Reasoning Visual Dialog with Sparse Graph Learning and Knowledge Transfer

Gi-Cheon Kang
Seoul National University, AIIS
chonkang@snu.ac.kr

Junseok Park
Seoul National University
jspark227@snu.ac.kr

Hwaran Lee
NAVER AI Lab
hwaran.lee@navercorp.com

Byoung-Tak Zhang†
Seoul National University, AIIS
btzhang@snu.ac.kr

Jin-Hwa Kim†
NAVER AI Lab
j1nhwa.kim@navercorp.com

Abstract

Visual dialog is a task of answering a sequence of questions grounded in an image using the previous dialog history as context. In this paper, we study how to address two fundamental challenges for this task: (1) reasoning over underlying semantic structures among dialog rounds and (2) identifying several appropriate answers to the given question. To address these challenges, we propose a Sparse Graph Learning (SGL) method to formulate visual dialog as a graph structure learning task. SGL infers inherently sparse dialog structures by incorporating binary and score edges and leveraging a new structural loss function. Next, we introduce a Knowledge Transfer (KT) method that extracts the answer predictions from the teacher model and uses them as pseudo labels. We propose KT to remedy the shortcomings of single ground-truth labels, which severely limit the ability of a model to obtain multiple reasonable answers. As a result, our proposed model significantly improves reasoning capability compared to baseline methods and outperforms the state-of-the-art approaches on the VisDial v1.0 dataset. The source code is available at https://github.com/gicheonkang/SGLKT-VisDial.

1 Introduction

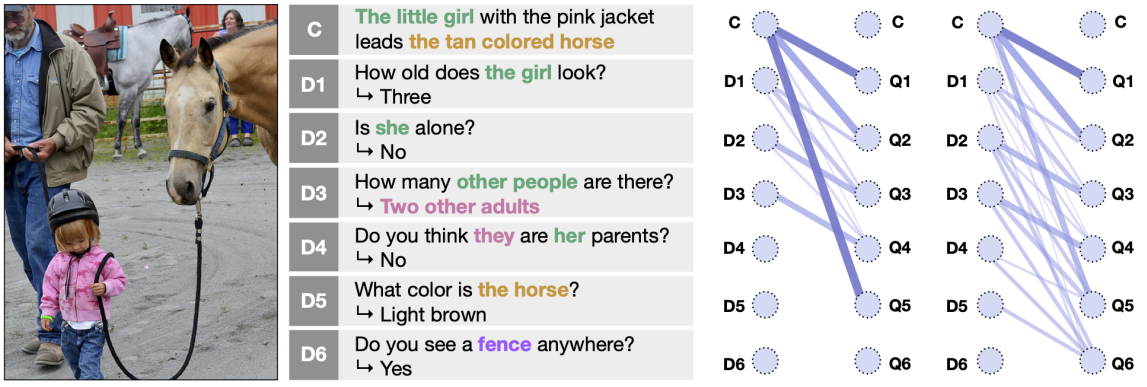
Recently, visually-grounded dialogue (Das et al., 2017; De Vries et al., 2017; Kottur et al., 2019; Kim et al., 2019) has attracted increasing research interest due to its potential impact on many real-world applications (*e.g.*, aiding visually impaired user). Notably, Visual Dialog (VisDial) (Das et al., 2017), which extends visual question answering (VQA) (Antol et al., 2015; Kim et al., 2018; Seo et al., 2021) to multi-round dialog, has been introduced to the research community, along with a large scale dataset. Unlike VQA, VisDial is designed to answer a *sequence* of questions grounded in an image utilizing a dialog history as context. This task requires a deep understanding of multi-modal inputs and the temporal nature of a human conversation. To infer an appropriate answer to the question, a dialog agent should attend to meaningful context from the dialog history as well as the given image.

There are two fundamental challenges in VisDial: (1) reasoning over underlying semantic structures among a series of utterances (*i.e.*, dialog rounds) and (2) identifying several appropriate answers to the given question. Previous approaches have implicitly addressed the first challenge by using the *soft-attention mechanism* (Bahdanau et al., 2014). Typically, the soft-attention mechanism is utilized to discover semantic relationships between the given question and previous utterances (*i.e.*, dialog history) while extracting rich contextual representations (Gan et al., 2019; Agarwal et al., 2020). Next, most of the previous work has not explicitly tackled the second challenge since there are no labels for prediction of multiple possible answers. For this reason, they have mostly focused on finding the single ground-truth answer by leveraging standard one-hot encoded labels.

We argue that existing approaches in VisDial show limited reasoning capability due to the way they approach the task: *soft-attention* and *one-hot encoded labels*. First, soft-attention restricts the ability to represent various types of semantic relationships in the dialog. As we illustrate in Figure 1, some questions in the dialog (Q1-Q4) are semantically dependent on previous utterances, while others (Q6)

† corresponding authors.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 327–339
November 7–11, 2021.



(a) Image **(b) Dialogs** **(c) Sparse Structure (Ours)** **(d) Dense Structure**

Figure 1: An example from the VisDial dataset. (a): a given image. (b): dialogue regarding the image, including image caption (C), and each round of dialog (D1-D6). (c) and (d): the semantic structures from our proposed model and the soft attention-based model, respectively. The left and right column in each figure denote the dialog history and the current question, respectively. The thicker and darker links indicate the higher semantic dependencies.

are independent, due to an abrupt change in topic. Furthermore, previous topics could be readdressed later in the dialog (Q5). However, soft-attention, which is based on the softmax function, always assigns a *non-zero* weight to all previous utterances, which results in dense (*i.e.*, fully-connected) relationships. Moreover, the sum of attention weights should be *one* due to the sum-to-1 constraint of the softmax function. Herein lies the problem: even for questions that are partly dependent (Q5 in Figure 1) or independent (Q6 in Figure 1) from the dialog history, all previous utterances are still considered and integrated into the contextual representations. As a consequence, the dialog agent could overly rely on the dialog history, even when the dialog history is irrelevant to the given question. Second, the model that utilizes the one-hot encoded labels learns to predict the single ground-truth answer only. However, similar to VQA, the given question is associated with one or several answers from a set of candidate answers. Therefore, the one-hot labels could suppress several plausible answers, assigning unreasonably low prediction probabilities to them.

In this paper, we propose two methods to remedy the conceptual shortcomings of the current approaches discussed above. First, we introduce a Sparse Graph Learning (SGL) method that predicts sparse structures of the visually-grounded dialog. In the graph structure, each node corresponds to a round of the dialog, and edges represent the semantic relationships between the rounds. SGL constructs the representations of each node by embedding the given image and each round of dialog in a joint fashion. SGL then infers two types of edge weights: binary (*i.e.*, 0 or 1) and score edges. It ultimately discovers the sparse and weighted structures (*e.g.*, (c) in Figure 1) by incorporating the two edge weights. Furthermore, we design a new structural loss function to encourage SGL to infer explicit and reliable dialog structures by leveraging a structural supervision. Next, to identify multiple possible answers, we treat VisDial as a regression task that predicts the correctness of each candidate answer individually, instead of a traditional setting that estimates the sum-to-1 scores over the candidate answers. To this end, we propose a Knowledge Transfer (KT) method that extracts the soft scores of each candidate answer from the teacher model (Qi et al., 2020). The soft scores are used to optimize for multiple possible answers. We expect this work to shed light on the above challenges that have not been explicitly addressed in visual dialog.

The main contributions of our paper are as follows. First, we propose a Sparse Graph Learning (SGL) approach that builds sparse structures of the visually-grounded dialog. By leveraging a new structural loss function, SGL learns the semantic relationships among dialog rounds in an explicit way. Second, we introduce a Knowledge Transfer (KT) method to encourage the model to find multiple possible answers to the given question. Third, the model that utilizes SGL and KT achieves the new state-of-the-art results on the VisDial v1.0 dataset. We perform comprehensive analysis to validate the effectiveness of SGL and KT. Finally, we conduct a qualitative analysis of each proposed method.

2 Related Work

Visual Dialog (Das et al., 2017) has been introduced as a temporal extension of VQA (Antol et al., 2015). In this task, a dialog agent should answer a sequence of questions by using an image and the dialog history as a clue. We carefully categorize the previous studies on visual dialog into three groups: (1) soft attention-based methods that compute the interactions among entities, including an input image, questions, and dialog history (Gan et al., 2019; Schwartz et al., 2019; Agarwal et al., 2020; Murahari et al., 2020; Wang et al., 2020), (2) a visual coreference resolution method (Seo et al., 2017; Kottur et al., 2018; Niu et al., 2019; Kang et al., 2019) that clarifies ambiguous expressions (*e.g.*, it, them) in the question and links them to the specific entities in the image, and (3) a structural inference method (Zheng et al., 2019) that attempts to discover dialog structures based on graph neural networks. Our approach belongs to the third group. Similar to the soft attention-based methods, Zheng et al. (2019) infer the dense semantic structures using a softmax function. Moreover, they attempt to find the structures without any explicit optimization for the structural inference. To tackle these aspects, we propose SGL which explicitly infers sparse structures with a structural loss function.

Graph Neural Networks (Scarselli et al., 2008) have sparked a tremendous interest at the intersection of deep neural networks and structural learning approaches. Recently, graph learning networks (GLNs) were proposed by (Pilco and Rivera, 2019; On et al., 2020), with the goal of reasoning over underlying structures of input data. GLNs consider unstructured data and dynamic domains (*e.g.*, time-varying domain). Our method belongs to the group of GLNs. CB-GLNs (On et al., 2020) attempt to discover the compositional structure of long video data with a graph-cut algorithm (Shi and Malik, 2000). However, SGL is different from previous studies in that SGL learns to build sparse structures *adaptively*, not relying on a predefined algorithm, and the dataset we use is highly multimodal.

Knowledge Transfer technique has been mainly explored to compress a large model into a small model (Buciluǎ et al., 2006; Ba and Caruana, 2014) without a significant drop in accuracy. The idea of knowledge transfer was later popularized under the name of knowledge distillation (KD) (Hinton et al., 2014). In KD, the knowledge of the large model (*i.e.*, teacher model) is transferred to the small model (*i.e.*, student model) as a form of supervision signal. Then, the student model learns to mimic the behavior of the teacher model by using the supervision signal and a pre-defined distillation loss function. Our Knowledge Transfer (KT) approach shares this same spirit. However, we repurpose KT to cast VisDial as a regression of scores for candidate answers. Accordingly, the soft targets from the teacher model are utilized as supervision for the correctness of each candidate answer which was originally unlabeled.

3 Sparse Graph Learning

The visual dialog task (Das et al., 2017) is defined as follows: given an image $\mathcal{I}$, a caption c describing the image, a dialog history $\mathcal{H} = \{\underbrace{c}_{h_0}, \underbrace{(q_1, a_1^{gt})}_{h_1}, \cdots, \underbrace{(q_{t-1}, a_{t-1}^{gt})}_{h_{t-1}}\}$, and a question q_t at current round t, the goal is to find an appropriate answer to the question among the N answer candidates, $\mathcal{A}_t = \{a_t^1, \cdots, a_t^N\}$.

In our approach, we consider the task as a graph $G_t = (V_t, E_t)$ with $t+1$ nodes (*i.e.*, vertices), where $(v_0, v_1, ..., v_{t-1})$ and (v_t) correspond to the node for the previous dialog history and the current question, respectively. Each node $v_i \in V_t$ is associated with a feature vector $\mathbf{x}_i$. The semantic dependencies among the nodes are represented as weighted edges $E_t = \{(v_i, v_j) : v_i, v_j \in V_t\}$. The goal of our approach is to discover a sparse and weighted adjacency matrix $\mathbf{A}_t \in \mathbb{R}^{(t+1)\times(t+1)}$ which represents the semantic dependencies among dialog rounds.

To implement the pipeline above, we propose a Sparse Graph Learning (SGL) method that consists of two modules (see Figure 2): (1) a node embedding module that embeds the visual-linguistic representations for each round of the dialog and (2) a sparse graph learning module that estimates a sparse and weighted structures of the dialog.

3.1 Input Features

Visual Features. In the given image $\mathcal{I}$, we extract the d_v-dimensional visual features of K objects by employing a pre-trained Faster R-CNN model (Ren et al., 2015; Anderson et al., 2018). Then, we project the visual features into dimension d_h using a linear matrix $\mathbf{W}_f \in \mathbb{R}^{d_v \times d_h}$, which results in $\mathbf{M}^v \in \mathbb{R}^{K \times d_h}$. We use $\mathbf{M}^v$ as visual features.

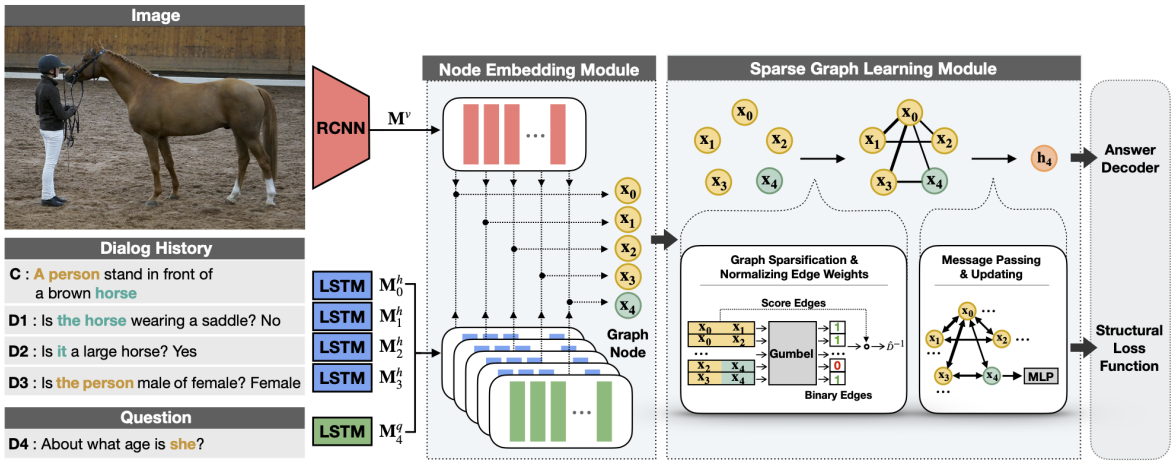


Figure 2: An overview of Sparse Graph Learning (SGL) framework. Please see Section 3 for details.

Language Features. In the t-th dialog round, we first encode the question q_t which is a word sequence of length L, $(w_1, ..., w_L)$, by using a LSTM (Hochreiter and Schmidhuber, 1997). Specifically, we use all hidden states of the LSTM as the question features, which results in $\mathbf{M}_t^q \in \mathbb{R}^{L \times d_h}$. Likewise, each round of the dialog history $\{h_i\}_{i=0}^{t-1}$ is encoded into $\{\mathbf{M}_i^h\}_{i=0}^{t-1} \in \mathbb{R}^{t \times L \times d_h}$. To reduce computational complexity, we embed all the answer candidates $\{a_t^i\}_{i=1}^{N}$ with sentence-level features by extracting the last hidden states of the LSTM, which results in $\mathbf{M}_t^a \in \mathbb{R}^{N \times d_h}$.

3.2 Node Embedding Module

The node embedding module aims to embed rich visual-linguistic joint representations for each round of the dialog. To implement these processes, we take inspiration from Modular Co-Attention Networks (MCAN) (Yu et al., 2019) which are based on the multi-head attention mechanism (Vaswani et al., 2017). Given the object-level visual features $\mathbf{M}^v \in \mathbb{R}^{K \times d_h}$ and the question features $\mathbf{M}_t^q \in \mathbb{R}^{L \times d_h}$, the node embedding module f_{ne} computes the joint representations $\mathbf{x}_t \in \mathbb{R}^{1 \times d_h}$.

$$\mathbf{x}_t = f_{ne}(\mathbf{M}^v, \mathbf{M}_t^q) \tag{1}$$

Each round of the dialog history $\{\mathbf{M}_i^h\}_{i=0}^{t-1}$ is also embedded by the module, which results in $\{\mathbf{x}_i\}_{i=0}^{t-1}$. Consequently, as shown in Figure 2, we obtain $(t+1)$ joint representations including the question features $\mathbf{x}_t$ and the dialog features $\{\mathbf{x}_i\}_{i=0}^{t-1}$. We use these features as the nodes of the graph which can be represented in matrix-form as $\mathbf{X} \in \mathbb{R}^{(t+1) \times d_h}$. A detailed architecture of the node embedding module can be found in the supplementary materials.

3.3 Sparse Graph Learning Module

The sparse graph learning module infers the underlying sparse and weighted graph structure among nodes, where the edge weights are estimated based on the node features. To make the graph structure to be sparse, we propose two types of edges on the graph G_t: binary edges E_t^b and score edges E_t^s, whose corresponding adjacency matrices are $\mathbf{A}_t^b$ and $\mathbf{A}_t^s$ respectively. To simplify the notation, we omit the subscript t in the following equations.

Binary Edges. We first define a binary edge between two nodes v_i and v_j as a binary random variable $z_{ij} \in \{0, 1\}$, for all $i, j \in [0, t]$ and $i < j$. The sparse graph learning module estimates the likelihood of the binary variables given the node features, where the probability implies whether the two nodes are semantically related or not. We regard the binary variable as a two-class categorical variable and define the probability distribution as:

$$\mathbf{A}_{ij}^b = z_{ij} \sim Categorical(\mathbf{p}_{ij}) \tag{2}$$

$$\mathbf{p}_{ij} = \text{softmax}\Big(\mathbf{W}_c(\mathbf{x}_i \circ \mathbf{x}_j)^\top / \tau\Big) \tag{3}$$

where $\mathbf{W}_c \in \mathbb{R}^{2 \times d_h}$ is a learnable parameter, $\circ$ denotes the hadamard product, and τ is the softmax temperature. Since z_{ij} is discrete and non-differentiable, we employ a Straight-Through Gumbel-Softmax estimator (*i.e.*, ST-Gumbel) (Jang et al., 2017) to ensure end-to-end training. During

forward propagation, ST-Gumbel makes a discrete decision by using the Gumbel-Max trick:

$$z_{ij} = \begin{cases} 1, & \text{if } \underset{k\in\{0,1\}}{\operatorname{argmax}}\big(\log(p_k) + g_k\big) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where the random variable g_k is drawn from a Gumbel distribution. In the backward pass, ST-Gumbel utilizes the derivative of the probabilities by approximating $\nabla_\theta z \approx \nabla_\theta p$, thus enabling the backpropagation and end-to-end training.

Score Edges. We define score edges to measure the extent to which two nodes are related, and the relevance is computed as:

$$\mathbf{A}^s_{ij} = (\mathbf{x}_i \mathbf{x}_j^\top)^2 \tag{5}$$

Following Yang et al. (2018), we also employ the squared dot product for stabilized training.

Sparse Weighted Edges. The sparse graph learning module multiplies the binary edges and score edges, finally yielding a sparse and weighted adjacency matrix as:

$$\hat{\mathbf{A}}_{ij} = \mathbf{A}^b_{ij}\mathbf{A}^s_{ij} = z_{ij}(\mathbf{x}_i \mathbf{x}_j^\top)^2 \tag{6}$$

With the above edge weight estimations, this module is able to model three types of relationships on v_i: (1) dense relationships similar to the previous conventional softmax-based approaches if $\sum_j z_{ij} = t$ (*i.e.*, all entries in z_i are one), (2) sparse relationships if $0 < \sum_j z_{ij} < t$, and (3) no relationships if $\sum_j z_{ij} = 0$ (*i.e., isolated node)*.

Message-passing and Update. Based on the sparse weighted adjacency matrix $\hat{\mathbf{A}}$, the sparse graph learner updates the hidden states of all nodes through a message-passing framework (Gilmer et al., 2017). Similar to graph convolutional networks (Kipf and Welling, 2017), we simply implement the message-passing layer F_M as the normalized weighted sum according to the adjacent weight, followed by a linear transformation.

$$\mathbf{M} = F_M(\mathbf{X}, \hat{\mathbf{A}}) = \hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}\mathbf{X}\mathbf{W}_m \tag{7}$$

where $\mathbf{W}_m \in \mathbb{R}^{d_h \times d_h}$. Note that $\hat{\mathbf{D}}$ is the degree matrix of $\hat{\mathbf{A}}$. The hidden node features are calculated via the update layer F_U which adds the input feature and aggregated messages and subsequently feeds them into a non-linear function f_u.

$$\mathbf{H} = F_U(\mathbf{X}, \mathbf{M}) = f_u(\mathbf{X} + \mathbf{M}) \tag{8}$$

f_u is two-layer feed-forward networks with a ReLU in between. The model can perform multi-step reasoning by conducting a set of equations (*i.e.*, Eq. 7 and Eq. 8) multiple times. Finally, SGL returns the adjacency matrix $\hat{\mathbf{A}}$ and the hidden node features $\mathbf{H} \in \mathbb{R}^{(t+1)\times d_h}$. The features for the current round, $\mathbf{H}[t,:] = \mathbf{h}_t$, is used to decode answers. Note that SGL as described above computes all interactions among $t+1$ nodes for every dialog round, although the edge weights among $\{\mathbf{x}_i\}_{i=0}^{t-1}$ are estimated in the previous dialog round. For the sake of computational efficiency, we can construct $\hat{\mathbf{A}}_t$ by combining the adjacency matrix of the previous round $\hat{\mathbf{A}}_{t-1}$ with the edge weights between $\mathbf{x}_t$ and $\{\mathbf{x}_i\}_{i=0}^{t-1}$ in the t-th round. This decreases the computational complexity, from $O(t^2)$ to $O(t)$.

3.4 Structural Learning

We introduce a structural loss function $\mathcal{L}_{sgl}$ to encourage SGL to infer explicit, reliable dialog structures. Inspired by Coref-NMN (Kottur et al., 2018) that employs the off-the-shelf neural coreference resolution tool[1] for visual coreference resolution, we repurpose this tool for structural learning. Specifically, we automatically obtain the semantic dependencies between rounds by using the coreference resolution tool and leverage this information as structural supervision. The one-valued entries in the structural supervision indicate that both dialog rounds include at least one noun phrase or a pronoun referring to the same entity. Otherwise, the entries are filled with a zero-value. SGL minimizes the distance between the structural supervision $\mathbf{C}_T$ and the binary matrix $\mathbf{A}^b_T \in \mathbb{R}^{(T+1)\times(T+1)}$ finally predicted from SGL:

$$\mathcal{L}_{sgl} = \parallel \mathbf{C}_T - \mathbf{A}^b_T \parallel^2_F \tag{9}$$

where T and $\|\cdot\|^2_F$ denote the total number of rounds for each dialog and the squared Frobenius norm (*i.e.*, element-wise mean squared error), respectively. Here, $\mathcal{L}_{sgl}$ encourages SGL to predict a reliable dialog structure. Note that SGL uses the structural supervision only while training, and infers the dialog structures at test time.

[1] https://github.com/huggingface/neuralcoref based on the work (Clark and Manning, 2016).

4 Knowledge Transfer

The conventional assumption in VisDial is that there is one correct answer for each question from a set of candidate answers. Accordingly, the one-hot encoded single ground-truth label is used as standard supervision. However, the given question can indeed be associated with one or several answers. For this reason, a few works (Qi et al., 2020; Murahari et al., 2020) have applied an additional fine-tuning strategy on dense labels[2] for the validation split to improve the model's ability to predict multiple correct answers. Instead of using the fine-tuning approach, we propose a Knowledge Transfer (KT) method to optimize several correct answers simultaneously in a single training procedure. KT extracts the soft scores of each candidate answer from the fine-tuned teacher model, P1+P2 (Qi et al., 2020), and uses these scores as pseudo labels. We choose the P1+P2 for their strong performance on retrieving several appropriate answers for the given question. Specifically, we combine the dense score vector $\mathbf{y}_t^{dense} \in \mathbb{R}^N$ from the teacher model with the one-hot vector $\mathbf{y}_t^{sparse} \in \mathbb{R}^N$ for the t-th question as:

$$\hat{y}_{tn} = \max_{n \in \{1,\dots,N\}} \left(y_{tn}^{sparse},\ y_{tn}^{dense}\right) \tag{10}$$

where N is the number of candidate answers. Note that $\mathbf{y}_t^{dense}$ is a sigmoid output of the teacher model. As a result, $\hat{y}_{tn}$ contains a score of 1.0 for the ground-truth answer and soft scores ranging from 0 to 1 for the other candidates. Based on the combined labels $\hat{y}_{tn}$, we cast VisDial as a regression task that predicts the correctness of each candidate answer individually. The predicted score vector for N candidates is computed as:

$$\mathbf{s_t} = \sigma\left(\mathbf{M}_t^a \mathbf{h}_t^\top\right) \tag{11}$$

where $\mathbf{M}_t^a \in \mathbb{R}^{N \times d_h}$ (in Sec. 3.1) and $\mathbf{h}_t \in \mathbb{R}^{1 \times d_h}$ are feature vectors for candidate answers and the hidden node feature for current round from SGL, respectively. σ denotes a sigmoid function. Finally, we design a loss function for KT as:

$$\mathcal{L}_{kt} = -\sum_{t=1}^{T}\sum_{n=1}^{N} \hat{y}_{tn} \ln\left(s_{tn}\right) - \left(1 - \hat{y}_{tn}\right) \ln\left(1 - s_{tn}\right) \tag{12}$$

which is similar to a binary cross-entropy loss except that we use a *soft* target score $\hat{y}_{tn}$. $\mathcal{L}_{kt}$ and the sigmoid activation function allow optimization for multiple correct answers. We believe KT is an efficient approach to distill the prior knowledge of dense labels from the teacher model for the training split, rather than directly fine-tuning the model on those dense labels only for validation split.

[2]The densely annotated relevance scores for all candidate answers are released in the VisDial v1.0 validation & test split.

5 Experiments

5.1 Experimental Setup

Dataset. We benchmark our proposed model on the VisDial v1.0 dataset (Das et al., 2017). The VisDial v1.0 dataset contains 1.2M, 20k, and 44k question-answer pairs as train, validation, and test splits, respectively. The 123,287 images from COCO (Lin et al., 2014), 2,064, and 8k images from Flickr are used to collect the dialog data for each split, respectively. A list of N =100 answer candidates accompanies each question-answer pair.

Evaluation. We follow the standard protocol (Das et al., 2017) for evaluating visual dialog models: mean reciprocal rank (MRR), recall@k (R@k), mean rank (Mean), and normalized discounted cumulative gain (NDCG). The first three measure the performance of retrieving the single ground-truth answer, while NDCG considers all relevant answers from the 100-answers list by using the densely annotated scores. There is a growing consensus among recent works (Kim et al., 2020; Murahari et al., 2020) that MRR and NDCG are regarded as the primary metrics and a balance of the two is important. For this reason, we additionally report the average of MRR and NDCG as *overall* performance. The overall performance is also used as a selection criterion of VisDial challenge winner.

5.2 Quantitative Analysis

Compared Methods. We compare our methods with the state-of-the-art approaches on VisDial v1.0 dataset, including GNN (Zheng et al., 2019), CorefNMN (Kottur et al., 2018), RvA (Niu et al., 2019), Synergistic (Guo et al., 2019), ReDAN (Gan et al., 2019), DAN (Kang et al., 2019), HACAN (Yang et al., 2019), FGA (Schwartz et al., 2019), MCA (Agarwal et al., 2020), P1+P2 (Qi et al., 2020), VisDial-BERT (Murahari et al., 2020), VD-BERT (Wang et al., 2020).

Comparison with State-of-the-art. We evaluate our proposed methods with three different settings: (1) single model that utilizes the one-hot encoded labels (*i.e.*, SGL), (2) single model with dense

Model	Overall↑	NDCG↑	MRR↑	R@1↑	R@5↑	R@10↑	Mean↓
GNN	57.10	52.82	61.37	47.33	77.98	87.83	4.57
CorefNMN	58.10	54.70	61.50	47.55	78.10	88.80	4.40
RvA	59.31	55.59	63.03	49.03	80.40	89.83	4.18
Synergistic	59.76	57.32	62.20	47.90	80.43	89.95	4.17
Synergistic‡	60.65	57.88	63.42	49.30	80.77	90.68	3.97
ReDAN	57.50	61.86	53.13	41.38	66.07	74.50	8.91
ReDAN+‡	59.10	64.47	53.73	42.45	64.68	75.68	6.63
DAN	60.40	57.59	63.20	49.63	79.75	89.35	4.30
DAN‡	62.14	59.36	64.92	51.28	81.60	90.88	3.92
HACAN	60.70	57.17	64.22	50.88	80.63	89.45	4.20
FGA	57.90	52.10	63.70	49.58	80.97	88.55	4.51
FGA‡	60.90	54.50	**67.30**	**53.40**	**85.28**	**92.70**	**3.54**
MCA†	55.08	72.47	37.68	20.67	56.67	72.12	8.89
P1+P2†	60.09	71.60	48.58	35.98	62.08	77.23	7.48
P1+P2†‡	63.32	74.02	52.62	40.03	68.85	79.15	6.76
VisDial-BERT†	62.60	74.47	50.74	37.95	64.13	80.00	6.28
VD-BERT	62.70	59.96	65.44	51.63	82.23	90.68	3.90
VD-BERT†	60.63	74.54	46.72	33.15	61.58	77.15	7.18
VD-BERT†‡	63.26	**75.35**	51.17	38.90	62.82	77.98	6.69
SGL	62.13	61.97	62.28	48.15	79.65	89.10	4.34
SGL+KT†	65.31	72.60	58.01	46.20	71.01	83.20	5.85
SGL+KT†‡	**66.03**	73.70	58.36	46.63	71.28	84.15	5.57

Table 1: Test-std performance of the discriminative model on the VisDial v1.0 dataset. ↑ indicates higher is better. ↓ indicates lower is better. † denotes the use of dense labels. ‡ denotes ensemble model.

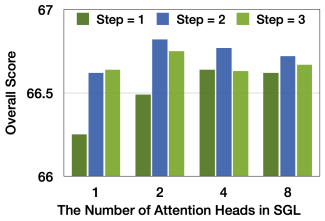


Figure 3: Ablation study on VisDial v1.0 val split.

Model	F1-Score
Edgeless	0.0
Dense	0.246
Sparse-hard	0.279
SGL	0.714
SGL+KT	**0.748**

Table 3: Graph inference on VisDial v1.0 val split.

labels (*i.e.*, SGL+KT), and (3) ensemble model with dense labels (*i.e.*, 5×(SGL+KT)). As shown in Table 1, (2) and (3) outperform the existing models on overall performance by 4.68% (65.31 vs. 60.63) and 2.71% (66.03 vs. 63.32), respectively. The results indicate that our methods show higher and more balanced performance than all other methods on NDCG and MRR. The single model also shows competitive performance compared with VD-BERT that utilizes BERT (Devlin et al., 2018) as a backbone. We observe that the use of dense labels yields huge improvements on NDCG and counter-effect on other metrics. Specifically, VD-BERT shows nearly 14% improvements on NDCG with dense labels (59.96 → 74.54) while dramatically dropping MRR (65.44 → 46.72). However, KT still boosts NDCG (61.97 → 72.60), yet notably with limited MRR drop (62.28 → 58.01). We conjecture that optimizing the loss on the combined labels (see Sec. 4) mitigates the counter-effect.

Model	Overall	NDCG	MRR
Edgeless	60.75	61.96	59.54
Dense	61.05	58.85	63.25
Sparse-hard	61.44	59.71	63.16
P1+P2† (teacher model)	61.65	73.42	49.88
SGL w/o RPN	61.56	61.25	61.86
SGL w/o SS	61.66	62.46	60.85
SGL w/o MR	62.11	62.42	61.79
SGL	63.38	63.41	**63.34**
SGL+KT†	**66.82**	**74.54**	59.10

Table 2: Comparison with the baseline models on the VisDial v1.0 validation split. MR, SS, and RPN denote the use of multi-step reasoning, structural supervision, and region proposal network, respectively. † denotes the use of dense labels.

Comparison with Baselines. We compare our methods to the baseline models in Table 2. First, we define three models as baselines for SGL: Edgeless, Dense, and Sparse-hard. The Dense model utilizes a soft-attention mechanism, which yields the fully-connected graph. Contrary to the Dense model, the Sparse-hard model picks exactly one edge weights for each node by applying the Gumbel-Softmax to all nodes in the graph. Note that the structural supervision is provided in the Sparse-hard model. Finally, the Edgeless model yields a graph consisting only of isolated nodes. This indicates that the Edgeless model does not utilize the dialog history at all. As shown in Table 2, SGL achieves better performance than the baseline models on all metrics. Furthermore, we report the performance of ablative models: SGL w/o RPN, SGL w/o SS, and SGL w/o MR. SGL w/o RPN employs ImageNet pre-trained with VGG-16 model (Simonyan and Zisserman, 2015), and uses the spatial grids of *pool5* feature map as visual features. SGL w/o SS is the model that does not use the structural supervision (*i.e.*, L_{sgl}). SGL w/o MR denotes the model that uses single-step reasoning in the sparse graph learning module. We identify that all three components (*i.e.*, RPN, SS, and MR) in SGL play a crucial role in boosting the performance. Next, comparing SGL with SGL+KT, we observe that KT significantly improves NDCG score from 63.41 to 74.54. It demonstrates that the knowledge of the teacher model – which helps to find multiple correct or relevant answers – is successfully transferred to SGL. In Table 2, SGL+KT even surpasses the NDCG score of the teacher model, P1+P2, by 1.12%. From this observation, we conjecture that SGL enriches the distilled knowledge from the

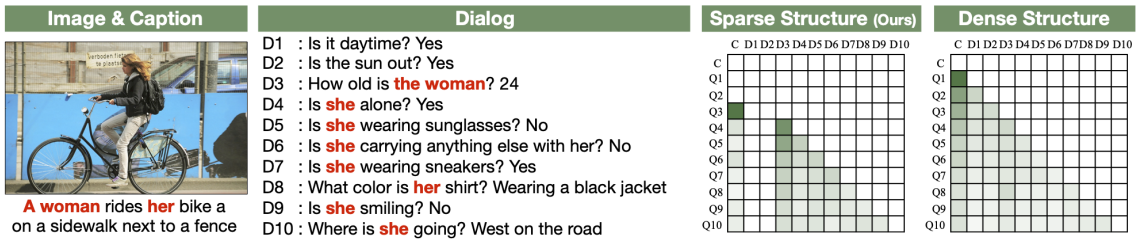


Figure 4: A visualization of the inferred semantic structures from the validation set. From the left, the given image and caption, the dialog history, and the structures of ours and baseline. The darker the color, the higher the score.

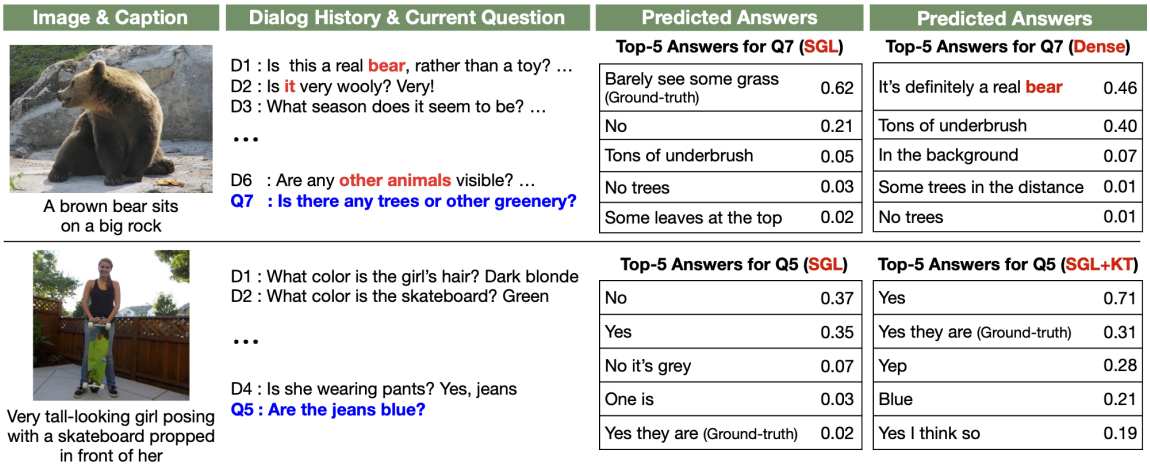


Figure 5: A visualization of the top five predicted answers from SGL+KT, SGL, and Dense. Note that SGL+KT utilizes the sigmoid activation function to compute the answer scores while the others use the softmax function.

teacher model, which results in better performance than the teacher model. Although boosting NDCG results in decreasing MRR score due to their trade-off relationship (Murahari et al., 2020; Kim et al., 2020), the MRR drop of KT is considerably smaller than other methods.

Reasoning Steps & Attention Heads. Based on SGL+KT model, we perform ablation experiments with different number of reasoning steps (1, 2, and 3) in the sparse graph learning module and attention heads (1, 2, 4, and 8) in the node embedding module. As shown in Figure 3, the model with two-step reasoning with two attention heads performs the best among all models in the experiments, recording 66.82 on overall performance.

Is SGL inferring the right graph? We investigate this question by measuring the agreement between the binary edges $\mathbf{A}^b$ inferred from our model and the structural supervision $\mathbf{C}$, assuming that $\mathbf{C}$ is the ground-truth graph. We use F1-score as an evaluation metric. Then, we employ Edgeless, Dense, and Sparse-hard as baselines. Note that the Dense model itself is not compatible with the evaluation metric since it does not predict the binary edges. To make it compatible, we create the binary edges by replacing the top edge weight for each node with one. The rest are replaced with zero. In Table 3, SGL and SGL+KT show significantly better F1-scores than the baselines. It might indicate that SGL infers more reliable semantic structures. Furthermore, comparing SGL with SGL+KT in Table 3, we observe that KT improves the performance of graph inference. It indicates that KT contributes to an accurate inference of sparse graphs.

5.3 Qualitative Results

In Figure 4, we visualize the images, the corresponding dialogs in the validation split, and the inferred adjacency matrices as well as the ones from the Dense model as a counter. Compared to the dense structure in the baseline, the proposed SGL indeed learns the innate sparse structures, and the question nodes receive the information from the other nodes in a selective fashion. For instance, the questions from Q3 to Q10 have non-zero binary edges to all previous contexts except D1 and D2, which do not contain relevant information about 'the woman'. On the contrary, Q1 and Q2 are not connected to any other nodes, because they can be

answered solely without additional context. We visualize additional examples regarding the graph inference in the supplementary materials. Next, to demonstrate the advantages of SGL and KT, we visualize the top five predicted answers for each question from the Dense model, SGL, and SGL+KT in Figure 5. In the first example, SGL retrieves the ground-truth answer by not using the dialog history, while the Dense model provides the wrong answer – containing the word *bear* – to the top. We conjecture that relying on the dialog history – even when the history is not required – leads to this phenomenon. In the next example, the answers predicted by SGL+KT are semantically exchangeable with each other, whereas the answers from SGL are not. It shows that the teacher model's knowledge enforces the ability to find multiple correct answers and resultant consistency of answer prediction.

6 Conclusions

We propose SGL and KT to remedy the shortcomings of previous work: soft-attention and one-hot labels. Experimental results illustrate the effectiveness of our approach. SGL with KT achieves the new state-of-the-art performance on the VisDial v1.0 dataset. We believe that the idea of selectively paying attention to desired information is widely applicable to various research fields, and KT can be generally adopted to improve answer prediction.

Acknowledgement The authors would like to thank Woo-Suk Choi and Björn Bebensee for helpful comments and editing. This work was supported in part by SK Telecom when Gi-Cheon Kang, Hwaran Lee, and Jin-Hwa Kim worked at SK Telecom. The Korean government (2015-0-00310-SW.StarLab, 2017-0-01772-VTT, 2018-0-00622-RMI, 2019-0-01371-BabyMind) partly supports this work as well.

References

Shubham Agarwal, Trung Bui, Joon-Young Lee, Ioannis Konstas, and Verena Rieser. 2020. History for visual dialog: Do we really need it? In *ACL*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. In *NIPS Deep Learning Symposium*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *ACM SIGKDD*, pages 535–541.

Kevin Clark and Christopher D Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *ACL*.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *CVPR*.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multimodal dialogue. In *CVPR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Zhe Gan, Yu Cheng, Ahmed EI Kholy, Linjie Li, Jingjing Liu, and Jianfeng Gao. 2019. Multi-step reasoning via recurrent dual attention for visual dialog. In *ACL*.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*.

Dalu Guo, Chang Xu, and Dacheng Tao. 2019. Image-question-answer synergistic network for visual dialog. In *CVPR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the Knowledge in a Neural Network. In *NIPS 2014 Deep Learning Workshop*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. In *Neural computation*. MIT Press.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

Gi-Cheon Kang, Jaeseo Lim, and Byoung-Tak Zhang. 2019. Dual attention networks for visual reference resolution in visual dialog. In *EMNLP*.

Hyounghun Kim, Hao Tan, and Mohit Bansal. 2020. Modality-balanced models for visual dialogue. In *AAAI*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *NeurIPS*.

Jin-Hwa Kim, Nikita Kitaev, Xinlei Chen, Marcus Rohrbach, Byoung-Tak Zhang, Yuandong Tian, Dhruv Batra, and Devi Parikh. 2019. Codraw: Collaborative drawing as a testbed for grounded goal-driven communication. In *ACL*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2019. Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. In *NAACL*.

Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2018. Visual coreference resolution in visual dialog using neural module networks. In *ECCV*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, and Dhruv Batra. 2017. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *NIPS*.

Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. 2020. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. In *ECCV*.

Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. 2019. Recursive visual attention in visual dialog. In *CVPR*.

Kyoung-Woon On, Eun-Sol Kim, Yu-Jung Heo, and Byoung-Tak Zhang. 2020. Cut-based graph learning networks to discover compositional structure of sequential video data. In *AAAI*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Darwin Saire Pilco and Adín Ramírez Rivera. 2019. Graph learning network: A structure learning algorithm. *arXiv preprint arXiv:1905.12665*.

Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. 2020. Two causal principles for improving visual dialog. In *CVPR*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. In *IEEE Transactions on Neural Networks*. IEEE.

Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G Schwing. 2019. Factor graph attention. In *CVPR*.

Ahjeong Seo, Gi-Cheon Kang, Joonhan Park, and Byoung-Tak Zhang. 2021. Attend what you need: Motion-appearance synergistic networks for video question answering. In *ACL*, pages 6167–6177.

Paul Hongsuck Seo, Andreas Lehrmann, Bohyung Han, and Leonid Sigal. 2017. Visual reference resolution using attention memory for visual dialog. In *NIPS*.

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. In *IEEE*. Ieee.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Yue Wang, Shafiq Joty, Michael R Lyu, Irwin King, Caiming Xiong, and Steven CH Hoi. 2020. Vd-bert: A unified vision and dialog transformer with bert. *arXiv preprint arXiv:2004.13278*.

Tianhao Yang, Zheng-Jun Zha, and Hanwang Zhang. 2019. Making history matter: History-advantage sequence training for visual dialog. In *ICCV*.

Zhilin Yang, Jake Zhao, Bhuwan Dhingra, Kaiming He, William W Cohen, Russ R Salakhutdinov, and Yann LeCun. 2018. Glomo: unsupervised learning of transferable relational graphs. In *NIPS*.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *CVPR*, pages 6281–6290.

Zilong Zheng, Wenguan Wang, Siyuan Qi, and Song-Chun Zhu. 2019. Reasoning visual dialogs with structural and partial observations. In *CVPR*.

Appendix Overview

The supplementary materials are organized as:

- Sec. A shows a detailed architecture of the node embedding module.
- Sec. B presents our experiments with a generative model.
- Sec. C presents implementation details.
- Sec. D shows qualitative examples from SGL.

A Node Embedding Module

Subcomponents. A detailed architecture of the node embedding module is presented in Figure 6. The module consists of three subcomponents: self-attention (*i.e.*, SA), guided-attention (*i.e.*, GA), and attention flat (*i.e.*, AF). First, SA and GA are based on the multi-head attention mechanism (*i.e.*, MHA) (Vaswani et al., 2017). MHA computes h parallel attention heads and aggregates them with a linear matrix. Each head corresponds to the output of the scaled dot-product attention. It is formulated as:

$$\mathrm{A}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V} \quad (13)$$

$$\mathrm{MHA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = [head_1,...,head_h]\mathbf{W}^o \quad (14)$$

$$head_n = \mathrm{A}(\mathbf{Q}\mathbf{W}_n^Q, \mathbf{K}\mathbf{W}_n^K, \mathbf{V}\mathbf{W}_n^V) \quad (15)$$

where $\mathbf{W}_n^Q, \mathbf{W}_n^K, \mathbf{W}_n^V$ are the projection matrices for the n-th head. $\mathbf{W}^O$ is the linear matrix. Then, the residual connection (He et al., 2016), layer normalization (Ba et al., 2016), and the two-layer feed-forward networks (*i.e.*, FFN) are applied in SA and GA (see Figure 6). The inputs of SA are from the same features, while GA takes two groups of input features – the query and the key-value pairs. Next, AF performs an attentional reduction to flatten the inputs to the vector representation. Given the input matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_m] \in \mathbb{R}^{m \times d_h}$, AF yields the vector $\tilde{\mathbf{x}} \in \mathbb{R}^{1 \times d_h}$ as follows:

$$\mathrm{AF}(\mathbf{X}) = \tilde{\mathbf{x}} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i \quad (16)$$

$$\alpha = \mathrm{softmax}(\mathrm{MLP}(\mathbf{X})) \quad (17)$$

where MLP projects $\mathbf{X}$ to m-dimensional vector. $\alpha = [\alpha_1, \cdots, \alpha_m] \in \mathbb{R}^m$ are the attention weights.

Overview. First, the object-level visual features $\mathbf{M}^v \in \mathbb{R}^{K \times d_h}$ and the question features $\mathbf{M}_t^q \in \mathbb{R}^{L \times d_h}$ are given to SA, yielding $\mathbf{Z}^v \in \mathbb{R}^{K \times d_h}$ and $\mathbf{Z}_t^q \in \mathbb{R}^{L \times d_h}$, respectively. Then, GA takes $\mathbf{Z}^v$ and $\mathbf{Z}_t^q$ as inputs and computes the pair-wise relationship between the visual features and the linguistic features. $\hat{\mathbf{Z}}^v \in \mathbb{R}^{K \times d_h}$ is obtained from GA. Finally, $\hat{\mathbf{Z}}^v$ and $\mathbf{Z}_t^q$ are passed through to AF$(\cdot)$ and the two-layer feed-forward networks, resulting in $\mathbf{z}^v \in \mathbb{R}^{1 \times d_h}$ and $\mathbf{z}_t^q \in \mathbb{R}^{1 \times d_h}$, respectively. Consequently, the visual-linguistic representation $\mathbf{x}_t \in \mathbb{R}^{1 \times d_h}$ is obtained by adding $\mathbf{z}^v$ and $\mathbf{z}_t^q$. From this pipeline, the node embedding module f_{ne} embeds the high-level abstraction of the visual and linguistic inputs in a joint fashion. Note that the module also embeds each round of the dialog history in the same way as the question features.

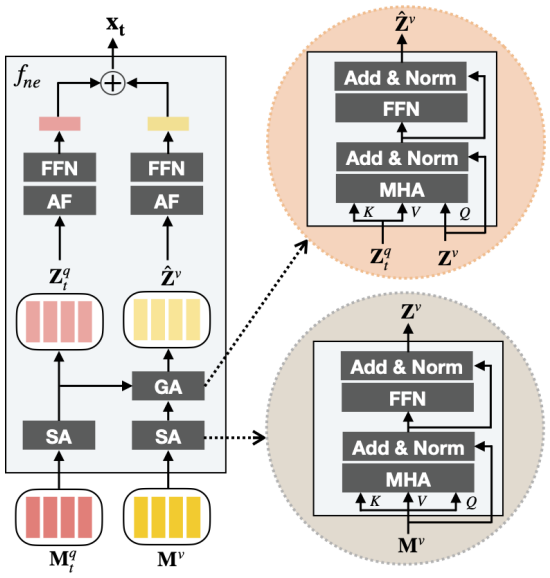


Figure 6: A detailed architecture of the node embedding module. SA, GA, AF, MHA, and FFN denote self-attention, guided-attention, attention flat, multi-head attention, and feed-forward networks, respectively.

B Generative Model

Overview. The authors of (Das et al., 2017) have also proposed a generative model which is trained for generating an answer without access to the answer candidates. Specifically, the generative model aims to generate the ground-truth answer's word sequence auto-regressively via a LSTM:

$$\begin{aligned}\mathcal{L}_{gen} &= -\sum_{t=1}^{T} \log p(a_t^{gt}|\mathbf{h}_t) \\ &= -\sum_{t=1}^{T}\sum_{l=1}^{L} \log p(w_l|w_{<l}, \mathbf{h}_t)\end{aligned} \quad (18)$$

Model	Overall	NDCG	MRR
MN (Das et al., 2017)	52.41	56.99	47.83
HCIAE (Lu et al., 2017)	54.39	59.70	49.07
ReDAN (Gan et al., 2019)	55.25	60.47	**50.02**
SGL	55.30	61.42	49.17
SGL+KT ($I = 2$)	55.42	63.80	47.03
SGL+KT ($I = 3$)	**56.21**	**65.74**	46.67

Table 4: VisDial v1.0 validation performance of the generative models.

where $\mathbf{h}_t$ is the hidden node feature for the current round from SGL and a_t^{gt} denotes the ground-truth answer consisting of L words $(w_1, ..., w_L)$. T is the number of rounds for each dialog. We initialize the hidden states of the LSTM with $\mathbf{h}_t$. Then, the generative model is optimized by minimizing negative log-likelihood of the ground-truth answer. In inference time, following Das et al. (2017), we utilize the log-likelihood scores to determine the rank of candidate answers for the process of evaluation.

Generative Model with Knowledge Transfer. We further apply the Knowledge Transfer (KT) technique to the generative model. Based on the combined labels $\hat{\mathbf{y}}_t$, which were discussed in Sec. 4, we extract the top-I answer candidates for the given question and use them to train the model. Formally,

$$\begin{aligned}\mathcal{L}_{gen,\,kt} &= -\sum_{t=1}^{T} \hat{\mathbf{y}}_t \log p(\hat{\mathcal{A}}_t|\mathbf{h}_t) \\ &= -\sum_{t=1}^{T}\sum_{i=1}^{I} \hat{y}_{ti} \log p(a_t^i|\mathbf{h}_t) \\ &= -\sum_{t=1}^{T}\sum_{i=1}^{I} \hat{y}_{ti} \sum_{l=1}^{L} \log p(w_{i,l}|w_{i,<l}, \mathbf{h}_t)\end{aligned} \tag{19}$$

where $\hat{\mathcal{A}}_t = \{a_t^i\}_{i=1}^{I}$ is a set of selected candidate answers and a_t^i consists of L words $(w_{i,1}, ..., w_{i,L})$. I implies the number of candidate answers that the generative model can access. Accordingly, $I = 1$ is equivalent to the standard generative model described in Eq. 18 since the ground-truth answer contains the highest score (*i.e.*, 1.0). Note that $\mathcal{L}_{gen,\,kt}$ computes a *weighted* negative log-likelihood loss because each selected candidate answer a_t^i has a different confidence score.

Experimental Results. We report the performance of the generative model on the VisDial v1.0 validation split. As shown in Table 4, SGL shows slightly better performance than ReDAN (Gan et al., 2019) on overall performance. Furthermore, we find that only a small subset of the knowledge of the teacher model is also effective for this generative approach. As observed in the discriminative model in Sec. 5, the use of teacher knowledge also leads to huge NDCG improvements and the counter-effect on other metrics.

C Implementation Details

We use pre-trained Glove (Pennington et al., 2014) to embed all the language inputs. The maximum sequence length of the questions, answers, and captions is 20, 20, and 40, respectively. Based on this maximum length, each language input is padded or truncated. We use $K = 10 \sim 100$ object-level visual features for reflecting the complexity of each image. The dimension of each feature is $d_v = 2048$ and the number of attention heads in multi-head attention is $h = 2$. The dimension of d_h is 512. The total number of rounds for each dialog T is 10 and the number of candidate answers N is 100. The softmax temperature for computing the binary edges τ is 0.5. We employ the Adam optimizer (Kingma and Ba, 2014) with initial learning rate 1×10^{-4}. The learning rate is warmed up to 4×10^{-4} until epoch 4 and is halved every three epochs from 12 to 24 epochs.

D Qualitative Examples

We visualize the inferred graph structures from our proposed model and the ones from the Dense model as a comparison. As shown in Figure 7, our proposed model indeed captures semantic structures among a series of utterances by selectively attending to the dialog history. On the other hand, the Dense model yields fully-connected graphs due to two constraints of the softmax function: (1) the softmax function always assigns non-zero values to all edge weights, and (2) the sum of the edge weights for each node should be one. However, SGL can assign zero values to all edge weights if needed (*e.g.*, Q4 in the first example of Fig. 7). We believe this ability is crucial to prevent the visual dialog model from overly relying on the dialog history.

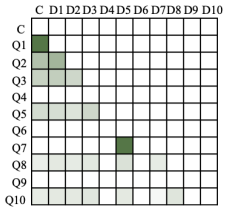
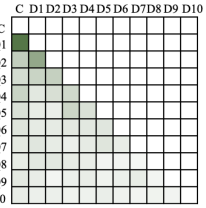
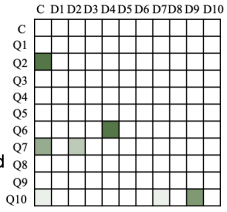
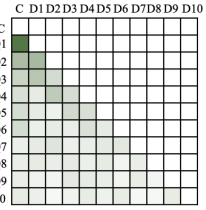
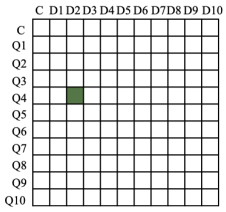
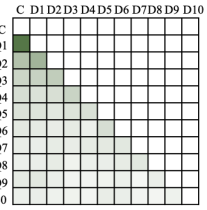
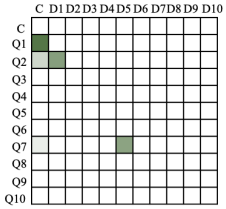
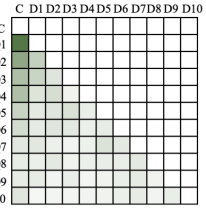
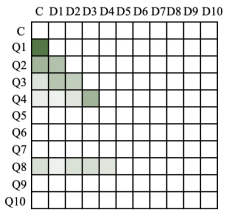
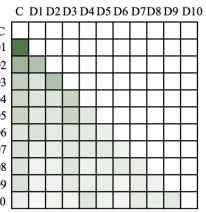
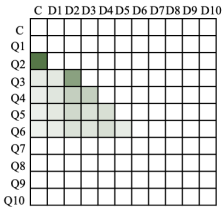
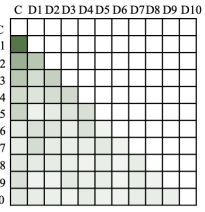
Image & Caption	Dialog	Sparse Structure (Ours)	Dense Structure
 The dog is pacing along the very grassy area	D1 : Is the dog large? No, very small D2 : What color is the dog? Tan and gray D3 : Is the dog a puppy? Nope D4 : Are there any people? I can't see any, no D5 : Does the dog have on a collar? Yes, or a bandana D6 : Are there any trees? Not that I can see D7 : What color is the collar? Red D8 : Is the dog fenced in? I can't tell, but I don't think so D9 : Can you see a house? No D10 : Is the dog standing? Yes		
 A bunk bed is in an old, wooden cabin with no sheets	D1 : Are there any people? No D2 : What color is the bunk bed? Brown D3 : What is the floor made of? Carpet D4 : Is there a window? Yes D5 : Is it night time? No D6 : What do you see through the window? A tree D7 : Is there a pillow on the bed? No D8 : Do you see other furniture? A table and another bed D9 : Is there a chair? Yes D10 : What's the chair made of? Wood		
 The red and yellow buses are on the sidewalk by a busy street	D1 : Can you see any people? Lots of people D2 : Can you see the sky? Yes a little D3 : Is there any logos or writing? Some logo on the car D4 : What color is the sky? Light blue D5 : Do you see any animals? No it is in the big city D6 : Do you see any trees? Lots of trees behind the bus D7 : Anything made out of wood? No D8 : Anything made out of glass? Cars, bus windows D9 : Anything made out of metal? Cars, bus, motorcycle D10 : Anything made out of plastic? Helmet, signs		
 A living room is filled with a lot of christmas decorations and a tree	D1 : Can any present be seen under the tree? Correct D2 : Is the tree lighted? Like on fire D3 : Are any people present? No people D4 : Is there any smoke in the view? No smoke D5 : Can you see a couch? I see a couch on the left D6 : Is there a television present in the scene? No D7 : What color is the couch? Mainly green tanish D8 : Is there a table in view? Somewhat D9. : Decorations on the wall? Lots of decorations D10 : Can you tell what time of day or night? Morning		
 Two girls playing soccer on a grass pitch	D1 : How old are the girls? Teens D2 : Is there anything besides the girls? Yes some cars D3 : What color are their shirts? One yellow, one white D4 : What color are their shorts? Blue one white one D5 : Can you see the street? No D6 : Is there a parking lot? Yes D7 : Is there a fence? Yes D8 : What color is their hair? Both blonde D9 : Can you see a ball? Yes D10 : Are there any bleachers? No		
 A skateboarder performing a trick in front of an audience	D1 : Is this in color? Yeah, this photo is in color D2 : What gender is the skateboarder? Man D3 : Is he young? I'd say he's young D4 : What is his race? He appears to be white D5 : Is he wearing a helmet? He's not wearing a helmet D6 : Is he wearing pads? It's hard to say D7 : Are there ramps? He's going off a mini ramp D8 : Is this indoors? This photo is indoors D9 : Are there a lot of people? Good amount of people D10 : Is this a competition? Looks like it's a competition		

Figure 7: The additional examples of the inferred semantic structures from the validation split. From the left, the given image and caption, the dialog history, and the structures of ours and the baseline. The darker the color, the higher the score.