type	Caption	Type			
CLEVR-Dialog [99]		CLEVR [85]	Syn	✓	Syn	NS	Acc	4
Minecraft-Dialog [1]		Minecraft [185]	Syn	✓	Syn	NS	Acc	4
DVD [108]		CATER [61]	Syn	✗	Syn	Cls	Acc	5
SIMMC [138, 97]		–	Syn	✗	Syn	Gen	BLEU	5, 7
VisDial [48]		MS-COCO [37]	Real	✓	AMT	Rnk/Gen	NDCG, R@k, MRR	6, 8
VisPro [203]		VisDial [48]	Real	✓	AMT	Rnk	NDCG, R@k, MRR	6
VisDialConv [7]		VisDial [48]	Real	✓	AMT	Rnk	NDCG, R@k, MRR	6
AVSD [9]		Charades [166]	Real	✓	AMT	Gen	BLEU, ROUGE METEOR, CIDEr	7, 8

TABLE 3.1: Overview of the multimodal conversational datasets used in the course of this thesis. ( = video data,  = image data, Syn = Synthetic, NS = Neuro-Symbolic, Cls = Classification, Rnk = Ranking, Gen = Generation, Acc = Accuracy).

For **generation**, we use established Natural Language Generation (NLG) metrics such as Bilingual Evaluation Understudy (BLEU) [147], Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [124], Metric for Evaluation of Translation with Explicit Ordering (METEOR) [19], and Consensus-based Image Description Evaluation (CIDEr) [178]. Although **neuro-symbolic** approaches are a mix between classification and generation, we use accuracy for benchmarking. Specifically, we differentiate between three types of accuracies: (1) the program generation accuracy, (2) the executor accuracy, and (3) the question answer accuracy. That said, we use the latter as the main performance measure of our models.

We provide a comprehensive overview of all<sup>1</sup> the datasets we used in the course of this thesis in [Table 3.1](#).

---

<sup>1</sup>We only cover multimodal *conversational* datasets here.



# 4

## Neuro-Symbolic DST

*This chapter is based on the following publication and includes extracts thereof without explicit quotation:*

### **Neuro-Symbolic Visual Dialog**

Adnen Abdessaied, Mihai Băce, Andreas Bulling

In *Proceedings of the International Conference on Computational Linguistics (COLING)*. Gyeongju, Republic of Korea, 2022.

## 4.1 Introduction & Contributions

In this chapter, we present `NSVD`, the first method to combine deep learning and symbolic program execution for multimodal DST.

Existing connectionist models [98, 163] suffer from several limitations. First, they require large amounts of training data, which is prohibitive for most settings. Second, these models are not explainable, making it difficult to troubleshoot their logic when answering a given question. Finally, current models lack generalizability, in particular for dialogs that include incomplete or inaccurate histories, longer dialogs than those seen during training, or unseen question types. While neuro-symbolic hybrid models have proven effective as a more robust, explainable, and data-efficient alternative, e.g. for VQA [197, 136], VideoQA [196], or commonsense reasoning [16], they have not yet been explored for DST.

We fill this gap by proposing `NSVD` which combines three novel contributions to disentangle vision and language understanding from DST. First, it introduces two different program generators: a caption and a question program generator. The former induces a program from the caption to initialize the dynamic knowledge base of the executor at the beginning of each dialog. Second, a question program generator predicts a program in each round using not only the current question but also the dialog history. Third, a symbolic executor with a dynamic knowledge base keeps track of all mentioned entities.

Through extensive experiments on CLEVR-Dialog [99], we show that our models are significantly better at resolving coreferences and at maintaining performance over many rounds compared to fully connectionist models. Our results further suggest that `NSVD` is more data efficient, has a higher Normalized First Failure Round (NFFR), and generalizes better to Out-of-Distribution (OOD) scenarios.

## 4.2 Method

As shown in Figure 4.1, our method consists of *four* components: a scene understanding method, a program generator with caption and question encoders and a decoder, and finally a symbolic program executor with a dynamic knowledge base.

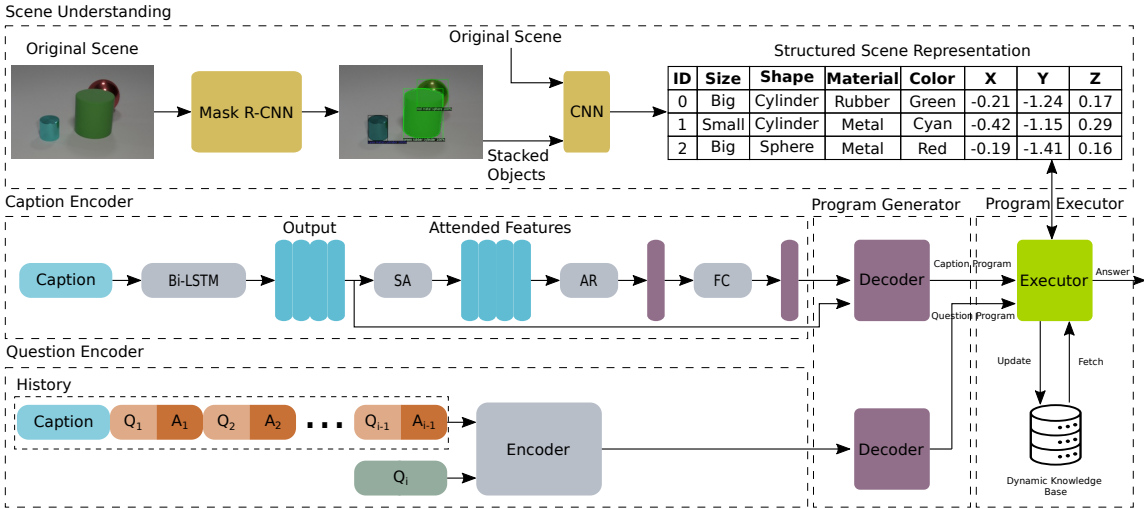


FIGURE 4.1: Overview of NSVD. First, we learn a structured representation of the visual scene. Then, we induce a caption program that can be run by our executor to initialize its knowledge base. In each subsequent round, we use the question and history to induce a program that answers the question and updates the dynamic knowledge base. (SA = Self-Attention, AR = Attention Reduction, FC = Fully Connected).

### 4.2.1 Scene Understanding

We use a Mask R-CNN [71] pre-trained on the CLEVR-mini dataset [197] to predict segmentation masks and attributes (color, shape, material, and size) for each entity in the visual scene. We then learn the 3D coordinates of each segment using a ResNet-34 model [72], resulting in a structured representation of the visual scene.

### 4.2.2 DSL for CLEVR-Dialog

Since CLEVR-Dialog implements its own grammar and vocabulary, we designed a novel DSL for it by implementing a collection of deterministic functions in Python that our symbolic executor can run over the structured CLEVR scenes. In previous works [86, 197, 136], these functional modules shared the same input/output interface and were arranged one after another to predict the answer. Instead, we followed a stricter approach by executing only one function that expects a *different* number of input arguments to answer a particular question. The full list of our functions, their arguments, and expected output can be found in [Appendix A.1](#).

### 4.2.3 Program Generation

Semantic parsing methods have been shown to be effective in mapping sentences to logical forms through a knowledge base or a program [66, 122, 171]. We adopt this approach and train a sequence-to-sequence model with an encoder-decoder structure to generate the programs.

#### Caption Encoder

The caption encoder first embeds the caption tokens to give  $\{\mathbf{w}_{\mathbf{c}_i}\}_{i=1}^{n_c}$  that are then fed into a bi-directional Long Short-Term Memory (LSTM) [73]. The self-attended outputs  $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_{n_c}]$  are reduced following:

$$\mathbf{a} = \text{softmax}(\text{MLP}(\mathbf{C})) \quad (4.1)$$

$$\bar{\mathbf{z}}_{\mathbf{c}} = \sum_{i=1}^{n_c} \alpha_i \mathbf{c}_i. \quad (4.2)$$

Finally, the latent vector  $\mathbf{z}_{\mathbf{c}} = \text{FC}(\bar{\mathbf{z}}_{\mathbf{c}})$  and the LSTM output  $\mathbf{C}$  are passed to the decoder.



### Question Encoders

To generate the question program in the current round  $t$ , we use not only the question  $\mathcal{Q}_t$  but also the history  $\mathcal{H}_t$  of previous question-answer pairs. We propose two different encoders based on how the question interacts with the history.

**Concat Encoder.** The concat encoder is similar in structure to the caption encoder. The tokens of the current question and history are embedded into a dense vector space to give  $\{\mathbf{w}_{\mathbf{q}_j}^{(t)}\}_{j=1}^{n_q}$  and  $\{\mathbf{w}_{\mathbf{h}_j}^{(t)}\}_{j=1}^{n_h}$ , respectively, which are then processed by two separate bi-directional LSTMs as shown in [Figure 4.2](#). The self-attended question and history features  $\bar{\mathbf{z}}_q$  and  $\bar{\mathbf{z}}_h$  are obtained similarly to  $\bar{\mathbf{z}}_c$ . Then,  $\bar{\mathbf{z}}_q$  and  $\bar{\mathbf{z}}_h$  are concatenated and linearly transformed to produce the question latent vector  $\mathbf{z}_q = \text{FC}([\bar{\mathbf{z}}_q, \bar{\mathbf{z}}_h])$ . Finally,  $\mathbf{z}_q$  and the question LSTM output  $\mathcal{Q}_t = [\mathbf{q}_1^t, \dots, \mathbf{q}_{n_q}^t]$  are passed to the decoder.

**Stack Encoder.** The approach of concatenating the question-answer pairs to form the history suffers from two main drawbacks. First, since the history is processed by an LSTM, its encoding becomes inefficient, in particular for later rounds, as the LSTM tends to forget crucial information that was mentioned in the first rounds. Second, this approach does not scale well for longer dialogs as it becomes computationally and memory demanding. To overcome these limitations, we introduce the stack encoder that separately encodes each question-answer pair in order to equally preserve the information from all previous rounds. The question and each round of the dialog history are embedded and then processed by separate bidirectional LSTMs ([Figure 4.2](#)). The last hidden states are used as feature representations of the question and previous rounds, i.e.

$$\mathbf{q} = [\overrightarrow{\mathbf{h}}_{\mathcal{Q}_t}, \overleftarrow{\mathbf{h}}_{\mathcal{Q}_t}] \text{ and } \mathbf{r}_{j < t} = [\overrightarrow{\mathbf{h}}_{\mathcal{H}_{j < t}}, \overleftarrow{\mathbf{h}}_{\mathcal{H}_{j < t}}], \quad (4.3)$$

where  $\overrightarrow{\mathbf{h}}_{(\cdot)}$  and  $\overleftarrow{\mathbf{h}}_{(\cdot)}$  are the bi-directional LSTM's last forward and backward hidden states, respectively.  $\bar{\mathbf{z}}_h$  is obtained by applying an inner-product attention between the question and history features:

$$\mathbf{a} = \text{softmax}(\mathbf{q}^T \mathbf{H}), \quad (4.4)$$

$$\mathbf{H} = [\mathbf{r}_0, \dots, \mathbf{r}_{t-1}]. \quad (4.5)$$

Finally,  $\mathbf{q}$  and  $\bar{\mathbf{z}}_{\mathbf{h}} = \sum_{j=1}^{t-1} \alpha_j \mathbf{r}_j$  are concatenated and linearly transformed to produce the latent question vector  $\mathbf{z}_{\mathbf{q}} = \text{FC}([\mathbf{q}, \bar{\mathbf{z}}_{\mathbf{h}}])$ . Similarly to the concat encoder,  $\mathbf{z}_{\mathbf{q}}$  and the question LSTM output  $\mathbf{Q} = [\mathbf{q}_1^t, \dots, \mathbf{q}_{n_q}^t]$  are passed to the decoder.

### Decoder

We use the same decoder architecture to generate all the caption and question programs. First, the ground truth program sequence  $\mathcal{Y}_t$  of the  $t$ -th dialog round is embedded to give  $\{\mathbf{w}_{\mathbf{y}_j}^{(t)}\}_{j=1}^{n_y}$  which are then processed by a simple LSTM whose hidden states are initialized by the encoder latent vector, i.e.  $\mathbf{z}_{\mathbf{c}}$  or  $\mathbf{z}_{\mathbf{q}}$ . The output  $\mathbf{P}$  of the LSTM is used with the encoder output, i.e.  $\mathbf{C}$  or  $\mathbf{Q}$ , to generate a context vector  $\Delta$  following:

$$\mathbf{A} = \text{softmax}(\mathbf{Q}^T \mathbf{P}), \quad (4.6)$$

$$\Delta = \mathbf{A}^T \mathbf{Q}. \quad (4.7)$$

Finally, the context vector  $\Delta$  is concatenated with the program output and the result is mapped to the program vocabulary dimension followed by a softmax function to obtain a distribution for the current program token  $y_j$ , i.e.

$$p(y_j | \mathcal{H}_t, \mathcal{Q}_t, \mathcal{Y}_t^{j-1}) \sim \text{softmax}(\text{FC}(\tanh([\mathbf{P}, \Delta]))), \quad (4.8)$$

where  $\mathcal{Y}_t^{j-1}$  is the sequence of previous ground truth program tokens. Our complete DSL grammar and qualitative samples can be found in [Appendix A.1](#) and [Appendix C.1](#), respectively.

### Executor

We equip the symbolic executor with a dynamic knowledge base that performs DST and keeps track of the entities previously mentioned in the dialog. It is initialized at the beginning of each dialog by executing the caption program. For instance, by executing the caption program

```
extreme-center(cylinder, small),
```

the executor searches for the center entity satisfying the function's arguments and stores it in the knowledge base under the handle

small-cylinder. We define two main operations of how the executor interacts with its knowledge base.

**Fetch.** The fetch operation is performed when executing a function requiring a coreference resolution. Given a set of attributes, the executor retrieves the appropriate entity from the knowledge base by searching the stored handles. For example

```
seek-attribute-early(color, cylinder)
```

first searches the previously stored handles and fetches the corresponding entity (in our example that is the cylinder mentioned in the caption with the handle `small-cylinder`) and then queries its color to answer the question.

**Update.** The update operation is performed after the execution of every question program. We differentiate between four update types:

1. *Handle update:* If a fetched entity is referenced by a new attribute, its handle in the knowledge base has to be updated accordingly. If the color of the previous cylinder is red, then its handle changes from `small-cylinder` to `small-cylinder-red`.
2. *Conversation subject update:* If the question program addresses a new entity, it becomes the new conversation subject. In our example, the conversation subject is still the small red cylinder. However, the question program

```
exist-obj-exclude-early(color, small, cylinder)
```

searches for other potential entities that share the same color as the previous small cylinder. If there is one, it becomes the new conversation subject.

3. *Seen entities update:* Each time a new entity is addressed, the executor saves it in its knowledge base together with the appropriate handle.
4. *Groups update:* Some questions refer to a group of entities, e.g. `count-attribute(red)` counts all red entities in the scene. These sets might be relevant for subsequent questions. For example,

```
count-attribute-group(large)
```

counts how many of the previous red entities are large.

## 4.3 Experiments

We modified CLEVR-Dialog [99] to generate datasets with ground truth caption and question programs required to train our program generators. Similarly to [99], we used the 70k training and 15k validation images of CLEVR [85] as our visual groundings. We left out the CLEVR test images because they lack ground truth scene annotations. For each image, we generated five dialogs each consisting of  $L = 10$  question-answer rounds as in [99]. We used 1k training images and their corresponding dialogs to create a validation set and tested our models and the baselines on the dialogs generated using the CLEVR validation images.

### 4.3.1 Performance Evaluation

Alongside the answer accuracy, the First Failure Round (FFR), i.e. the number of dialog rounds necessary for a model to make its first mistake, is commonly used to evaluate visual dialog models. Although popular, this metric has one major limitation, as it only allows us to compare the performance of models across datasets with the same dialog length but not across datasets with different ones. Thus, we propose the NFFR  $\in [0, 1]$  as an improvement and use it alongside the answer accuracy to assess the performance of all models. We refer to [Appendix A.2](#) for more details about its derivation.

### 4.3.2 History during Evaluation

One key limitation in the way visual dialog models are currently evaluated is the use of *ground truth* answers when calculating the correctness of an answer in any given round [163]. The problem of this approach is that it leads to overly-optimistic performance that do not reflect the true capabilities of the models in real-world scenarios: in real-world dialogs, full information on which previous answers were correct or not is typically not available. We instead propose to condition the generation of the current answer on all previous *predicted* answers. We call these evaluation schemes *Hist. + GT* and *Hist. + Pred*, respectively.

Model	Hist. + GT		Hist. + Pred	
	Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$
MAC-CQ [163]	97.34 $\dagger$	0.92	41.10	0.15
+ CAA	97.87 $\dagger$	0.94	89.39 $\dagger$	0.75
+ MTM	97.58 $\dagger$	0.92	70.39 $\dagger$	0.46
HCN [183]	75.88	0.34	74.42 $\dagger$	0.32
NSVD -concat	<u>99.59<math>\dagger</math></u>	<u>0.98</u>	<u>99.59<math>\dagger</math></u>	<u>0.98</u>
NSVD -stack	<b>99.72<math>\dagger</math></b>	<b>0.99</b>	<b>99.72<math>\dagger</math></b>	<b>0.99</b>

TABLE 4.1: Performance comparison on CLEVR-Dialog *test*.  $\dagger$  represents  $p < 0.01$  compared to the second best score in the respective column. Best and second best performances are in **bold** and underlined, respectively.

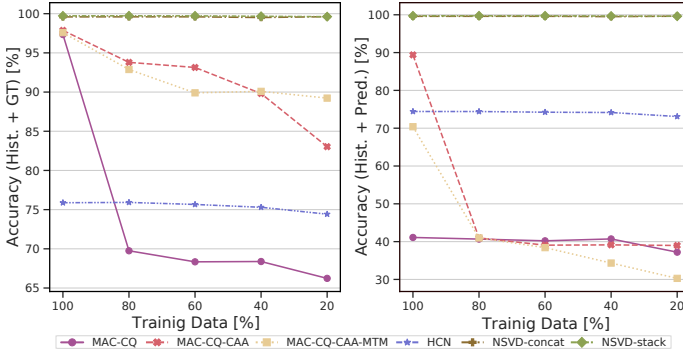


FIGURE 4.3: Accuracy of NSVD when trained on limited amounts of data (20%, 40%, 60%, 80%, and 100% of the overall dataset).

### 4.3.3 Visual Dialog Performance

After validating our implemented logic (see [Appendix A.3](#)), we compared the performance of our models with the visual dialog MAC networks [163]. We limited our comparison to their top three performing models, given that these outperformed the previous state of the art [98] by a 30% accuracy margin. Furthermore, we compared our models to the Hybrid Code Networks (HCN) [183] that also operate on symbolic dialog state representation but follow a different approach to parse programs than our generative one. They represent programs as templates in an action space and select the one with the highest probability during inference. This action space might become intractable if the DSL has many functions and arguments.

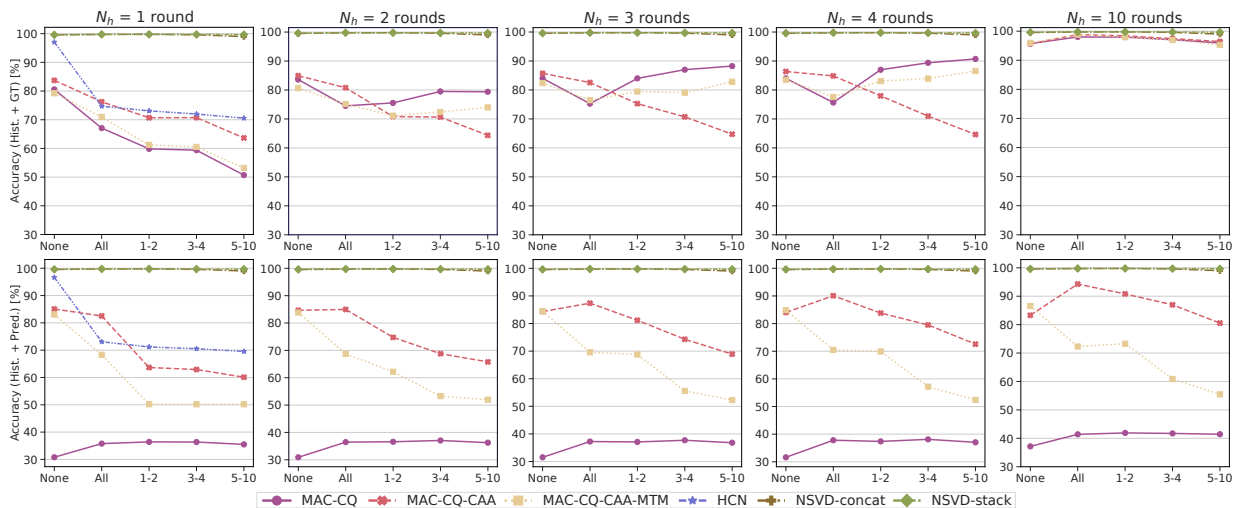


FIGURE 4.4: Robustness for different coreference distance bins and varying number of rounds in the history. All models were trained with full histories.

As shown in [Table 4.1](#), our models achieve a new state-of-the-art performance with *NSVD-stack* reaching an overall accuracy and a NFFR of 99.72% and 0.99, respectively. The high value of NFFR demonstrates our models' ability to answer correctly across all rounds of the dialogs with only a few failures in between.

While [Table 4.1](#) shows the results obtained when training on the entire dataset, our method achieves the same performance when trained on only 20% of the data, while the performance of other methods deteriorates significantly with fewer data, as shown in [Figure 4.3](#).

### 4.3.4 History Length vs Coreference Distance

CLEVR-Dialog provides coreference distances for each question, i.e. the number of rounds between the current and earlier mention of an entity in a question. A coreference distance of *one* means that the coreferent was mentioned in the previous question, while a coreference distance of 10 means that the question in round 10 refers to an entity in the caption. *All* and *None* mean that the question either depends on all previous rounds or is stand-alone, i.e. it does not depend on the history. To assess performance with respect to the coreference distance, we evaluated accuracy on different coreference distance bins. In our evaluations, we further limited the histories to the last  $N_h$  rounds to assess the robustness of the models to incomplete dialog histories.

As can be seen from [Figure 4.4](#), our models consistently achieve a performance of more than 99% across all coreference distance bins, independent of the evaluation scheme. Furthermore, their performance is only slightly affected by incomplete dialog histories. In contrast, performance of all baselines deteriorates quickly without access to the complete history. This deterioration is more visible when the *Hist. + Pred* evaluation scheme is used (second row of [Figure 4.4](#)).

However, their performance is consistent with the difficulty levels of the coreference distance bins, i.e. the accuracy decreases with increasing coreference distance. In contrast, this behavior is not reflected in the popular evaluation approach *Hist. + GT* (three middle plots of the first row of [Figure 4.4](#)). The most likely reason for this is that, as is currently common practice when evaluating visual dialog models, the ground truth answers of all previous rounds are used for prediction.

Model		$N_{\text{objects}} = 10$				$N_{\text{objects}} = 15$				$N_{\text{objects}} = 20$			
		Without FT		After FT		Without FT		After FT		Without FT		After FT	
		Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$
Hist. + GF	MAC-CQ [163]	38.52 $\ddagger$	0.14	53.49*	0.21	<b>36.87*</b>	<b>0.13</b>	48.89*	0.19	<b>36.12*</b>	<b>0.13</b>	47.44*	0.18
	+ CAA	37.72 $\ddagger$	0.14	53.35 $\ddagger$	0.21	36.82 $\ddagger$	0.13	48.67 $\ddagger$	0.18	35.52 $\ddagger$	0.13	47.43 $\ddagger$	0.18
	+ MTM	<b>38.59*</b>	<b>0.14</b>	52.62	0.21	36.22 $\ddagger$	0.13	47.41	0.17	36.01*	0.13	46.54	0.17
	HCN [183]	19.59	0.11	73.07 $\ddagger$	0.30	14.42	0.11	56.65 $\ddagger$	0.22	12.33	0.11	53.14 $\ddagger$	0.19
Hist. + Pred.	NSVD-concat	25.05*	0.12	99.32 $\ddagger$	0.97	18.51*	0.11	70.59 $\ddagger$	0.44	15.67 $\ddagger$	0.11	64.82 $\ddagger$	0.38
	NSVD-stack	24.95 $\ddagger$	0.12	<b>99.33*</b>	<b>0.98</b>	18.45 $\ddagger$	0.11	<b>70.62*</b>	<b>0.44</b>	15.65*	0.11	<b>64.95*</b>	<b>0.38</b>
	MAC-CQ [163]	<b>37.74*</b>	<b>0.14</b>	52.26 $\ddagger$	0.21	36.32*	0.12	47.58*	0.18	35.36 $\ddagger$	0.13	46.30 $\ddagger$	0.18
	+ CAA	37.70 $\ddagger$	0.13	51.36 $\ddagger$	0.21	<b>36.76*</b>	0.13	47.08 $\ddagger$	0.18	<b>35.57*</b>	<b>0.13</b>	45.56 $\ddagger$	0.17
Hist. + Pred.	+ MTM	36.29 $\ddagger$	0.14	50.58	0.20	35.62 $\ddagger$	<b>0.14</b>	45.67	0.17	33.98 $\ddagger$	0.13	44.02	0.17
	HCN [183]	19.50	0.11	71.55 $\ddagger$	0.29	14.40	0.11	55.55 $\ddagger$	0.21	12.26	0.11	51.95 $\ddagger$	0.19
	NSVD-concat	25.05*	0.12	99.32 $\ddagger$	0.97	18.51*	0.11	70.59 $\ddagger$	0.44	15.67 $\ddagger$	0.11	64.82 $\ddagger$	0.38
	NSVD-stack	24.95 $\ddagger$	0.12	<b>99.33*</b>	<b>0.98</b>	18.45 $\ddagger$	0.11	<b>70.62*</b>	<b>0.44</b>	15.65*	0.11	<b>64.95*</b>	<b>0.38</b>

TABLE 4.2: Results when training on simple scenes and testing on more complex ones.  $\ddagger$ ,  $\ddagger$ , and \* represent  $p < 0.01$ ,  $p < 0.05$  and  $p \geq 0.05$  compared to the second best score in each column, respectively.

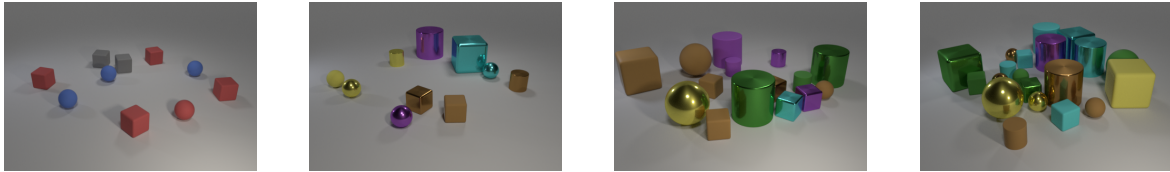


FIGURE 4.5: **From left to right:** Samples of a training image, fine-tuning image with 10 objects, fine-tuning image with 15 objects, and fine-tuning image with 20 objects.

### 4.3.5 Generalization to Unseen Scenes and Attributes

In previous experiments, our training and test sets had similar distributions in both the number of objects (between *three* and 10) and in their sizes, shapes, colors, and materials. To further test generalizability, we created a new training set consisting of 1.5k images, in which we restricted the type of objects to small, rubber cubes and spheres with the colors gray, red, or blue. We kept the number of objects in this dataset between *three* and 10. For testing, we generated three datasets consisting of 1k images each in which we allowed all classes of CLEVR objects (cubes, spheres, and cylinders) and materials (rubber and metal) to appear. However, we excluded the training colors and increased the number of objects  $N_{\text{objects}}$  in each one to 10, 15, and 20, respectively. Finally, we generated three fine-tuning datasets containing 1.5k images each similarly to the testing ones. [Figure 4.5](#) illustrates some examples of our new images. As in [\[99\]](#), all dialogs had a length of 10 rounds.

As we can see in [Table 4.2](#), the connectionist baselines outperform our neuro-symbolic models without fine-tuning in all scene complexities. This outcome is expected since these models rely on a Mask R-CNN to reconstruct the scenes. By increasing their complexities, i.e. more objects and attributes, the Mask R-CNN fails to accurately reconstruct these scenes which is reflected by the poor accuracies. However, after fine-tuning, our neuro-symbolic models perform best with NSVD -*stack* scoring 99.33%, 70.62%, and 64.95% on the test datasets with 10, 15, and 20 objects, respectively.

### 4.3.6 Generalization to Longer Dialogs

In order to assess the generalization capabilities of our models to longer dialogs, we used the testing images of the previous experiment to generate three dialog datasets with increasing numbers of rounds, i.e.  $L = 15, 20, 25,$  and  $30$ , respectively. That is, our test datasets for this experiment contain not only dialogs that are up to three times the length of the training dialogs but also visual scenarios never seen during training. Finally, we evaluated the best performing models of the previous experiment on this data *without* fine-tuning them again on longer dialogs. As shown in [Figure 4.6](#), our models generalize better across all datasets for both evaluation schemes. As expected, the performance of all models decreases with longer dialogs and more

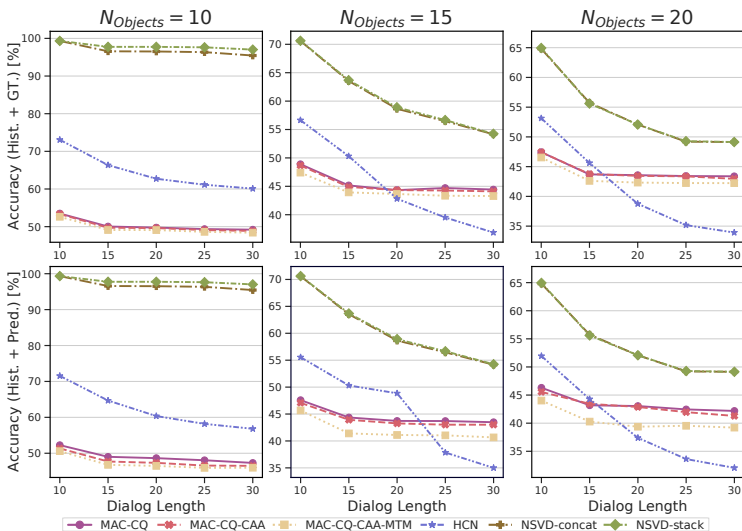


FIGURE 4.6: Answer accuracy for different dialog lengths and scene complexities. Our models generalize better to longer dialogs *without* the need for fine-tuning.

complex scenes. However, our models suffer less from this and still significantly outperform all baselines in all test scenarios.

### 4.3.7 Generalization to Unseen Questions Types

Contrarily to prior works [85, 197, 136] that merely addressed the generalizability to new scenes and object combinations, we further explore generalizability to unseen question types. To address this, we created two splits (AA and BB) based on CLEVR-Dialog as follows: First, we split the CLEVR validation images into two disjoint halves, A and B. We then split the question types into Split A and Split B. We randomly assigned half of the question types in each category to Split A and the other half to Split B to avoid biasing either one to a particular question category. For each image in both splits, we generated five dialogs consisting of 10 rounds as in [99]. Split AA contains a training set and a validation set based on 6k and 1.5k images, respectively. Split BB has a fine-tuning, a validation, and a test set based on 2k, 500, and 5k images, respectively. The desired

Model		Without FT		FT on split BB	
		Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$
Hist. + GT	MAC-CQ [163]	36.12 $\ddagger$	0.14	40.33 $\dagger$	0.16
	+ CAA	35.09 $\dagger$	0.14	40.36*	0.16
	+ MTM	34.73	0.15	35.09	0.13
	HCN [183]	47.67 $\ddagger$	0.14	70.43 $\ddagger$	0.27
	NSVD -concat	64.07 $\ddagger$	0.24	99.44 $\ddagger$	0.96
	NSVD -stack	<b>71.55<math>\ddagger</math></b>	<b>0.28</b>	<b>99.51<math>\dagger</math></b>	<b>0.97</b>
Hist. + Pred.	MAC-CQ [163]	35.09 $\ddagger$	0.13	39.53 $\ddagger$	0.15
	+ CAA	36.40 $\ddagger$	0.14	37.72 $\ddagger$	0.15
	+ MTM	6.19	0.09	7.03	0.09
	HCN [183]	46.91 $\ddagger$	0.13	68.82 $\ddagger$	0.25
	NSVD -concat	64.07 $\ddagger$	0.24	99.44 $\ddagger$	0.96
	NSVD -stack	<b>71.55<math>\ddagger</math></b>	<b>0.28</b>	<b>99.51<math>\dagger</math></b>	<b>0.97</b>

TABLE 4.3: Results when training on split AA and testing on split BB.  $\ddagger$ ,  $\dagger$ , and \* represent  $p < 0.01$ ,  $p < 0.05$  and  $p \geq 0.05$  compared to the second best score in each column, respectively.

behavior for a model that generalizes well is to perform well on split BB while only being trained on split AA.

As shown in Table 4.3, NSVD -concat and NSVD -stack achieve an accuracy of 64.07% and 71.55%, respectively, when tested on split BB without fine-tuning, thereby significantly outperforming all baselines. However, low NFFR values indicate that failures occur early in the dialogs. After fine-tuning all models on the fine-tuning dataset of Split BB, our models achieve accuracies and NFFR values comparable to previous experiments. In stark contrast, the baselines’ performance only improves by a small margin, with the highest jump of 5.26% being achieved by MAC-CQ-CAA. This shows an impressive data efficiency of our neuro-symbolic models that, in contrast to the data-hungry baselines, are able to learn and adapt efficiently from a very small amount of data.

### 4.3.8 Generalization to Other Scene Domains

In this experiment, we show that our method could be extended to other scene domains. To this end, we rendered new scenes in Minecraft that, as can be seen from Figure 4.7, are drastically different in terms of context and scene constellations from CLEVR. Specifically, they come with more entities (12 vs. 3 in CLEVR) that have different visual appearances.

Model	Hist. + GT		Hist. + Pred.	
	Acc.	NFFR $\uparrow$	Acc.	NFFR $\uparrow$
MAC-CQ [163]	64.30*	0.27	59.96 $\ddagger$	0.24
+ CAA	64.28 $\ddagger$	0.27	57.69 $\ddagger$	0.23
+ MTM	61.55 $\ddagger$	0.25	52.04 $\ddagger$	0.20
HCN [183]	47.31	0.14	46.50	0.14
NSVD-concat	91.57 $\ddagger$	0.76	91.57 $\ddagger$	0.76
NSVD-stack	92.46 $\ddagger$	0.83	92.46 $\ddagger$	0.83

TABLE 4.4: Performance comparison on Minecraft-Dialog.  $\ddagger$  and \* represents  $p < 0.01$  and  $p \geq 0.05$  compared to the second best score in the respective column, respectively.



FIGURE 4.7: Sample images from the Minecraft dataset.

Specifically, we rendered 10k images using [185]. The scenes consist of *three* to *six* objects in a 2D plane that are sampled from 12 entities with 4 different facing directions. Finally, we filtered out scenes that contain fully occluded objects. We used 5k images for training, 1.5k images for validation, and 1k for testing. Furthermore, we adapted the dialog generation tool [99] to be able to account for the different scene properties.

Similar to previous experiments, we generated five dialogs for every image consisting of 10 rounds each. We call this dataset *Minecraft-Dialog*. As shown in Table 4.4, we compared our models to the same baselines as in previous experiments in terms of accuracy and NFFR. Once again, our neuro-symbolic models managed to significantly outperform all baselines by achieving 92.64% and 91.57% accuracies for NSVD-stack and NSVD-concat, respectively, while maintaining high NFFR values. In contrast, the best baseline achieved test accuracies of 64.30% and 59.96% using the *Hist. + GT* and *Hist. + Pred* evaluation schemes, respectively. Compared to CLEVR-Dialog (Table 4.1), the performance of our models witnessed which could be attributed to the difficulty of the Minecraft scenes that come with occluded and diverse objects. However, the promising results of our models

underline their ability to generalize to new scene domains other than CLEVR. We refer to [Appendix C.1](#) for some qualitative samples of this new dataset.

## 4.4 Limitations & Closing Remarks

In this chapter, we proposed `NSVD`, the *first* model that leverages neuro-symbolic program execution to perform DST. Specifically, our model uses questions in natural language to generate programs in a DSL that can be run by an external executor. This executor is equipped with a dynamic knowledge base that keeps track of the entities mentioned in the dialog. We showed the effectiveness of our approach on a suitable diagnostic dataset, i.e. CLEVR-Dialog, and demonstrated its ability to generalize to OOD scenarios.

Despite its great potential, `NSVD` suffers from one key limitation. Due to its dependence on a DSL, our approach cannot straightforwardly tackle other datasets without a prior construction of a suitable DSL, which is not feasible for some datasets. Thus, we need to transition to more flexible approaches that learn differentiable latent DST representations.

# 5

## Attention-Based DST

*This chapter is based on the following publication and includes extracts thereof without explicit quotation:*

**OLViT: Multimodal State Tracking via Attention-Based Embeddings**

Adnen Abdessaied, Manuel von Hochmeister, Andreas Bulling  
In *Proceedings of the International Conference on Computational Linguistics (COLING)*. Turin, Italy, 2024.

## 5.1 Introduction & Contributions

In this chapter, we explore the idea of latent DST and present the Object Language Video Transformer (OLViT), a novel model for video dialog operating over a multimodal attention-based dialog state tracker. Existing models struggle with questions that require both spatial and temporal localization within videos, long-term temporal reasoning, and accurate object tracking across multiple dialog turns. OLViT alleviates these challenges by maintaining a global dialog state based on the output of an Object State Tracker (OST) and a Language State Tracker (LST): while the OST attends to the most important objects within the video, the LST keeps track of the most important linguistic references to previous dialog turns.

After each turn, both trackers compute continuous object and language state vectors that are used to update a global latent dialog state maintained over the course of the entire dialog. We study different ways of integrating the state vectors in an end-to-end manner within LLMs and demonstrate the applicability of our approach in both a discriminative setting, where the model predicts a full answer, and a generative one where response tokens are predicted one after another. Specifically, OLViT achieves new state-of-the-art performance on DVD [108], outperforming the previous best model by a **3.75%** margin. In addition, it outperforms strong baselines on the SIMMC 2.1 generative task [97] by reaching BLEU-4 scores of **28.30** and **25.20** on the *test-dev* and *test-std* splits, respectively.

## 5.2 Method

As shown in Figure 5.1, OLViT consists of six main components: An *Object Encoder* uses the Multi-Object Network (MONet) [27] to decompose the video frames into multiple masks that are used to compute the embeddings of scene objects. Complementing the object encoder, a *Text Encoder* uses a DistilRoBERTa model [159] to generate a textual embedding of the current question. The *OST and LST* take the object and language state vectors  $s_o^{(t-1)}$  and  $s_l^{(t-1)}$  from the previous dialog round ( $t - 1$ ) as input and generate updated state vectors  $s_o^{(t)}$  and  $s_l^{(t)}$ . A *Combiner* merges both state vectors with the special [CLS] token and the remaining multimodal features. Finally, a transformer *Encoder* block applies a sequence of self-attention and normalization operations on top of the combiner’s output and uses

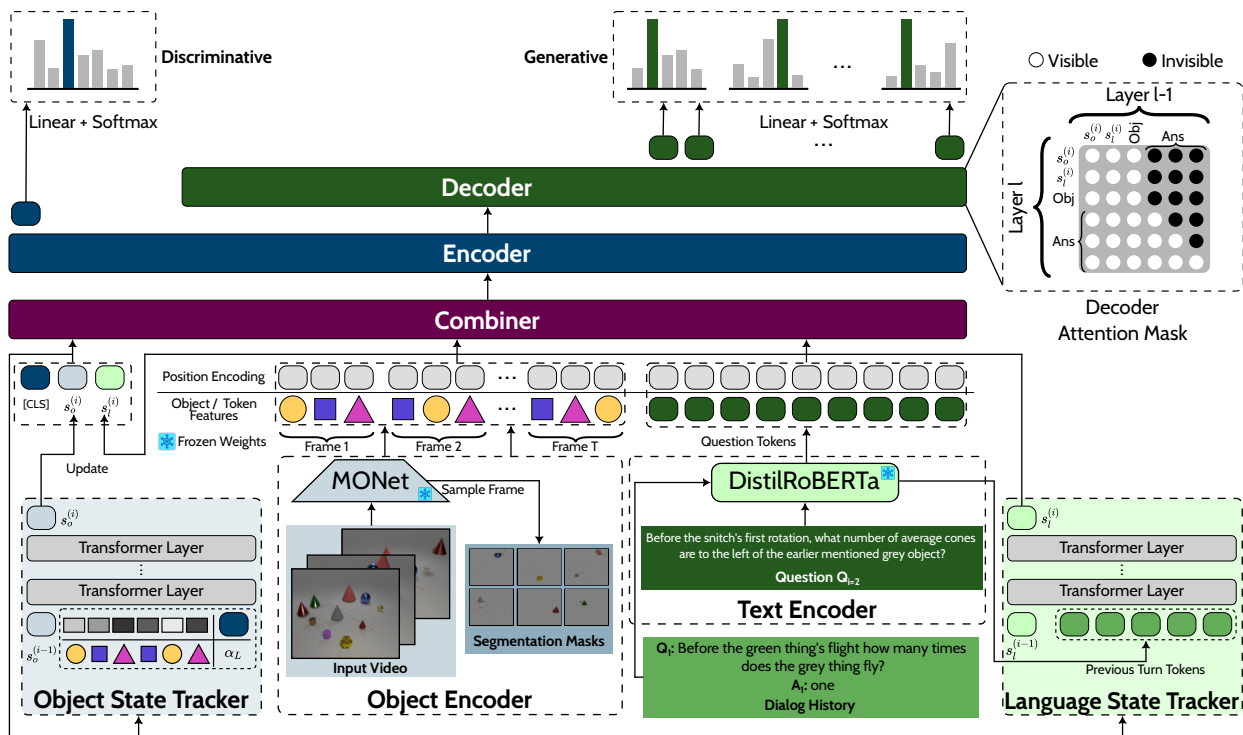


FIGURE 5.1: We use MONet and DistilRoBERTa-Base to generate the object and text embeddings, respectively. Then, we add position encoding and append the special [CLS] token to the multimodal features. Finally, we combine the object and language latent state vectors of the current  $t$ -th turn ( $s_o^{(t)}$  and  $s_l^{(t)}$ ) with the rest of the input, which will be processed by the subsequent transformer layers.

the [CLS] token to predict the correct answer in the discriminative setting. For the generative task, a *Decoder* block is added on top of the encoder and used to predict the answer tokens autoregressively.

### 5.2.1 Object Encoder

The encoder first samples  $F$  frames equidistantly from the video and uses a frozen MONet to obtain  $N_o$  object segmentation masks per frame. Some sample video frames and segmentation masks are illustrated in [Figure 5.1](#). The masks can be thought of as probabilities of each pixel belonging to a particular object and are encoded in latent variables with mean  $\mu_f^n \in \mathbb{R}^{d_{obj}}$ , where  $n$  indexes the object and  $f$  the video frame in which that object occurred. To obtain the initial object embeddings  $H_0^{obj} \in \mathbb{R}^{(FN_o) \times d}$ ,  $\{\mu_f^n\}$  are mapped to the same dimension  $d$  as the transformer block using a linear layer. Position embeddings are added to learn object-frame relationships:

$$H_0^{obj} = [W^{obj} \mu_1^1, \dots, W^{obj} \mu_1^{N_o}, \dots, W^{obj} \mu_T^{N_o}] + O_{pos}, \quad (5.1)$$

where  $W^{obj} \in \mathbb{R}^{d \times d_{obj}}$  and  $O_{pos} \in \mathbb{R}^{(FN_o) \times d}$  are a learnable parameter and the position embedding, respectively.

### 5.2.2 Text Encoder

The current question at turn  $t$  is first tokenized and the resulting  $N_w$  tokens are processed by a frozen pre-trained DistilRoBERTa model [159]. The  $d_w$ -dimensional embeddings from the last DistilRoBERTa layer are mapped to match the input dimension  $d$  of the subsequent transformer layers and used as the final word embeddings  $H_0^w \in \mathbb{R}^{N_w \times d}$ , that is:

$$H_0^w = [W^w h_1, \dots, W^w h_{N_w}] + W_{pos}, \quad (5.2)$$

where  $W^w \in \mathbb{R}^{d \times d_w}$  is a learnable parameter,  $h_j$  is the embedding of the  $j$ -th token and  $W_{pos} \in \mathbb{R}^{N_w \times d}$  is a learnable positional embedding.

### 5.2.3 Dialog State Tracking

The dialog state tracker is a key novelty of our method. Its purpose is to track relevant objects and dialog rounds within the video and over the course of the dialog, respectively. Specifically, this tracking

is learned in a latent space using two separate sub-components, i.e. OST and LST.

### Object State Tracker

As shown in the bottom left of **Figure 5.1**, the OST takes the latent object state vector  $s_o^{(t-1)}$  and the  $k$  most important object embeddings from the previous turn as input. It consists of  $L_{ost}$  transformer layers and uses Multi-Head Self-Attention (MSA) with Layer Normalization (LN) to output an updated latent object state vector  $s_o^{(t)}$ :

$$H_0^{ost} = [s_o^{(t-1)}, h_{L,1}^{obj}, \dots, h_{L,k}^{obj}], \quad (5.3)$$

$$h_{L,j}^{obj} = H_L^{obj}[j, :] \quad \forall j \in \text{top}_k(\alpha_L(h_L^{[CLS]}, H_L^{obj})), \quad (5.4)$$

$$H_l^{ost} = \text{MSA}(\text{LN}(H_{l-1}^{ost})) + H_{l-1}^{ost}, \quad (5.5)$$

$$s_o^{(t)} = H_{L_{ost}}^{ost}[0, :] \in \mathbb{R}^d, \quad (5.6)$$

where  $L$  is the number of subsequent transformer layers and

$$\alpha_L(h_L^{[CLS]}, H_L^{obj}) \in \mathbb{R}^{TN_o}$$

is the attention values between the final embedding of the [CLS] token and the object embeddings. As such, the updated latent object state vector  $s_o^{(t)}$  holds information about the most relevant objects for the current turn  $t$ . For the first turn, it is initialized with a zero vector, i.e.  $s_o^{(0)} = 0$ . For the generative setting where we do not use the [CLS] token, we summed the attention values of the object embeddings over all textual tokens in order to determine the most important objects for the given question, i.e.

$$h_{L,j}^{obj} = H_L^{obj}[j, :] \quad \forall j \in \text{top}_k(\alpha_L(h_L^{txt}, H_L^{obj})), \quad (5.7)$$

where  $h_L^{txt} = \sum_{k=1}^{N_w} h_k^w$  and  $\{h_k^w\}$  are the question token embeddings of the last decoder layer.

### Language State Tracker

As shown in the bottom right corner of **Figure 5.1**, the LST is a transformer-based module with  $L_{lst}$  layers that learns important information previously mentioned in the dialog history. The input of its first layer is the previous latent language state vector  $s_l^{(t-1)}$  and

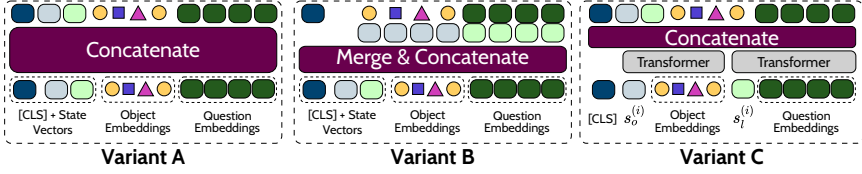


FIGURE 5.2: Overview of the different variants of our combiner.

the history token embeddings. To calculate the updated language state vector  $s_l^{(t)}$  for the current turn  $t$ , the LST has to be executed on the dialog history comprising of the previous  $(t - 1)$  turn embeddings. These embeddings are generated using the same frozen DistilRoBERTa model as before, that is:

$$H_0^{\text{lst}} = \left[ s_l^{(t-1)}, W_{\text{lst}}^w h_1^w, \dots, W_{\text{lst}}^w h_{n_{i-1}}^w \right], \quad (5.8)$$

$$H_l^{\text{lst}} = \text{MSA}(\text{LN}(H_{l-1}^{\text{lst}})) + H_{l-1}^{\text{lst}} \quad (5.9)$$

$$s_l^{(i)} = H_{L_{\text{lst}}}^{\text{lst}} [0, :] \in \mathbb{R}^d, \quad (5.10)$$

where  $W_{\text{lst}}^w$  is a learnable parameter,  $h_j^w$  is the  $j$ -th token embedding and  $n_i$  is the length of  $i$ -th dialog turn. For the first turn, the latent language state vector is initialized with a zero vector, i.e.  $s_l^{(0)} = 0$ .

### Combiner

The combiner is responsible for merging the  $[CLS]$  token embedding  $h_{[CLS]}$  and the two latent dialog state vectors with the rest of the multimodal input, i.e. the object and text embeddings. We propose three variants of how this merging is performed, as illustrated in [Figure 5.2](#).

**Variant A** simply concatenates all inputs to form the input of the current dialog turn, i.e.

$$I_A = \left[ h_{[CLS]}, s_o^{(t)}, s_l^{(t)}, H_0^{\text{obj}}, H_0^w \right]. \quad (5.11)$$

**Variant B** first appends  $s_o^{(t)}$  to the object embeddings and  $s_l^{(t)}$  to the text embeddings. Then, it maps them to the  $d$ -dimensional space

using a linear layer and concatenates all inputs like in Variant A, i.e.

$$I_B = [h^{[\text{CLS}]}, \tilde{H}_0^{\text{obj}}, \tilde{H}_0^w], \quad (5.12)$$

$$\tilde{H}_0^{\text{obj}} = [W^b[s_o^{(t)}; h_1^{\text{obj}}], \dots, W^b[s_o^{(t)}; h_{T_{N_o}}^{\text{obj}}]], \quad (5.13)$$

$$\tilde{H}_0^w = [W^b[s_l^{(t)}; h_1], \dots, W^b[s_l^{(t)}; h_{N_w}]], \quad (5.14)$$

where  $W^b \in \mathbb{R}^{d \times (2d)}$  is a learnable parameter.

**Variant C** uses two small transformers operating on the concatenation of the object state vector with the object embeddings and the language state vector with the text embeddings. Then, it concatenates the outputs of their final layers ( $H^{\text{obj}}$  and  $H^w$ ) with the [CLS] token embeddings and the state vectors, i.e.

$$I_C = [h^{[\text{CLS}]}, s_o^{(t)}, s_l^{(t)}, H^{\text{obj}}, H^w]. \quad (5.15)$$

## 5.2.4 Encoder/Decoder

The last component of our model, the encoder, consists of  $L$  transformer layers and takes the output of the combiner as input. The output of its final layer is used to either predict or generate the answer of the current question.

**Prediction.** We use a linear layer with softmax to map the final  $d$ -dimensional [CLS] token embeddings to the  $N$ -dimensional answer space. We train our model end-to-end using cross-entropy loss. During testing, we choose the answer  $\hat{a}$  with the highest likelihood score as described in [Chapter 3.2.1](#).

**Generation.** For answer generation, we couple the encoder with a decoder that has the same number of transformer layers and attention heads and append the ground truth answer to the question. We then train the model end-to-end as described in [Chapter 3.2.3](#) while applying a generative mask to the input as shown in [Figure 5.1](#).

## 5.3 Experiments

### 5.3.1 Datasets & Metrics

To evaluate the performance of our model for both discriminative and generative scenarios, we used the DVD [\[108\]](#) and SIMMC 2.1 [\[97\]](#)

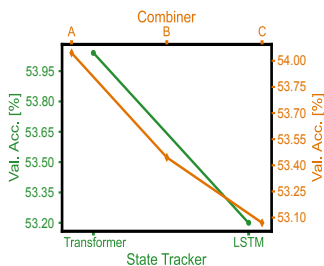


FIGURE 5.3: Performance comparison of OLViT with different combiners and state tracker variants.



FIGURE 5.4: Performance comparison of OLViT with different numbers of objects and history turns.

benchmarks. DVD was recently proposed with the goal of assessing higher-order spatio-temporal reasoning capabilities of video dialog models. It is based on 11k videos from the challenging CATER dataset [61] and contains over 100k dialogs and 1M question-answer pairs with detailed spatio-temporal annotations. SIMMC 2.1 is a task-oriented dataset that was proposed for realistic virtual assistance scenarios. It contains 11k dialogs from the shopping domain. As discussed in Table 3.1, we use accuracy and BLEU-4 to assess the performance of our model on DVD and SIMMC 2.1, respectively.

### 5.3.2 Combiner

As can be seen from Figure 5.3, Variant A of the combiner, despite having the simplest architecture, outperformed all other variants with a validation accuracy of 54.01% on DVD. We hypothesize that this is because concatenating the latent state vectors to all respective embeddings (Variant B) prevents the model from applying attention over the raw state vectors that contain rich information about the previous relevant objects and dialog turns. On the other hand, using additional transformer layers increases complexity and the risk of overfitting (Variant C). We also experimented with an LSTM-based state tracker which reached a validation accuracy of 53.20% compared to 54.01% achieved by the transformer-based one.

### 5.3.3 Number of Objects and History Turns

To find the best hyperparameters (number of the most important previous objects and history turns) of the state tracker (**Variant A**),

we first used the full history, i.e. all previous turns, and increasingly varied the number of objects. We considered these two variables independently to keep the size of the search space tractable. Once the optimal number of objects was found in the OST, we optimized the number of previous turns in the LST. As can be seen from [Figure 5.4](#), we increased the number of object embeddings in each step (i.e. None, 1, 2, 3, and 4) while keeping the entire dialog history. `OLViT` achieved its peak validation accuracy of 55.10% when using two object embeddings per round. With more embeddings, performance started to decrease and reached 54.86% and 54.72% with three and four object embeddings, respectively. We then fixed the number of object embeddings to two and varied the history length. The best validation accuracy of 55.39% was reached when we used a history of *seven* previous rounds. These optimal values were then fixed for the rest of the experiments unless it is explicitly stated otherwise.

### 5.3.4 Baselines

We compared `OLViT` against the baselines of [\[108\]](#) and other strong models that (1) were recently published for video dialog state tracking, i.e. `VDTN` [\[104\]](#), and (2) maintain the current state-of-the-art performance on both DVD and SIMMC 2.1 datasets, i.e. `MTN` [\[106\]](#) and the GPT-2 [\[154\]](#) based MM-DST model [\[138\]](#).

### 5.3.5 Results on DVD

#### Quantitative Analysis

We first evaluated `OLViT` on the discriminative video dialog task. As can be seen from [Table 5.1](#), our model reached an overall accuracy of 54.85%, thereby outperforming the state-of-the-art model by 3.75% absolute points. Not only did our model reach a new state-of-the-art test accuracy, it did so by improving the performance across *all* question categories. This is in stark contrast to previous methods (e.g. `MTN` [\[106\]](#) and more importantly the recent `VDTN` model [\[104\]](#)) which typically only improved performance for a subset of categories. Moreover, our method performed particularly well on challenging categories that require accurate object tracking and that existing models tend to struggle with. Specifically, `OLViT` improved the performance of *action count*, *action query* and *attribute query* by 4.62%, 5.84%, and 7.67% over the state of art, respectively. Furthermore, our model

Accuracy [%]	Answer	Q-type	Q-type	Q-retrieval	RNN	HRNN	HRNN	VDTN	MTN	OLViT	$\Delta$ [%]
	Prior	(Random)	(Freq)		(Q)	(Q + H)	(Q + H + V)				
<b>Action count</b>	0.0	9.30	23.40	19.80	16.30	28.20	36.00	38.78	38.80	<b>43.42</b>	<b>+4.62</b>
<b>Action query</b>	0.0	12.70	23.70	20.60	25.80	33.10	38.60	39.37	<u>39.40</u>	<b>45.24</b>	<b>+5.84</b>
<b>Attribute query</b>	0.0	32.90	38.70	39.40	38.10	39.20	<u>45.10</u>	42.93	43.10	<b>52.77</b>	<b>+7.67</b>
<b>Compare action seq</b>	33.40	34.10	37.30	35.10	45.50	52.50	57.50	61.57	<u>61.60</u>	<b>61.88</b>	<b>+0.28</b>
<b>Compare action set</b>	25.10	28.20	36.30	28.20	32.80	40.00	44.30	<u>45.41</u>	45.40	<b>51.71</b>	<b>+6.30</b>
<b>Compare action freq</b>	48.50	50.00	50.50	44.40	58.40	56.90	65.20	66.42	<u>67.10</u>	<b>67.91</b>	<b>+0.81</b>
<b>Object count</b>	0.0	9.10	23.30	18.80	26.20	38.60	<u>40.20</u>	39.86	39.90	<b>43.34</b>	<b>+3.14</b>
<b>Object exist</b>	48.90	49.80	51.10	54.40	66.40	67.00	69.40	<u>69.42</u>	69.00	<b>71.25</b>	<b>+1.83</b>
<b>None</b>	0.0	32.10	38.30	39.00	38.30	39.50	<u>45.10</u>	43.51	43.40	<b>52.74</b>	<b>+7.64</b>
<b>Atomic (non-spatial)</b>	18.80	26.30	31.90	42.40	47.20	47.80	<u>50.70</u>	48.88	48.90	<b>56.54</b>	<b>+5.84</b>
<b>Atomic (spatial)</b>	21.20	27.30	35.50	27.60	36.80	46.00	<u>47.60</u>	47.12	47.10	<b>49.61</b>	<b>+2.01</b>
<b>Compositional</b>	22.80	28.00	35.40	32.10	40.00	45.80	51.40	53.18	<u>53.20</u>	<b>56.70</b>	<b>+3.50</b>
<b>Transfer (attribute)</b>	0.0	30.70	45.50	37.10	40.80	45.70	57.30	<u>57.70</u>	<u>57.70</u>	<b>61.28</b>	<b>+3.58</b>
<b>Transfer (spatial)</b>	49.80	42.40	44.90	26.40	29.60	<u>48.10</u>	47.40	47.86	48.00	<b>50.50</b>	<b>+2.40</b>
<b>Transfer (temporal)</b>	28.90	38.40	22.60	3.00	30.20	53.50	64.60	68.72	69.00	<b>74.83</b>	<b>+5.83</b>
<b>All</b>	21.30	27.80	35.30	32.10	39.70	45.80	50.20	51.02	<u>51.10</u>	<b>54.85</b>	<b>+3.75</b>

TABLE 5.1: Performance comparison on DVD. Best and second best performances are in **bold** and underlined, respectively.  $Q$ ,  $C$ , and  $V$  denote question, history/context, and visual input, respectively, and † denotes training with additional supervision.

	test-dev			test-std		
	MM-DST	MTN	OLViT	MM-DST	MTN	OLViT
<b>BLEU-4</b>	19.20	<u>21.70</u>	<b>28.30</b>	19.20	<u>21.00</u>	<b>25.20</b>
Rel. $\Delta$ [%]	<b>+47.4</b>	<b>+30.4</b>	–	<b>+31.3</b>	<b>+20.0</b>	–

TABLE 5.2: Performance comparison on SIMMC 2.1. Best and second best performances are in **bold** and underlined, respectively.

outperformed the state of the art across all *transfer* categories (penultimate section of Table 5.1) underlining the efficiency of the proposed DST mechanism in improving its spatio-temporal reasoning capabilities. Specifically, it reached accuracies of 61.28%, 50.50%, and 74.83% on the *transfer (attribute)*, *transfer (spatial)*, and *transfer (temporal)* categories, respectively.

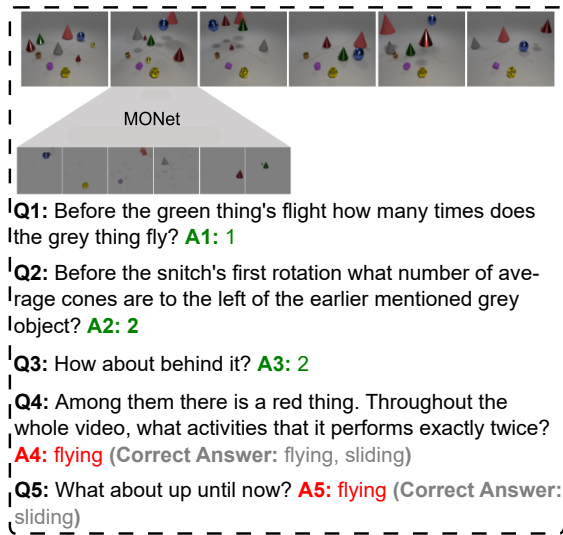
### Qualitative Analysis

In Figure 5.5a, we show sample predictions of OLViT on the DVD test split together with some video frames and MONet object masks. Although our model correctly answered the first three questions, it failed in the last two. Specifically, when asked about the actions the red metal cone performs, it predicted “*flying*” instead of “*flying, sliding*”. This might be attributed to the difficulty of the CATER videos: deciding between “*flying*” and “*sliding*” requires the model to reason about the object’s shadow, which is hard to acquire from the object embeddings.

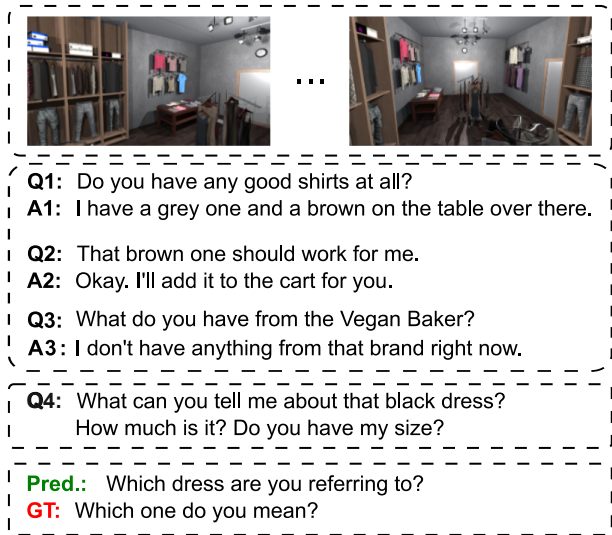
## 5.3.6 Results on SIMMC 2.1

### Quantitative Analysis

For the generative setting, we evaluated OLViT on the SIMMC 2.1 dataset that comes with two test splits (*test-dev* and *test-std*). The ground-truth answers of the former are publicly available, but those of the latter are withheld by the creators of the dataset. As can be seen in Table 5.2, OLViT outperforms strong baselines on the *test-dev* split by reaching a BLEU-4 score of 28.30 compared to 19.20 and 21.70 achieved by MM-DST and MTN, respectively. Table 5.2 also shows that our model outperforms these baselines by a considerable margin by achieving a BLEU-4 score of 25.20 corresponding to relative improvements of 31.3% and 20.0% over MM-DST and MTN, respectively.



(A) Answer classification on a random DVD test sample.



(B) Answer generation on a random SIMMC 2.1 test.

FIGURE 5.5: Qualitative samples of OLViT.

Ablated $\text{OLViT}$				State	
DVD		SIMMC 2.1		Trackers	
Acc. [%]	$\Delta$ [%]	BLEU-4	Rel. $\Delta$ [%]	OST	LST
44.11	11.28	21.50	22.79	$\times$	$\times$
46.52 <sup>‡</sup>	8.87	21.70 <sup>†</sup>	21.66	$\checkmark$	$\times$
54.94 <sup>‡</sup>	0.45	26.30 <sup>‡</sup>	0.38	$\times$	$\checkmark$
<b>55.39<sup>†</sup></b>	–	<b>26.40<sup>†</sup></b>	–	$\checkmark$	$\checkmark$

TABLE 5.3: Performance comparison of different ablations of  $\text{OLViT}$  on the DVD and SIMMC 2.1 *val* splits.  $\dagger$  and  $\ddagger$  represent significant improvement with  $p < 0.05$  and  $p < 0.01$  compared to the second best score.

### Qualitative Analysis

Figure 5.5b shows the generated answer of our model on a randomly sampled example from the SIMMC 2.1 *test-dev* split. Although there is not a big overlap in the used words between our model’s prediction (green) and the ground-truth (red), the two responses are semantically similar and lead to the same action, i.e. ask for more information from the user.

### 5.3.7 Ablation Study

**Performance without OST & LST.** We first evaluated a version of our model without any state trackers (first row in Table 5.3). It is important to note that in this case the model still has access to the MONet object embeddings but not to the previous history turns. This ablated version performed poorly, only reaching 44.11% validation accuracy on DVD and 21.50 BLEU-4 score on SIMMC.

**Performance with the OST.** Adding the OST (second row in Table 5.3) resulted in a notable performance improvement of our model by reaching an accuracy of 46.52% on DVD and a BLEU-4 score of 21.70 on SIMMC. We emphasize that this version still does not have access to the previous history turns and treats the task as a simple single-round question answering task. This finding underlines the positive influence of performing latent state tracking of the object embeddings by the OST which helps the model focus more visual cues relevant to the question at hand.

**Performance with the LST.** Adding the LST (third row in Table 5.3) lifted the performance on both DVD and SIMMC 2.1 by a considerable

LLM	# of LLM param.	# of Layers L	Accuracy [%]
None	-	4	54.46
None	-	16	54.54
DistillRoBERTa	81.5M	4	55.39
RoBERTa	354M	<b>4</b>	<b>55.41</b>

TABLE 5.4: The effect of pre-trained language models on the performance of our model. Performance is measured on the DVD *val* split.

margin compared to the version with only the OST. We note here that this version, in stark contrast to the previous one, not only had access to the visual input in the form of MONet embeddings but also to the previous dialog turns. This variant reached a validation accuracy of 54.94% and a BLEU-4 score of 26.30. However, it still underperformed our full model with both dialog trackers in action (last row in Table 5.3) underlining the complementary benefits of performing latent multimodal DST on each input modality separately.

**Performance with Pre-trained LLM.** To assess the effect of pre-trained LLM on OLViT, we completely removed the DistillRoBERTa model and trained the encoder layers from scratch on the DVD dataset. Thereby, we increased the number of the encoder transformer layers from *four* to 16 to compensate for the removal of the pre-trained LLM. As can be seen from Table 5.4, our model’s DVD validation accuracy dropped from 55.39% to 54.46% when DistillRoBERTa was replaced by *four* encoder layers. However, its performance improved when we increased the number of these layers to 16 and reached a validation accuracy of 54.54%. Finally, using a full RoBERTa model only improved the model’s accuracy by 0.02%. Thus, we decided to use a DistillRoBERTa model in the previous experiments to be able to train efficiently. We refer to Appendix A.4 for further details about the design choices of our model.

## 5.4 Limitations & Closing Remarks

In this chapter, we proposed OLViT, the first model equipped with latent multimodal DST. Specifically, our model uses two attention-based modules, i.e. the OST and LST, to keep track of the most relevant objects in the visual scene and linguistic information in the

dialog. Due to its differentiable nature, our DST approach can be seamlessly integrated within LLMs to improve their capabilities on different multimodal dialog tasks. Through extensive evaluations on two challenging benchmarks, we showed that OLViT manages to reach new state-of-the-art results on the discriminative and generative tasks of DVD [108] and SIMMC [97], respectively.

That said, one of the major limitations of this work is that the proposed approach was only beneficial for synthetic and diagnostic datasets that do not adequately reflect the complexity of real-world data. Addressing more challenging realistic datasets necessitates the transition to more sophisticated latent and differentiable DST approaches which we discuss in the next chapters.



# 6

## **GNN-Based DST with Static Graph Topologies**

*This chapter is based on the following publication and includes extracts thereof without explicit quotation:*

**VD-GR: Boosting Visual Dialog with Cascaded Spatial-Temporal Multimodal Graphs**

Adnen Abdessaied, Lei Shi, Andreas Bulling

*In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Hawaii, USA, 2024.

## 6.1 Introduction & Contributions

In this chapter, we present a more advanced multimodal latent DST approach capable of addressing the challenges of real-world datasets and propose **VDGR** – a novel visual dialog model that combines pre-trained LLMs with GNNs. Previous works mainly focused on one class of models at the expense of the other, thus missing out on the opportunity of combining their respective benefits. At the core of **VDGR** is a novel integration mechanism that alternates between spatial-temporal multimodal GNNs and transformer layers covering three distinct contributions: First, we use multimodal GNNs to perform latent DST by processing the features of each modality (image, question, and dialog history) and exploiting their local structures before performing transformer global attention. We posit that it is essential to exploit both local intra-modal structures and the global inter-modal context: Each modality is composed of smaller entities whose relationships have to be separately captured and tracked by the model (i.e. the objects in the image, the words that constitute the question, and the rounds that form the history). At the same time, the inter-dependency of the modalities and the global context are equally important for a model to be able to answer the current question efficiently. Second, we propose hub-nodes that link to all other nodes within one modality graph, allowing the model to propagate information from one GNN (modality) to the other in a cascaded manner. Third, we augment the LLM hidden states with fine-grained multimodal latent DST features obtained from the respective GNN before passing them to the next **VDGR** layer. Through extensive evaluations on *four* challenging benchmarks, we show that **VDGR** outperforms strong baselines and achieves new state-of-the-art results across all datasets.

## 6.2 Method

As shown in [Figure 6.1](#), **VDGR** consists of *four* main components: a) a graph construction and pruning module, (b) a backbone VLM operating on alternating cascaded multimodal graphs and BERT layers, (c) multiple loss heads (NSP, Masked Language Modeling (MLM), Masked Region Modeling (MRM), Graph Edge Masking (GEM)), and (d) a prediction module to **rank** a set of 100 candidate answers as discussed in [Chapter 3.2.2](#).

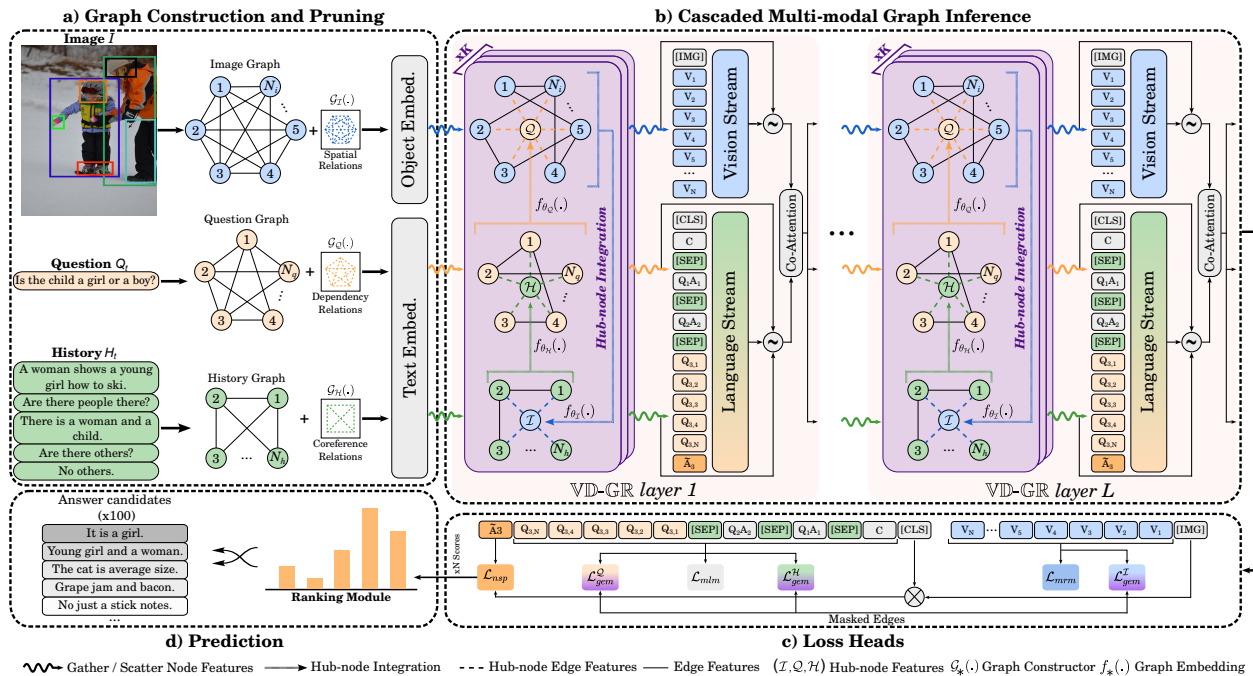


FIGURE 6.1: VDGR takes an image, a question, and a history as input and constructs their corresponding graph structures. Each of its layers is composed of cascaded multimodal graphs that learn latent DST representations that augment the hidden states of the subsequent LLM layer.

## 6.2.1 Graph Construction and Pruning

### Image Modality

The image graph constructor  $\mathcal{G}_I(\cdot)$  treats each object in the image as a node and relies on spatial relationships to construct the graph topology, i.e. the adjacency matrix, where each edge represents a relationship between two objects in the image. The objects features

$$\mathbf{I} = \{\mathbf{v}_1, \dots, \mathbf{v}_{N_i}\} \quad (6.1)$$

are obtained using Faster R-CNN [156] pre-trained on Visual Genome [100] where each object feature  $\mathbf{v}_i$  is a 2048 dimensional vector and  $N_i = 36$ . Similar to [195], we distinguish between 11 spatial relations based on the  $(x_1, y_1, x_2, y_2)$  coordinates of the object’s bounding box.

### Question Modality

The question graph constructor  $\mathcal{G}_Q(\cdot)$  treats each word in the question as a node and relies on dependency relations to construct the graph topology, where each edge represents a relationship between two words in the question. To extract these relationships, we use the neural dependency parser of Stanza [153] that yields 47 relations in total.

### History Modality

The history graph constructor  $\mathcal{G}_H(\cdot)$  treats each dialog round as a node and relies on coreference relations to construct the graph topology, i.e. the adjacency matrix, where each edge represents a relationship between two rounds in the history.

As can be seen in Figure 6.1, the topology of all multi-modal graphs is computed *once* during a pre-processing stage and is always kept constant over time. We refer to Appendix A.5 for additional details about the graph construction process.

## 6.2.2 VDGR Layer

### LLM Features

In line with previous work, we use VisDial-BERT [141] layers within our novel GNN-enhanced DST approach. For a given image-question pair at round  $t$ , we first concatenate the caption  $\mathcal{C}$ , the previous dialog rounds  $\mathcal{H}_t$ , the current question  $\mathcal{Q}_t$ , and a candidate answer  $\hat{\mathcal{A}}_t$  to

form the textual input

$$\mathbf{T} = \{[\text{CLS}] \mathcal{C} [\text{SEP}] \mathcal{Q}_1 [\text{SEP}] \mathcal{A}_1, \dots, \mathcal{Q}_t [\text{SEP}] \tilde{\mathcal{A}}_t\}, \quad (6.2)$$

where [CLS] and [SEP] are the special classification and separation tokens, respectively.

As in [141, 181, 202, 32], we use a special learnable token [IMG], and initialize it using mean pooling of the object features  $\mathbf{I}$ . The hidden features of the [IMG] token are used in conjunction with those of [CLS] by means of element-wise multiplication to produce the final input features of the NSP head. Finally, the two sequences  $\mathbf{T}$  and  $\mathbf{I}$  are used as initial inputs for the language and vision streams to obtain the hidden states of the  $l$ -th VDGR layer, i.e.  $\mathbf{T}_h^{(l)}$  and  $\mathbf{I}_h^{(l)}$ .

### GNN Features

**Node Features.** As illustrated in Figure 6.1, the multimodal GNNs of the  $l$ -th VDGR layer get their node features from the hidden states of the previous VisDial-BERT layer (or from the embedding layers in the first step). It is worth noting that the special tokens [IMG] and [CLS] are not included in the graph features. The nodes of the image and question graphs are gathered from the image and question embeddings of the  $(l-1)$ -th layer, i.e.  $\{\mathbf{I}_{h,i}^{(l-1)}\}_{i=1}^{N_i}$  and  $\{\mathbf{T}_{h,i}^{(l-1)}\}_{i=s_q}^{s_q+N_q}$ , where  $s_q$  and  $N_q$  denote the index of the first question token and the length of the question, respectively.

We use the special [SEP] tokens to represent each dialog round in the history and gather their hidden states from  $\mathbf{T}_h^{(l-1)}$  to get the node features of the history graph. The node features of the question and history graphs have to be extracted carefully since the textual input, i.e. where the question starts and ends and where the [SEP] tokens are located, varies within the features  $\mathbf{T}_h^{(l-1)}$  for each dialog round.

**Hub-node Features.** As illustrated in Figure 6.1, we introduce hub-nodes to make each modality aware of the other in a cascaded manner. As a result, the history becomes aware of the image, the question becomes aware of the history, and finally the image becomes aware of the question. The hub-node within each graph links to all other nodes using a special edge feature to propagate information on a *local scale* from one modality to the other. To obtain these features, we train attention-based graph embeddings for each modality, i.e.  $f_{\theta_x}(\cdot)$ ,

$f_{\theta_{\mathcal{Q}}}(\cdot)$ , and  $f_{\theta_{\mathcal{H}}}(\cdot)$ , which take the node features of the corresponding graph as input and output a single vector representing it. For example, the image hub-node  $\mathcal{I}$ , is computed from the image graph node features  $\mathbf{I}_G$  of the  $l$ -th layer as follows:

$$\mathcal{I} = f_{\theta_{\mathcal{I}}}(\mathbf{I}_G) = \sum_i \alpha_i \mathbf{v}_i \text{ for } \mathbf{v}_i \in \mathbf{I}_G, \quad (6.3)$$

$$\alpha = \{\alpha\}_i = \text{softmax}(\text{MLP}(\mathbf{I}_G)), \quad (6.4)$$

where  $\theta_{\mathcal{I}}$  is the set of learnable parameters and the Multilayer Perceptron (MLP) computes linear weights by learning a mapping between the node features' vector space and  $\mathbb{R}$ . The question hub-node  $\mathcal{Q}$  and history hub-node  $\mathcal{H}$  are obtained following the same steps.

### Graph Aggregation

Our multimodal graphs are a variant of graph attention networks [179], although other types of GNNs can be used within each VDGR layer. Each multimodal graph uses  $K$  layers (purple boxes in Figure 6.1) to propagate information between its nodes. For example, the  $k$ -th image graph layer receives the node features from the previous graph layer and updates them as follows:

$$\mathbf{I}_G^{(k)} = \{\mathbf{v}_1^{(k)}, \dots, \mathbf{v}_{N_i}^{(k)}, \underbrace{\mathbf{v}_{N_i+1}^{(k)}}_{=\mathcal{Q}}\} = \text{GNN}_{\mathcal{I}}(\mathbf{I}_G^{(k-1)}), \quad (6.5)$$

$$\mathbf{v}_i^{(k)} = \text{GeLU}\left(\left\| \bigg\|_{h=1}^H \tilde{\mathbf{v}}_i^{(k,h)} + \mathbf{v}_i^{(k-1)}\right.\right), \quad (6.6)$$

$$\tilde{\mathbf{v}}_i^{(k,h)} = f\left(\sum_{v_s \in \mathcal{N}_{v_t}} \alpha_{s \rightarrow t}^{(k,h)} \mathbf{m}_{s \rightarrow t}^{(k,h)}\right) \forall 1 \leq h \leq H, \quad (6.7)$$

where  $\|$  and  $H$  are the concatenation operation and the number of GNN attention heads, respectively.  $\mathcal{N}_{v_t}$  represents the neighbourhood of node  $v_t$ ,  $\alpha_{s \rightarrow t}$  denotes the attention weight that scales the message  $\mathbf{m}_{s \rightarrow t}^{(k,h)}$  between a source node  $\mathbf{v}_s$  and a target node  $\mathbf{v}_t$ , and  $f$  is a linear layer. The messages  $\mathbf{m}_{s \rightarrow t}^{(k,h)}$  between the nodes are computed following:

$$\mathbf{m}_{s \rightarrow t}^{(k,h)} = g_h(\mathbf{v}_s^{(k-1)}, \mathbf{e}_{s \rightarrow t}), \quad (6.8)$$

where  $g_h$  is a linear layer and  $\mathbf{e}_{s \rightarrow t}$  is the edge feature between the nodes  $\mathbf{v}_s$  and  $\mathbf{v}_t$ . We omitted the  $l$  index of the  $\text{VDGR}$  layer in the previous equations for brevity. The  $k$ -th question and history graph layers update their node features  $\mathbf{g}_G^{(k)}$  and  $\mathbf{H}_G^{(k)}$  in the same manner.

### Latent DST & Hidden States Enhancement

The outputs of the multimodal graphs of the last  $K$ -th layer, i.e.  $\mathbf{I}_G^{(l,K)}$ ,  $\mathbf{g}_G^{(l,K)}$  and  $\mathbf{H}_G^{(l,K)}$ , hold more fine-grained information about the corresponding modalities by exploiting their local structures and form the latent DST representations that we use to enhance the hidden states of the following BERT layer. First, the GNN features are scattered back to their corresponding indices within the original hidden states. Then, we apply a fusion operation inspired by the idea of residual connections [72] as illustrated in Figure 6.1. Our experiments show that this step is crucial and leads to significant improvements in performance. These operations can be summarized as follows:

$$\hat{\mathbf{I}}_h^l = \mathbf{I}_h^l \oslash (\mathbf{I}_G^{(l,K)}, \text{Idx}_v), \quad \hat{\mathbf{I}}_h^l = \lambda \mathbf{I}_h^l + (1 - \lambda) \hat{\mathbf{I}}_h^l, \quad (6.9)$$

$$\tilde{\mathbf{T}}_h^l = (\mathbf{T}_h^l \oslash (\mathbf{g}_G^{(l,K)}, \text{Idx}_q)) \oslash (\mathbf{H}_G^{(l,K)}, \text{Idx}_h), \quad (6.10)$$

$$\hat{\mathbf{T}}_h^l = \lambda \mathbf{T}_h^l + (1 - \lambda) \tilde{\mathbf{T}}_h^l \quad \text{for } \lambda \in [0, 1], \quad (6.11)$$

where  $\oslash$  denote the scatter operation and  $\text{Idx}_*$  the indices of the graph nodes features with respect to the BERT hidden states. The final DST-enhanced features  $\hat{\mathbf{I}}_h^l$  and  $\hat{\mathbf{T}}_h^l$  are passed to the next BERT layer.

### 6.2.3 Loss Heads

We complement the traditional losses used for the visual dialog task (MLM, MRM, and NSP) with GEM to improve the learning of the local structure of each modality, and thus to enhance the feature representation of our multimodal graphs.

### Masked Language & Region Modeling

Similar to MLM [50], we randomly mask 10% of the text tokens and image objects with the special token [MASK] and task the model to recover them based on the surrounding tokens and cross-modal clues,

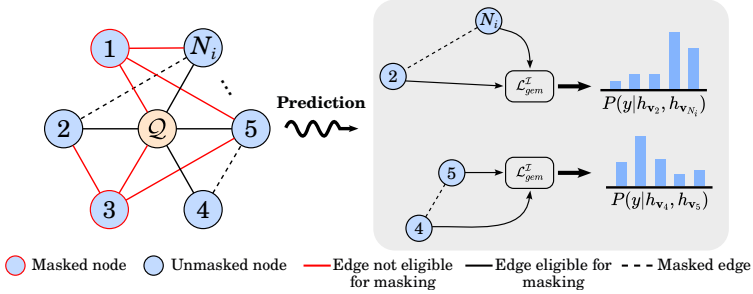


FIGURE 6.2: Only edges connecting two unmasked node features are eligible for masking with a 15% probability.

i.e.

$$\mathcal{L}_{\text{mlm}} = -\mathbb{E}_{(\mathbf{w}, \mathbf{I})} [\log P(\mathbf{w}_m | \mathbf{w}_{\setminus m}, \mathbf{h}_{[\text{IMG}]})], \quad (6.12)$$

$$\mathcal{L}_{\text{mrm}} = -\mathbb{E}_{(\mathbf{w}, \mathbf{I})} [\log P(\mathbf{v}_m | \mathbf{I}_{\setminus m}, \mathbf{h}_{[\text{CLS}]})], \quad (6.13)$$

where  $\mathbf{w}_m$  and  $\mathbf{v}_m$  denote the masked words and the masked regions, respectively.

### Next Sentence Prediction

The aim of NSP is to identify whether or not the appended answer  $\tilde{\mathcal{A}}_t$  of equation Equation 6.2 is adequate to answer the question at hand, i.e.

$$\mathcal{L}_{\text{nsf}} = -\mathbb{E}_{(\mathbf{w}, \mathbf{I})} [\log P(y | \mathcal{N}(\mathbf{h}_{[\text{CLS}]}, \mathbf{h}_{[\text{IMG}]}))], \quad (6.14)$$

where  $y \in \{0, 1\}$  is a ground-truth binary label, and  $\mathcal{N}(\dots)$  is the binary answer prediction head operating on the element-wise product of the [CLS] and [IMG] token representations.

### Graph Edge Masking

We introduce multimodal GEM to alleviate the potential problem of having well-trained LLM layers but not sufficiently trained GNNs. Given that, by design, our GNNs receive their features from the previous LLM layer, they inherit some masked node representations. In order to make the edge prediction task stable, we only mask 15% of the edges connecting two unmasked node features as illustrated in Figure 6.2. The representations of these nodes are then used to predict the masked edge classes. For the example of the image graph,

this results in the following loss:

$$\mathcal{L}_{\text{gem}}^{\mathcal{I}} = -\mathbb{E}_{(\mathbf{w}, \mathbf{I})} \left[ \log P(y^{(i,j)} | \mathbf{h}_{\mathbf{v}_i}, \mathbf{h}_{\mathbf{v}_j}) \right], \quad (6.15)$$

where  $y^{(i,j)}$  is the ground-truth edge type between the nodes  $\mathbf{v}_i$  and  $\mathbf{v}_j$ . We follow the same procedure to obtain the question and history GEM losses  $\mathcal{L}_{\text{gem}}^{\mathcal{Q}}$  and  $\mathcal{L}_{\text{gem}}^{\mathcal{H}}$ .

### Total Loss

We adopt a two-stage approach to train our model. First, we train it on a warm-up task of masked token and graph edge prediction, i.e. using the total loss:

$$\mathcal{L}_{\text{warm}} = \alpha_1 (\mathcal{L}_{\text{mlm}} + \mathcal{L}_{\text{mrm}}) + \alpha_2 \mathcal{L}_{\text{GEM}}, \quad (6.16)$$

$$\mathcal{L}_{\text{GEM}} = \mathcal{L}_{\text{GEM}}^{\mathcal{I}} + \mathcal{L}_{\text{GEM}}^{\mathcal{Q}} + \mathcal{L}_{\text{GEM}}^{\mathcal{H}}. \quad (6.17)$$

Then, we train it using the established visual dialog loss

$$\mathcal{L}_{\text{VD}} = \mathcal{L}_{\text{mlm}} + \mathcal{L}_{\text{mrm}} + \mathcal{L}_{\text{nspr}}. \quad (6.18)$$

## 6.3 Experiments

### 6.3.1 Datasets

We evaluated VDGR on the popular datasets of VisDial v0.9 and VisDial v1.0. VisDial v0.9 has circa 83k training and 40k validation dialogs and the more recent v1.0 version consists of about 123k, 2k, and 8k images for training, validation, and testing, respectively. Each image comes with a caption and 10 question-answer pairs and each question is associated with 100 candidate answers. The validation data and part of the training data of VisDial v1.0 provide dense annotations for the candidate answers. In addition, we evaluated our model on two recent and more challenging datasets, i.e. VisPro [203] and VisDialConv [7].

### 6.3.2 Quantitative Results

#### VisDialConv & VisPro

First, we evaluated our model on VisPro and VisDialConv which were introduced to verify the role of dialog history in answering the current

Method	VisPro <sup>a</sup>						VisDialConv					
	NDCG↑	MRR↑	R@1↑	R@5↑	R@10↑	Mean↓	NDCG↑	MRR↑	R@1↑	R@5↑	R@10↑	Mean↓
MCA-I [7]	59.80	57.88	45.39	72.24	82.76	5.84	52.07	55.55	41.65	72.47	83.81	5.92
MCA-I-HConcQ [7]	61.08	<u>61.79</u>	<u>48.95</u>	<u>77.50</u>	86.58	<u>4.72</u>	54.84	62.06	47.42	<u>80.10</u>	<u>88.87</u>	<u>4.37</u>
MCA-I-HGuidedQ [7]	61.35	60.13	47.11	75.26	86.18	5.23	53.81	<u>62.29</u>	<u>48.35</u>	<u>80.10</u>	88.76	4.42
MCA-I-VGH [7]	61.68	59.33	46.18	75.53	<u>86.71</u>	5.07	55.48	58.48	44.54	74.95	86.19	5.18
MCA-I-H [7]	<u>61.72</u>	59.62	45.92	77.11	<u>86.45</u>	4.85	53.01	61.24	47.63	79.07	87.94	4.77
Student [87]	–	–	–	–	–	–	<u>59.30</u>	–	–	–	–	–
<b>VDGR</b>	<b>67.09</b>	<b>66.82</b>	<b>54.47</b>	<b>81.71</b>	<b>91.44</b>	<b>3.54</b>	<b>60.35</b>	<b>69.89</b>	<b>57.21</b>	<b>85.97</b>	<b>92.68</b>	<b>3.15</b>

<sup>a</sup>Same subset as in [7].

TABLE 6.1: Performance comparison on VisPro and VisDialConv. The best and second-best results are in **bold** and underlined, respectively. ↑ indicates higher is better and ↓ indicates lower is better.

Model	MRR↑	R@1↑	R@5↑	R@10↑	Mean↓
MN [48]	59.65	45.55	76.22	85.37	5.46
CoAtt [206]	63.98	50.29	80.71	88.81	4.47
HCIAE [132]	62.22	48.48	78.75	87.59	4.81
CorefNMN [98]	64.10	50.92	80.18	88.81	4.45
RvA [144]	66.34	52.71	82.97	90.73	3.93
Student [87]	60.03	50.40	70.74	77.15	12.13
DVAN [62]	66.67	53.62	82.85	90.72	3.93
VD-BERT [181]	70.04	57.79	85.34	92.68	4.04
VisDial-BERT [141]	<u>71.99</u>	<u>59.41</u>	<u>87.92</u>	<u>94.59</u>	<u>2.87</u>
<b>VDGR</b>	<b>74.50</b>	<b>62.10</b>	<b>90.49</b>	<b>96.37</b>	<b>2.45</b>

TABLE 6.2: Performance comparison on the *val* split of VisDial v0.9 dataset. NDCG is not supported in this version of the dataset.

Model	NDCG↑	MRR↑	R@1↑	R@5↑	R@10↑	Mean↓
LTMI [143]	62.72	62.32	48.94	78.65	87.88	4.86
VD-BERT [181]	63.22	67.44	54.02	83.96	92.33	3.53
VisDial-BERT [141]	60.96	67.17	53.42	84.41	92.62	3.41
MCA [7]	60.27	64.33	51.12	80.91	89.65	4.24
UniMM-UL [182]	62.86	53.49	42.70	65.03	74.58	10.65
UTC [32]	63.22	68.58	55.48	85.38	93.20	3.28
Student [87]	<b>65.47</b>	53.19	43.08	64.09	71.51	14.34
VD-PCR [202]	64.16	<u>69.71</u>	<u>56.79</u>	<u>85.82</u>	<u>93.64</u>	<u>3.15</u>
<b>VDGR</b>	<u>64.32</u>	<b>69.91</b>	<b>57.01</b>	<b>86.14</b>	<b>93.74</b>	<b>3.13</b>

TABLE 6.3: Performance comparison on the *val* split of VisDial v1.0 dataset.

question at hand  $Q_t$ . We compared VDGR to the baselines introduced in [7] as well as the most recent Student model [87]. As can be seen from Table 6.1, VDGR managed to outperformed all MCA [7] variants across all metrics on both datasets. Specifically, it increased the performance of the baselines by over 5 absolute points on NDCG and MRR on VisPro. On VisDialConv, VDGR increased the top performance by over 4 absolute points on the same metrics. Table 6.1 also shows that our model managed to surpass the Student model [87] by over 1 NDCG absolute point although it was trained on circa 13M additional images underlining the efficacy of our GNN-based DST approach.

### VisDial v0.9

Second, we compared our model with the state of the art on VisDial v0.9. As can be seen from Table 6.2, VDGR outperformed all previous models and achieved new state-of-the-art results across all metrics. Specifically, it outperformed pre-training methods such

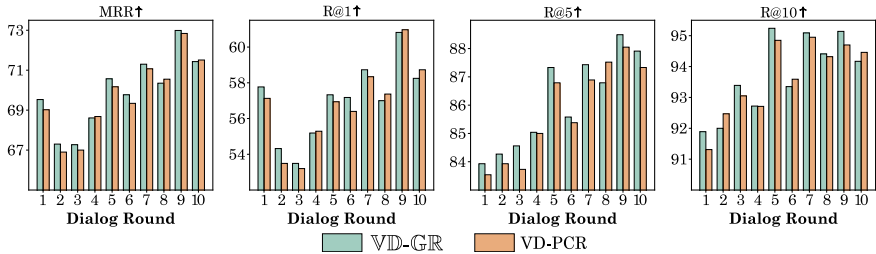


FIGURE 6.3: Performance comparison on different dialog rounds of the VisDial v1.0 *val* split. We only considered sparse metrics since the dense annotations used to compute the NDCG score are not defined on all 10 rounds of the validation dialogs.

as VisDial-BERT [141] and VD-BERT [181] by a large margin. Even more importantly, it managed to also surpass more recent models such as Student [87]. Specifically,  $\text{VDGR}$  improved the MRR and  $\text{R@1}$  scores by over 2.5 absolute points compared to the second best model.

### VisDial v1.0 *val*

We then compared  $\text{VDGR}$  with the state of the art on the *val* split of VisDial v1.0. As can be seen from Table 6.3, our model outperformed all previous models across all five sparse metrics. Specifically, it outperformed the seminal works of VisDial-BERT [141] and VD-BERT [181] by a considerable margin. Even more importantly, it managed to also surpass more recent models such as UniMM-UL [182], VD-PCR [202], UTC [32], and Student [87].  $\text{VDGR}$  improved MRR,  $\text{R@1}$ , and  $\text{R@5}$  by over 0.2 absolute points compared to the second best VD-PCR model [202]. Furthermore, we compared their performance on individual dialog rounds using the sparse metrics (MRR,  $\text{R@1}$ ,  $\text{R@5}$ , and  $\text{R@10}$ ). As can be seen from Figure 6.3,  $\text{VDGR}$  managed to outperform VD-PCR [202] on almost all rounds of the dataset further showcasing the benefits of our novel DST approach.

### VisDial v1.0 *test-std*

Finally, we compared our model with state-of-the-art published baselines on the *test-std* split of the VisDial v1.0 dataset.

**State-of-the-art Results on Sparse Metrics.** As can be seen from the first section of Table 6.4,  $\text{VDGR}$  lifted the state-of-the-art  $\text{R@5}$ ,

Model	NDCG $\uparrow$	MRR $\uparrow$	R@1 $\uparrow$	R@5 $\uparrow$	R@10 $\uparrow$	Mean $\downarrow$
MN [48]	47.50	55.49	46.98	72.30	83.30	5.92
CorefNMN [98]	54.70	61.50	47.55	78.10	88.80	4.40
FGA [160]	56.90	66.20	52.75	82.92	91.07	3.80
DAN [88]	57.59	63.20	49.63	79.75	89.35	4.30
LTM [143]	59.03	64.08	50.20	80.68	90.35	4.05
CAG [63]	56.64	63.49	49.85	80.63	90.15	4.11
GOG [33]	61.04	63.52	50.01	80.13	89.28	4.31
VD-BERT [181]	59.96	65.44	51.63	82.23	90.68	3.90
VisDial-BERT [141]	63.87	67.50	53.85	84.68	93.25	3.32
UTC [32]	<u>64.60</u>	<u>68.70</u>	<b>55.73</b>	84.93	93.08	3.32
VD-PCR [202]	<u>63.55</u>	<b>68.73</b>	<u>55.45</u>	<u>85.38</u>	<u>93.53</u>	<u>3.21</u>
UniMM-UL [182]	63.90	68.14	54.57	85.15	93.13	3.27
‡Student [87]	<b>64.91</b>	68.44	55.05	85.18	93.35	3.23
<b>VDGR</b>	<b>63.49</b>	<b>68.65</b>	<b>55.33</b>	<b>85.58</b>	<b>93.85</b>	<b>3.20</b>
♣MCA [7]	72.47	37.68	20.67	56.67	72.12	8.89
♣VD-BERT [181]	74.54	50.74	33.15	61.58	77.15	7.18
♣VisDial-BERT [141]	74.47	50.74	37.95	64.13	80.00	6.28
♣UTC [32]	74.32	50.24	37.12	63.98	79.88	6.48
♣VD-PCR [202]	75.30	56.17	<u>45.32</u>	68.05	80.98	6.15
♣UniMM-UL [182]	<u>75.92</u>	<u>56.18</u>	43.70	<u>71.03</u>	<b>84.80</b>	<u>5.42</u>
♣VDGR	<b>75.95</b>	<b>58.30</b>	<b>46.55</b>	<b>71.45</b>	<u>84.52</u>	<b>5.32</b>
◇♣P1+P2 [152]	74.91	49.13	36.68	62.96	78.55	7.03
◇♣VD-BERT [181]	75.35	51.17	38.90	62.82	77.98	6.69
◇♣VD-PCR [202]	76.14	56.05	<u>44.75</u>	<u>68.40</u>	<u>82.75</u>	<u>5.72</u>
◇♣UniMM-UL [182]	<u>76.17</u>	<b>56.42</b>	44.32	<b>70.23</b>	<b>84.52</b>	<b>5.47</b>
◇♣VDGR	<b>76.43</b>	<u>56.35</u>	<b>45.18</b>	68.13	82.18	5.79

TABLE 6.4: Performance comparison on the *test-std* split of VisDial v1.0 dataset. ♣ indicates fine-tuning on dense annotations and ◇ denotes ensemble model. ‡ denotes the use of extra large datasets for training.

R@10, and Mean scores from 85.38, 93.53, and 3.21 achieved by VD-PCR [202] to 85.58, 93.85, and 3.20, respectively. On the remaining metrics, our models performed on par with the state of the art. Specifically, it reached respective scores of 68.65 and 55.33 on MRR and R@1, only third to UTC [32] and VD-PCR [202].

**Fine-tuning on Dense Annotations.** As in previous works, we fine-tuned our model on the dense annotations released by [141] in order to improve the NDCG score. As illustrated in the second section of

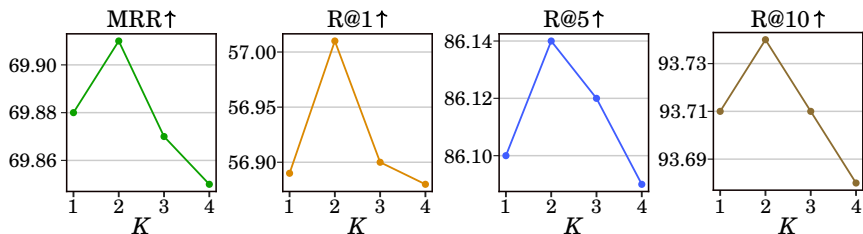


FIGURE 6.4: Performance comparison with different number of GNN layers  $K$  on the *val* split of VisDial v1.0. Results are shown for the *val* split of VisDial v1.0.

Table 6.4, the NDCG score sharply increased from 63.49 to 75.95, outperforming all previous models in the single-model setting. However, by fine-tuning on dense annotations, we decreased the performance on the sparse metrics (MRR, R@1, R@5, R@10, and Mean). This well-known phenomenon of the dataset is due to the misalignment of the dense and sparse annotations as previously observed by [141, 181]. In contrast to other baselines,  $\text{VDGR}$  managed to keep relatively high sparse scores (4/5 metrics are the new state of the art) after fine-tuning.

**Ensemble Setting.** As it is common practice, we fine-tuned an ensemble comprised of *eight* pre-trained  $\text{VDGR}$  models to further improve the NDCG score. One set of four models was fine-tuned with CE and with a varying number of GNN layers  $K$ , i.e.  $K = \{1, 2, 3, 4\}$ . Each model of the second set was trained using the ListNet [29] ranking optimization method. As can be seen in the last section of Table 6.4, our ensemble model reached an NDCG score of 76.43 outperforming the closest competitor UniMM-UL [182] with only 76.17 and, thus setting a new state-of-the-art benchmark performance.

### 6.3.3 Ablation Study

#### Number of GNN layers $K$

This is an important hyperparameter of our model that greatly affects the quality of latent DST learning: If  $K$  is too small, then the expressive power of the GNNs will be hampered. Contrarily, if  $K$  is too large, the GNNs will suffer from over-smoothing [114]. To this end, we increased  $K$  incrementally from *one* to *four* and used the sparse metrics of

<b>Ablation</b>	<b>NDCG<math>\uparrow</math></b>	<b>MRR<math>\uparrow</math></b>	<b>R@1<math>\uparrow</math></b>	<b>R@5<math>\uparrow</math></b>	<b>R@10<math>\uparrow</math></b>	<b>Mean<math>\downarrow</math></b>
VDGR w/ $\lambda = 0$	56.69	67.71	54.07	85.03	92.84	3.33
VDGR w/o $\mathcal{L}_{\text{warm}}$	63.76	69.83	56.84	<u>86.05</u>	<u>93.70</u>	3.15
VDGR w/o sharing	<u>64.15</u>	69.79	56.73	86.02	<u>93.68</u>	3.15
VDGR w/o HN	64.11	<u>69.86</u>	<u>56.88</u>	85.97	93.67	<u>3.14</u>
VDGR (Full)	<b>64.32</b>	<b>69.91</b>	<b>57.01</b>	<b>86.14</b>	<b>93.74</b>	<b>3.13</b>

TABLE 6.5: Performance comparison of our model ablations on the *val* split of VisDial v1.0.

the task (MRR, R@1, R@5, R@10) for validation. As illustrated in Figure 6.4, the performance of our model peaked at  $K = 2$  for all four metrics. Thus, we kept this value fixed in all previous experiments unless explicitly stated otherwise.

### Model Ablations

In addition to the full model, we evaluated the following ablated versions:

- **VDGR w/ $\lambda = 0$** : This variant fully excludes the latent DST features by setting  $\lambda = 0$ , i.e. it does not apply the proposed residual connection of Equation 6.11 while augmenting the BERT hidden states.
- **VDGR w/o  $\mathcal{L}_{\text{warm}}$** : This variant was not trained on the warm-up task of edge prediction as discussed in Section 6.2.3.
- **VDGR w/o sharing**: This variant did not share the weights of the GNNs in each layer of our model.
- **VDGR w/o HN**: This variant did not use hub-nodes to propagate the information between the multimodal GNNs.

As can be seen from Table 6.5, the latent DST features are essential for high performance. Without them, VDGR achieved the lowest performance across all metrics. The same applies to the warm-up training stage: Although the performance of this ablated version improved over the previous one, it still performed significantly worse than our best model. The results also underline the importance of sharing the GNN weights within the different VDGR layers: Although this version has more weights, it still performed worse than our best model on all metrics. This finding was to be expected given that the local structure of each modality does not change from one VDGR layer to another. Finally, the importance of the inter-modal feature propagation using hub-nodes is highlighted by the two last rows

of [Table 6.5](#): The hub-nodes enabled our model to achieve the best performance across all metrics of the *val* split of the VisDial v1.0 dataset.

### 6.3.4 Qualitative Results

Finally, in [Figure 6.5](#) we show selected qualitative samples (with more in the supplementary material) from the *val* split of VisDial v1.0 alongside the ground truth answers, as well as the top-1 predictions of our VDGR model and VD-PCR [202] for comparison since it achieved the second best results on this split. We make two interesting observations: (1) Our model deals better with questions that require exploiting local structure within modalities. For example, it managed to correctly answer the question

$Q_2$  = How many skiers can you see?

in the first dialog sample (see [Figure 6.5a](#)) by predicting 11 whereas VD-PCR [202] predicted I can only see two. We hypothesize that this is due to the fact that our model is equipped with a DST mechanism that exploits the spatial structure of the visual input more effectively using the image graph compared to VD-PCR [202], although the latter has access to the same visual features. (2) VDGR has more accurate semantic understanding of the question. This is highlighted in answering

$Q_2$  = How old does she seem to be?

(referring to the woman) in the second dialog sample (see [Figure 6.5b](#)). Whereas our model correctly predicted Early 20’s, VD-PCR [202] failed by answering Grey, which is not a semantically-sound response. We refer to [Appendix C.2](#) for additional qualitative samples.

## 6.4 Limitations & Closing Remarks

In this chapter we continued our efforts towards developing more capable multimodal latent differentiable DST approaches by proposing VDGR, a model that alternates between GNN and LLM layers. Specifically, it leverages GNNs to exploit the local structures of the

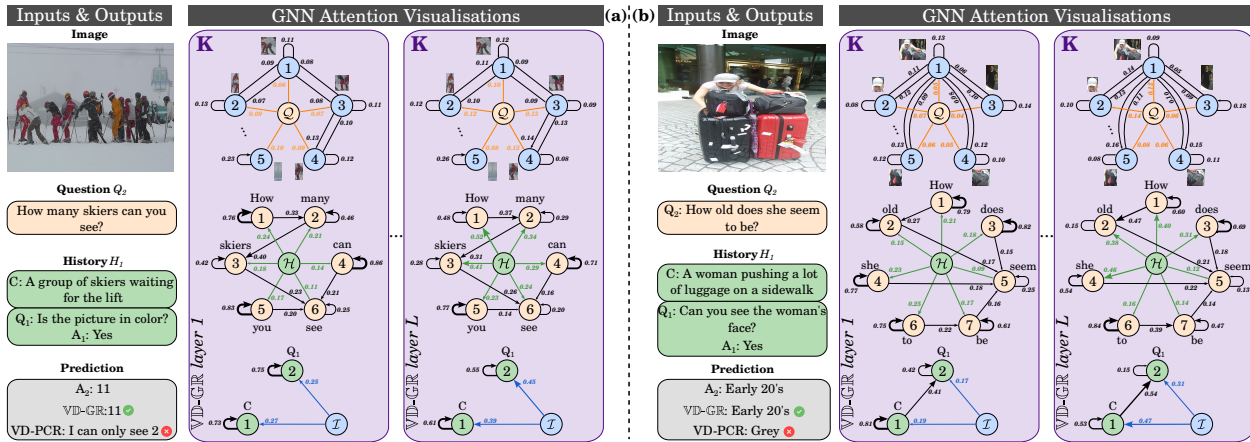


FIGURE 6.5: Qualitative samples from the VisDial v1.0 *val* split. The attention weights correspond to the  $K$ -th GNN of the first and last VDGR layers. We first average them across all attention heads before normalizing them for each node using a softmax function. For clarity, we only show a subset of the image graph nodes.

input modalities and produce fine-grained multimodal features that help the model focus on the relevant relationships between the constituents of each modality. These GNN features hold important information about the state of the dialog and are used to augment the hidden states of the subsequent LLM layer. Through extensive evaluations on *four* real-world visual dialog datasets, we demonstrated the effectiveness of our model in tackling the challenges of such datasets.

That said, VDGR suffers from some key limitations. First, it relies on an ad hoc construction of the graph topologies using deterministic rules or external tools. This poses a challenge for other datasets such as AVSD [9] for which this construction is not feasible, i.e. we cannot pre-defined clear relationships between its modalities' constituents. Second, it keeps the same graph topologies constant throughout the whole architecture. We posit that this is suboptimal, as the semantic relationships between the features of the input modalities might shift in deeper layers of the model.

# 7

## **GNN-Based DST with Latent Learnable Graphs**

*This chapter is based on the following publication and includes extracts thereof without explicit quotation:*

**Multimodal Video Dialog State Tracking in the Wild**

Adnen Abdessaied, Lei Shi, Andreas Bulling

In *Proceedings of the European Conference on Computer Vision (ECCV)*. Milan, Italy, 2024.

## 7.1 Introduction & Contributions

In this chapter, we build on our previous works on latent multimodal DST (Chapter 5 and Chapter 6). Specifically, we attempt to alleviate their limitations by introducing a latent DST approach capable of dealing with the complexity real-world datasets. To this end, we present  $\text{MST}_{\text{MIXER}}$ , a novel video dialog model operating over a generic GNN-based multimodal state tracking scheme. Current models that claim to perform multimodal state tracking fall short in two major aspects: (1) They either track only one modality (mostly the visual input) or (2) they target synthetic datasets that do not reflect the complexity of real-world in-the-wild scenarios. Our model addresses these two limitations in an attempt to close this crucial research gap. Specifically, it first tracks the most important constituents of each input modality in a similar manner to  $\text{OLViT}$ . Then, it predicts the *missing* underlying structure of the selected constituents of each modality by learning local latent graphs using a novel multimodal graph structure learning method. This is in stark contrast to  $\text{VDGR}$ , which assumes that the topologies of the ad hoc constructed graphs are kept constant throughout the whole architecture. Subsequently, the learned local graphs and features are parsed together to form a global graph operating on the mix of all modalities, further refining its structure and node embeddings. Finally, the fine-grained DST graph node features are used to enhance the hidden states of the backbone VLM similarly to  $\text{VDGR}$ . Extensive evaluations on *five* challenging VL benchmarks showcase the effectiveness of our proposed approach.

## 7.2 Method

As can be seen from Figure 7.1,  $\text{MST}_{\text{MIXER}}$  complements a BART layer [112] with a custom  $\text{MIXER}$  layer that learns latent DST features which are used to enhance the hidden states of the subsequent layer. The output of the modified encoder is input to a decoder that autoregressively **generates** a free-form answer to the question at hand as introduced in Chapter 3.2.3.

### 7.2.1 Input Representation Learning

#### Visual Representations

As it is standard for this task, the visual representations are extracted for a given video using I3D-rgb and I3D-flow models [30] pre-trained

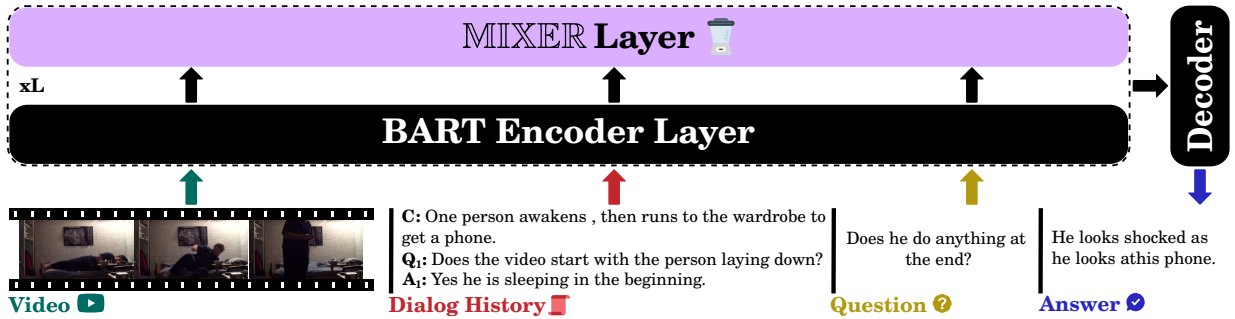


FIGURE 7.1:  $MST_{MIXER}$  takes a video, a dialog history, and a question as input. It uses a BART backbone adapted to deal with multimodal input features and enhances it with our graph-based DST approach.

on YouTube videos and the Kinetics dataset [89]. Formally, a video  $\mathcal{V}$  is first split into  $l_v$  segments using a sliding window of  $n$  frames. Then, each segment  $S = \{f_1, f_2, \dots, f_n\}$ , where  $f_i$  represents one video frame, is fed to the pre-trained I3D models to extract the  $d_v$ -dimensional video features  $\mathbf{V}_{\text{rgb}}, \mathbf{V}_{\text{flow}} \in \mathbb{R}^{l_v \times d_v}$ . Additionally, we extract object features  $\mathbf{V}_{\text{sam}} \in \mathbb{R}^{l_v \times d_s}$  from the middle frame of the video using SAM [94]. Finally, We project these features to match the hidden dimension  $d$  of BART using linear layers with weight matrices  $\mathbf{W}_{\text{rgb}}, \mathbf{W}_{\text{flow}}, \mathbf{W}_{\text{sam}}$ .

### Audio Representations

Similar to previous works [199, 102, 121], we use audio features extracted from a pre-trained VGGish model [167]. Since video and audio are synchronous, we use the same splits to generate the  $d_a$ -dimensional audio features  $\mathbf{A}_{\text{vggish}} \in \mathbb{R}^{l_v \times d_a}$ . Similarly to the video features, we map the audio features to the BART embedding space using a linear projection with a weight matrix  $\mathbf{W}_a \in \mathbb{R}^{d \times d_a}$ . We refer to [74] for further details about the feature extraction procedure.

### Textual Representations

We use the dialog history composed of the video caption, the previous question-answer pairs, and the current question as additional input to the encoder. We separate each segment with the special token  $\langle /s \rangle$ . Subsequently, we embed their concatenation into a dense representation  $\mathbf{T} = [\mathbf{T}_{\mathcal{H}}, \mathbf{T}_{\mathcal{Q}}] \in \mathbb{R}^{l_{\text{txt}} \times d}$  using a word embedding matrix  $\mathbf{W}_{\text{txt}} \in \mathbb{R}^{|\Omega_v| \times d}$ , where  $l_{\text{txt}}$ ,  $\Omega_v$ ,  $\mathbf{T}_{\mathcal{H}}$ , and  $\mathbf{T}_{\mathcal{Q}}$  are the length of the textual input, the vocabulary, the dense representation of the history and question, respectively. Finally, we input a shifted ground truth answer into the decoder and embed it using the same word matrix.

### State Tokens

We insert special *state* tokens  $\langle s_i \rangle$  at the beginning of each modality ( $\mathbf{V}_{\text{rgb}}, \mathbf{V}_{\text{flow}}, \mathbf{V}_{\text{sam}}, \mathbf{A}_{\text{vggish}}, \mathbf{T}_{\mathcal{H}}, \mathbf{T}_{\mathcal{Q}}$ ) and used them to keep track of the most relevant constituents.

## 7.2.2 Latent Multimodal DST

The main idea of  $\text{MST}_{\text{MIXER}}$  is to keep track of the most relevant constituents at different semantic levels (e.g. across modalities and encoder layers) and learn a latent GNN-based DST representations that we use to augment the hidden states of the VLM as we did in

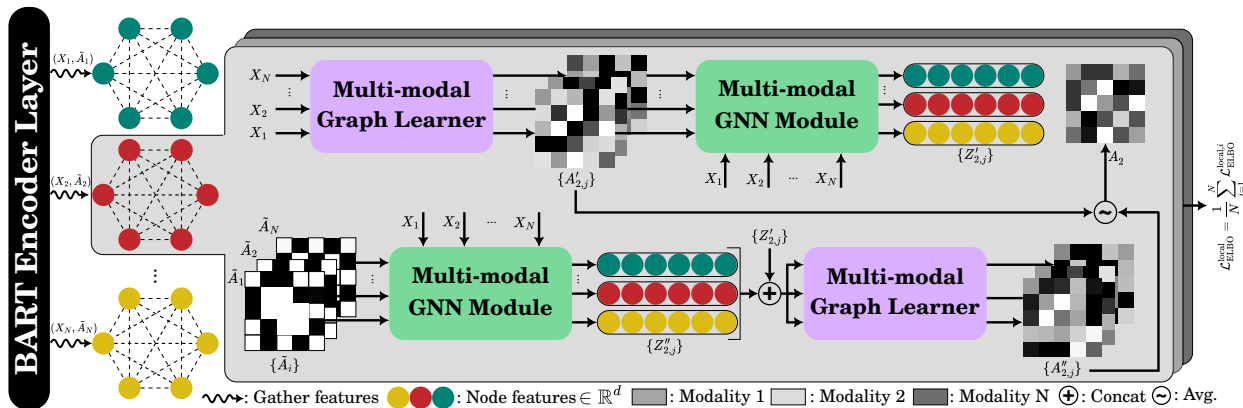


FIGURE 7.2: In Stage I,  $\text{MST}_{\text{MIXER}}$  first gathers multimodal features  $\{\mathbf{X}_i\}$  from the previous BART layer and computes their respective initial local structures  $\{\hat{\mathbf{A}}_i\}$ . Then, it simultaneously learns the local latent multimodal graphs and refines the features using a two-stream framework, i.e.,  $\{\mathbf{A}'_{i,j}, \mathbf{A}''_{i,j}\}$  and  $\{\mathbf{Z}'_{i,j}, \mathbf{Z}''_{i,j}\}$ , respectively. Finally, it outputs the final multimodal latent graph  $\mathbf{A}_i$  used to compute the local ELBO loss

$$\mathcal{L}_{\text{ELBO}}^{\text{local}} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{ELBO}}^{\text{local},i}.$$

**Chapter 6.** Therefore, we insert a MIXER layer after every  $\Delta$  encoder layer as shown in Figure 7.1. Our approach follows a two-stage divide and conquer scheme where we first learn the underlying *local* structures of the individual modalities before learning the *global* inter-modal structure of the mix of all available modalities. We posit that directly learning the latter might be daunting for such a high multimodal task.

### Multimodal Feature Tracking

We take advantage of the special state tokens  $\langle s_i \rangle$  to keep track of the most relevant modality-specific features at different embedding levels of the encoder. Specifically, for each modality, we select the  $K$  tokens with the highest attention values with respect to the corresponding state token, i.e.

$$\mathbf{X}_i = \text{top}_K(\alpha_{\text{avg}}(\mathbf{h}_{\langle s_i \rangle}, \mathbf{H}_i)) \in \mathbb{R}^{K \times d}, \quad (7.1)$$

where  $\alpha_{\text{avg}}(\mathbf{h}_{\langle s_i \rangle}, \mathbf{H}_i)$  is the attention values between the state token embedding and the remaining features  $\mathbf{H}_i$  of the  $i$ -th modality averaged across all attention heads.

### Mixing Stage I (Divide)

We posit that the selected features  $\{\mathbf{X}_i\}$  of all  $N$  modalities encapsulate rich information about the dialog state that could be leveraged to improve the learning capabilities of our model. As mentioned previously, we take advantage of the power of GNNs to refine these features based on their local structures, as previous work has highlighted the merit of integrating GNNs with transformer-based models [5, 194, 198]. However, the underlying structures that govern  $\{\mathbf{X}_i\}$  are missing in contrast to our previous work (Chapter 6). To this end, we propose a novel multimodal graph structure learning approach that simultaneously learns the graph weights and the adjacency matrix in the form of latent graphs. We posit that we can divide the adjacency matrix  $\mathbf{A}_i$  of the  $i$ -th modality into an initial (observable) part  $\tilde{\mathbf{A}}_i$  and a missing (sought-after) part  $\mathbf{A}'_i$  where  $\tilde{\mathbf{A}}_i$  is a *binary matrix* constructed using a kNN ( $k = 4$ ) approach based on  $\mathbf{X}_i$ . Thus,

$$P(\mathbf{X}_i, \mathbf{A}_i) = P(\mathbf{A}_i | \mathbf{X}_i) P(\mathbf{X}_i) \quad (7.2)$$

$$= P(\mathbf{A}'_i, \tilde{\mathbf{A}}_i | \mathbf{X}_i) P(\mathbf{X}_i). \quad (7.3)$$

Although the conditional distribution  $P(\mathbf{A}'_i, \tilde{\mathbf{A}}_i | \mathbf{X})$  can be modeled by a parametric families of distributions  $p_{\hat{\theta}}^i(\mathbf{A}'_i, \tilde{\mathbf{A}}_i | \mathbf{X})$ , the optimal parameter set  $\hat{\theta}$  is not known making the computations of the marginal

$$p_{\hat{\theta}}^i(\tilde{\mathbf{A}}_i | \mathbf{X}_i) = \int p_{\hat{\theta}}^i(\mathbf{A}'_i, \tilde{\mathbf{A}}_i | \mathbf{X}_i) d(\mathbf{A}'_i) \quad (7.4)$$

and therefore, the posterior of each modality

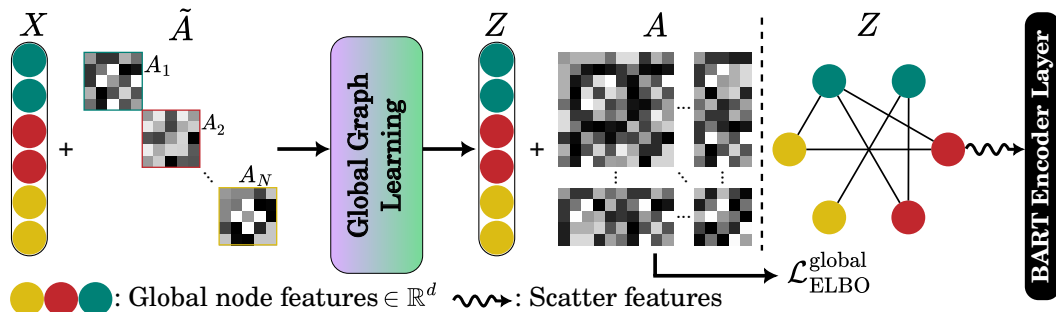
$$p_{\hat{\theta}}^i(\mathbf{A}'_i | \tilde{\mathbf{A}}_i, \mathbf{X}_i) = \frac{p_{\hat{\theta}}^i(\mathbf{A}'_i, \tilde{\mathbf{A}}_i | \mathbf{X}_i)}{p_{\hat{\theta}}^i(\tilde{\mathbf{A}}_i | \mathbf{X}_i)} \quad (7.5)$$

intractable. To be able to infer the missing part of the local adjacency matrix, we take advantage of VI to learn an approximation  $q_{\phi}^i(\mathbf{A}'_i | \tilde{\mathbf{A}}_i, \mathbf{X}_i)$  of the posterior. We postulate that the missing adjacency matrix of modality  $i$  depends on its own features  $\mathbf{X}_i$  and the features of other modalities  $\mathbf{X}_{j \neq i}$ . Therefore, we propose Multimodal Conditioning (MMC) of Equation 7.5 on all  $\mathbf{X}_{j \neq i}$  in addition to  $\mathbf{X}_i$ . We also follow the idea of [40] that better graph structures lead to better features, and better features lead to better graph structures. Therefore, as shown in Figure 7.2, we use a two-stream approach where one stream uses enhanced features to learn the latent multimodal graphs, and the other uses the predicted graphs to infer fine-grained features to learn both  $q_{\phi}^i$  and  $p_{\hat{\theta}}^i$  for each modality.

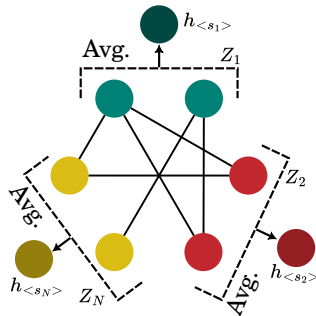
Specifically, in the purple module of the upper stream, we estimate an edge of latent graph  $\mathbf{A}'_{i,j}$  using cosine similarity as

$$a'_{mn} = \frac{1}{H} \sum_{h=1}^H \cos(\mathbf{w}_j^h \odot \mathbf{x}_m, \mathbf{w}_j^h \odot \mathbf{x}_n), \quad (7.6)$$

where  $\mathbf{x}_m, \mathbf{x}_n \in \mathbf{X}_i$ ,  $\{\mathbf{w}_j^h\}$  are learnable weights for each modality,  $H$  is the number of attention heads, and  $\odot$  denotes element-wise multiplication. Then, in the green module, we update the multimodal node features using an APPNP [60] module and the predicted latent graphs for modality  $i$  to get  $\{\mathbf{Z}'_{i,j}\}$ . For the lower stream, we first start by updating the node features similarly to the upper stream by using the initial graphs  $\{\tilde{\mathbf{A}}_i\}$  to get  $\{\mathbf{Z}''_{i,j}\}$ . Then, we use the enhanced node features  $\{[\mathbf{Z}'_{i,j}, \mathbf{Z}''_{i,j}]\}$  to predict the second set of local graphs  $\{\mathbf{A}''_{ij}\}$ .



(A) We use the predicted local latent graphs  $\{\mathbf{A}_i\}$  to initialize  $\tilde{\mathbf{A}} = \text{diag}([\mathbf{A}_1, \dots, \mathbf{A}_N], \mathbf{0})$  in order to learn the final global latent graph  $\mathbf{A}$ . The updated node features  $\mathbf{Z}$  hold important information about the global dialog state and are scattered back to their initial positions in the BART layer.



(B) We update the state embeddings  $\mathbf{h}_{\langle s_i \rangle}$  by averaging the corresponding features  $\mathbf{Z}$ .

FIGURE 7.3: Overview of mixing stage II.

Finally, we output the final local latent graph of modality  $i$  as

$$\mathbf{A}_i = \underbrace{\frac{1}{2}\tilde{\mathbf{A}}_i}_{\text{IB}} + \underbrace{\frac{1}{2}\sum_{j=1}^N \frac{1}{N}(\mathbf{A}'_{i,j} + \mathbf{A}''_{i,j})}_{\text{VI approximation via MMC}} \in \mathbb{R}^{K \times K}, \quad (7.7)$$

### Mixing Stage II (Conquer).

This stage tries to infer the global latent graph structure that governs the mix of all modalities  $\{\mathbf{X}_i\}$ . As seen in [Figure 7.3a](#), it depends on the previously predicted local latent graphs to build the initial global graph as

$$\tilde{\mathbf{A}} = \text{diag}([\mathbf{A}_1, \dots, \mathbf{A}_N], \mathbf{0}) \in \mathbb{R}^{N_K \times N_K}. \quad (7.8)$$

Similar to Stage I, we use a two-stream approach to learn the global  $p_\theta$  and  $q_\phi$  and thus the global latent graph  $\mathbf{A}$  and node features

$$\mathbf{z} = \frac{1}{2}(\mathbf{z}' + \mathbf{z}''), \quad (7.9)$$

that hold important information about the global dialog state, where  $\mathbf{z}'$  and  $\mathbf{z}''$  are obtained from the upper and lower streams, respectively. Finally, we update the state tokens embeddings  $\mathbf{h}_{\langle s_i \rangle}$  by averaging the corresponding features from  $\mathbf{z}$  (see [Figure 7.3b](#)) and integrate the latter back into the hidden state of the corresponding BART layer following

$$\mathbf{H} = (1 - \lambda)(\mathbf{H} \oslash (\mathbf{z}, \text{Idx})) + \lambda\mathbf{H}, \quad (7.10)$$

where  $\lambda \in (0, 1)$  is a hyperparameter and  $\oslash$ ,  $\mathbf{H}$ , and  $\text{Idx}$  denote the scatter operation, the hidden state of the BART layer and the indices of the nodes features  $\mathbf{z}$  relative to  $\mathbf{H}$ , respectively.

**Loss Function.** Since we rely on VI to infer the local and global latent graphs, we used two Evidence Lower Bound (ELBO) losses to optimize (1) the local multimodal graph learners  $\{q_\phi^i, p_\theta^i\}$  and (2) the global learners  $q_\phi, p_\theta$ . We refer to [Appendix A.6](#) for further details about the derivation of these losses. We trained our model end-to-end using a combination of the generative loss  $\mathcal{L}_{\text{gen}}$  as discussed in

Chapter 3.2 and both ELBO losses, i.e.

$$\mathcal{L} = \alpha_1 \mathcal{L}_{\text{gen}} - \alpha_2 \mathcal{L}_{\text{ELBO}}^{\text{local}} - \alpha_3 \mathcal{L}_{\text{ELBO}}^{\text{global}}, \quad (7.11)$$

$$\mathcal{L}_{\text{ELBO}}^{\text{local}} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{ELBO}}^{\text{local},i}, \quad (7.12)$$

where  $\{\alpha_k\}$  are hyperparameters and  $\mathcal{L}_{\text{ELBO}}^{\text{local},i}$  is the local ELBO loss for the  $i$ -th modality.

## 7.3 Experiments

### 7.3.1 Datasets

We mainly evaluated our model on the popular and challenging AVSD dataset [9]. Each of its dialogs comes with 10 question-answer pairs as well as a short description/caption based on a video. Each video is collected from the Charades dataset [166] and the dialogs are generated by human annotators. We considered all three benchmarks of the dataset, i.e. AVSD-DSTC7 [200], AVSD-DSTC8 [91], and AVSD-DSTC10 [161], which were respectively released for the Dialog System Technology Challenge (DSTC). To assess the generalizability of our model, we further experimented with the generative task of SIMMC 2.0 [97] and the recent open-ended video question answering dataset of NExT-QA [189].

### 7.3.2 Metrics

We used the established official metrics for each dataset in order to fairly compare  $\text{MST}_{\text{MIXER}}$  with the previous models. Specifically, for all *three* AVSD datasets, we used BLEU (**B-n**) [147], ROUGE-L (**R**) [124], METEOR (**M**) [19], and CIDEr (**C**) [178]. Whereas for SIMMC and NExT-QA, we used **B-4** and **WUPS** [135] scores, respectively.

### 7.3.3 Main Results

#### AVSD-DSTC7

As shown in Table 7.1, our model managed to achieve new state-of-the-art results across all evaluation metrics, thereby outperforming strong baselines, including PDC [102], DialogMCF [41], THAM [199], and ITR [208]. Specifically,  $\text{MST}_{\text{MIXER}}$  outperformed the latest ITR [208] model by over 1.5% (relative improvement) on B-2, B-3, B-4,

Model	AVSD-DSTC7							AVSD-DSTC8						
	B-1	B-2	B-3	B-4	M	R	C	B-1	B-2	B-3	B-4	M	R	C
Baseline [74]	62.1	48.0	37.9	30.5	21.7	48.1	73.3	61.4	46.7	36.5	28.9	21.0	48.0	65.1
MTN [106]	71.5	58.1	47.6	39.2	26.9	55.9	106.6	–	–	–	–	–	–	–
JMAN [44]	66.7	52.1	41.3	33.4	23.9	53.3	94.1	64.5	50.4	40.2	32.4	23.2	52.1	87.5
VGD [105]	74.9	62.0	52.0	43.6	28.2	58.2	119.4	–	–	–	–	–	–	–
BiST [107]	75.5	61.9	51.0	42.9	28.4	58.1	119.2	68.4	54.8	45.7	37.6	27.3	56.3	101.7
SCGA [90]	74.5	62.2	51.7	43.0	28.5	57.8	120.1	71.1	59.3	49.7	41.6	27.6	56.6	112.3
RLM [121]	76.5	64.3	54.3	45.9	29.4	60.6	130.8	74.6	62.6	52.8	44.5	28.6	59.8	124.0
PDC [102]	77.0	65.3	53.9	44.9	29.2	60.6	129.5	74.9	62.9	52.8	43.9	28.5	59.2	120.1
AV-TRN [161]	–	–	–	40.6	26.2	55.4	107.9	–	–	–	39.4	25.0	54.5	99.7
VGNMN [103]	–	–	–	42.9	27.8	57.8	118.8	–	–	–	–	–	–	–
COST [151]	72.3	58.9	48.3	40.0	26.6	56.1	108.5	69.5	55.9	46.5	3.82	27.8	57.4	105.1
MRLV [8]	–	59.2	49.3	41.5	26.9	56.9	115.9	–	–	–	–	–	–	–
♦THAM [199]	77.8	65.4	54.9	46.8	<u>30.8</u>	<u>61.9</u>	133.5	<u>76.4</u>	<u>64.1</u>	53.8	45.5	<u>30.1</u>	<u>61.0</u>	<u>130.4</u>
DialogMCF [41]	77.7	65.3	54.7	45.7	30.6	61.3	<u>135.2</u>	75.6	63.3	53.2	44.9	29.3	60.1	125.3
ITR [208]	<u>78.2</u>	<u>65.5</u>	<u>55.2</u>	<u>46.9</u>	30.5	<u>61.9</u>	133.1	76.2	<u>64.1</u>	<u>54.3</u>	<u>46.0</u>	29.8	60.7	128.5
MST <sub>MIXER</sub>	<b>78.7</b>	<b>66.5</b>	<b>56.3</b>	<b>47.6</b>	<b>31.3</b>	<b>62.5</b>	<b>138.8</b>	<b>77.5</b>	<b>66.0</b>	<b>56.1</b>	<b>47.7</b>	<b>30.6</b>	<b>62.4</b>	<b>135.4</b>
w/o $\mathbf{V}_{\text{sam}}$	78.6	66.3	56.0	47.4	31.2	62.2	137.3	77.4	65.8	56.0	47.3	30.6	62.1	134.8
w/o $\mathbf{A}_{\text{vggish}}$	78.4	66.0	55.8	47.1	31.0	62.0	136.5	77.1	65.6	55.7	47.1	30.2	61.8	133.6

TABLE 7.1: Results on AVSD-DSTC7 and AVSD-DSTC8. Best and second best performances are in **bold** and underlined, respectively. ♦ = Two-stage training.

Model	B-1	B-2	B-3	B-4	M	R	C
AV-TRN [161]	–	–	–	24.7	19.1	43.7	56.6
+ Ext. [161]	–	–	–	37.1	24.5	53.5	86.9
DSTC10 [77]	67.3	54.5	44.8	37.2	24.3	53.0	91.2
DialogMCF [41]	69.3	55.6	45.0	36.9	24.9	53.6	91.2
$MST_{MIXER}$	<b>70.0</b>	<b>57.4</b>	<b>47.6</b>	<b>40.0</b>	<b>25.7</b>	<b>54.5</b>	<b>99.8</b>
w/o $V_{sam}$	69.8	57.4	47.5	39.8	25.6	54.3	97.6
w/o $A_{vggish}$	69.7	57.1	47.2	39.5	25.1	54.0	96.9

TABLE 7.2: Results on AVSD-DSTC10.

and M scores. Since some previous models did not use SAM [94] and audio features, we trained two additional versions of our model where we only removed SAM features before additionally removing the audio features. Both versions are denoted by “w/o  $V_{sam}$ ” and “w/o  $A_{vggish}$ ”, respectively. As seen from Table 7.1, both versions still outperform all previous models across all evaluation metrics.

### AVSD-DSTC8

As depicted in Table 7.1, models tend to struggle more on this more recent benchmark. However,  $MST_{MIXER}$  scored new state-of-the-art results with higher relative improvements compared to DSTC7, thereby lifting the B-2, B-3, B-4, and C scores by over 3% relative to the second best models ITR [208] and THAM [199]. Similarly to AVSD-DSTC7, our ablated versions surpassed these models as well on all evaluation metrics of the benchmark.

### AVSD-DSTC10

We then evaluated  $MST_{MIXER}$  on the latest AVSD-DSTC10 benchmark. Contrary to the previous versions, AVSD-DSTC10 does not include human-generated video descriptions during inference, since these are unavailable in real-world applications. As depicted in Table 7.2, models struggle the most on this version of the benchmark. However, not only our full  $MST_{MIXER}$  model but also its two ablated versions managed to outperform the latest models on all evaluation metrics.

### ♣SIMMC

To assess the generalizability of our model, we additionally tested it on the generative task of SIMMC 2.0 [138]. As seen from Table 7.3,

Model	B-4
MTN [106]	21.7
GPT-2 [97]	19.2
BART [111]	33.1
PaCE [120]	<u>34.1</u>
<b>MST<sub>MIXER</sub></b>	<b>44.7</b>

TABLE 7.3: Results on SIMMC.

Model	WUPS <sub>C</sub>	WUPS <sub>T</sub>	WUPS <sub>D</sub>	WUPS
HCRN [109]	16.05	17.68	49.78	23.92
HGA [81]	<u>17.98</u>	<u>17.95</u>	<u>50.84</u>	24.06
Flamingo [10]	-	-	-	28.40
KcGA [84]	-	-	-	28.20
EMU [174]	-	-	-	23.40
<b>MST<sub>MIXER</sub></b>	<b>22.12</b>	<b>22.20</b>	<b>55.64</b>	<b>29.50</b>

TABLE 7.4: Results on open-ended NEXT-QA<sup>◇</sup>.

MST<sub>MIXER</sub> outperformed the latest published models such as PaCE [120] by achieving a B-4 score of 44.7.

### ♣NEXT-QA

Finally, we tested our model on the recent open-ended NEXT-QA benchmark [189]. As depicted in Table 7.4, MST<sub>MIXER</sub> not only outperformed HCRN [109] and HGA [81] on all WUPS scores [135] but also surpassed latest models such as Flamingo [10], KcGA [84], and EMU [174]. Specifically, it lifted the overall WUPS score by 1.1 absolute points compared to the seminal Flamingo-9B model with x18 more parameters.

## 7.3.4 Ablation Study

### Effect of $\lambda$ and $\Delta$

We independently optimized these hyperparameters based on the validation perplexity (PPL). First, we fixed  $\Delta = 4$  to guarantee a reasonable training time on our hardware setup and varied  $\lambda \in \{0, 0.1, 0.5, 0.9, 1\}$ . As seen in Table 7.5a, the best performance was achieved when using  $\lambda = 0.9$ . Thereafter, we varied  $\Delta \in \{2, 3, 4, 5\}$  while keeping  $\lambda = 0.9$  and achieved the best results for  $\Delta = 4$  as can be seen from Table 7.5b.

### GNN Types

We experimented with different types of GNNs within our full model. As depicted in Table 7.5c, the combination of MST<sub>MIXER</sub> with APPNP [60] led to the best overall performance compared to other GNNs such as GAT [179], GCN [93], and SAGE [68].

♣: Models trained with optimal hyperparameters from AVSD and without  $\mathbf{V}_{\text{sam}}$ .

◇ C, T, and D denote causal, temporal, and descriptive questions, respectively.

$\lambda$	PPL (val)	AVSD-DSTC7			AVSD-DSTC8		
		B-4	R	C	B-4	R	C
0.0		Training unstable					
0.1	11.03	17.3	29.0	35.1	11.4	24.3	21.2
0.5	5.48	44.6	60.3	126.4	44.7	59.4	123.8
0.9	<b>5.16</b>	<b>47.6</b>	<b>62.5</b>	<b>138.8</b>	<b>47.7</b>	<b>62.4</b>	<b>135.4</b>
1.0	5.30	45.1	60.8	131.3	42.3	61.1	126.9

(A) Influence of the value of  $\lambda$ .

MST <sub>MIXER</sub>	AVSD-DSTC7			AVSD-DSTC8		
	B-4	R	C	B-4	R	C
w/ GAT	46.7	61.5	135.4	46.5	60.9	129.4
w/ GCN	46.6	61.9	136.7	46.7	61.6	131.6
w/ SAGE	46.0	61.2	133.4	45.8	60.9	129.3
w/ APPNP	<b>47.6</b>	<b>62.5</b>	<b>138.8</b>	<b>47.7</b>	<b>62.3</b>	<b>134.9</b>

(c) Performance comparison of our best model using different GNN types.

$\Delta$	PPL (val)	AVSD-DSTC7			AVSD-DSTC8		
		B-4	R	C	B-4	R	C
$\leq 2$		Training too long					
3	5.19	45.7	61.5	134.1	46.7	61.5	131.8
4	<b>5.16</b>	<b>47.6</b>	<b>62.5</b>	<b>138.8</b>	<b>47.7</b>	<b>62.4</b>	<b>135.4</b>
5	5.21	45.0	61.1	133.6	44.6	60.5	129.1

(B) Influence of the value of  $\Delta$ .

MST <sub>MIXER</sub>	AVSD-DSTC7			AVSD-DSTC8		
	B-4	R	C	B-4	R	C
Base ( $\Delta = 2$ )	39.8	60.0	113.9	40.1	55.4	110.2
Large ( $\Delta = 4$ )	<b>47.6</b>	<b>62.5</b>	<b>138.8</b>	<b>46.7</b>	<b>61.6</b>	<b>131.6</b>

(d) Performance comparison between different model sizes. “Base” and “Large” mean that MST<sub>MIXER</sub> uses a base or a large backbone, respectively.TABLE 7.5: Ablation experiments of MST<sub>MIXER</sub>.

### Mode Size

Moreover, we experimented with different sizes our model. As depicted in [Table 7.5d](#), the variant of  $\text{MST}_{\text{MIXER}}$  that is based on BART-base significantly under-performed the large variant across all evaluation metrics of both datasets.

### Latent Graph Size $K$

As illustrated in the first section of [Table 7.6](#), we varied  $K$  from 7 to 16 in three-step intervals. The overall performance of  $\text{MST}_{\text{MIXER}}$  peaked when using  $K = 10$  features from each modality as the graphs' node embeddings. Using higher values of  $K$  rendered the learning of the global latent graphs with  $K \times N$  nodes more difficult and thus hurt the overall performance of our model. This is underlined by the behavior of the global ELBO loss  $\mathcal{L}_{\text{ELBO}}^{\text{global}}$  as illustrated in [Figure 7.4a](#). Using  $K = 7$  hurt the performance of our model across almost all metrics. We posit that low values of  $K$  are insufficient to capture each modality's most influential constituents. Therefore, we set  $K = 10$  in the rest of the experiments.

### Multimodal State Tracking GNNs

In each row of the middle section of [Table 7.6](#), we ablated *one* GNN-based tracking module and kept the remaining ones unchanged. Our full model outperformed all these ablated versions *despite them having access to the same input features*. The comparable results of all these ablated versions validate the use of a uniform graph size  $K$  for all available modalities. Finally, we replaced all GNNs (local and global) with vanilla transformer layers. As can be seen from the [last row](#) of the middle section of [Table 7.6](#), this version was outperformed by our full model as well, underlining the efficacy of our proposed multimodal graph learning approach.

### ELBO Losses

As can be seen in the third section of [Table 7.6](#), we conducted extensive experiments with different combinations of the ELBO losses: (1) We first ablated the learning of both global and local latent graphs and, therefore, both ELBO losses resulting in a plain BART model [112] without any DST capabilities. (2) We then only used the initial graphs  $\hat{\mathbf{A}}_i$  as the final latent graph approximations in both training stages I and II leading to improvements compared to plain BART.

K	GNNs	$\mathcal{L}_{\text{ELBO}}^{\text{local}}$	$\mathcal{L}_{\text{ELBO}}^{\text{global}}$	# Params.	AVSD-DSTC7				AVSD-DSTC8			
					B-1	B-4	R	C	B-1	B-4	R	C
7	All	✓	✓	~ 511M	77.8	47.0	61.8	136.2	76.6	47.0	61.5	131.8
10	All	✓	✓	~ 511M	<b>78.7</b>	<b>47.6</b>	<b>62.5</b>	<b>138.8</b>	<b>77.5</b>	<b>47.7</b>	<b>62.4</b>	<b>135.4</b>
13	All	✓	✓	~ 511M	77.0	45.4	60.6	131.9	75.7	45.2	60.4	127.0
16	All	✓	✓	~ 511M	76.6	45.4	60.7	132.6	75.8	45.9	60.5	128.4
10	w/o GNN <sub>rgb</sub>	✓	✓	~ 495M	78.4	<u>47.2</u>	62.4	137.2	77.3	<u>47.4</u>	62.0	133.2
10	w/o GNN <sub>flow</sub>	✓	✓	~ 495M	78.5	47.1	<u>62.5</u>	<u>138.5</u>	76.9	47.2	61.9	<u>134.1</u>
10	w/o GNN <sub>sam</sub>	✓	✓	~ 495M	78.1	46.1	<u>62.2</u>	<u>137.2</u>	<u>77.5</u>	46.5	61.7	<u>132.7</u>
10	w/o GNN <sub>vggish</sub>	✓	✓	~ 495M	78.0	45.8	61.4	134.9	<u>76.8</u>	46.5	61.0	131.0
10	w/o GNN <sub>H</sub>	✓	✓	~ 495M	78.1	45.7	61.8	134.1	77.4	46.7	<u>62.2</u>	134.0
10	w/o GNN <sub>Q</sub>	✓	✓	~ 495M	78.2	47.1	62.1	138.5	77.0	47.0	61.8	133.6
10	TRN	✗	✗	~ 500M	77.8	46.9	61.8	136.6	76.8	46.7	61.4	131.8
–	–	✗	✗	~ 411M	76.6	45.1	60.8	131.3	74.2	42.3	61.1	126.9
–	w/ only $\tilde{A}_i$	✗	✗	~ 413M	76.5	45.4	60.9	131.7	75.2	45.5	60.7	130.3
10	All	✓	✓	~ 416M	75.9	44.5	59.8	127.8	74.3	44.2	59.2	122.8
10	All	✓	✗	~ 506M	77.5	46.4	61.4	134.9	76.2	46.6	60.9	130.6
10	All	RAND	RAND	~ 448M	73.0	42.1	57.3	119.2	71.4	41.6	57.1	114.2

TABLE 7.6: Comparison between different ablated versions of our model. All ablations use SAM and audio features. TRN means that the model replaces the global and local multimodal GNNs with vanilla transformer layers, and RAND denotes that it uses random latent graphs instead of learning them. Our full model is highlighted in blue.

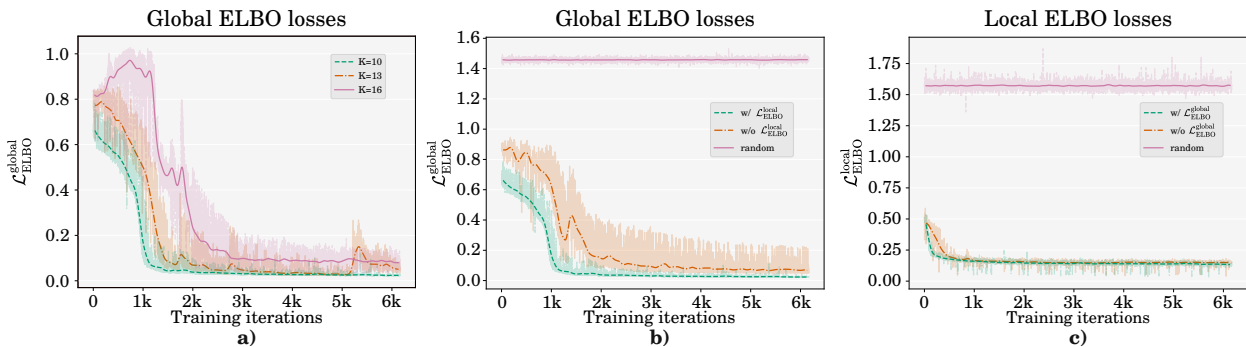


FIGURE 7.4: **a)** Larger values of  $K$  make the learning of the global latent graphs more challenging. **b)** The local ELBO loss  $\mathcal{L}_{ELBO}^{local}$  facilitates the learning of the global latent graphs. **c)** The global ELBO loss  $\mathcal{L}_{ELBO}^{global}$  facilitates the learning of the local latent graphs. All models use SAM and audio features.

MST <sub>MIXER</sub>	# Params.	AVSD-DSTC7				AVSD-DSTC8			
		B-1	B-4	R	C	B-1	B-4	R	C
w/o MMC	~ 500M	76.9	46.6	61.4	135.5	75.8	46.1	60.5	130.9
w/o IB	~ 511M	77.6	47.0	61.8	136.2	76.3	46.2	61.2	131.1
<b>Full</b>	~ 511M	<b>78.7</b>	<b>47.6</b>	<b>62.5</b>	<b>138.8</b>	<b>77.5</b>	<b>47.7</b>	<b>62.4</b>	<b>135.4</b>

TABLE 7.7: Comparison between different ablated versions of our model. All ablations were trained with SAM and audio features and with the optimal hyperparameters as the full model. **IB** = Initialization Bias, **MMC** = Multimodal Conditioning.

This ablated version keeps all graph topologies constant similarly to VDGR (Chapter 6). (3) Thereafter, we ablated the local ELBO loss and directly learned the global latent graphs. This version of our model underperformed BART, which validates our hypothesis that directly learning the global latent graphs is daunting and might lead to performance drops. As illustrated in Figure 7.4b,  $\mathcal{L}_{\text{ELBO}}^{\text{global}}$  converged faster and reached lower values when optimized jointly with  $\mathcal{L}_{\text{ELBO}}^{\text{local}}$ . (4) Then, we ablated the global ELBO loss and only learned the local latent graphs, leading to performance increases compared to the previous versions. This underlines that learning the local latent graphs is less sensitive to  $\mathcal{L}_{\text{ELBO}}^{\text{global}}$  than learning the global latent graphs is to  $\mathcal{L}_{\text{ELBO}}^{\text{local}}$  as can be seen in Figure 7.4c. (5) Finally, we evaluated a version that has a comparable computational complexity to our full model but uses random latent graphs instead of learning them. As can be seen from Figure 7.4b, Figure 7.4c, and the last row of Table 7.6, both ELBO losses remained constant, and the model scores the worst results among all ablated versions empirically showcasing the importance of our latent graph learning approach.

### Latent Graph Learning

Lastly, we considered two additional ablations of  $\text{MST}_{\text{MIXER}}$ . Specifically, we first ablated the multimodal conditioning (MMC) of Equation 7.5 and learned the local latent graphs of modality  $i$  based only on its features  $\mathbf{X}_i$ . This reduces Equation 7.7 to

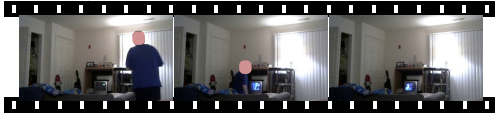
$$\mathbf{A}_i = \frac{1}{2}\tilde{\mathbf{A}}_i + \frac{1}{2}(\mathbf{A}'_i + \mathbf{A}''_i). \quad (7.13)$$

Then, we trained a version without the initialization bias (IB) of Equation 7.7. As can be seen in Table 7.7, MMC is essential for high performance. Without it  $\text{MST}_{\text{MIXER}}$  achieves the lowest performance across all metrics. The same applies to IB since not incorporating  $\tilde{\mathbf{A}}_i$  and only using the posterior approximation impeded performance across all evaluation metrics.

### 7.3.5 Qualitative Results

Finally, in Figure 7.5 we give a qualitative comparison of  $\text{MST}_{\text{MIXER}}$  with different ablated versions on response generation and global latent graph inference: Our full model manages to accurately answer

## Video (id = C2IRU)



## Dialog History

C: A person walks into the room with a blanket.  
 He lowers down and throws the blanket onto  
 a couch while he turns on the television.  
 He stands up and walks away .

Q<sub>1</sub>: How many people are there ?

A<sub>1</sub>: There is just one person.  
 ⋮

## Question

Q<sub>10</sub>: So the last thing that happens is the man leaves?

## Predictions

MST<sub>MIXER</sub>: Yes that is the last thing he does (BLEU-4 = 69.14)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : He gets up and walks away (BLEU-4 = 1.69e-10)

MST<sub>MIXER</sub> (Rand): He stands up and walks away (BLEU-4 = 1.69e-10)

## Latent Global Graphs

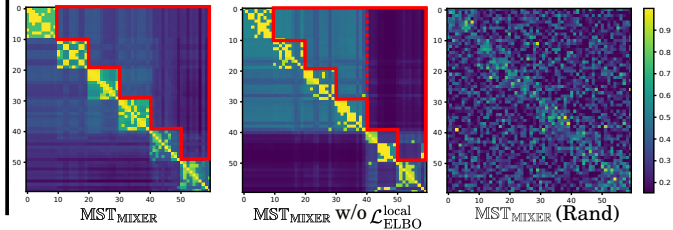


FIGURE 7.5: Qualitative comparison of different model ablations. on response generation and latent global graph inference of  $q_\phi$  obtained from the last encoder layer. The diagonal blocks (from upper left to lower right) correspond to  $\mathbf{V}_{rgb}$ ,  $\mathbf{V}_{flow}$ ,  $\mathbf{V}_{sam}$ ,  $\mathbf{A}_{vggish}$ ,  $\mathbf{T}_H$ , and  $\mathbf{T}_Q$ , respectively.

the question whereas both ablated version fail to generate reliable responses. Furthermore, we can see how our full model with multi-modal DST capabilities better captures the local interactions within each modality (structured diagonal blocks) as well as the global ones across modalities: Whereas the off-diagonal region (bordered in red) of the version “w/o  $\mathcal{L}_{\text{ELBO}}^{\text{local}}$ ” show a clear divide between the modalities (dotted line), the full model mitigates this by producing more homogeneous values indicating better inter-modal interactions. We provide more qualitative examples and failure cases in [Appendix C.3](#).

## 7.4 Limitations & Closing Remarks

In this chapter, we proposed a more sophisticated GNN-based DST scheme that alleviates some of the limitations of the previous two works. In contrast to `OLViT`, this approach is capable of handling the complexities of real-world datasets and is not only limited to synthetic and diagnostic benchmarks. `MSTMIXER` also improves on `VDGR` by proposing a more relaxed GNN tracking approach that simultaneously learns graph topologies alongside their node embeddings. Thus, it can be applied on input modalities for which the construction of ad hoc graphs is not feasible. Equipped with this novel DST scheme, our model managed to outperform strong baselines on *five* VL benchmarks.

That said, we can identify one major limitation of this work. Specifically, we evaluated `MSTMIXER` only on video-based benchmarks using pre-extracted features. Although this weakness is valid for our previous works, there is evidence [42] that combining data of different types (e.g. video and image) is beneficial for VLMs. We regard this training paradigm as highly relevant to our research topics as it enables the unification of two of the most important dialog tasks, i.e. video and visual/image dialog, which to this date evolved independently from one another. We address this limitation in the final chapter of this thesis.

# 8

## Unification of Multimodal Conversational Tasks

*This chapter is based on the following publication and includes extracts thereof without explicit quotation:*

**V<sup>2</sup>Dial: Unification of Video and Visual Dialog via Multimodal Experts**

Adnen Abdessaied, Anna Rohrbach, Marcus Rohrbach, Andreas Bulling

*In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA, 2025.*

## 8.1 Introduction & Contributions

In this chapter, we shift our attention to the task unifying multimodal conversational models and present  $\mathbb{V}^2\text{Dial}$ , a novel expert-based model specifically geared towards simultaneously handling image and video input data for multimodal dialog tasks.

Powered by the availability of large-scale training data [31, 137, 18] and advances in model design [177, 52, 162, 20, 126], the field of VL learning saw unprecedented success in recent years. However, current multimodal foundational models [126, 10, 168, 180, 115] still mainly focus on single-round tasks (e.g., VQA [15], VideoQA [190], video-text and text-video retrieval [191]). In contrast, the significantly more challenging conversational tasks, such as visual [48] and video dialog [9], received considerably less attention. Furthermore, methods for these different tasks have advanced independently of each other despite the apparent structural similarities between them. Whereas visual dialog models [5, 143, 32, 33, 182] have been primarily trained to rank a list of candidate answers using a NSP head, video dialog models [4, 102, 151, 8, 199, 41, 43] are trained to autoregressively predict the next answer token.

To this end, we propose to unify both tasks using a single model that for the first time jointly learns the spatial and temporal features of images and videos by routing them through dedicated experts and aligns them using matching and contrastive learning techniques. Furthermore, we systemically study the domain shift between the two tasks by investigating whether and to what extent these seemingly related tasks can mutually benefit from their respective training data.

Extensive evaluations on the widely used video and visual dialog datasets of AVSD [9] and VisDial [48] show that our model achieves new state-of-the-art results across *four* benchmarks both in zero-shot and fine-tuning settings.

## 8.2 Method

### 8.2.1 Joint Problem Formulation

We use a fully generative formulation to unify both video and visual dialog as discussed in [Chapter 3.2.3](#). In the rest, we use the term *context* to refer to the concatenation of the history  $\mathcal{H}_t$  and the current question  $\mathcal{Q}_t$ .

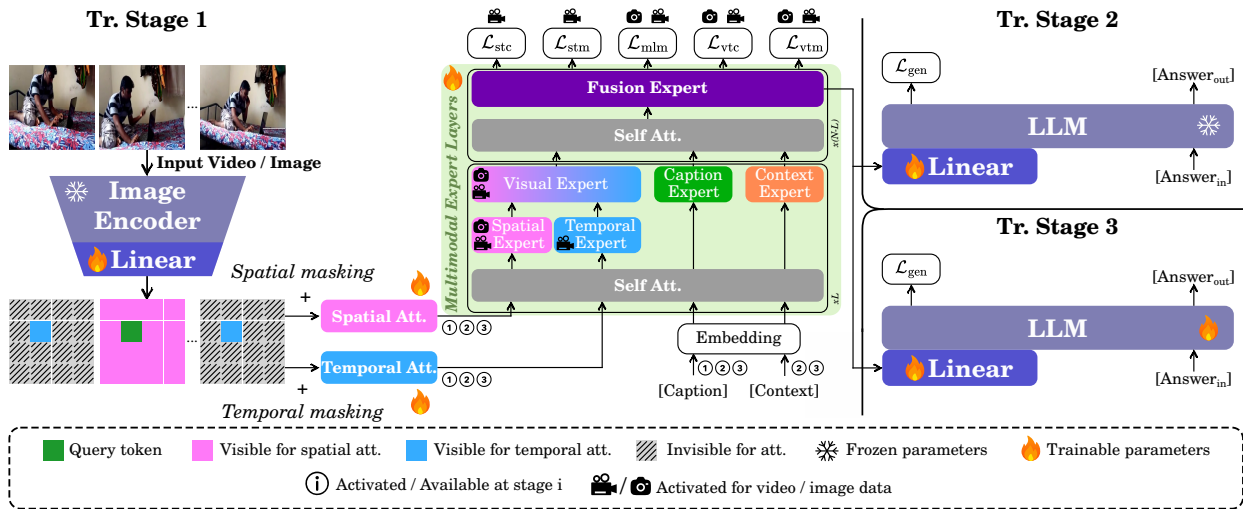


FIGURE 8.1: Architectural overview of V<sup>2</sup>Dial. We adopt a training strategy composed of three stages. *First*, we only train the multimodal expert layers using spatial-temporal and video/image text matching losses ( $\mathcal{L}_{stm}$ ,  $\mathcal{L}_{vtm}$ ), spatial-temporal and video/image contrastive learning losses ( $\mathcal{L}_{stc}$ ,  $\mathcal{L}_{vtc}$ ), and masked language modeling loss ( $\mathcal{L}_{mmlm}$ ). *Second*, we couple the expert layers with a frozen pre-trained LLM and fine-tune them end-to-end using a generative loss  $\mathcal{L}_{gen}$  to align their hidden representations. *Finally*, we additionally fine-tune the LLM weights on the downstream datasets. Each expert is a FFN composed of two fully connected layers.

### 8.2.2 $V^2$ Dial

#### Overview

As can be seen from [Figure 8.1](#), our model takes an image/video  $\mathcal{V} \in \mathbb{R}^{F \times 3 \times H \times W}$  as input, where  $F$  is the number of frames and is set to *one* for images, and  $(H, W)$  is the resized resolution. Then it processes every frame using a pre-trained EVA-CLIP [173] Image Encoder and concatenates every four spatially adjacent visual patches into a single one<sup>1</sup> similar to [17, 36]. Thereafter, a linear layer maps each visual token into a lower dimensional vector  $\mathbf{v}$  of dimension  $D$  to obtain

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1^1 & \mathbf{v}_1^2 & \cdots & \mathbf{v}_1^F \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_P^1 & \mathbf{v}_P^2 & \cdots & \mathbf{v}_P^F \end{bmatrix} \in \mathbb{R}^{F \times P \times D}, \quad (8.1)$$

where  $P = \frac{1}{4} \frac{H \times W}{14^2}$  and  $D$  denote the visual input length and the joint hidden dimension, respectively. In stark contrast to previous works [42, 22] that performed spatial and temporal attention in series, our model *separately* performs these operations using the masks  $\mathbf{M}^{\text{spa}}$  and  $\mathbf{M}^{\text{tmp}}$  as shown in [Figure 8.1](#) on the visual features  $\mathbf{V}$  to obtain  $\mathbf{V}^{\text{spa}}$  and  $\mathbf{V}^{\text{tmp}}$ :

$$\mathbf{V}^{\text{spa}} = \text{SA}(\mathbf{V}, \mathbf{M}^{\text{spa}}) \in \mathbb{R}^{(FP) \times D} \quad (8.2)$$

$$\mathbf{V}^{\text{tmp}} = \text{SA}(\mathbf{V}, \mathbf{M}^{\text{tmp}}) \in \mathbb{R}^{(FP) \times D} \quad (8.3)$$

$$\mathbf{M}_{m,n}^{\text{spa}}(\mathbf{v}_i^j) = \delta_{nj}, \quad \mathbf{M}_{m,n}^{\text{tmp}}(\mathbf{v}_i^j) = \delta_{mi}, \quad (8.4)$$

where SA and  $\delta$  denote self-attention and Kronecker delta, respectively. Subsequently, the textual input in the form of a caption and a context is processed by an embedding layer to obtain  $\mathbf{T}^{\text{cap}/\text{ctx}} \in \mathbb{R}^{N_{\text{cap}/\text{ctx}} \times D}$  where  $N_{\text{cap}}$  and  $N_{\text{ctx}}$  are the respective lengths of the caption and context. These visual and textual features form the initial input to the multimodal expert layers which are pre-trained using a combination of matching, contrastive learning, and masked language modeling losses. Finally, they are coupled with a pre-trained LLM and are fine-tuned end-to-end using a generative loss.

<sup>1</sup>Before patch concatenation, every frame is encoded with a tensor of shape  $(256 \times 1408)$ . Thereafter, its shape becomes  $(64 \times 5632)$ .

### Multimodal Expert Layers

These consist of  $N$  layers of stacked MSA with LN, and *several* modality-specific and *one* modality-agnostic FFN that we refer to as *experts*. As shown in [Figure 8.1](#), we propose to use a set of *six* experts denoted as  $\{\mathcal{E}_*\}$ : *three* of which are vision-specific and *two* are language-specific and are activated in the first  $L$  layers. The *remaining* expert  $\mathcal{E}_{\text{fus}}$  is the fusion expert and is only activated in the last  $(N - L)$  layers and operates on the concatenation of all available modalities ([Equation 8.8](#)). To the best of our knowledge, we propose for the first time to learn the spatial and temporal features using dedicated experts (i.e., the spatial  $\mathcal{E}_{\text{spa}}$  and temporal  $\mathcal{E}_{\text{tmp}}$  experts, respectively) as shown in [Equation 8.10](#). This allows our model to unify video and visual dialog by jointly learning from image and video data. The visual expert  $\mathcal{E}_{\text{vis}}$  operates on top of the concatenation of  $\mathcal{E}_{\text{spa}}$  and  $\mathcal{E}_{\text{tmp}}$  to learn a joint spatial-temporal video representation ([Equation 8.9](#)). Similarly, the textual experts  $\mathcal{E}_{\text{cap}}$  and  $\mathcal{E}_{\text{ctx}}$  operate on the caption and context embeddings  $\mathbf{T}^{\text{cap}}$  and  $\mathbf{T}^{\text{ctx}}$  ([Equation 8.11](#)). As seen in [Table 8.1](#), the availability of the multimodal features depends on the visual input type (i.e., videos vs images) and the training stage. However, without the loss of generality, we can formulate the multimodal expert operations as:

$$\mathbf{X}_0 = [\mathbf{V}^{\text{spa}}, \mathbf{V}^{\text{tmp}}, \mathbf{T}^{\text{cap}}, \mathbf{T}^{\text{ctx}}], \quad (8.5)$$

$$\tilde{\mathbf{X}}_l = [\tilde{\mathbf{V}}_l^{\text{spa}}, \tilde{\mathbf{V}}_l^{\text{tmp}}, \tilde{\mathbf{T}}_l^{\text{cap}}, \tilde{\mathbf{T}}_l^{\text{ctx}}] \quad (8.6)$$

$$= \text{MSA}(\mathbf{X}_{l-1}) + \mathbf{X}_{l-1} \quad (8.7)$$

$$\mathbf{X}_l = \begin{cases} [\mathbf{V}_l^{\text{vis}}, \mathbf{T}_l^{\text{cap}}, \mathbf{T}_l^{\text{ctx}}] & \text{if } 1 \leq l \leq L \\ \mathcal{E}_{\text{fus}}(\tilde{\mathbf{X}}_l) + \tilde{\mathbf{X}}_l & \text{if } L < l \leq N \end{cases}, \quad (8.8)$$

$$\mathbf{V}_l^{\text{vis}} = \mathcal{E}_{\text{vis}}(\tilde{\mathbf{V}}_l^{\text{vis}}) + \tilde{\mathbf{V}}_l^{\text{vis}}, \quad \tilde{\mathbf{V}}_l^{\text{vis}} := [\mathbf{V}_l^{\text{spa}}, \mathbf{V}_l^{\text{tmp}}], \quad (8.9)$$

$$\mathbf{V}_l^{\text{spa}} = \mathcal{E}_{\text{spa}}(\tilde{\mathbf{V}}_l^{\text{spa}}) + \tilde{\mathbf{V}}_l^{\text{spa}}, \quad \mathbf{V}_l^{\text{tmp}} = \mathcal{E}_{\text{tmp}}(\tilde{\mathbf{V}}_l^{\text{tmp}}) + \tilde{\mathbf{V}}_l^{\text{tmp}}, \quad (8.10)$$

$$\mathbf{T}_l^{\text{cap}} = \mathcal{E}_{\text{cap}}(\tilde{\mathbf{T}}_l^{\text{cap}}) + \tilde{\mathbf{T}}_l^{\text{cap}}, \quad \mathbf{T}_l^{\text{ctx}} = \mathcal{E}_{\text{ctx}}(\tilde{\mathbf{T}}_l^{\text{ctx}}) + \tilde{\mathbf{T}}_l^{\text{ctx}}. \quad (8.11)$$

When dealing with images and non-dialog data, we drop the temporal features  $\mathbf{V}_l^{\text{tmp}}$  and the context features  $\mathbf{T}_l^{\text{cap}}$  from the previous equations and deactivate their respective experts.

	Tr. Stage ①	Tr. Stage ②	Tr. Stage ③
<b>Videos</b> 📺	$\mathbf{V}^{\text{spa}}, \mathbf{V}^{\text{tmp}}, \mathbf{T}^{\text{cap}}$	$\mathbf{V}^{\text{spa}}, \mathbf{V}^{\text{tmp}}, \mathbf{T}^{\text{cap}}, \mathbf{T}^{\text{ctx}}$	$\mathbf{V}^{\text{spa}}, \mathbf{V}^{\text{tmp}}, \mathbf{T}^{\text{cap}}, \mathbf{T}^{\text{ctx}}$
<b>Images</b> 📷	$\mathbf{V}^{\text{spa}}, \mathbf{T}^{\text{cap}}$	-	$\mathbf{V}^{\text{spa}}, \mathbf{T}^{\text{cap}}, \mathbf{T}^{\text{ctx}}$

TABLE 8.1: Overview of the available features for each training stage and visual input type.

## 8.2.3 Training

### Stage 1

In the first stage, we only pre-train the multimodal expert layers, the vision encoder linear layer, and the spatial-temporal attention modules. Since we are the first to suggest learning the spatial and temporal features of videos and images using dedicated experts, we propose to train our model using Spatial-Temporal Contrastive Learning (STC) and Spatial-Temporal Matching (STM). In addition, we use the MLM, Vision-Text Contrastive Learning (VTC), and Vision-Text Matching (VTM) similar to [42, 115, 116]. In our context, vision can refer to either a video or an image depending on the dataset.

**Spatial-Temporal Contrastive Learning** aims to better align the spatial and temporal features of video data. To this end, we use output features of the last multi-modal expert layer<sup>2</sup> and learn a cosine similarity function

$$s(\mathbf{V}^{\text{spa}}, \mathbf{V}^{\text{tmp}}) = \Theta_{\text{spa}}(\mathbf{V}^{\text{spa}})^\top \Theta_{\text{tmp}}(\mathbf{V}^{\text{tmp}}), \quad (8.12)$$

so that aligned spatial-temporal features result in higher similarity scores, where  $\Theta_*$  are linear layers that map the features to a normalized lower dimensional vector space. Then, given spatial and temporal feature pairs, we compute the softmax normalized spatial-to-temporal and temporal-to-spatial similarities as

$$p_i^{\text{s2t}}(\mathbf{V}^{\text{spa}}) = \frac{\exp(\tilde{s}(\mathbf{V}^{\text{spa}}, \mathbf{V}_i^{\text{tmp}})/\tau)}{\sum_{k=1}^K \exp(\tilde{s}(\mathbf{V}^{\text{spa}}, \mathbf{V}_k^{\text{tmp}})/\tau)}, \quad (8.13)$$

<sup>2</sup>We drop the layer index for clarity.

$$p_i^{t2s}(\mathbf{V}^{\text{tmp}}) = \frac{\exp(\tilde{s}(\mathbf{V}^{\text{tmp}}, \mathbf{V}_i^{\text{spa}})/\tau)}{\sum_{k=1}^K \exp(\tilde{s}(\mathbf{V}^{\text{tmp}}, \mathbf{V}_k^{\text{spa}})/\tau)}, \quad (8.14)$$

where  $\tau$  is learnable temperature parameters, and  $\tilde{s}$  is the maximum value of  $s$  as in [115]. Finally, we can compute the loss as the CE between  $\mathbf{p}$  and  $\mathbf{y}$ :

$$\mathcal{L}_{\text{stc}} = \frac{1}{2} \mathbb{E}_{(\mathbf{v}^{\text{spa}}, \mathbf{v}^{\text{tmp}})} \left[ \text{CE}(\mathbf{y}^{\text{s2t}}, \mathbf{p}^{\text{s2t}}) + \text{CE}(\mathbf{y}^{\text{t2s}}, \mathbf{p}^{\text{t2s}}) \right], \quad (8.15)$$

where  $\mathbf{y}^{\text{s2t}}$  and  $\mathbf{y}^{\text{t2s}}$  are the golden one-hot similarities.

**Spatial-Temporal Matching** complements STC and teaches the model to distinguish between positive and negative spatial-temporal feature pairs. Specifically, a matched feature pair originates from the same video, whereas an unmatched pair is constructed using negative sampling from a different video. We use a classification token as a proxy of the joint spatial-temporal representations to learn a binary classification problem using the STM loss

$$\mathcal{L}_{\text{stm}} = \mathbb{E}_{(\mathbf{v}^{\text{spa}}, \mathbf{v}^{\text{tmp}})} \left[ \text{CE}(\mathbf{y}^{\text{stm}}, \mathbf{p}^{\text{stm}}) \right], \quad (8.16)$$

where  $\mathbf{p}^{\text{stm}}$  and  $\mathbf{y}^{\text{stm}}$  are the predicted and the ground-truth two-class probabilities, respectively.

We provide more details about the remaining established objectives we used (i.e. MLM, VTC, VTM) in [Appendix A.7](#).

## Stages 2 & 3

In the subsequent stages, we couple the multimodal expert layers with a pre-trained Flan-T5<sub>large</sub> [45] via a linear layer. Specifically, Stage 2 aims to align the hidden states of the proposed layers with those of the pre-trained LLM. To this end, we keep the LLM weights frozen and train the whole architecture end-to-end using the generative loss (i.e., next token prediction) on large scale video dialog data<sup>3</sup>, i.e.,

$$\mathcal{L}_{\text{gen}} = \mathbb{E}_{\mathbf{x}^{\text{gen}}} \left[ \text{CE}(\mathbf{y}_{\rightarrow}^{\text{gen}}, \mathbf{p}^{\text{gen}}) \right], \quad (8.17)$$

$$\mathbf{x}^{\text{gen}} = \Theta_{\text{gen}} \left( \text{LLM}_{\text{dec}}([\mathbf{X}^{\text{enc}}, \mathbf{T}^{\text{ans}}]) \right), \quad (8.18)$$

<sup>3</sup>The weights of  $\mathcal{E}_{\text{ctx}}$  are initialized with those of  $\mathcal{E}_{\text{cap}}$  from Stage 1.

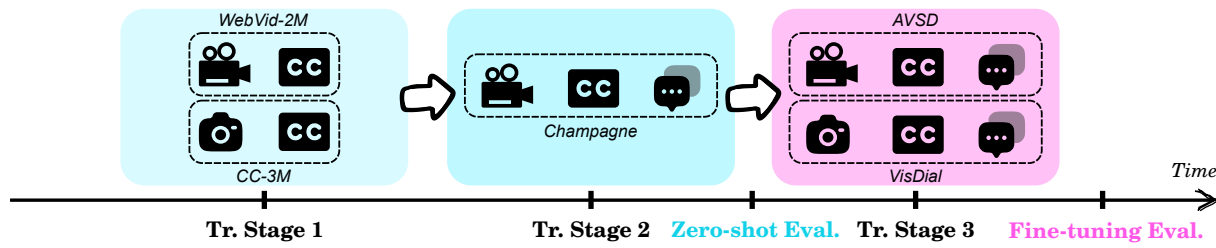


FIGURE 8.2: **Overview of the training and evaluation pipeline of V<sup>2</sup>Dial.** We show the different datasets used to train our model at each stage. Evaluations are conducted on the most popular video and visual dialog datasets of AVSD and VisDial, respectively. (🎥 = video data, 📷 = image data, 🗣️ = closed / visual captioning data, 💬 = dialog data).

where  $\mathbf{X}^{\text{enc}}$ ,  $\mathbf{T}^{\text{ans}}$  and  $\Theta_{\text{gen}}$  are the LLM encoder output, the answer token embeddings, and a linear layer that maps the features to the vocabulary space, respectively.  $\mathbf{y}_{\rightarrow}^{\text{gen}}$  and  $\mathbf{p}^{\text{gen}}$  denote the right-shifted ground-truth answer tokens and the predicted text token probabilities. Finally, in Stage 3, we unfreeze the LLM weights and fine-tune our model end-to-end on the downstream tasks of video and visual dialog using the same generative loss of Equation 8.17.

## 8.3 Experiments

### 8.3.1 Datasets

As shown in Figure 8.2, we simultaneously use the video and image captioning datasets of WebVid-2M [18] and CC-3M [164] to pre-train the multimodal expert layers in Stage 1. Then in the *second* stage, we use 25% of the recent large-scale video dialog dataset Champagne [70] before performing *zero-shot* evaluation on the widely used video and visual dialog datasets of AVSD [9] and VisDial [48], respectively. Finally, in the *third* stage, we perform a domain shift evaluation based on different combinations of AVSD and VisDial to quantify whether and to what extent these seemingly similar benchmarks benefit from each other in both *zero-shot* and *fine-tuning* evaluation settings.

### 8.3.2 Evaluation Metrics

We use the established official metrics for each dataset to fairly benchmark  $\mathbb{V}^2\text{Dial}$  with previous works as discussed in Chapter 3.3. However, since we are jointly tackling both tasks with a fully generative model, we propose to rank the VisDial candidate answers by means of cosine similarity with respect to the generated answer using the embeddings of a pre-trained sentence transformer. In contrast, previous works [34, 33, 32, 181, 95] ranked the candidate answers used NSP or perplexity values. We posit that our approach is more natural, caters to the current advances in generative models, and appropriately captures the semantic similarities between the generated and the candidate answers. In addition, it allows for a seamless unification of AVSD and VisDial without any training or architectural modifications. As shown in Figure 8.3, our proposed adaptation does *not* alter the computation of the sparse metrics itself and *only rethinks* the ranking of the candidate answers allowing for a fair comparison with previous works.

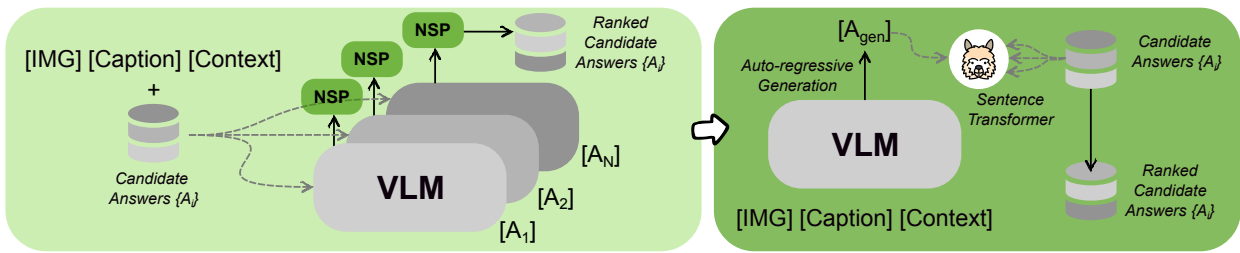


FIGURE 8.3: Instead of training a dedicated NSP head, we propose a ranking scheme based on the cosine similarity of the candidate answers’ embeddings with the respect to those of the generated ones. We used RoBERTa<sub>large</sub> [127] and OpenAI Text Embedding-3 to generate these embeddings.

Model	AVSD-DSTC10							AVSD-DSTC8							AVSD-DSTC7						
	B-1	B-2	B-3	B-4	M	R	C	B-1	B-2	B-3	B-4	M	R	C	B-1	B-2	B-3	B-4	M	R	C
♦MoE-LLAVA [123]	35.8	18.9	10.1	5.9	15.4	27.1	12.8	39.8	23.9	15.2	10.1	18.7	32.2	23.7	44.7	29.1	19.6	13.8	<u>21.8</u>	37.3	33.2
♦MiniGPT4-video [17]	37.9	19.9	11.3	6.8	16.2	28.7	17.7	34.8	17.6	9.7	5.8	15.8	26.3	13.3	37.8	21.2	12.7	8.2	18.4	30.2	17.7
♦Video-ChatGPT [133]	24.5	14.7	8.8	5.4	16.7	25.2	3.9	25.5	16.0	10.1	6.4	18.4	27.1	9.1	28.5	18.5	11.8	7.6	20.4	32.1	19.1
♦MST <sub>MIXER</sub> [4]	0.1	0.0	0.0	0.0	3.1	6.8	3.0	0.2	0.1	0.1	0.0	3.3	7.1	4.3	0.2	0.1	0.0	0.0	3.4	6.9	4.6
♦VideoChat2 [117]	<b>42.5</b>	<u>25.9</u>	<u>16.0</u>	<u>10.3</u>	<u>18.7</u>	<u>33.1</u>	<u>25.4</u>	<u>43.9</u>	<u>28.1</u>	<u>18.5</u>	<u>12.6</u>	<b>20.8</b>	<u>34.5</u>	<u>29.2</u>	<u>46.7</u>	<u>31.1</u>	<u>20.9</u>	<u>14.4</u>	<b>22.9</b>	<u>37.6</u>	<u>31.4</u>
V <sup>2</sup> Dial	<b>54.6</b>	<b>34.8</b>	<b>24.0</b>	<b>17.2</b>	<b>19.7</b>	<b>38.3</b>	<b>53.8</b>	<b>53.2</b>	<b>33.8</b>	<b>23.5</b>	<b>16.7</b>	<u>18.8</u>	<b>37.7</b>	<b>49.7</b>	<b>55.5</b>	<b>36.7</b>	<b>26.2</b>	<b>18.7</b>	20.0	<b>39.2</b>	<b>50.8</b>

TABLE 8.2: Zero-shot performance comparison on AVSD-DSTC10, AVSD-DSTC8 and AVSD-DSTC7. Best and second-best performances are in **bold** and underlined, respectively. ♦ indicates that we evaluated the model. (B-n = BLEU-n, M = METEOR, R = ROUGE-L, C = CIDEr).

Model	Sentence Embeddings	R@1	R@5	R@10	MRR	NDCG
FROMAGe [95]	n/a	17.6	20.1	25.1	22.0	16.5
ESPER [204]		14.6	–	–	25.7	22.3
Champagne [70]		–	–	–	–	25.5
♦MoE-LLAVA [123]	RoBERTa <sub>large</sub>	10.6	25.4	36.4	19.6	26.7
♦MiniGPT-video [17]		7.4	17.4	26.5	14.6	23.2
♦Video-ChatGPT [133]		10.0	22.5	31.5	18.1	24.8
♦MST <sub>MIXER</sub> [4]		<u>18.2</u>	22.1	25.7	21.9	24.6
♦VideoChat2 [117]		12.7	<u>29.0</u>	<u>39.9</u>	22.3	30.9
V <sup>2</sup> Dial	RoBERTa <sub>large</sub>	<u>20.0</u>	<u>30.2</u>	<u>39.3</u>	<u>26.9</u>	<b>33.3</b>
	OpenAI TE-3 <sub>large</sub>	<b>22.1</b>	<b>41.2</b>	<b>48.1</b>	<b>32.7</b>	<u>32.0</u>

TABLE 8.3: Zero-shot performance comparison in the generative setting on the VisDial v1.0 val split. OpenAI TE-3 = OpenAI Text Embedding-3.

### 8.3.3 Zero-shot Evaluation

**AVSD.** We first assessed V<sup>2</sup>Dial in a zero-shot setting on AVSD. This is in stark contrast to previous models that were exclusively evaluated in a fine-tuning setting. We instead advocate for complementing the fine-tuning evaluation with a zero-shot one, as it results in a more rigorous and challenging testbed for the proposed models. To this end, we establish; to the best of our knowledge; the *first* zero-shot benchmark comparison on AVSD comprised of recent capable Multimodal Large Language Models (MLLMs). As can be seen from Table 8.2, our model outperforms all baselines by a considerable margin across 6/7 metrics of AVSD-DSTC8 and AVSD-DSTC7. On the more recent and challenging<sup>4</sup> version of the benchmark (i.e., AVSD-DSTC10), V<sup>2</sup>Dial ranks first across all metrics. For instance, it more than doubles the CIDEr score compared to VideoChat2 [117].

**VisDial.** Additionally, we assessed the same model checkpoint on VisDial v1.0. As can be seen from Table 8.3, V<sup>2</sup>Dial managed to outperform previous models such as FROMAGe [95] by a considerable margin across all metrics of the dataset. In addition, it outperformed Champagne [70] that was trained on x4 more dialog data by 7.8 absolute NDCG points. Furthermore, our model outperformed the more recent MLLMs of the previous section on 4/5 metrics, underlining its capability of dealing with both video and image input data types. Finally, replacing the sentence embeddings generated by RoBERTa<sub>large</sub>

<sup>4</sup>AVSD-DSTC10 comes with no caption data.

Model	AVSD-DSTC10							AVSD-DSTC8							AVSD-DSTC7						
	B-1	B-2	B-3	B-4	M	R	C	B-1	B-2	B-3	B-4	M	R	C	B-1	B-2	B-3	B-4	M	R	C
PDC [102]	-	-	-	-	-	-	-	74.9	62.9	52.8	43.9	28.5	59.2	120.1	77.0	65.3	53.9	44.9	29.2	60.6	129.5
THAM [199]	-	-	-	-	-	-	-	76.4	64.1	53.8	45.5	30.1	61.0	130.4	77.8	65.4	54.9	46.8	30.8	61.9	133.5
DialogMCF [41]	69.3	55.6	45.0	36.9	24.9	53.6	91.2	75.6	63.3	53.2	44.9	29.3	60.1	125.3	77.7	65.3	54.7	45.7	30.6	61.3	135.2
*VideoLLAMA 2 [43]	50.2	35.0	24.9	18.1	21.8	42.8	57.5	53.3	39.0	29.1	22.2	24.8	46.3	74.0	56.2	41.1	30.7	23.2	26.4	48.5	79.2
MSTMIXER [4]	<u>69.7</u>	<u>57.1</u>	<u>47.2</u>	<u>39.5</u>	<u>25.1</u>	<u>54.0</u>	<u>96.9</u>	<b>77.1</b>	<b>65.6</b>	<u>55.7</u>	<u>47.1</u>	<u>30.2</u>	<u>61.8</u>	<u>133.6</u>	<u>78.4</u>	<u>66.0</u>	<u>55.8</u>	<u>47.1</u>	<u>31.0</u>	<u>62.0</u>	<u>136.5</u>
V <sup>2</sup> Dial	<b>70.7</b>	<b>58.2</b>	<b>48.2</b>	<b>40.3</b>	<b>26.0</b>	<b>55.4</b>	<b>103.3</b>	<u>76.8</u>	<u>65.5</u>	<b>55.8</b>	<b>47.5</b>	<b>30.4</b>	<b>62.1</b>	<b>135.7</b>	<b>78.9</b>	<b>66.5</b>	<b>56.1</b>	<b>47.4</b>	<b>31.2</b>	<b>62.3</b>	<b>139.8</b>

TABLE 8.4: Fine-tuning performance comparison on AVSD-DSTC10, AVSD-DSTC8 and AVSD-DSTC7. VideoLLAMA 2 [43] was trained on AVSD amongst other datasets. Additional model comparisons can be found in Table 7.1 and Table 7.2 of Chapter 7.

Model	Sentence Embeddings	R@1	R@5	R@10	MRR	NDCG
LTM1 [143]		40.4	<u>61.6</u>	<b>69.7</b>	50.7	<b>63.5</b>
LTM1-LG [34]		41.3	<u>61.6</u>	69.0	<u>51.3</u>	<b>63.2</b>
GoG [33]	n/a	41.2	<b>61.8</b>	<u>69.4</u>	<u>51.3</u>	62.6
UTC [32]		41.3	59.8	66.3	50.6	61.0
Champagne [70]		–	–	–	–	62.5
♣MST <sub>MIXER</sub> [4]	RoBERTa <sub>large</sub>	42.2	51.6	57.8	47.7	52.5
V <sup>2</sup> Dial	RoBERTa <sub>large</sub>	<u>44.2</u>	53.3	59.5	49.6	52.3
	OpenAI TE-3 <sub>large</sub>	<b>44.9</b>	58.6	65.1	<b>52.4</b>	52.0

TABLE 8.5: Fine-tuning performance comparison on the VisDial v1.0 val split. ♣ indicates that we trained and evaluated the model.

with those from OpenAI Text Embedding-3 improved the external ranking of the candidate answers and resulted in higher scores across all metrics, as can be seen in the last row of Table 8.3.

### 8.3.4 Fine-tuning Evaluation

**AVSD.** Similar to almost all previous works on AVSD, we assessed V<sup>2</sup>Dial in a fine-tuning setting on all *three* benchmarks of the dataset. As can be seen in Table 8.4, our model managed to maintain its competitiveness ahead of recent models and outperformed them on the latest and most challenging AVSD-DSTC10 benchmark across all evaluation metrics. For instance, it lifted CIDEr by over 6 absolute points compared to the second-best model. Furthermore, our model managed to maintain an on-par performance with the state of the art on AVSD-DSTC8 and AVSD-DSTC7. As shown in Table 8.4, V<sup>2</sup>Dial increased their respective CIDEr scores by more than 2 and 3 absolute points compared to the second-best model.

**VisDial.** Finally, we fine-tuned our model and MST<sub>MIXER</sub> [4] that had the closest AVSD performance on Visdial v1.0 using the same fully-generative approach. As can be seen from Table 8.5, V<sup>2</sup>Dial managed to outperform all previous models on the strictest metric of the dataset by achieving a R@1 score of 44.2. However, when using OpenAI Text Embedding-3 our model managed to increase the R@1 and MRR scores to 44.9 and 52.4, respectively, thereby setting new state-of-the-art results. As expected and due to the more challenging aspect of tackling VisDial as a fully generative task, our model performed slightly worse than the previous fine-tuned models on the remaining metrics of the dataset. However, when comparing our model with

MST<sub>MIXER</sub> that was trained using the same paradigm (i.e., the last penultimate two rows of Table 8.5), we can see that our model outperformed it across 4/5 metrics of the task and scored almost equally on NDCG.

### 8.3.5 Domain Shift Evaluation

#### Zero-shot setting

First, we fine-tuned our model’s checkpoint from Stage 2 on AVSD and zero-shot evaluated it on VisDial. As can be seen in the second section of Table 8.6, our model’s performance was lifted by a considerable margin across most metrics. In particular, the NDCG score improved by 9 absolute points compared to the results of Table 8.3. Then, we replicated the same experiment on AVSD after having fine-tuned the model on VisDial. Interestingly, our model’s performance deteriorated across all metrics of the benchmark. This behavior could be explained by the nature of both datasets. Whereas AVSD encourages the model to produce long and elaborate responses, VisDial teaches it to produce brief answers instead, which diminishes its performance on the language generation metrics. The qualitative examples of Figure 8.4 clearly illustrate this phenomenon on both datasets.

#### Fine-tuning Setting

We first experimented with a curriculum learning strategy where we used one dataset for pre-training before finally fine-tuning on the other. As can be seen in the last section of Table 8.6, this training paradigm resulted in performance drops on both datasets compared to Table 8.4 and Table 8.5 where the model was only trained on the data of the respective benchmark. This indicates that the weights of the converged model on one dataset do not offer a good initialization for training on the remaining one. Allowed by our model design that can jointly handle video and image input data, we finally fine-tuned one single model on both datasets simultaneously. As seen in the last row of Table 8.6, this resulted in the best joint performance of our model across the two datasets. Although the results on AVSD slightly dropped compared to Table 8.4, our model lifted its performance on VisDial by a considerable margin. This could largely be attributed to the same previous observation, as training on VisDial incentivizes

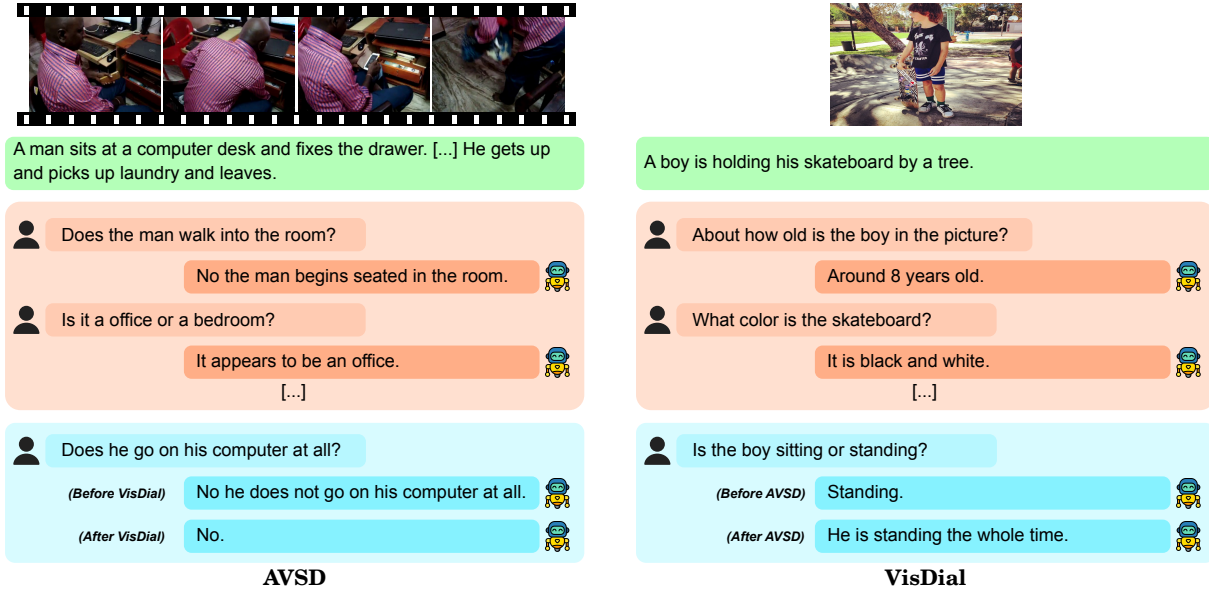


FIGURE 8.4: Zero-shot qualitative examples of  $V^2Dial$  before and after fine-tuning on VisDial and AVSD. The former teaches the model to answer question with brief responses whereas the latter teaches it to produce longer and more elaborate answers.

Fine-tuning data		AVSD-DSTC10				AVSD-DSTC8				AVSD-DSTC7				VisDial			
AVSD	VisDial	B-1	M	R	C	B-1	M	R	C	B-1	M	R	C	R@1	R@5	R@10	NDCG
$\times$	$\times$	<i>Zero-shot (from Table 8.2 and Table 8.3)</i>															
		54.6	19.7	38.3	53.8	53.2	18.8	37.7	49.7	55.5	20.0	39.2	50.8	20.0	30.2	39.3	33.3
$\checkmark$	$\times$	<i>Fine-tuning (from Table 8.4)</i>												<i>Zero-shot</i>			
		70.7	26.0	55.4	103.3	76.8	30.4	62.1	135.7	78.9	31.2	62.3	139.8	12.8	36.7	50.8	42.3
$\times$	$\checkmark$	<i>Zero-shot</i>												<i>Fine-tuning (from Table 8.5)</i>			
		11.5	6.8	20.1	14.6	11.5	7.3	20.7	20.9	7.9	6.2	17.4	18.2	44.2	53.3	59.5	52.3
		<i>Fine-tuning</i>															
$\checkmark$	$\rightarrow$	$\checkmark$	-	-	-	-	-	-	-	-	-	-	-	42.2	50.1	56.3	51.3
$\checkmark$	$\leftarrow$	$\checkmark$	69.6	25.7	55.0	100.5	75.9	29.8	61.4	132.1	77.6	30.4	61.5	134.5	-	-	-
$\checkmark$	$\&$	$\checkmark$	69.3	25.4	54.8	99.9	75.1	29.3	61.1	130.0	77.3	30.0	61.7	134.5	45.4	54.7	61.1

TABLE 8.6: Domain shift evaluation between the respective *most prominent* video and visual dialog datasets of AVSD and VisDial.  $\square \rightarrow \triangle$  means that the model was pre-trained on dataset  $\square$  before fine-tuning on dataset  $\triangle$ . RoBERTa<sub>large</sub> was used to generate the sentence embeddings for ranking.

Expert Swapping	AVSD-DSTC7				VisDial	
	B-1	M	R	C	R@1	NDCG
Original	78.9	31.2	62.3	139.8	44.2	52.3
<i>Swapping experts of the same modality (vision / language)</i>						
$\mathcal{E}_{\text{spa}} \leftrightarrow \mathcal{E}_{\text{tmp}}$	77.0	29.5	61.2	133.7	-	-
$\mathcal{E}_{\text{cap}} \leftrightarrow \mathcal{E}_{\text{ctx}}$	76.1	29.6	60.3	131.1	42.8	51.9
<i>Swapping experts of different modalities</i>						
$\mathcal{E}_{\text{spa}} \leftrightarrow \mathcal{E}_{\text{cap}}$	28.5	10.7	22.0	10.1	35.7	45.3
$\mathcal{E}_{\text{tmp}} \leftrightarrow \mathcal{E}_{\text{ctx}}$					-	-
$\mathcal{E}_{\text{spa}} \leftrightarrow \mathcal{E}_{\text{ctx}}$	34.4	12	25.4	11.7	32.4	42.9
$\mathcal{E}_{\text{tmp}} \leftrightarrow \mathcal{E}_{\text{cap}}$					-	-

TABLE 8.7: Expert swapping results.  $\mathcal{E}_{\square} \leftrightarrow \mathcal{E}_{\Delta}$  means that the  $\square$  features are *inadequately* routed at test time through  $\mathcal{E}_{\Delta}$  and vice versa. The other experts remain unchanged. RoBERTa<sub>large</sub> was used to generate the sentence embeddings for ranking.

our model to shorten its responses on AVSD. Additional qualitative examples can be found in [Appendix C.4](#).

### 8.3.6 Expert Swapping Experiment

In order to validate the specialization of each expert, we conducted a swapping experiment where we routed some features through *inadequate* experts. We first swapped experts of the same modality (i.e., experts operating on vision or language data). As shown in [Table 8.7](#), this resulted in performance drops across all metrics of both datasets, indicating that experts of the same modality are able to capture the semantic nuances of the data they specialize on. More interestingly, the performance of our model dropped more significantly when swapping experts of different modalities, as seen from the last section of [Table 8.7](#). This showcases their ability to adjust to the nature of the data they process and to capture its modality specific features.

### 8.3.7 Ablation Study

#### Effect of Pre-training Data

To assess the effectiveness of the pre-training data in the first two stages, we trained two versions of our model, where one was only pre-trained on Stage 1 using WebVid-2M & CC-3M and the other only on Stage 2 with a subset of Champagne. As can be seen from the middle section of [Table 8.8](#), our model witnessed a comparable

Model Ablations	AVSD-DSTC7				VisDial	
	B-1	M	R	C	R@1	NDCG
Full	78.9	31.2	62.3	139.8	44.2	52.3
w/o Tr. Stage 1	76.9	30.0	61.4	134.0	34.5	44.6
w/o Tr. Stage 2	77.8	30.7	61.7	134.7	32.6	44.1
w/o $\mathcal{L}_{\text{stc}}$ & $\mathcal{L}_{\text{stm}}$	77.2	29.9	61.1	133.2	33.1	44.6
w/o separate $\mathcal{E}_{\text{spa}}$ & $\mathcal{E}_{\text{tmp}}$	77.0	30.1	61.1	133.8	32.9	43.8
w/o experts $\{\mathcal{E}_*\}$	77.5	30.0	61.4	134.8	30.6	42.2

TABLE 8.8: Ablation results of  $V^2Dial$ . RoBERTa<sub>large</sub> was used to generate the sentence embeddings for ranking.

drop in performance compared to the full model. This underlines the equal importance of these proposed training stages to the joint down-stream performance on AVSD and VisDial. We did not conduct ablations using either WebVid-2M or CC-3M in Stage 1 as this was sufficiently explored by other recent works [42] that showed the benefit of pre-training on both image and video data.

### Effect of Pre-training Objectives & Model Design

To evaluate the effect of the newly introduced spatial-temporal objectives, we trained a version of our model without  $\mathcal{L}_{\text{stc}}$  and  $\mathcal{L}_{\text{stm}}$  in Stage 1 using the same schedule and training data as our full model. As shown in the *fourth* row of Table 8.8, this ablated version suffered from a drop in performance not only on AVSD but also on VisDial as well. This indicates that these losses improve not only the temporal capabilities of our model but also its spatial ones. Then, we trained a version that sequentially applies spatial and temporal attention, as in [42, 22]. Since this version does not have separate spatial temporal experts, we also omitted the previous two objectives. As seen in the penultimate row of Table 8.8, this version underperformed our full model on both datasets, showcasing the effectiveness of our approach. Finally, we trained a version without all the expert layers. As shown in the last row, its performance dropped compared to our full model and performed the worst on VisDial.

## 8.4 Limitations & Closing Remarks

We dedicated this chapter to researching the potential of unifying the multimodal conversational tasks of video and visual dialog using a

joint generative model. To this end, we presented  $\mathbb{V}^2\text{Dial}$ , a model that uses multimodal experts to be able to learn from both image and video data. As a core novelty of our approach, we proposed learning the spatial and temporal features of images and videos through dedicated experts. In addition, we studied *for the first time* the cross-learning benefits of the two prominent video and visual dialog datasets of AVSD and VisDial. Through extensive experiments, we were able to show the benefits of our approach on both datasets. Specifically,  $\mathbb{V}^2\text{Dial}$  managed to outperform recent strong baselines on both datasets in zero-shot and fine-tuning settings.

That said, these performance benefits are not to be seen in isolation of extensive amounts of training data (CC-3M, WebVid-2M, Champagne) and powerful GPU cards (Nvidia A100 80GB) needed for the training of our model. Furthermore, our results indicate a misalignment between AVSD and VisDial, which necessitates a rethinking of how we need to construct multimodal datasets in the future. We still believe in the potential of a unifying model that jointly learns from images and videos. However, future datasets should also follow the same paradigm and by design include data from both vision types instead of simply aggregating existing datasets.



# 9

## Conclusion

### 9.1 Summary of Findings

In this thesis, we mainly focused on a challenging class of tasks at the intersection of CV and NLP, namely visual/image dialog and video dialog. Specifically, we researched and contributed several novel ideas that advanced the field, which we summarize in the following paragraphs.

In [Chapter 4](#), we proposed `NSVD`, the first hybrid method to combine deep learning and symbolic program execution for DST. Our method outperformed state-of-the-art connectionist baselines on CLEVR-Dialog by achieving near-perfect test accuracies. Furthermore, `NSVD` has a higher NFFR, is more robust to incomplete dialog histories, and generalizes better to OOD scenarios, e.g. longer dialogs and unseen question types and scenes. Additional evaluations showed that our models was able to generalize to other scene domains, namely Minecraft-dialog. These performance improvements are not to be seen in isolation of the strict supervision our models have, as

they require a fine-tuning of a Mask-RCNN in addition to the training of the program parsers. Furthermore, NSVD faces generalizability limitations on datasets with unknown DSLs.

Thus, in the subsequent chapters, we proposed latent, differentiable, and end-to-end DST approaches that can be seamlessly integrated into pre-trained LLMs. Specifically, in [Chapter 5](#), we introduced OLViT, a model operating over a novel two-stream multimodal attention-based dialog state tracker that jointly learns visual object representations and textual embeddings. Through extensive experiments on two challenging datasets, we demonstrated significant improvements over strong baselines not only on DVD [108] where our model has to predict an answer from a pool of candidate responses but also on SIMMC [138] where it needs to generate the answer tokens.

To be able to deal with the challenges of real-world datasets, we proposed VDGR in [Chapter 6](#), a novel visual dialog model that performs latent DST using multimodal GNNs. Specifically, VDGR alternates between multimodal graphs and LLM layers, and augments the hidden states of the latter with the fine-grained features obtained by the former. VDGR propagated information from one modality graph to another in a cascaded manner using hub-nodes that link to all other nodes within each modality, effectively alleviating the lack of intermodal context. Extensive analyzes underlined its effectiveness, while experiments on four challenging visual dialog datasets (VisDial v1.0, VisDial v0.9, VisDialConv, and VisPro) demonstrated its superior performance over existing methods.

Then in [Chapter 7](#), we proposed MST<sub>MIXER</sub>, a universal GNN-based DST approach that, in contrast to VDGR, jointly learns the graph structures and their node embeddings. MST<sub>MIXER</sub> first identifies the most influential constituents at different semantic levels (e.g., across modalities and encoder layers). Then, it relies on a two-stage divide and conquer approach to infer the missing underlying structure of the mix of all modalities and leverages it to augment the hidden states of the backbone LLM using GNNs. Extensive ablations experiments and evaluations on *five* VL benchmarks, (i.e. AVSD-DSTC7, AVSD-DSTC8, AVSD-DSTC10, SIMMC, and NEXT-QA) showed the effectiveness and generalization capabilities of our approach.

Finally, in [Chapter 8](#), we turned our attention to unifying the divergent works on visual/image dialog and video dialog by presenting V<sup>2</sup>Dial. Our model can jointly tackle video and visual conversational

tasks using a multimodal expert-based approach that *for the first time* disentangles the learning of the spatial and temporal features of images and videos using separate experts. In addition, we conducted the *first* domain shift evaluation of the prominent visual and video dialog datasets (i.e. VisDial and AVSD) by providing insight on how to optimally leverage their respective training data. Extensive evaluation on these two datasets showed that our model achieves new state-of-the-art performance both in zero-shot and fine-tuning settings.

All in all, we believe that this thesis advanced the fields of VL learning in general and multimodal conversational systems in particular. We provided valuable contributions to the topics of multimodal DST and identified ways how to unify and take advantage of the available multimodal conversational data.

## 9.2 Future Work

Although we managed to push the state of the art in a multitude of VL domains, we can still identify some challenges that need further investigation. Thus, in the following, we provide an outlook for future work and discuss interesting research directions moving forward.

### Neuro-Symbolic Methods for Real-World Dialog

As discussed in [Chapter 4](#), our neuro-symbolic method reached almost perfect performance on both CLEVR-Dialog [99] and Minecraft-Dialog [1] which come with a limited set of operations and predefined object characteristics, i.e. DSL (see [Table A.1](#)). However, in real-world scenarios, we lack such a closed and pre-specified DSL or even a consistent object ontology. Thus, it becomes challenging to automatically induce or learn compositional reasoning structures (i.e. logical forms or programs needed for neuro-symbolic methods) from unconstrained language and visual groundings. This is one of the main reasons why most real-world multimodal dialog datasets such as VisDial [48] and AVSD [9] lack symbolic annotations or reasoning traces that we can leverage to both supervise and evaluate neuro-symbolic models. We see multiple interesting research directions moving forward that could alleviate these shortcomings and demystify real-world tasks for neuro-symbolic models.

First, since applying program synthesis from input-output examples is not feasible in our setting because we lack a proper specification of a generalizable DSL, it would be interesting to pursue the idea of

latent program execution for our multimodal setting. Although Chen et al. [38] showed promising results when dealing with standard programming languages like C, it is still an open challenge how such an approach could align multiple input modalities in a joint latent space for meaningful program synthesis and execution. This could be seen as a *continuous* equivalent to visual programming agents [65, 175] that execute programs in a *discrete* space.

Second, since we lack ground-truth programs for real-world scenarios that we could leverage to infer a proper DSL, an interesting research direction would be investigating how we can generate reliable programs at scale. We believe that visual program distillation [76] offers a promising direction moving forward as it showed potential on multimodal single-round tasks; mainly VQA. That said, it is still unknown how such approaches could be extended to multi-round scenarios where reasoning over dynamic visual inputs is essential.

### **Modern Evaluation of Multimodal Dialog Tasks**

When dealing with established multimodal datasets such as VisDial [48] and AVSD [9], we have been primarily measuring the performance of our models using the official metrics of these established benchmarks such as BLEU, ROUGE, METERO, and CIDEr (see Table 3.1). However, these automatic language generation metrics have been known to suffer from fundamental limitations. Specifically, since they are based on n-gram overlap with reference texts, they poorly correlate with human judgment and thus penalize valid paraphrases or creative answers. Furthermore, they suffer from reference bias as they compare the generated output to one of few human-written references encouraging generic responses such as "*I don't know*" and "*It is not clear*". Therefore, as multimodal models continue to evolve, the need for more reliable and suitable evaluation paradigms has become critically important.

We posit that using an LLM as a judge could alleviate many of the aforementioned limitations. This approach has been shown to correlate better with humans [211] as frontier LLMs have better semantic understanding and contextual reasoning making them able to recognize nuances and paraphrases. Furthermore, they are highly customizable using detailed and rich prompts making them flexible for task-specific evaluations.

However, this approach is not without its own challenges that we need to address in future research. LLMs themselves can hallucinate and are prone to multiple biases such as prompt wording, misinformation, gender, authority, etc [35]. Furthermore, evaluations may suffer from high variance across model judges [21] which make it even more challenging to apply them on multimodal multi-turn tasks such as visual and video dialog.

### **Multimodal Sparse Mixture-of-Experts**

Our novel approach presented in [Chapter 8](#) already validated the effectiveness of using multimodal experts to unify multiple tasks by being able to process and learn from different input modalities. In our approach, we proposed dense multimodal experts in the form of FFNs where each expert processes a specific type of input data. Another equally valid and promising line of work [217, 123, 142] consists in training sparse modality agnostic MoEs using soft-routers. Specifically, each expert can process visual and textual features without any distinction. Despite their initial success, we posit that these models could be further improved when equipped with modality-aware MoEs resulting in a hybrid approach that could take advantage of the two previously mentioned research directions of multimodal dense experts and sparse modality agnostic experts.

### **Unified Vision-and-Language Datasets**

In [Chapter 8](#), we made progress in unifying visual and video dialog by proposing a model able to jointly learn from image and video input data. We strongly believe that this training paradigm has great potential to be unlocked and should be further investigated in future work. Adopting this approach is becoming more and more urgent with the increase in popularity of video and language learning. Although we proposed initial methodological contributions, more work has yet to be done regarding the quality of the training and evaluation data. This aspired unification has to be established not only on the model architecture side but also on the data side. As a result, novel VL datasets should by design include image and video data to build strong foundations for a more unifying evaluation of MLLM and thus move away from merely combining existing benchmarks into new ones. Recent benchmarks still suffer from this dichotomy by focusing on either image [57, 128] or video [47, 134, 58] data.





# Details & Derivations

## A.1 The CLEVR-Dialog DSL

Our CLEVR-Dialog DSL is depicted in [Table A.1](#). We present each function with its expected argument types, output, and knowledge base operations. The argument types are defined in [Table A.2](#). We use the variables `attr`, `attr_obj_1`, `attr_obj_2` and `attr_i` for  $i = 1, \dots, 4$  to denote the set of possible CLEVR attributes, i.e. color, material, shape, and size. Furthermore, the variable `pos` denote one possible position, i.e. right, left, front, or behind. Finally, the variable `num` denotes the set of possible numerical values between 0 and  $N$ , where  $N$  is the maximum number of objects in the scene. If not explicitly stated, we assume  $N = 10$ .

	Func. Name	Func. Args.	Func. Out.	fetch	update			
					Handle	Conv. Subj.	Seen Objs.	Groups
Caption Programs	count-att	attr	none	X	✓	X	✓	✓
	extreme-right	⊔ [attr_1, ..., attr_4]	none	X	✓	✓	✓	X
	extreme-left	⊔ [attr_1, ..., attr_4]	none	X	✓	✓	✓	X
	extreme-behind	⊔ [attr_1, ..., attr_4]	none	X	✓	✓	✓	X
	extreme-front	⊔ [attr_1, ..., attr_4]	none	X	✓	✓	✓	X
	extreme-centre	⊔ [attr_1, ..., attr_4]	none	X	✓	✓	✓	X
	unique-obj	⊔ [attr_1, ..., attr_4]	none	X	✓	✓	✓	X
	obj-relation	attr_obj_1, pos, attr_obj_2	none	X	✓	✓	✓	X
Question Programs	count-all	-	num	X	X	X	X	✓
	count-other	-	num	X	X	✓	✓	X
	count-all-group	-	num	X	X	X	X	X
	count-attribute	attr	num	X	✓	✓	✓	✓
	count-attribute-group	attr	num	X	✓	✓	✓	✓
	count-obj-rel-imm	pos	num	X	X	✓	✓	✓
	count-obj-rel-imm-2	pos	num	X	X	✓	✓	✓
	count-obj-rel-early	pos, attr	num	✓	✓	✓	✓	✓
	count-obj-exclude-imm	attr_type	num	X	X	✓	✓	✓
	count-obj-exclude-early	attr_type, attr	num	✓	X	✓	✓	✓
	exist-other	-	yes/no	X	X	X	✓	✓
	exist-attribute	attr	yes/no	X	✓	X	✓	✓
	exist-attribute-group	attr	yes/no	X	✓	✓	✓	✓
	exist-obj-rel-imm	pos	yes/no	X	X	✓	✓	✓
	exist-obj-rel-imm2	pos	yes/no	X	X	✓	✓	✓
	exist-obj-rel-early	pos, attr	yes/no	✓	✓	✓	✓	✓
	exist-obj-exclude-imm	attr_type	yes/no	X	X	✓	✓	✓
	exist-obj-exclude-early	attr_type, attr	yes/no	✓	X	X	X	✓
	seek-attr-imm	attr_type	attr	X	✓	X	X	X
	seek-attr-imm2	attr_type	attr	X	✓	X	X	X
seek-attr-early	attr_type, attr	attr	✓	✓	✓	✓	X	
seek-attr-sim-early	attr_type, attr	attr	✓	✓	✓	✓	X	
seek-attr-rel-imm	attr_type	attr	X	✓	✓	✓	X	
seek-attr-rel-early	attr_type, pos, attr	attr	✓	✓	✓	✓	X	

TABLE A.1: Our CLEVR-Dialog DSL. We present each function with its expected arguments types, output, and knowledge base operations. Given a set of  $n$  arguments, the  $\cup$  operator selects  $m \leq n$  possible ones.

$\text{attr\_1} \in \text{COLOURS} = [\text{blue}, \text{brown}, \text{cyan}, \text{grey}, \text{green}, \text{purple}, \text{red}, \text{yellow}]$ ,  
 $\text{attr\_2} \in \text{MATERIALS} = [\text{rubber}, \text{metal}]$ ,  
 $\text{attr\_3} \in \text{SHAPES} = [\text{cube}, \text{cylinder}, \text{sphere}]$ ,  
 $\text{attr\_4} \in \text{SIZES} = [\text{large}, \text{small}]$ ,  
 $\text{attr}, \text{attr\_obj\_1}, \text{attr\_obj\_2} \in \bigcup \{\text{COLOURS}, \text{MATERIALS}, \text{SHAPES}, \text{SIZES}\}$ ,  
 $\text{attr\_type} \in [\text{colour}, \text{material}, \text{shape}, \text{size}]$ ,  
 $\text{pos} \in [\text{right}, \text{left}, \text{front}, \text{behind}]$ ,  
 $\text{num} \in [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ .

TABLE A.2: Argument types of our DSL.

## A.2 The Derivation of the NFFR

The NFFR is calculated as

$$\text{NFFR} = \frac{1}{N} \sum_{i=1}^N \frac{1}{L+1} \sum_{j=1}^L \Delta_j^{(i)} \delta_{\text{pred}_j^{(i)}, \text{gt}_j^{(i)}} \alpha_j^{(i)},$$

where  $N$  is the total number of dialogs,  $L$  is the length of each dialog, and  $\text{pred}_j^{(i)}$  and  $\text{gt}_j^{(i)}$  are the predicted and ground truth answers at round  $j$  of dialog  $i$ , respectively. Furthermore, for each round  $j$  of every dialog  $i$ , we define  $\Delta_j^{(i)}$ ,  $\delta_{\text{pred}_j^{(i)}, \text{gt}_j^{(i)}}$ , and  $\alpha_j^{(i)}$  as:

$$\Delta_j^{(i)} = \begin{cases} j & \text{if } j \leq L \\ L+1 & \text{if } j = L \wedge \delta_{\text{pred}_j^{(i)}, \text{gt}_j^{(i)}} = 1 \end{cases},$$

$$\delta_{\text{pred}_j^{(i)}, \text{gt}_j^{(i)}} = \begin{cases} 1 & \text{if } \text{pred}_j^{(i)} \neq \text{gt}_j^{(i)} \\ 0 & \text{otherwise} \end{cases},$$

$$\alpha_j^{(i)} = \begin{cases} 0 & \text{if } \exists k < j \text{ s.t. } \delta_{\text{pred}_k^{(i)}, \text{gt}_k^{(i)}} = 1 \\ 1 & \text{otherwise} \end{cases}.$$

By definition, we set the NFFR of a model to be  $L+1$  if it correctly answers all  $L$  dialog rounds.

Model	Prog. Acc.		Executor Acc.
	Caption	Question	
Caption-Net	99.79	-	
NSVD-concat	-	99.87	99.99
NSVD-stack	-	99.99	

TABLE A.3: Quantitative analysis of our DSL. The high program accuracies demonstrate that our models follow the implemented logic to predict the correct answer, i.e. they do not execute false programs that by chance might lead to correct predictions. When tested with the ground truth scene annotations and programs, our executor reaches an answer-accuracy of 99.99% showcasing its highly reliable execution logic.

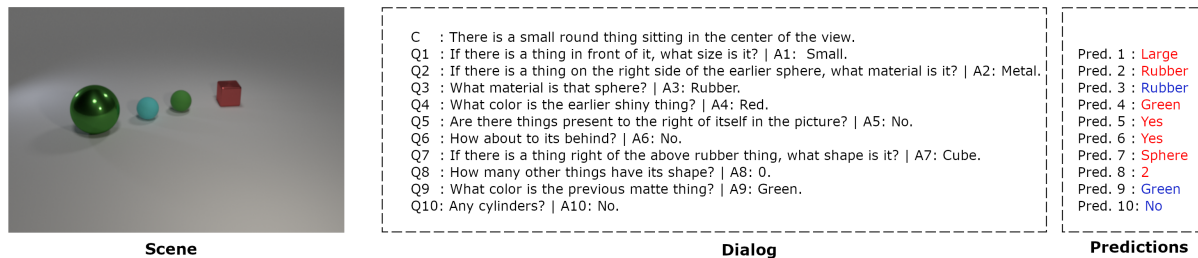


FIGURE A.1: The program `extreme-center(cylinder, small)` induced from the caption “there is a small round thing sitting in the center of the view” does not lead to a unique initialization of our executor’s knowledge base as there are *two* small spheres in the center of the scene. By our logic, we consider it to be the cyan one. Incorrectly initializing the knowledge base leads to confusion when answering the subsequent questions. The blue and red colors indicate a match or a mismatch between the predicted answer and the ground truth, respectively.

## A.3 DSL Logic Validation

To quantify the logical capabilities of our models, we measure the caption and question program accuracies on the test split. As we can see from [Table A.3](#), the caption and question program generation accuracies are 99.79% and 99.99%, respectively. This means that they do not follow a flawed logic, i.e. wrong programs, to predict correct answers. Furthermore, we evaluated the logic of our program executor by measuring its answer accuracy when provided with the ground truth programs and scene annotations of the test split. The last column of [Table A.3](#) shows that it reaches 99.99% accuracy, underlining the flawless implementation of our DSL. The reason why it does not reach the 100% mark is that some scene captions cannot be uniquely interpreted leading to potential confusions when answering the subsequent questions. [Figure A.1](#) illustrates a concrete example of such a case. The caption “there is a small round thing sitting in the centre of the view” induces the program `extreme-centre(cylinder, small)`. However, this can be interpreted in two different ways since there are *two* small spheres in the centre of the scene, i.e. the cyan and the green ones. Therefore, the performance of our executor at answering the following dialog questions depends on which object is considered as the central one. By our logic, we consider it to be the cyan one. The subsequent questions, ground truth answers and predictions are also shown in [Figure A.1](#).

## A.4 Additional Details about OLViT

### A.4.1 Encoder Configuration

To find the optimal configuration of our encoder, i.e. number of transformer layers, number of attention heads, and hidden dimension, we conducted the following experiment in which we independently optimized these variables to keep the search space tractable *while keeping all history rounds and MONet objects*. It should be noted that during this experiment, we also fixed number of the state tracker layers to *two*, i.e.  $L_{\text{ost}} = L_{\text{lst}} = 2$ , and used the same number of attention heads and hidden dimension  $d$  as in the encoder.

#### Number of Attention Heads

First, we fixed the number of transformer layers to *four* and varied the number of attention heads from *two* to *six*, thus varying the hidden

	# of layers <b>L</b>	# of attention heads	Hidden dim. <b>d</b>	Accuracy [%]
A.4.1	4	2	72	52.63
	4	4	144	53.74
	<b>4</b>	<b>6</b>	<b>216</b>	<b>54.01</b>
A.4.1	<b>4</b>	<b>6</b>	<b>216</b>	<b>54.01</b>
	8	6	216	53.15
	16	6	216	52.89

(A) Performance comparison using different number of encoder transformer layers, attention heads, and embedding dimensions.

TABLE A.4: Encoder and state tracker configuration experiment. Performance is measured on the DVD *val* split.

$L_{\text{ost}} = L_{\text{lst}}$	# of attention heads	Hidden dim. <b>d</b>	Accuracy [%]
1	6	216	53.20
<b>2</b>	<b>6</b>	<b>216</b>	<b>54.01</b>
3	6	216	53.91
4	6	216	53.83

(B) Performance comparison using different number of state tracker transformer layers  $L_{\text{ost}}$  and  $L_{\text{lst}}$ . To reduce the search space, we set  $L_{\text{ost}} = L_{\text{lst}}$ .

dimension  $d$  from 72 to 216 in 36 steps. As can be seen in the first section of [Table A.4a](#), our  $\text{OLViT}$  model achieves the best validation accuracy of 54.01% when using six attention heads corresponding to a hidden dimension  $d = 216$ .

### Number of Transformer Layers

Then, we doubled the number of transformer layers starting from *four* until 16 while keeping the optimal values of the attention heads and hidden dimension fixed. As illustrated in the second section of [Table A.4a](#), the performance of  $\text{OLViT}$  on DVD started to decrease with more transformer layers, i.e. from 54.01% validation accuracy with four layers to only 52.89% with 16 layers. This finding was expected as increasing the complexity of the model renders it more prone to overfitting.

### A.4.2 State Tracker Configuration

To optimize our transformer-based state trackers, we experimented with different configurations mainly varying the number of transformer layers  $L_{\text{ost}}$  and  $L_{\text{lst}}$  while inheriting the previous optimal values of the encoder hyperparameters, i.e.  $L = 4$  and  $d = 216$ . For both state trackers, we used the same number of transformer layers, i.e.  $L_{\text{ost}} = L_{\text{lst}}$ . As can be seen in [Table A.4b](#), our model achieved the best DVD validation accuracy of 54.01% when using *two* layers and its performance started to decrease with more layers reaching 53.83% with  $L_{\text{ost}} = L_{\text{lst}} = 4$ . Similarly to the previous experiment, this finding was expected as increasing the number of state tracker layers makes the model more prone to overfitting.

## A.5 VDGR Graph Construction

### A.5.1 Image Modality

Given two object features  $\mathbf{v}_j$  and  $\mathbf{v}_k$ , and their bounding box coordinates  $(x_j, y_j)$  and  $(x_k, y_k)$ , we compute the value of their intersection over unions  $\text{IoU}_{jk}$  and relative angle  $\phi_{jk}$ . As shown in [Figure A.2](#), there are two spacial cases: The first occurs when the bounding box of  $\mathbf{v}_j$  completely includes the bounding box of  $\mathbf{v}_k$  and this class is denoted as *inside* with index  $i = 1$ . The second occurs when the bounding box of  $\mathbf{v}_j$  is entirely covered by the bounding box of  $\mathbf{v}_k$ . This class is denoted as *cover* with index  $i = 2$ .

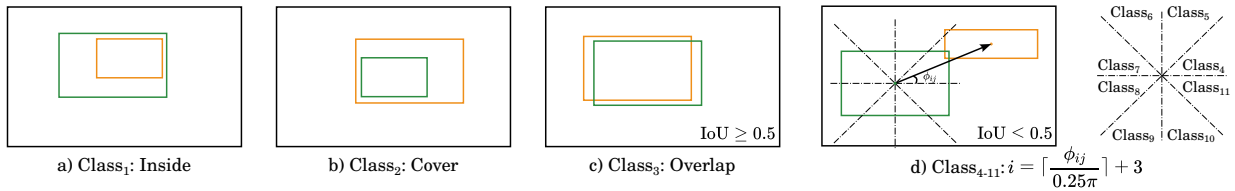


FIGURE A.2: The different spatial relationships used in constructing the image graph. The orange and green rectangles correspond to the bounding boxes of two objects within the scene.

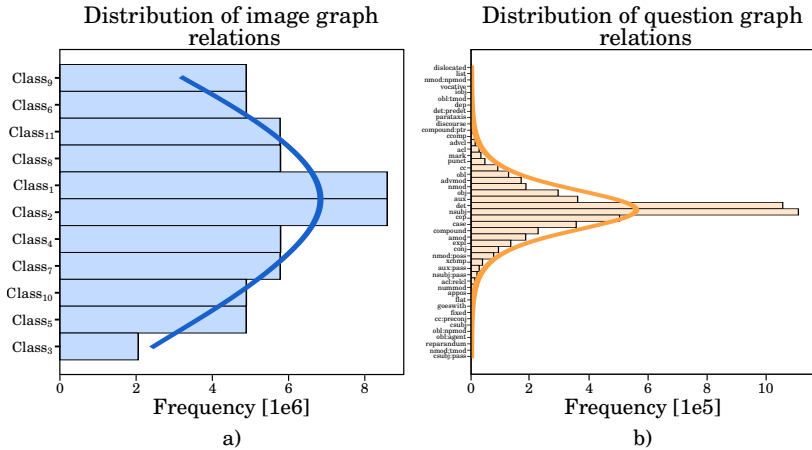


FIGURE A.3: **Right:** The distribution of the image graph relationship classes within the training split of VisDial v1.0. **Left:** The distribution of the question graph relationship classes within the training split of VisDial v1.0.

The remaining classes are solely determined by the value of  $\text{IoU}_{jk}$ . If  $\text{IoU}_{jk} \geq 0.5$ , then the relationship between the objects is denoted as *overlap* and has the index  $i = 3$ . Finally, if  $\text{IoU}_{jk} < 0.5$ , the class index is computed as

$$i = \lceil \frac{\Phi_{jk}}{0.25\pi} \rceil + 3. \quad (\text{A.1})$$

By construction, all classes of index  $i \neq 3$  are pairwise symmetric as can be seen from [Figure A.3a](#) where we plotted the distribution of the different image graph relationship classes over the training split of VisDial v1.0.

## A.5.2 Question Modality

The question graph relationship classes are determined by the dependency between the question words. To this end, each question is input into the Stanza dependency parser [153] that outputs the classes between the different word pairs resulting in a total of 47 classes. As shown in [Figure A.3b](#), the distribution of these classes in the VisDial v1.0 training split is not uniform with *det* and *nsubj* being the most frequent. We illustrate a qualitative sample in [Figure A.4](#).

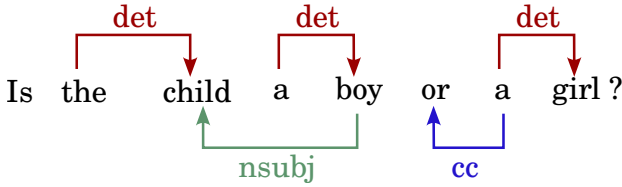


FIGURE A.4: A sample of the dependency relationships between question word pairs.

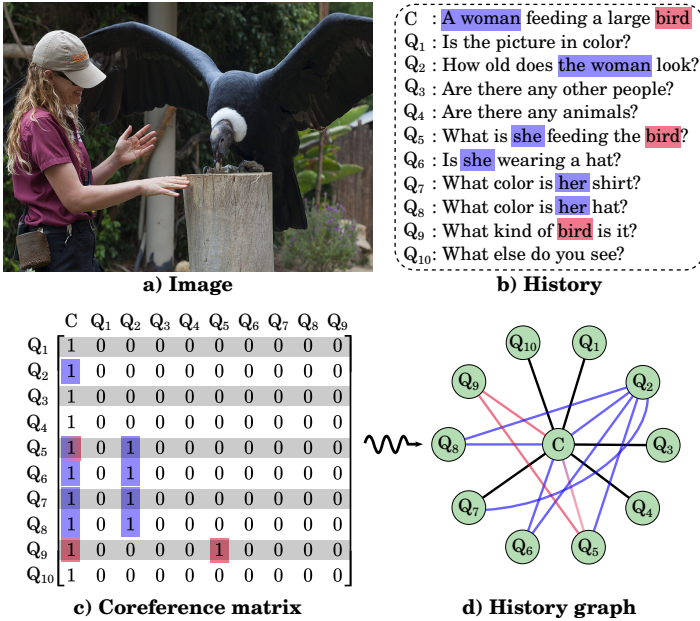


FIGURE A.5: A qualitative sample of the coreference relationships between different dialog rounds. The hub-node was not visualized for clarity.

### A.5.3 History Modality

We rely on coreference resolution to construct the history graph. Specifically, an edge exists between two rounds  $i$  and  $j$  ( $i > j$ ) if and only if a word in round  $j$  references another in round  $i$ . The only exception is the caption  $\mathcal{C}$  that links to all upcoming rounds in the history even if there is no explicit coreference between them. We posit that the caption is complementary to the visual input and helps the

model better understand the scene. We illustrate a qualitative sample in [Figure A.5](#).

## A.6 ELBO loss

In this section, we derive the ELBO loss and show how it can be used as an optimization term in [Equation 7.11](#). Without the loss of generality, we only consider the global ELBO loss as the local ones of [Equation 7.12](#) can be derived in the same manner. Given the intractable posterior  $p_\theta(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})$  and its approximation  $q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})$ , it holds that

$$D_{\text{KL}}\left(q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})\|p_\theta(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})\right) = \mathbb{E}_{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})} \left[ \log \frac{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})}{p_\theta(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})} \right] \quad (\text{A.2})$$

$$= \mathbb{E}_{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})} \left[ \log \frac{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})p_\theta(\tilde{\mathbf{A}}|\mathbf{X})}{p_\theta(\mathbf{A}', \tilde{\mathbf{A}}|\mathbf{X})} \right] \quad (\text{A.3})$$

$$= \mathbb{E}_{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})} \left[ \log \frac{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})}{p_\theta(\mathbf{A}', \tilde{\mathbf{A}}|\mathbf{X})} \right] + \log p_\theta(\tilde{\mathbf{A}}|\mathbf{X}) \quad (\text{A.4})$$

$$= \underbrace{\log p_\theta(\tilde{\mathbf{A}}|\mathbf{X})}_{\text{Evidence}} - \underbrace{\mathbb{E}_{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})} \left[ \log \frac{p_\theta(\mathbf{A}', \tilde{\mathbf{A}}|\mathbf{X})}{q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})} \right]}_{=:\mathcal{L}_{\text{ELBO}}^{\text{global}}} \geq 0 \quad (\text{A.5})$$

Thus, ELBO serves as a lower bound of the evidence. As a result, VI tries to maximize the ELBO which is equivalent to minimizing the Kullback-Leibner divergence between  $q_\phi(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})$  and the intractable posterior  $p_\theta(\mathbf{A}'|\tilde{\mathbf{A}}, \mathbf{X})$  leading to a better estimation of the latter. Since we used ELBO terms in the total loss  $\mathcal{L}$  that we want to minimize, we had to use the opposite value of each one of them, which explains the minus sign in [Equation 7.11](#). Since  $q_\phi$  and  $p_\theta$  only output normalized scores as the prediction for each edge, we appended the zero vectors to both predictions in order to convert the raw scores to a two-value probability before applying the log-softmax function. In [Listing A.1](#), we provide a code-snippet of our PyTorch implementation of the ELBO loss.

## A.7 Training Objectives of $V^2Dial$

In addition to the proposed STC and STM, we trained our model with the following established VL objectives.

**Masked Language Modeling** teaches the model to predict masked text tokens given both the visual and textual context. As in [42, 116] we mask 15% of the tokens and minimize the loss

$$\mathcal{L}_{\text{mlm}} = \mathbb{E}_{(\mathbf{v}^{\text{vis}}, \hat{\mathbf{t}}^{\text{cap}})} \left[ \text{CE}(\mathbf{y}^{\text{mlm}}, \mathbf{p}^{\text{mlm}}) \right], \quad (\text{A.6})$$

where  $\mathbf{y}^{\text{mlm}}$  and  $\mathbf{p}^{\text{mlm}}$  denote the ground-truth and predicted probabilities of the masked tokens whereas  $\mathbf{V}^{\text{vis}}$  and  $\hat{\mathbf{T}}^{\text{cap}}$  are the visual and masked caption token embeddings, respectively.

**Vision-Text Contrastive Learning** helps the model better align the video/image and the text features and is defined similarly to STC as

$$\mathcal{L}_{\text{vtc}} = \frac{1}{2} \mathbb{E}_{(\mathbf{v}^{\text{vis}}, \mathbf{t}^{\text{cap}})} \left[ \text{CE}(\mathbf{y}^{\text{v2t}}, \mathbf{p}^{\text{v2t}}) + \text{CE}(\mathbf{y}^{\text{t2v}}, \mathbf{p}^{\text{t2v}}) \right], \quad (\text{A.7})$$

where  $\mathbf{p}^{\text{v2t}}$  and  $\mathbf{p}^{\text{t2v}}$  are the softmax normalized vision-to-text and text-to-vision similarities defined as in Equation 8.13 and Equation 8.14.  $\mathbf{y}^{\text{v2t}}$  and  $\mathbf{y}^{\text{t2v}}$  are their respective ground-truth one-hot similarities.

**Vision-Text Matching** is defined similarly to STM as a binary classification problem and complements VTC by teaching the model to distinguish between matched and unmatched paired vision-text features. We use a video/image and its corresponding caption as a positive example. The negative examples are constructed via negative sampling of captions from different visual inputs. Formally,

$$\mathcal{L}_{\text{vtm}} = \mathbb{E}_{(\mathbf{v}^{\text{vis}}, \mathbf{t}^{\text{cap}})} \left[ \mathcal{H}(\mathbf{y}^{\text{vtm}}, \mathbf{p}^{\text{vtm}}) \right], \quad (\text{A.8})$$

where  $\mathbf{p}^{\text{stm}}$  and  $\mathbf{y}^{\text{stm}}$  are the predicted and the ground-truth two-class probabilities, respectively.

LISTING A.1: PyTorch implementation of the ELBO loss.

```

import torch
import torch.nn as nn
import torch.nn.functional as F
class ELBO(nn.Module):
    def __init__(self):
        super(ELBO, self).__init__()
    def forward(self, Aq, Ap):
        """
        Args:
            Aq: The predicted latent graph of q_phi
                (batch_size, K, K) -- local graphs
                (batch_size, NK, NK) -- global graphs
            Ap: The predicted latent graph of p_theta
                (batch_size, K, K) -- local graphs
                (batch_size, NK, NK) -- global graphs

        Returns:
            The ELBO loss
        """
        Aq_flat = Aq.view(-1).unsqueeze(-1)
        Ap_flat = Ap.view(-1).unsqueeze(-1)

        Aq_flat = torch.cat(
            [torch.zeros_like(Aq_flat), Aq_flat],
            dim=-1)
        Ap_flat = torch.cat(
            [torch.zeros_like(Ap_flat), Ap_flat],
            dim=-1)

        log_Aq = F.log_softmax(QA_flattened, dim=1)
        log_Ap = F.log_softmax(PA_flattened, dim=1)
        Aq_dist = torch.exp(log_Aq)
        loss_Aq = torch.mean(log_Aq * Aq_dist)
        loss_Ap = torch.mean(log_Ap * Ap_dist)
        elbo_loss = loss_Aq - loss_Ap

    return elbo_loss

```



# B

## Training Details

### B.1 OLViT

We implemented our model in PyTorch [148] and conducted our experiments on *two* NVIDIA Tesla V100 (32GB) GPUs and used MONet to generate  $N_o = 12$  object masks from  $T = 20$  equidistantly-sampled frames and DistilRoBERTa-base to generate the  $d_w = 768$  dimensional word embeddings. During training, we optimized the model weights using AdamW [130] with a batch size of 50. We set the initial learning rate to  $1e^{-4}$  and used a learning rate schedule with warm-up for 4k iterations and cosine-decay until convergence for a total of 200K iterations.

### B.2 VDGR

We implemented VDGR using PyTorch [148] and trained it on a server with *eight* NVIDIA Tesla V100 (32GB) GPUs using an effective batch size of 64 and Adam optimizer [92] with a linear decay learning rate schedule with warm-up. We set the initial learning rates of the BERT

Hyperparameter	Training Stage		
	Warm-up	Sparse fine-tuning	Dense fine-tuning
Number of GNN layers K	2	2	2
Number of GNN heads H	4	4	4
Residual connection coefficient $\lambda$	0.5	0.5	0.5
Dimension of $\text{GNN}_{\mathcal{I}}$ node features	1024	1024	1024
Dimension of $\text{GNN}_{\mathcal{Q}}$ node features	768	768	768
Dimension of $\text{GNN}_{\mathcal{H}}$ node features	768	768	768
Dimension of $\text{GNN}_{\mathcal{I}}$ edge features	12	12	12
Dimension of $\text{GNN}_{\mathcal{Q}}$ edge features	48	48	48
Dimension of $\text{GNN}_{\mathcal{H}}$ edge features	2	2	2
Dimension of $\text{Linear}_{\mathcal{I} \rightarrow \mathcal{H}}(\cdot)$	(1024, 768)	(1024, 768)	(1024, 768)
Dimension of $\text{Linear}_{\mathcal{Q} \rightarrow \mathcal{I}}(\cdot)$	(768, 1024)	(768, 1024)	(768, 1024)
Maximum number of text tokens	256	256	256
Maximum number of image regions	37	37	37
Text token mask probability	0.1	0.1	–
Image region mask probability	0.1	0.1	–
Graph edge mask probability	0.15	–	–
Optimiser	Adam	Adam	Adam
Minimum lr of BERT parameters	0	0	$1 \times 10^{-5}$
Minimum lr of GNN parameters	0	0	$1 \times 10^{-5}$
Maximum lr of BERT parameters	$5 \times 10^{-6}$	$5 \times 10^{-6}$	$2 \times 10^{-5}$
Maximum lr of GNN parameters	$5 \times 10^{-4}$	$5 \times 10^{-4}$	$1 \times 10^{-4}$
Learning rate warm-up	True	True	True
Learning rate schedule	Linear	Linear	Linear
Training Loss	$\mathcal{L}_{\text{warm}}$	$\mathcal{L}_{\text{VD}}$	$\mathcal{L}_{\text{CE}} / \mathcal{L}_{\text{ListNet}}$
Number of epochs	5	20	3
Effective batch size	64	64	100
GPU Model	Tesla V100-32GB	Tesla V100-32GB	Tesla V100-32GB
Number of GPUs	8	8	8
Distributed training	Apex	Apex	PyTorch DP

TABLE B.1: Detailed hyperparameter setting for the different training stages of VDGR.  $\text{Linear}_{\mathcal{I} \rightarrow \mathcal{H}}(\cdot)$  and  $\text{Linear}_{\mathcal{Q} \rightarrow \mathcal{I}}(\cdot)$  denote the linear layers that produce the history and image hub-node features, respectively.

and GNN weights to  $5 \times 10^{-6}$  and  $5 \times 10^{-4}$ , respectively. Furthermore, we set the loss coefficients  $\alpha_1 = \alpha_2 = 1$  and the residual connection coefficient  $\lambda = 0.5$ . We refer to Table B.1 for a complete overview of our experimental setup.

### B.3 MST<sub>MIXER</sub>

We implemented MST<sub>MIXER</sub> in PyTorch [148] and trained it end-to-end on a cluster consisting of 8 Nvidia Tesla V100 (32GB) GPUs using AdamW [130] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1e - 8$  and a linear learning rate schedule with warm-up for a maximum of 12 epochs.

<b>Hyperparameter</b>	
Dimension of I3D rgb / I3D flow / SAM features $d_v$	2048
Dimension of SAM features $d_s$	512
Maximum length of I3D rgb / I3D flow / SAM features $d_l$	36
Dimension of audio features $d_a$	128
Maximum length of audio features $l_a = l_v$	36
Maximum total length of multi-modal input	1024
Dimension of hidden features $d$	1024/768
Number of node features in local GNNs $K$	10
Number of node features in global GNNs $K \times N$	60
Number for kNNs in $\{\tilde{\mathbf{A}}_i\}$	4
Number of heads $H$ of Equation 7.6	8
Input dimension of GNNs in Table 7.5c	1024
Output dimension of GNNs in Table 7.5c	1024
$K$ value of APPNP	2
$\alpha$ value of APPNP	0.1
Number of attention heads in local GATs	2
Number of attention heads in global GATs	4
$\lambda$ value	0.9
$\Delta$ value	4
Optimizer	AdamW
Learning rate of parameters in the VLM backbone $lr_{\text{BART}}$	$1e-5$
Learning rate of other parameters $lr_{\text{rest}}$	$1e-4$
Values of $\{\alpha_1, \alpha_2, \alpha_3\}$	{1, 100, 100}
Learning rate schedule	linear
Dropout rate	0.1
Value of gradient clipping	1.0
Effective batch size	96
Number of epochs	12
GPU model	Tesla V100-32GB
Number of GPUs	8
Distributed training	PyTorch DDP

TABLE B.2: Detailed hyperparameter setting of the training and inference of our best MST<sub>MIXER</sub> model.

We used a learning rate  $lr_{\text{BART}} = 1e-5$  for the weights of the BART model and a learning rate  $lr_{\text{rest}} = 1e-4$  for the rest of the parameters of our model. Similarly to  $\lambda$  and  $\Delta$ , we validated the choice of the ELBO loss coefficients  $\alpha_2$  and  $\alpha_3$  based on the validation perplexity. The training of our full model takes approximately 20 hours. Complete details about the hyperparameter values are listed in Table B.2.

<b>Hyperparameter</b>	
Number of expert-based layers N	12
Number of multimodal experts layers L	9
Number of fusion experts layers (N - L)	3
Joint hidden dimension D	1024
Number of frames F	4
Number of patches per frame P	64
Hidden dimension of LLM	1024
Dimension of LLM linear layer	(1024, 1024)
Dimension of linear layers $\Theta_*$	(1024, 256)
Optimizer	AdamW
Learning rate schedule	linear
Minimum learning rate value	$5e - 5$
Base learning rate value	$1e - 4$
Weight decay	0.01
Gradient clipping value	1.0
Effective batch size	48
GPU model	A100
Number of GPUs	8
Distributed training	DDP

TABLE B.3: Detailed hyperparameter setting of  $\mathbb{V}^2\text{Dial}$ .

## B.4 $\mathbb{V}^2\text{Dial}$

We implemented  $\mathbb{V}^2\text{Dial}$  in PyTorch [148] and trained it on a cluster consisting of *eight* A100 GPUs. In the first stage, we trained it for a maximum of *ten* epochs and applied early stopping based on a validation split to select the best checkpoint. In the subsequent stages, we trained it for up to *three* and *twelve* epochs, respectively. In all stages, we used the AdamW [130] optimizer with the default parameters and a weight decay value of 0.01. Furthermore, we applied a linear learning rate schedule with warm-up and minimum and base values of  $5e - 5$  and  $1e - 4$ , respectively. In Table B.3, we provide additional information about the hyperparameter setup of our model.

# C

## Qualitative Samples

### C.1 NSVD

We illustrate inference examples of our models from the experiments of [Chapter 4](#). As seen in [Figure C.1](#), both variant of our model answered all 10 rounds successfully. However, in [Figure C.2](#), *NSVD-concat* failed at answering round 7. Finally, we show a sample dialog of our model on the novel Minecraft-Dialog in [Figure C.3](#).

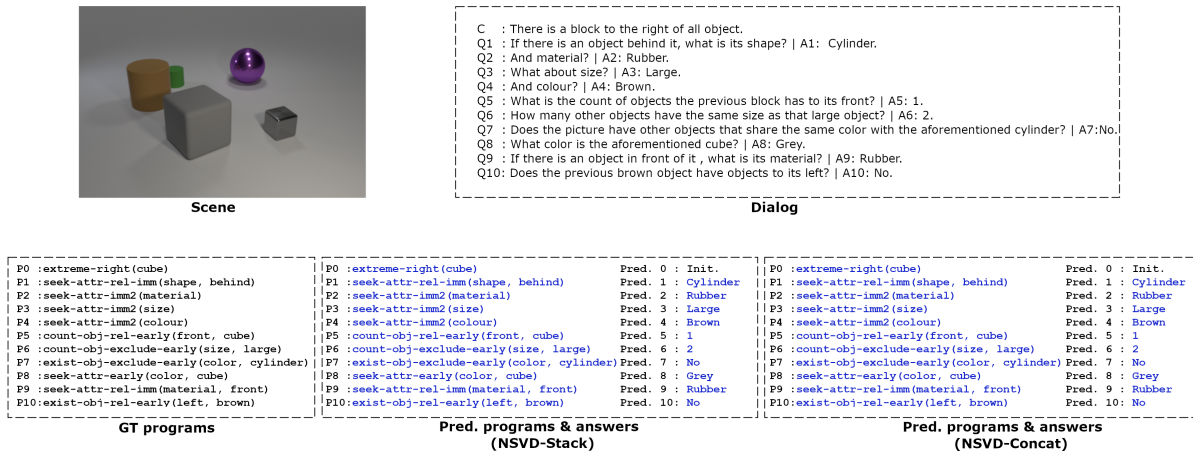
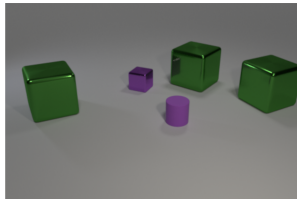


FIGURE C.1: Both of our models generated correct programs to predict the answers. The blue color indicates a match between the predicted program/answer and the ground truth.



Scene

C : There is a rubber thing in front of a large thing in the view.  
 Q1 : How many other things are in the image? | A1: 3.  
 Q2 : What color is the above large thing? | A2: Green.  
 Q3 : What is the shape of the previous matte thing? | A3: Cylinder.  
 Q4 : What material is the previous big thing? | A4: Metal.  
 Q5 : If there is a thing to the left of that matte thing , what is its color? | A5: Purple.  
 Q6 : Does it have things to the right of itself? | A6: Yes.  
 Q7 : And to its left? | A7: Yes.  
 Q8 : If there is a thing behind the earlier purple thing , what shape is it? | A8: Cube.  
 Q9 : If there is a thing in front of the previous purple thing , what is its shape? | A9: Cube.  
 Q10: What number of red things are present? | A10: 0.

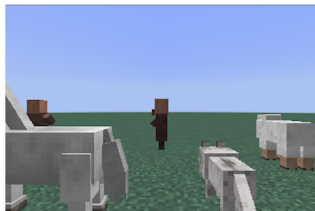
Dialog

P0 :obj-relation(rubber, large, front)	P0 :obj-relation(rubber, large, front)	Pred. 0 : Init.	P0 :obj-relation(rubber, large, front)	Pred. 0 : Init.
P1 :count-other()	P1 :count-other()	Pred. 1 : 3	P1 :count-other()	Pred. 1 : 3
P2 :seek-attr-early(color, large)	P2 :seek-attr-early(color, large)	Pred. 2 : Green	P2 :seek-attr-early(color, large)	Pred. 2 : Green
P3 :seek-attr-early(shape, rubber)	P3 :seek-attr-early(shape, rubber)	Pred. 3 : Cylinder	P3 :seek-attr-early(shape, rubber)	Pred. 3 : Cylinder
P4 :seek-attr-early(material, large)	P4 :seek-attr-early(material, large)	Pred. 4 : Metal	P4 :seek-attr-early(material, large)	Pred. 4 : Metal
P5 :seek-attr-rel-early(color, left, rubber)	P5 :seek-attr-rel-early(color, left, rubber)	Pred. 5 : Purple	P5 :seek-attr-rel-early(color, left, rubber)	Pred. 5 : Purple
P6 :exist-obj-rel-imm(right)	P6 :exist-obj-rel-imm(right)	Pred. 6 : Yes	P6 :exist-obj-rel-imm(right)	Pred. 6 : Yes
P7 :exist-obj-rel-imm2(left)	P7 :exist-obj-rel-imm2(left)	Pred. 7 : Yes	P7 :count-obj-rel-imm2(left)	Pred. 7 : 1
P8 :seek-attr-rel-early(shape, behind, purple)	P8 :seek-attr-rel-early(shape, behind, purple)	Pred. 8 : Cube	P8 :seek-attr-rel-early(shape, behind, purple)	Pred. 8 : Cube
P9 :seek-attr-rel-early(shape, front, purple)	P9 :seek-attr-rel-early(shape, front, purple)	Pred. 9 : Cube	P9 :seek-attr-rel-early(shape, front, purple)	Pred. 9 : Cube
P10 :count-attribute(red)	P10 :count-attribute(red)	Pred. 10 : 0	P10 :count-attribute(red)	Pred. 10 : 0

GT programs

Pred. programs & answers  
(NSVD-Stack)Pred. programs & answers  
(NSVD-Concat)

FIGURE C.2: While *NSVD-stack* answered all questions correctly, *NSVD-concat* failed at round 7. The blue and red colors indicate a match or a mismatch between the predicted program/answer and the ground truth, respectively.



Scene

C : The view has a filly.  
 Q1 : If present, how many humans are in the view? | A1: 2.  
 Q2 : If there is a thing to the left of the aforementioned horse, what is its class? | A2: Villager.  
 Q3 : Does the aforementioned mare have things to its front? | A3: No.  
 Q4 : What direction is the above villager facing | A4: Backward.  
 Q5 : What is the direction of the previous mare? | A5: Left.  
 Q6 : If there is a thing on the right side of it, what is its direction? | A6 : Left.  
 Q7 : What about class? | A7 : Villager.  
 Q8 : Are there other things that are of same direction as that object facing backward? | A8: Yes.  
 Q9 : What is the nature of that filly? | A9: Animal.  
 Q10: What is the count of other things that share its direction? | A10: 1

Dialog

```
P0 :unique-obj(horse)
P1 :count-attribute(human)
P2 :seek-attr-rel-early(class, left, horse)
P3 :exist-obj-rel-early(front, horse)
P4 :seek-attr-early(direction, villager)
P5 :seek-attr-early(direction, horse)
P6 :seek-attr-rel-imm(direction, right)
P7 :seek-attr-imm2(class)
P8 :exist-obj-exclude-early(direction, backward)
P9 :seek-attr-early(nature, horse)
P10:count-obj-exclude-imm(direction)
```

GT Programs

```
P0 :unique-obj(horse)          Pred.0 :Init
P1 :count-attribute(human)    Pred.1 :2.
P2 :seek-attr-rel-early(class, left, horse) Pred.2 :Villager.
P3 :exist-obj-rel-early(front, horse)    Pred.3 :No.
P4 :seek-attr-early(direction, villager) Pred.4 :Backward
P5 :seek-attr-early(direction, horse)    Pred.5 :Left.
P6 :seek-attr-rel-imm(direction, right)  Pred.6 :Left.
P7 :seek-attr-imm2(class)          Pred.7 :Villager.
P8 :exist-obj-exclude-early(direction, backward) Pred.8 :Yes.
P9 :seek-attr-early(nature, horse)    Pred.9 :Animal.
P10:count-obj-exclude-imm(direction)  Pred.10:1.
```

Pred. programs & answers  
(NSVD-Stack)

```
P0 :unique-obj(horse)          Pred.0 :Init
P1 :count-attribute(human)    Pred.1 :2.
P2 :seek-attr-rel-early(class, left, horse) Pred.2 :Villager.
P3 :exist-obj-rel-early(front, horse)    Pred.3 :No.
P4 :seek-attr-early(direction, villager) Pred.4 :Backward
P5 :seek-attr-early(direction, horse)    Pred.5 :Left.
P6 :seek-attr-rel-imm(direction, right)  Pred.6 :Left.
P7 :seek-attr-imm2(class)          Pred.7 :Villager.
P8 :exist-obj-exclude-early(direction, backward) Pred.8 :Yes.
P9 :seek-attr-early(nature, horse)    Pred.9 :Animal.
P10:count-obj-exclude-imm(direction)  Pred.10:1.
```

Pred. programs & answers  
(NSVD-Concat)

FIGURE C.3: Inference example of our models on a test instance of Minecraft-Dialog. The blue and red colors indicate a match or a mismatch between the predicted program/answer and the ground truth, respectively.

## C.2 VDGR

We present additional qualitative examples from the *val* split of VisDial v1.0 in [Figure C.4](#) and [Figure C.5](#). As in [Chapter 6](#), we compared the top-1 predictions of VDGR with the ground-truth and the predictions of VD-PCR since it achieved the second best performance on this split.

## C.3 MST<sub>MIXER</sub>

We provide additional positive qualitative examples of our best model and some of its ablated versions for comparison in [Figure C.6](#). In addition, we illustrate some failure cases in [Figure C.7](#).

## C.4 V<sup>2</sup>Dial

We provide additional qualitative samples comprising of both success and failure cases of our model. [Figure C.8](#) and [Figure C.9](#) illustrate some zero-shot samples for AVSD and VisDial, respectively. Additional fine-tuning examples for both datasets are shown in [Figure C.10](#) and [Figure C.11](#).








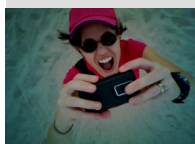

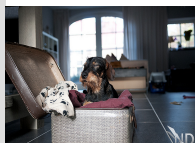
<p><b>Image &amp; Caption</b></p>  <p>A couple sitting on a wooden bench in front of a field of trees</p>	<p><b>Predictions</b></p> <p>Q: How old is the couple? A: It's hard to tell I can only see their backs Q: What color is the girl's hair? A: It's black Q: What color is the bench? A: Male Q: Is there a lot of trees? A: Yes, in the distance VD-GR: Yes, in the distance VD-PCR: Yes it is</p>	<p><b>Image &amp; Caption</b></p>  <p>An assortment of doughnuts on a white shelf</p>	<p><b>Predictions</b></p> <p>Q: How many doughnuts do you see? A: 1 Q: What kind of doughnuts? A: Baked Q: Any toppings on the doughnut? A: Yes Q: Anything else on the shelf? A: Sugars, other baked stuff VD-GR: Sugars, other baked stuff VD-PCR: The mirror</p>
<p><b>Image &amp; Caption</b></p>  <p>A white polar bear sleeping on a wet rock</p>	<p><b>Predictions</b></p> <p>Q: Is there only 1 polar bear? A: Yes Q: Is the rock large? A: I don't see the whole rock so I assume so Q: What color is the rock? A: Dark concrete Q: Does the bear look to be an adult? A: I am not sure, it is zoomed close Q: Can you see the water? A: No VD-GR: No VD-PCR: I don't think so</p>	<p><b>Image &amp; Caption</b></p>  <p>3 young men are smiling and looking at a scene to the left of the camera</p>	<p><b>Predictions</b></p> <p>Q: Are they inside or outside? A: Inside Q: Is this a house or a business? A: I can't quite tell but it looks like a stage Q: Are there any women? A: No Q: Are the men standing? A: Yes Q: Are the men wearing glasses? A: 1 of them VD-GR: 1 of them VD-PCR: There are no hats</p>
<p><b>Image &amp; Caption</b></p>  <p>A boy with a cowboy hat in front of a large street clock</p>	<p><b>Predictions</b></p> <p>Q: What color is the hat? A: It is light beige Q: What color is the clock? A: The pole is green and the face is white Q: What color is the boy's shirt? A: White Q: What color is his pants? A: I cannot see his pants Q: Is it sunny outside? A: It is partly sunny, yes VD-GR: It is partly sunny, yes VD-PCR: Overcast</p>	<p><b>Image &amp; Caption</b></p>  <p>A plate displays chocolate pastries molded to look like train cars winding around a hill</p>	<p><b>Predictions</b></p> <p>Q: Is the image in color? A: Yes Q: What color is the plate? A: Yellow Q: What color are the pastries? A: Mostly brown Q: What color is the hill? A: Brown with white icing VD-GR: Brown with white icing VD-PCR: It is white</p>
<p><b>Image &amp; Caption</b></p>  <p>A group of colorful boats tied up together on the water</p>	<p><b>Predictions</b></p> <p>Q: What color are the boats? A: white on the bottom and having different color trims: blue, green, red, and yellow Q: Are they sail boats? A: No, row boats Q: Are there people in the picture? A: No Q: Is it a sunny day? A: No, it looks somewhat cloudy Q: Are there people in the picture? A: No Q: Are the boats tied to a pier? A: To a dock Q: How many boats are there? A: At least 9 VD-GR: At least 9 VD-PCR: 4 visible and the arm of 1 just out of pic</p>	<p><b>Image &amp; Caption</b></p>  <p>A woman is holding her phone over her head while standing on a beach</p>	<p><b>Predictions</b></p> <p>Q: Is the woman taking a photo? A: Yes it seems so Q: Is the woman in the water? A: No Q: How old is this woman? A: Late twenties early thirties Q: Are any other people visible? A: No one else Q: Is it a sunny day? A: Can't tell Q: What is the woman wearing? A: A short sleeve shirt and a hat VD-GR: A short sleeve shirt and a hat VD-PCR: Red dress with patterns</p>
<p><b>Image &amp; Caption</b></p>  <p>A person has a kite on a beach</p>	<p><b>Predictions</b></p> <p>Q: How many people are in the picture? A: Just 1 person Q: Are there any animals in the picture? A: No Q: Is it daytime or nighttime? A: It's daytime Q: Is it sunny outside? A: It seems hazy outside Q: What color is the kite? A: It is red Q: Is the person male or female? A: The person seems to be male Q: What color is his hair? A: I can't tell VD-GR: I can't tell VD-PCR: 1</p>	<p><b>Image &amp; Caption</b></p>  <p>The dog is sitting in the basket in the dark room</p>	<p><b>Predictions</b></p> <p>Q: Are there other dogs? A: No, just the one Q: What kind of dog is it? A: Looks like a schmuser Q: Does the dog have a blanket? A: He does, it's next to him Q: What color is the dog? A: Black &amp; brown Q: Does it look happy? A: He looks super happy Q: Is there food near it? A: No Q: Are there doggy toys? A: In the background there appears to be 1 on the floor VD-GR: In the background there appears to be 1 on the floor VD-PCR: 1</p>

FIGURE C.4: Qualitative samples from the *val* split of the VisDial v1.0 dataset. We compared the predictions of VDGR (orange) with the ground truth answers (blue) and the predictions of VD-PCR (pink). The answers of both models correspond to the top-1 predictions.





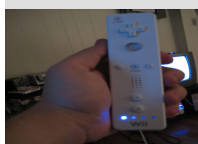





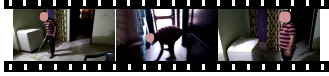
<p><b>Image &amp; Caption</b></p>  <p>An old black and white photo of cows in a pasture</p>	<p><b>Predictions</b></p> <p>Q: How many cows are there? A: More than 10 it's hard to see them</p> <p>Q: Does the photo look vintage? A: Slightly</p> <p>Q: Are there any people near? A: No</p> <hr/> <p>Q: Can you tell what the weather is like? A: Appears to be slightly cloudy</p> <p>Q: Are there any calves present? A: Possibly it is hard to tell</p> <p>VD-GR: Possibly it is hard to tell <span style="color: green;">✔</span></p> <p>VD-PCR: 1 other 1 in the background <span style="color: red;">✘</span></p>
<p><b>Image &amp; Caption</b></p>  <p>A cupboard full of hair supplies and lotion</p>	<p><b>Predictions</b></p> <p>Q: Is the picture in color? A: Full blown color</p> <p>Q: Are there any people? A: 0</p> <p>Q: What color is the cupboard? A: White pinkish</p> <p>Q: Are there any cupboards shown? A: Lots of utensils</p> <p>Q: Are the cupboard doors open? A: No doors</p> <p>VD-GR: No doors <span style="color: green;">✔</span></p> <p>VD-PCR: Small <span style="color: red;">✘</span></p>
<p><b>Image &amp; Caption</b></p>  <p>Yellow, gold like looking hair is surfacing the image</p>	<p><b>Predictions</b></p> <p>Q: Can you see the person? A: No</p> <p>Q: Is it just hair? A: Yes</p> <p>Q: Is it straight hair? A: Wavy</p> <hr/> <p>Q: Are they short and curly hairs? A: Short</p> <p>Q: Is it a coyote? A: It could be</p> <p>VD-GR: It could be <span style="color: green;">✔</span></p> <p>VD-PCR: Black <span style="color: red;">✘</span></p>
<p><b>Image &amp; Caption</b></p>  <p>The black and white photo shows a woman with an umbrella</p>	<p><b>Predictions</b></p> <p>Q: What color is the umbrella? A: Bright pink with little birds and flowers</p> <p>Q: What is the woman doing? A: Walking on the sidewalk</p> <p>Q: Do you see any animals? A: No animals</p> <p>A: Yes looks very clean</p> <p>VD-GR: Can't really tell from the picture <span style="color: red;">✘</span></p> <p>VD-PCR: Yes looks very clean <span style="color: green;">✔</span></p>
<p><b>Image &amp; Caption</b></p>  <p>A person holds a video game controllers in their hands</p>	<p><b>Predictions</b></p> <p>Q: Is the picture in color? A: Yes</p> <p>Q: Is the person a man or a woman? A: I can't see their face to tell</p> <p>Q: How old do they look? A: Can't tell since I can't see the face</p> <p>Q: What kind of video game? A: Can't tell but it's a wii</p> <p>Q: Can you see the tv? A: Yes</p> <p>VD-GR: No <span style="color: red;">✘</span></p> <p>VD-PCR: Yes <span style="color: green;">✔</span></p>
<p><b>Image &amp; Caption</b></p>  <p>2 men ride ocean waves together on separate surfboards</p>	<p><b>Predictions</b></p> <p>Q: Can you see the beach? A: Yes</p> <p>Q: Is it sunny? A: No</p> <p>Q: Are the men smiling? A: I think so, 1 is looking backwards</p> <p>Q: Can you see the sky? A: No</p> <p>Q: Are the waves tall? A: No there are medium size</p> <p>VD-GR: No there are medium size <span style="color: green;">✔</span></p> <p>VD-PCR: Yes, they are <span style="color: red;">✘</span></p>
<p><b>Image &amp; Caption</b></p>  <p>The indian woman is holding a motorola flip phone</p>	<p><b>Predictions</b></p> <p>Q: Is the woman young or old? A: The woman is a little bit old</p> <p>Q: Does she have long or short hair? A: She has a long tied hair</p> <p>Q: What color hair does she have? A: She has black hair</p> <p>Q: Is it daytime? A: Yes, it is</p> <p>Q: Is it inside or outside this picture? A: It is outdoors</p> <p>VD-GR: It is outdoors <span style="color: green;">✔</span></p> <p>VD-PCR: I don't think so <span style="color: red;">✘</span></p>
<p><b>Image &amp; Caption</b></p>  <p>A man in a blue jacket is skiing down the slope</p>	<p><b>Predictions</b></p> <p>Q: Is the man's jacket light or dark blue? A: Light blue</p> <p>Q: Does he have ski poles? A: Yes</p> <p>Q: Are they black? A: No</p> <hr/> <p>Q: What color are his boots? A: I can hardly see them</p> <p>Q: Can you see any trees? A: No</p> <p>VD-GR: Not really <span style="color: red;">✘</span></p> <p>VD-PCR: Not really <span style="color: red;">✘</span></p>
<p><b>Image &amp; Caption</b></p>  <p>A set of sinks with a large mirror above them</p>	<p><b>Predictions</b></p> <p>Q: Is the picture in color? A: It is</p> <p>Q: What color are the sinks? A: They are white</p> <p>Q: Is the mirror square? A: Rectangular</p> <p>Q: Does the mirror have a frame? A: They open, so yes</p> <p>Q: Can you see a reflection in the mirror? A: Just of the bathroom</p> <p>VD-GR: Yes <span style="color: green;">✔</span></p> <p>VD-PCR: Yes <span style="color: red;">✘</span></p>
<p><b>Image &amp; Caption</b></p>  <p>A group of people standing in front of a table with pizza on it</p>	<p><b>Predictions</b></p> <p>Q: About how many people are there? A: 6</p> <p>Q: Are they adults? A: Yes</p> <p>Q: What are their genders? A: 2 women and 4 men</p> <p>Q: Do people have plates in their hands? A: No</p> <p>Q: Can you see a reflection in the mirror? A: Yes</p> <p>VD-GR: Yes <span style="color: green;">✔</span></p> <p>VD-PCR: Yes <span style="color: green;">✔</span></p>

FIGURE C.5: Qualitative samples from the *val* split of the VisDial v1.0 dataset. We compared the predictions of VDGR (orange) with the ground truth answers (blue) and the predictions of VD-PCR (pink). The answers of both models correspond to the top-1 predictions.

## Video



## Dialog History

C: A guy is sweeping the floor when he decides to go into the hallway to retrieve 2 pairs of shoes to put into the washing machine.

Q<sub>1</sub>: What is the gender? A<sub>1</sub>: It looks like a male to me.  
⋮

## Question

Q<sub>10</sub>: Anything else before that?

## Video



## Dialog History

C: A man is sitting on top of his bed, typing on his laptop computer [...] He continues typing on the laptop even while talking on the phone.

Q<sub>1</sub>: Is the man typing on the computer?  
A<sub>1</sub>: Yes the man is typing at the computer.  
⋮

## Question

Q<sub>1</sub>: Does he talk after picking it up?

## Video



## Dialog History

C: A person is cleaning a window with a rag. Then, puts the rag down, takes the blankets off the bed and sits on the bed as they stare out the window.

Q<sub>1</sub>: How does the video start?  
A<sub>1</sub>: It starts with a man cleaning a window with a rag.  
⋮

## Question

Q<sub>1</sub>: Is there sound?

## Video



## Dialog History

C: A man stands in a room while eating something and watching tv. He then picks up a plate and slams it to the floor.

Q<sub>1</sub>: How many people are in the video?  
A<sub>1</sub>: One person is in the video.  
⋮

## Question

Q<sub>9</sub>: Does he exit the filming area?

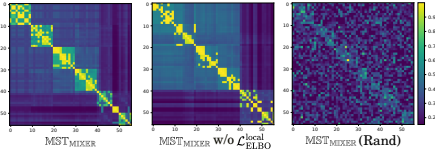
## Predictions

MST<sub>MIXER</sub>: He did not do anything else before that (B-4 = 70.71)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : He was sweeping the floor (B-4 = 2.23e-10)

MST<sub>MIXER</sub> (Rand): He went to the sink with his shoes (B-4 = 8.78e-7)

## Latent Global Graphs



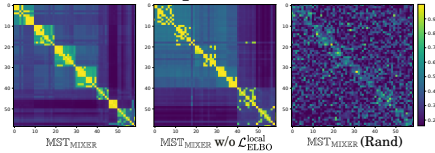
## Predictions

MST<sub>MIXER</sub>: Yes, he is talking on the phone (B-4 = 86.68)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : No, he does not talk at all (B-4 = 9.05e-11)

MST<sub>MIXER</sub> (Rand): No, he does not talk at all (B-4 = 9.05e-11)

## Latent Global Graphs



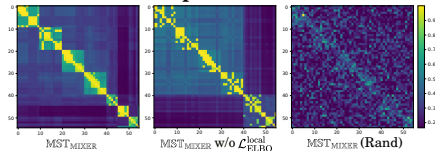
## Predictions

MST<sub>MIXER</sub>: Yes there is sound in the video (B-4 = 99.99)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : No there is no sound (B-4 = 9.05e-7)

MST<sub>MIXER</sub> (Rand): No there is no sound (B-4 = 9.05e-7)

## Latent Global Graphs



## Predictions

MST<sub>MIXER</sub>: Yes he does exit the filming area (B-4 = 80.91)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : No he does not exit the filming area (B-4 = 41.11)

MST<sub>MIXER</sub> (Rand): No he does not exit the room (B-4 = 9.87e-7)

## Latent Global Graphs

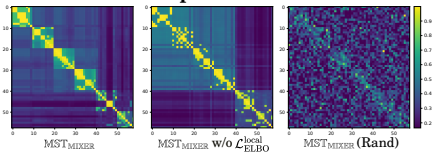


FIGURE C.6: Qualitative samples of MST<sub>MIXER</sub> from the test split of AVSD-DSTC7.

## Video



## Dialog History

C: A man sits in a chair next to a bed with his head on his shoulder . [...] he raises his head and stands up . [...] then picks up a cup and drinks from it .

Q<sub>1</sub>: What is happening?

A<sub>1</sub>: A man is sitting then wakes up .

## Question

Q<sub>2</sub>: Is the video set in the bedroom?

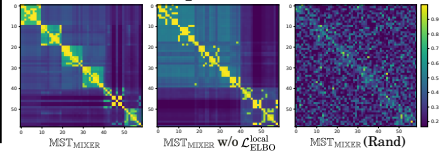
## Predictions

MST<sub>MIXER</sub>: Yes the video is set in the bedroom (B-4 = 69.14)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : Yes it is in the bedroom (B-4 = 99.99)

MST<sub>MIXER</sub> (Rand): Yes it is in the bedroom (B-4 = 99.99)

## Latent Global Graphs



## Video



## Dialog History

C: A man sits in a doorway watching tv in the room behind him. He puts several cups on a table and fills one with a metal pitcher.

Q<sub>1</sub>: What happens at the start of the video?

A<sub>1</sub>: A man is sitting in a doorway watching tv behind him .

## Question

Q<sub>2</sub>: Does he drink from the cup at all?

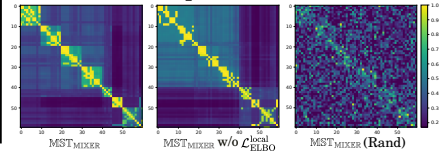
## Predictions

MST<sub>MIXER</sub>: Yes he does drink from the cup (B-4 = 42.38)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : No he does not drink from the cup (B-4 = 99.99)

MST<sub>MIXER</sub> (Rand): no he does not drink from the cup (B-4 = 99.99)

## Latent Global Graphs



## Video



## Dialog History

C: A man is watching tv as he grabs a piece of bread and takes a bite. He grabs a cup and drinks from it as he continues to watch tv.

Q<sub>1</sub>: What is happening in this video?

A<sub>1</sub>: A young man is watching tv .

## Question

Q<sub>2</sub>: What is he watching on tv ?

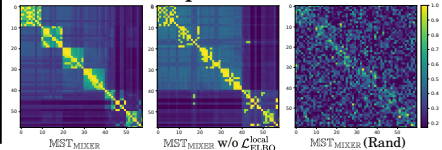
## Predictions

MST<sub>MIXER</sub>: I can't tell what is on the tv. (B-4 = 9.87e-7)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : He is watching tv (B-4 = 6.38e-3)

MST<sub>MIXER</sub> (Rand): He is watching tv (B-4 = 6.38e-3)

## Latent Global Graphs



## Video



## Dialog History

C: A man with an orange shoebox [...] . He then reaches to the side to grab a pair of jeans. He places the jeans inside of the shoebox, closes it, and walks away.

Q<sub>1</sub>: How many people are in the video ?

A<sub>1</sub>: It is just this one man .

## Question

Q<sub>2</sub>: What sort of room are they in?

## Predictions

MST<sub>MIXER</sub>: It looks like a living room (B-4 = 4.95e-3)

MST<sub>MIXER</sub> w/o  $\mathcal{L}_{ELBO}^{local}$ : It looks like a living room (B-4 = 4.95e-3)

MST<sub>MIXER</sub> (Rand): It looks like a living room (B-4 = 4.95e-3)

## Latent Global Graphs

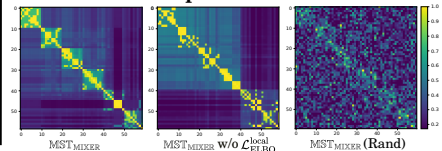


FIGURE C.7: Negative qualitative samples of MST<sub>MIXER</sub> from the test split of AVSD-DSTC7.







 <p>C A man is sitting in a room facing a stand with a laptop on it smiling at a camera, sneezes twice, grabs medicine and takes it.</p> <p>H<sub>r</sub> Is the boy in the room when the video starts ? Yes, the boy is in the room from the start.</p> <p>Q<sub>r</sub> What is he doing on the bed? He is sitting on the bed.</p> <p>A<sub>r</sub> He is sitting on the bed and looking at the camera.</p> <p>✓</p>	 <p>C A woman gets up out of bed and puts a sweatshirt on. She goes over to get a hairbrush and starts brushing her hair while looking in the mirror.</p> <p>H<sub>r</sub> Is the person sleeping to begin with? Yes, she is sleeping and then gets up from her bed.</p> <p>Q<sub>r</sub> Does she stretch once she gets up? Yes, she does stretch once she gets up.</p> <p>A<sub>r</sub> No, she never stretches in the video.</p> <p>✗</p>
 <p>C A man is in the bathroom, he takes off his shirt and drops it on the floor [...]</p> <p>H<sub>r</sub> What is the man doing in the beginning of the video? He's standing in the bathroom looking around. [...]</p> <p>Q<sub>r</sub> After he picks up the items what does he do? He starts wiping the cabinets.</p> <p>A<sub>r</sub> He sprayed the cabinet and wiped it down.</p> <p>✓</p>	 <p>C A man is sitting on a sofa and throwing clothes into a bag. He picks up the bag and walks out an open door.</p> <p>H<sub>r</sub> How many people are in the video? I can see one but someone else is talking.</p> <p>Q<sub>r</sub> What is the man in the video doing? Sitting on a couch.</p> <p>A<sub>r</sub> He's gathering clothes and putting them in a bag.</p> <p>✗</p>
 <p>C A woman holding a cup bends down to put it in a bucket next to her. She then grabs a broom and starts sweeping the floor .</p> <p>H<sub>r</sub> How does the video start? A lady puts a can in to a bucket. [...]</p> <p>Q<sub>r</sub> Then what does she do? She is sweeping the floor.</p> <p>A<sub>r</sub> She sweeps the floor with a broom.</p> <p>✓</p>	 <p>C A boy stands trying to fix something with a tool. He then comes into the hallway and stretches. Then he walks further to pick up a cup and drink.</p> <p>H<sub>r</sub> No dialog history available</p> <p>Q<sub>r</sub> Where does the video happens? In a home.</p> <p>A<sub>r</sub> This appears to be in a bathroom.</p> <p>✗</p>

FIGURE C.8: Zero-shot qualitative examples of  $V^2$ Dial on AVSD. 🤖 = generated answers, 🏆 = golden ground-truth answers, ✓ / ✗ = success / failure cases).

 <p>C The people are standing on the tennis courts outside.</p> <p>H<sub>r</sub> Is it sunny ?</p> <p>A<sub>r</sub> Yes, but i can't see the sky. [-]</p> <p>Q<sub>r</sub> Are they wearing shirts?</p> <p>A<sub>r</sub> Yes, they are wearing shirts. [1] Yes, they are all wearing shirts. [1]</p> <p>✓ Yes, they are all wearing shirts. [1]</p>	 <p>C A family of 3 is on a beach flying a kite.</p> <p>H<sub>r</sub> Is the photo in color?</p> <p>A<sub>r</sub> Yes. [-]</p> <p>Q<sub>r</sub> Is the family comprised of parents and 1 kid?</p> <p>A<sub>r</sub> Yes, there is a child. [1] A boy. [1]</p> <p>✗ Yes. [1]</p>
 <p>C A zebra looks down at the ground as it walks across a plainfield.</p> <p>H<sub>r</sub> Do the trees have leaves?</p> <p>A<sub>r</sub> Yes. [-]</p> <p>Q<sub>r</sub> Are they green or fall color?</p> <p>A<sub>r</sub> They appear to be green. [1] Green. [1]</p> <p>✓ Green. [1]</p>	 <p>C A man driving a horse drawn carriage down a road.</p> <p>H<sub>r</sub> What color is the carriage ?</p> <p>A<sub>r</sub> White with green seats. [-]</p> <p>Q<sub>r</sub> Does it have a top on it?</p> <p>A<sub>r</sub> Yes it does have a top on it. [1] Yes it does! [1]</p> <p>✗ Yes. [1]</p>
 <p>C A small child on the beach, walking past a stop sign.</p> <p>H<sub>r</sub> Is the child wearing a bathing suit ?</p> <p>A<sub>r</sub> No, a winter coat. [-]</p> <p>Q<sub>r</sub> Is she alone or with someone?</p> <p>A<sub>r</sub> She is alone on the beach. [1] Alone. [1]</p> <p>✓ Alone. [1]</p>	 <p>C A brown bench sits in a grassy field full of wildflowers.</p> <p>H<sub>r</sub> Are the flowers colorful?</p> <p>A<sub>r</sub> They are pink and white. [-]</p> <p>Q<sub>r</sub> Is there a path or sidewalk?</p> <p>A<sub>r</sub> I don 't see a path or sidewalk. [1] No there is no sidewalk. [1]</p> <p>✗ I can't see that. [1]</p>

FIGURE C.9: Zero-shot qualitative examples of  $V^2$ Dial on VisDial. [1] = top ranked candidate answers,







 <p><b>C</b> Man watches television on a rug placed on the floor. [...] and he stands to grab a pillow, which he drops to the floor.</p> <p><b>H<sub>i</sub></b> How many people are in the video? There is only one person, which is the man. [...]</p> <p><b>Q<sub>i</sub></b> What is the man doing on the floor? <b>A<sub>i</sub></b> He is sitting on the floor watching tv. 🤖 He was watching tv on the floor. 📊</p> <p>✓</p>	 <p><b>C</b> The man is in the kitchen making something. [...] The man is holding a book in his hand then walks away.</p> <p><b>H<sub>i</sub></b> How many people are in the video? One person is in the video. [...]</p> <p><b>Q<sub>i</sub></b> After he picks up the items what does he do? <b>A<sub>i</sub></b> He is opening the fridge. 🤖 It seems to me he is prepping something. 📊</p> <p>✗</p>
 <p><b>C</b> A woman gets up out of bed and puts a sweatshirt on. She goes over to get a hairbrush and starts brushing her hair while looking in the mirror.</p> <p><b>H<sub>i</sub></b> Is the person sleeping to begin with? Yes, she is sleeping and then gets up from her bed.</p> <p><b>Q<sub>i</sub></b> Does she stretch once she gets up? <b>A<sub>i</sub></b> No, she does not stretch. 🤖 No, she never stretches in the video. 📊</p> <p>✓</p>	 <p><b>C</b> A guy puts a lid back on a bottle and sets it down. He drinks something. Then he backs up and points to the cat on the floor and smiles.</p> <p><b>H<sub>i</sub></b> Is the man wearing one shoe? yes he only has one shoe. [...]</p> <p><b>Q<sub>i</sub></b> What does he do after laughing? <b>A<sub>i</sub></b> He sets the cup down. 🤖 That is how the video ends. 📊</p> <p>✗</p>
 <p><b>C</b> A person is working on something at a kitchen counter. He then walks and opens a fridge before closing it to reach for something on a shelf.</p> <p><b>H<sub>i</sub></b> How many people are in the video? There is just one man in the video. [...]</p> <p><b>Q<sub>i</sub></b> Is the kitchen clean or messy? <b>A<sub>i</sub></b> The kitchen appears to be clean. 🤖 The kitchen appears to be very clean. 📊</p> <p>✓</p>	 <p><b>C</b> This guy is sitting with his phone in the kitchen and reading it. He eats part of a snack then his friend joins him to look at it, too.</p> <p><b>H<sub>i</sub></b> Are there just the two people? Yeah there are just two people. [...]</p> <p><b>Q<sub>i</sub></b> Does anything else happen after that? <b>A<sub>i</sub></b> No that is all that happens. 🤖 He take a photo out of the window. 📊</p> <p>✗</p>

FIGURE C.10: Fine-tuning qualitative examples of  $V^2$ Dial on AVSD.



FIGURE C.1 1: Fine-tuning qualitative examples of  $V^2$ Dial on VisDial.



# Bibliography

- [1] Adnen Abdessaied, Mihai Băce, and Andreas Bulling. Neuro-Symbolic Visual Dialog. In *COLING*, 2022.
- [2] Adnen Abdessaied, Manuel Hochmeister, and Andreas Bulling. OLViT: Multi-modal state tracking via attention-based embeddings for video-grounded dialog. In *COLING*, 2024.
- [3] Adnen Abdessaied, Anna Rohrbach, Marcus Rohrbach, and Andreas Bulling. V<sup>2</sup>Dial: Unification of Video and Visual Dialog via Multimodal Experts. In *CVPR*, 2025.
- [4] Adnen Abdessaied, Lei Shi, and Andreas Bulling. Multi-modal video dialog state tracking in the wild. In *ECCV*, 2024.
- [5] Adnen Abdessaied, Lei Shi, and Andreas Bulling. VD-GR: Boosting Visual Dialog With Cascaded Spatial-Temporal Multi-Modal Graphs. In *WACV*, 2024.
- [6] Adnen Abdessaied, Ekta Sood, and Andreas Bulling. Video language co-attention with multimodal fast-learning feature fusion for VideoQA. In *Repl4NLP Workshop @ ACL*, 2022.
- [7] Shubham Agarwal, Trung Bui, Joon-Young Lee, Ioannis Konstas, and Verena Rieser. History for Visual Dialog: Do we really need it? In *ACL*, 2020.
- [8] Huda Alamri, Anthony Bilic, Michael Hu, Apoorva Beedu, and Irfan Essa. End-to-end multimodal representation learning for video dialog. In *NeurIPS*, 2022.
- [9] Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori

- Hori, Peter Anderson, et al. Audio visual scene-aware dialog. In *CVPR*, 2019.
- [10] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.
- [11] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reiter. Fusion of Detected Objects in Text for Visual Question Answering. In *EMNLP*, 2019.
- [12] Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitriy Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. Semantic machines et al. task-oriented dialogue as dataflow synthesis. *TACL*, 2020.
- [13] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Deep compositional question answering with neural module networks. In *CVPR*, 2016.
- [14] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *NAACL*, 2016.
- [15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

- [16] Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom M. Mitchell. Conversational neuro-symbolic commonsense reasoning. In *AAAI*, 2021.
- [17] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Esam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. MiniGPT4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. In *CVPRW*, 2024.
- [18] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021.
- [19] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.
- [20] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. VLMo: Unified vision-language pre-training with mixture-of-modality-experts. In *NeurIPS*, 2022.
- [21] Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K. Surikuchi, Ece Takmaz, and Alberto Testoni. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *CoRR*, 2024.
- [22] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.
- [23] Matteo Bortoletto, Constantin Ruhdorfer, Adnen Abdessaied, Lei Shi, and Andreas Bulling. Limits of Theory of Mind Modelling in Dialogue-Based Collaborative Plan Acquisition. In *ACL*, 2024.

- [24] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, et al. Language models are few-shot learners. In *NeurIPS*, 2020.
- [25] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral Networks and Locally Connected Networks on Graphs. In *ICLR*, 2014.
- [26] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *EMNLP*, 2018.
- [27] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. In *arXiv, 1901.11390*, 2019.
- [28] Min Cao, Shiping Li, Juntao Li, Liqiang Nie, and Min Zhang. Image-text retrieval: A survey on recent research and development. In *IJCAI*, 2022.
- [29] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to Rank: From Pairwise Approach to Listwise Approach. In *ICML*, 2007.
- [30] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [31] Soravit Changpinyo, Piyush Sharma, and Nan Ding and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *CVPR*, 2021.
- [32] Cheng Chen, Yudong Zhu, Zhenshan Tan, Qingrong Cheng, Xin Jiang, Qun Liu, and Xiaodong Gu. UTC: A Unified Transformer

- with Inter-Task Contrastive Learning for Visual Dialog. In *CVPR*, 2022.
- [33] Feilong Chen, Xiuyi Chen, Fandong Meng, Peng Li, and Jie Zhou. GoG: Relation-aware graph-over-graph network for visual dialog. In *Findings of ACL*, 2021.
- [34] Feilong Chen, Xiuyi Chen, Can Xu, and Daxin Jiang. Learning to ground visual objects for visual dialog. In *Findings of EMNLP*, 2021.
- [35] Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *EMNLP*, 2024.
- [36] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.
- [37] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, 2015.
- [38] Xinyun Chen, Dawn Song, and Yuandong Tian. Latent execution for neural program synthesis. In *NeurIPS*, 2021.
- [39] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.
- [40] Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *NeurIPS*, 2020.
- [41] Zhe Chen, Hongcheng Liu, and Yu Wang. DialogMCF: Multimodal Context Flow for Audio Visual Scene-Aware Dialog. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

- [42] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. VindLU: A Recipe for Effective Video-and-Language Pretraining. In *CVPR*, 2023.
- [43] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. VideoLLaMA 2: Advancing Spatial-Temporal Modeling and Audio Understanding in Video-LLMs. *arXiv preprint arXiv:2406.07476*, 2024.
- [44] Yun-Wei Chu, Kuan-Yen Lin, Chao-Chun Hsu, and Lun-Wei Ku. Multi-step joint-modality attention network for scene-aware dialogue system. In *DSTC Workshop @ AAI*, 2020.
- [45] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *JMLR*, 2024.
- [46] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 2007.
- [47] Daniel Cores, Michael Dorckenwald, Manuel Mucientes, Cees G. M. Snoek, and Yuki M. Asano. TVBench: Redesigning Video-Language Evaluation. *arXiv preprint arXiv:2410.07752*, 2024.
- [48] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. Visual Dialog. In *CVPR*, 2017.
- [49] Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. In *ICCV*, 2017.
- [50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.
- [51] David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matthew Botvinick. Attention over Learned Object Embeddings Enables Complex Visual Reasoning. In *NeurIPS*, 2021.

- [52] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.
- [53] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: Efficient scaling of language models with mixture-of-experts. In *ICML*, 2022.
- [54] Pantelis Elinas, Edwin V Bonilla, and Louis Tiao. Variational inference for graph convolutional networks in the absence of graph data and adversarial settings. *NeurIPS*, 2020.
- [55] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation. In *AAAI*, 2017.
- [56] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *ICML*, 2019.
- [57] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *arXiv preprint arXiv:2306.13394*, 2023.
- [58] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. *arXiv preprint arXiv:2405.21075*, 2024.
- [59] Zhe Gan, Yu Cheng, Ahmed El Kholy, Linjie Li, Jingjing Liu, and Jianfeng Gao. Multi-step reasoning via recurrent dual attention for visual dialog. In *ACL*, 2019.
- [60] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR*, 2019.

- [61] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for Compositional Actions and Temporal Reasoning. In *ICLR*, 2020.
- [62] Dan Guo, Hui Wang, and Meng Wang. Dual visual attention network for visual dialog. In *IJCAI*, 2019.
- [63] Dan Guo, Hui Wang, Hanwang Zhang, Zheng Jun Zha, and Meng Wang. Iterative Context-Aware Graph Inference for Visual Dialog. In *CVPR*, 2020.
- [64] Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogerio Feris. Dialog-based interactive image retrieval. *NeurIPS*, 31, 2018.
- [65] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023.
- [66] Kelvin Guu, Panupong Pasupat, Evan Liu, and Percy Liang. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *ACL*, 2017.
- [67] Christopher Hahn, Frederik Schmitt, Jens U. Kreber, Markus Norman Rabe, and Bernd Finkbeiner. Teaching Temporal Logics to Neural Networks. In *ICLR*, 2021.
- [68] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2017.
- [69] Chi Han, Jiayuan Mao, Chuang Gan, Joshua B. Tenenbaum, and Jiajun Wu. Visual Concept Metaconcept Learning. In *NeurIPS*, 2019.
- [70] Seungju Han, Jack Hessel, Nouha Dziri, Yejin Choi, and Young-jae Yu. CHAMPAGNE: Learning Real-world Conversation from Large-Scale Web Videos. In *ICCV*, 2023.
- [71] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.
- [72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

- [73] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 1997.
- [74] Chiori Hori, Huda Alamri, Jue Wang, Gordon Wichern, Takaaki Hori, Anoop Cherian, Tim K. Marks, Vincent Cartillier, Raphael Gontijo Lopes, Abhishek Das, Irfan Essa, Dhruv Batra, and Devi Parikh. End-to-end audio visual scene-aware dialog using multimodal attention-based video features. In *ICASSP*, 2019.
- [75] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, 2017.
- [76] Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *CVPR*, 2024.
- [77] Xin Huang, Hui Li Tan, Mei Chee Leong, Ying Sun, Liyuan Li, Ridong Jiang, and JJ Kim. Investigation on transformer-based multi-modal fusion for audio-visual scene-aware dialog. In *DSTC10 Workshop @ AAI*, 2022.
- [78] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. In *ICLR*, 2018.
- [79] Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. “Do you follow me?”: A Survey of Recent Approaches in Dialogue State Tracking. In *SIGDIAL*, 2022.
- [80] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Léo Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of Experts. *arXiv preprint arXiv:2401.04088*, 2024.

- [81] Pin Jiang and Yahong Han. Reasoning with heterogeneous graph alignment for video question answering. In *AAAI*, 2020.
- [82] Xiaoze Jiang, Jing Yu, Zengchang Qin, Yingying Zhuang, Xingxing Zhang, Yue Hu, and Qi Wu. DualVD: An Adaptive Dual Encoding Model for Deep Visual Understanding in Visual Dialogue. In *AAAI*, 2020.
- [83] Weike Jin, Zhou Zhao, Mao Gu, Jun Xiao, Furu Wei, and Yueting Zhuang. Video dialog via progressive inference and cross-transformer. In *EMNLP-IJCNLP*, 2019.
- [84] Yao Jin, Guocheng Niu, Xinyan Xiao, Jian Zhang, Xi Peng, and Jun Yu. Knowledge-Constrained Answer Generation for Open-Ended Video Question Answering. In *AAAI*, 2023.
- [85] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR*, 2017.
- [86] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017.
- [87] Gi-Cheon Kang, Sungdong Kim, Jin-Hwa Kim, Donghyun Kwak, and Byoung-Tak Zhang. The Dialog Must Go On: Improving Visual Dialog via Generative Self-Training. In *CVPR*, 2023.
- [88] Gi-Cheon Kang, Jaeseo Lim, and Byoung-Tak Zhang. Dual attention networks for visual reference resolution in visual dialog. In *EMNLP*, 2019.
- [89] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [90] Junyeong Kim, Sunjae Yoon, Dahyun Kim, and Chang D. Yoo. Structured co-reference graph attention for video-grounded dialogue. In *AAAI*, 2021.

- 
- [91] Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*, 2019.
- [92] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [93] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.
- [94] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, 2023.
- [95] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *ICML*, 2023.
- [96] Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? text-to-image coreference. In *CVPR*, 2014.
- [97] Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *EMNLP*, 2021.
- [98] Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *ECCV*, 2018.
- [99] Satwik Kottur, José M.F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. CLEVR-Dialog: A diagnostic dataset for multi-round reasoning in visual dialog. In *NAACL*, 2019.
- [100] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

- 
- [101] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *ICLR*, 2020.
- [102] Hung Le, Nancy F. Chen, and Steven Hoi. Learning reasoning paths over semantic graphs for video-grounded dialogues. In *ICLR*, 2021.
- [103] Hung Le, Nancy F. Chen, and Steven C. H. Hoi. VGNMN: video-grounded neural module network to video-grounded language tasks. In *NAACL*, 2022.
- [104] Hung Le, Nancy F. Chen, and Steven C.H. Hoi. Multimodal Dialogue State Tracking. In *NAACL*, 2022.
- [105] Hung Le and Steven C.H. Hoi. Video-Grounded Dialogues with Pretrained Generation Language Models. In *ACL*, 2020.
- [106] Hung Le, Doyen Sahoo, Nancy Chen, and Steven Hoi. Multimodal transformer networks for end-to-end video-grounded dialogue systems. In *ACL*, 2019.
- [107] Hung Le, Doyen Sahoo, Nancy Chen, and Steven C.H. Hoi. BiST: Bi-directional Spatio-Temporal Reasoning for Video-Grounded Dialogues. In *EMNLP*, 2020.
- [108] Hung Le, Chinnadhurai Sankar, Seungwhan Moon, Ahmad Beirami, Alborz Geramifard, and Satwik Kottur. DVD: A diagnostic dataset for multi-step reasoning in video grounded dialogue. In *ACL*, 2021.
- [109] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *CVPR*, 2020.
- [110] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *IEEE*, 1998.
- [111] Haeju Lee, Oh Joon Kwon, Yunseon Choi, Minho Park, Ran Han, Yoonhyung Kim, Jinhyeon Kim, Youngjune Lee, Haebin Shin, Kangwook Lee, and Kee-Eung Kim. Learning to embed multi-modal contexts for situated conversational agents. In *NAACL-Findings*, July 2022.

- [112] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.
- [113] Chaozhuo Li, Bochen Pang, Yuming Liu, Hao Sun, Zheng Liu, Xing Xie, Tianqi Yang, Yanling Cui, Liangjie Zhang, and Qi Zhang. AdsGNN: Behavior-Graph Augmented Relevance Modeling in Sponsored Search. In *ACM SIGIR*, SIGIR '21, 2021.
- [114] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. DeepGCNs: Can GCNs Go As Deep As CNNs? In *ICCV*, 2019.
- [115] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICLR*, 2023.
- [116] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *NeurIPS*, 2021.
- [117] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBenchmark: A Comprehensive Multi-modal Video Understanding Benchmark. In *CVPR*, 2024.
- [118] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. In *arXiv:1908.03557*, 2019.
- [119] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks. In *ICLR*, 2015.
- [120] Yunshui Li, Binyuan Hui, ZhiChao Yin, Min Yang, Fei Huang, and Yongbin Li. PaCE: Unified Multi-modal Dialogue Pre-training with Progressive and Compositional Experts. In *ACL*, 2023.
- [121] Zekang Li, Zongjia Li, Jinchao Zhang, Yang Feng, and Jie Zhou. Bridging text and video: A universal multimodal transformer

- for audio-visual scene-aware dialog. *Transactions on Audio, Speech, and Language Processing*, 2021.
- [122] Percy Liang, Michael I. Jordan, and Dan Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 2013.
- [123] Bin Lin, Zhenyu Tang, Yang Ye, Jiayi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Yatian Pang, Munan Ning, and Li Yuan. MoE-LLaVA: Mixture of Experts for Large Vision-Language Models. *arXiv preprint arXiv:2401.15947*, 2024.
- [124] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.
- [125] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van den Hengel. Deeply Learning the Messages in Message Passing Inference. In *NeurIPS*, 2015.
- [126] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.
- [127] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pre-training Approach. In *arXiv, 1907.11692*, 2019.
- [128] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, 2024.
- [129] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Fine-grained Fact Verification with Kernel Graph Attention Network. In *ACL*, 2020.
- [130] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. In *ICLR*, 2019.
- [131] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019.

- [132] Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, and Dhruv Batra. Best of Both Worlds: Transferring Knowledge from Discriminative Learning to a Generative Visual Dialog Model. In *NeurIPS*, 2017.
- [133] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In *ACL*, 2024.
- [134] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv preprint arXiv:2406.09418*, 2024.
- [135] Mateusz Malinowski and Mario Fritz. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In *NeurIPS*, 2014.
- [136] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *ICLR*, 2019.
- [137] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.
- [138] Seungwhan Moon, Satwik Kottur, Paul A. Crook, Ankita De, Shivani Poddar, Theodore Levin, David Whitney, Daniel Difranco, Ahmad Beirami, Eunjoon Cho, Rajen Subba, and Alborz Geramifard. Situated and interactive multimodal conversations. In *COLING*, 2020.
- [139] Xiangyang Mou, Brandyn Sigouin, Ian Steenstra, and Hui Su. Multimodal dialogue state tracking by QA approach with data augmentation. In *DSTC8 Workshop @ AACL*, 2020.
- [140] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *ACL*, 2017.

- [141] Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. In *ECCV*, 2020.
- [142] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal Contrastive Learning with LIMoE: the Language-Image Mixture of Experts. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *NeurIPS*, 2022.
- [143] Van Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. Efficient Attention Mechanism for Visual Dialog that Can Handle All the Interactions Between Multiple Inputs. In *ECCV*, 2020.
- [144] Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. Recursive Visual Attention in Visual Dialog. In *CVPR*, 2019.
- [145] Vahid Noroozi, Yang Zhang, Evelina Bakhturina, and Tomasz Kornuta. A fast and robust bert-based dialogue state tracker for schema-guided dialogue dataset. In *KDD workshop on Conversational Systems Towards Mainstream Adoption*, 2020.
- [146] Wei Pang and Xiaojie Wang. Visual dialogue state tracking for question generation. In *AAAI*, 2020.
- [147] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.
- [148] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019.
- [149] Badri N. Patro, Anupriy, and Vinay P. Namboodiri. Probabilistic framework for solving Visual Dialog. *Pattern Recognit.*, 2019.

- [150] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.
- [151] Hoang-Anh Pham, Thao Minh Le, Vuong Le, Tu Minh Phuong, and Truyen Tran. Video Dialog as Conversation about Objects Living in Space-Time. In *ECCV*, 2022.
- [152] Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. In *CVPR*, 2020.
- [153] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *ACL*, 2020.
- [154] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.
- [155] Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei. Linking people in videos with “their” names using coreference resolution. In *ECCV*, 2014.
- [156] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015.
- [157] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.
- [158] Anna Rohrbach, Marcus Rohrbach, Siyu Tang, Seong Joon Oh, and Bernt Schiele. Generating descriptions with grounded and co-referenced people. In *CVPR*, 2017.
- [159] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of the Workshop on Energy Efficient Machine Learning and Cognitive Computing, NeurIPS*, 2019.

- [160] Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G. Schwing. Factor Graph Attention. In *CVPR*, 2019.
- [161] Ankit Shah, Shijie Geng, Peng Gao, Anoop Cherian, Takaaki Hori, Tim K Marks, Jonathan Le Roux, and Chiori Hori. Audio-visual scene-aware dialog and reasoning using audio-visual transformers with joint student-teacher learning. In *ICASSP*, 2022.
- [162] Ankit P. Shah, Shijie Geng, Peng Gao, Anoop Cherian, Takaaki Hori, Tim K. Marks, Jonathan Le Roux, and Chiori Hori. Audio-Visual Scene-Aware Dialog and Reasoning using Audio-Visual Transformers with Joint Student-Teacher Learning. In *Inter-speech*, 2019.
- [163] Muhammad Shah, Shikib Mehri, and Tejas Srinivasan. Reasoning Over History: Context Aware Visual Dialog. In *ACL Workshop on Natural Language Processing Beyond Text*, 2020.
- [164] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.
- [165] Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. Visually grounded neural syntax acquisition. In *ACL*, 2019.
- [166] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.
- [167] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [168] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *CVPR*, 2022.
- [169] Florian Strub, Harm De Vries, Jeremie Mary, Bilal Piot, Aaron Courville, and Olivier Pietquin. End-to-end optimization of

- goal-driven and visually grounded dialogue systems. In *IJCAI*, 2017.
- [170] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *ICLR*, 2020.
- [171] Alane Suhr, Srinivasan Iyer, and Yoav Artzi. Learning to map context-dependent sentences to executable formal queries. In *NAACL*, 2018.
- [172] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *NeurIPS*, 2016.
- [173] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved Training Techniques for CLIP at Scale. *arXiv preprint arXiv:2303.15389*, 2023.
- [174] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative Pretraining in Multimodality. In *ICLR*, 2024.
- [175] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *CVPR*, 2023.
- [176] Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 2019.
- [177] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017.
- [178] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based Image Description Evaluation. In *CVPR*, 2015.
- [179] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.
- [180] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language:

- Beit pretraining for vision and vision-language tasks. In *CVPR*, 2023.
- [181] Yue Wang, Shafiq Joty, Michael R. Lyu, Irwin King, Caiming Xiong, and Steven C.H. Hoi. VD-BERT: A unified vision and dialog transformer with BERT. In *EMNLP*, 2020.
- [182] Zihao Wang, Junli Wang, and Changjun Jiang. Unified multimodal model with unlikelihood training for visual dialog. In *ACM MM*, 2022.
- [183] Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*, 2017.
- [184] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1989.
- [185] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural Scene De-rendering. In *CVPR*, 2017.
- [186] Qitian Wu, Wentao Zhao, Zenan Li, David Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. In *NeurIPS*, 2022.
- [187] Yaxiong Wu, Craig Macdonald, and Iadh Ounis. Multi-modal dialog state tracking for interactive fashion recommendation. In *ACM RecSys*, 2022.
- [188] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.
- [189] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.
- [190] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, 2017.

- [191] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.
- [192] Puyang Xu and Qi Hu. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *ACL*, 2018.
- [193] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph. In *NeurIPS*, 2021.
- [194] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph. In *NeurIPS*, 2021.
- [195] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018.
- [196] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision Events for Video Representation and Reasoning. In *ICLR*, 2020.
- [197] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*, 2018.
- [198] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *NeurIPS*, 2021.
- [199] Sunjae Yoon, Eunseop Yoon, Hee Suk Yoon, Junyeong Kim, and Chang Yoo. Information-theoretic text hallucination reduction for video-grounded dialogue. In *EMNLP*, 2022.
- [200] Koichiro Yoshino, Chiori Hori, Julien Perez, Luis Fernando D’Haro, Lazaros Polymenakos, Chulaka Gunasekara, Walter S Lasecki, Jonathan K Kummerfeld, Michel Galley, Chris Brockett, et al. Dialog system technology challenge 7. *arXiv preprint arXiv:1901.03461*, 2019.

- [201] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014.
- [202] Xintong Yu, Hongming Zhang, Ruixin Hong, Yangqiu Song, and Changshui Zhang. VD-PCR: Improving visual dialog with pronoun coreference resolution. *Pattern Recognition*, 2022.
- [203] Xintong Yu, Hongming Zhang, Yangqiu Song, Yan Song, and Changshui Zhang. What You See is What You Get: Visual Pronoun Coreference Resolution in Dialogues. In *EMNLP-IJCNLP*, 2019.
- [204] Youngjae Yu, Jiwan Chung, Heeseung Yun, Jack Hessel, Jae Sung Park, Ximing Lu, Rowan Zellers, Prithviraj Ammanabrolu, Ronan Le Bras, Gunhee Kim, et al. Fusing pre-trained language models with multimodal prompts through reinforcement learning. In *CVPR*, 2023.
- [205] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *ICML*, 2019.
- [206] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, 2017.
- [207] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From Recognition to Cognition: Visual Commonsense Reasoning. In *CVPR*, June 2019.
- [208] Haoyu Zhang, Meng Liu, Yaowei Wang, Da Cao, Weili Guan, and Liqiang Nie. Uncovering hidden connections: Iterative tracking and reasoning for video-grounded dialog. *arxiv preprint arxiv:2310.07259*, 2024.
- [209] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. GreaseLM: Graph REASONing Enhanced Language Models. In *ICLR*, 2022.
- [210] Zhou Zhao, Xinghua Jiang, Deng Cai, Jun Xiao, Xiaofei He, and Shiliang Pu. Multi-turn video question answering via multi-stream hierarchical attention context network. In *IJCAI*, 2018.

- [211] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *NeurIPS Datasets and Benchmarks Track*, 2023.
- [212] Zilong Zheng, Wenguan Wang, Siyuan Qi, and Song-Chun Zhu. Reasoning Visual Dialogs with Structural and Partial Observations. In *CVPR*, 2019.
- [213] Zexuan Zhong, Mengzhou Xia, Danqi Chen, and Mike Lewis. Lory: Fully Differentiable Mixture-of-Experts for Autoregressive Language Model Pre-training. In *COLM*, 2024.
- [214] Luwei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *AAAI*, 2020.
- [215] Cunjuan Zhu, Qi Jia, Wei Chen, Yanming Guo, and Yu Liu. Deep learning for video-text retrieval: a review. *International Journal of Multimedia Information Retrieval*, 2023.
- [216] Jason Zhu, Yanling Cui, Yuming Liu, Hao Sun, Xue Li, Markus Pelger, Tianqi Yang, Liangjie Zhang, Ruofei Zhang, and Huasha Zhao. TextGNN: Improving Text Encoder via Graph Neural Network in Sponsored Search. In *WWW*, 2021.
- [217] Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. Llama-moe: Building mixture-of-experts from llama with continual pre-training. In *EMNLP*, 2024.