Knowledge-Based Systems 259 (2023) 110069

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-task learning with graph attention networks for multi-domain task-oriented dialogue systems

Meng Zhao [a], Lifang Wang [a], Zejun Jiang [a,*], Ronghan Li [b], Xinyu Lu [a], Zhongtian Hu [a]

[a] School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, PR China
[b] School of Computer Science and Technology, Xidian University, Xi'an, PR China

ARTICLE INFO

Article history:
Received 28 July 2022
Received in revised form 20 October 2022
Accepted 22 October 2022
Available online 29 October 2022

Keywords:
Task-oriented dialogue system
Natural response generation
Graph attention network
Memory network

ABSTRACT

A task-oriented dialogue system (TOD) is an important application of artificial intelligence. In the past few years, works on multi-domain TODs have attracted increased research attention and have seen much progress. A main challenge of such dialogue systems is finding ways to deal with cross-domain slot sharing and dialogue act temporal planning. However, existing studies seldom consider the models' reasoning ability over the dialogue history; moreover, existing methods overlook the structure information of the ontology schema, which makes them inadequate for handling multi-domain TODs. In this paper, we present a multi-task learning framework equipped with graph attention networks (GATs) to probe the above two challenges. In the method, we explore a dialogue state GAT consisting of a dialogue context subgraph and an ontology schema subgraph to alleviate the cross-domain slot sharing issue. We further construct a GAT-enhanced memory network using the updated nodes in the ontology subgraph to filter out the irrelevant nodes to acquire the needed dialogue states. For dialogue act temporal planning, a similar GAT and corresponding memory network are proposed to obtain fine-grained dialogue act representation. Moreover, we design an entity detection task to improve the capability of soft gate, which determines whether the generated tokens are from the vocabulary or knowledge base. In the training phase, four training tasks are combined and optimized simultaneously to facilitate the response generation process. The experimental results for automatic and human evaluations show that the proposed model achieves superior results compared to the state-of-the-art models on the MultiWOZ 2.0 and MultiWOZ 2.1 datasets.

1. Introduction

The goal of task-oriented dialogue systems (TODs) is to complete the user's goals with multi-turn interactions through natural language, such as for restaurant reservations and ticket bookings [1,2]. With the increasing use of mobile phones for commercial applications and intelligent devices, building a human-like task-oriented dialogue system has attracted a rapid proliferation of interest from academia and industry.

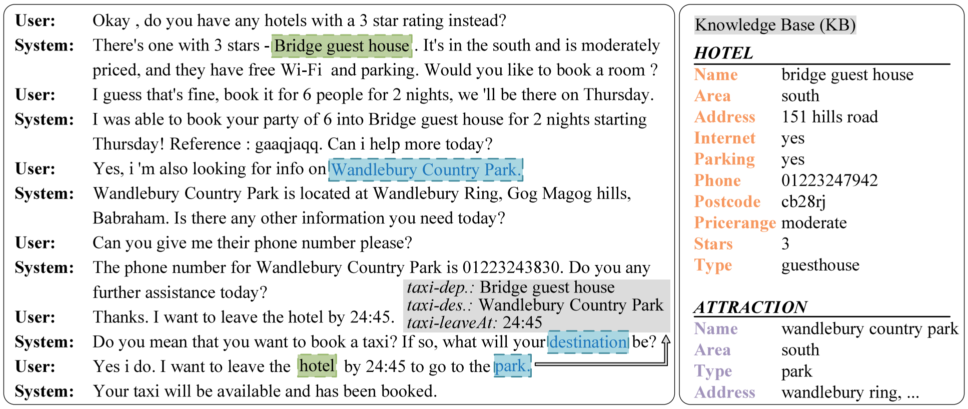
In the literature [3,4], a classic TOD generally consists of three modules, natural language understanding, dialogue management, and natural language generation. The dialogue management module is also separated into dialogue state tracking and dialogue policy learning. The entire procedure for a TOD to complete a dialogue can be described as follows. The natural language understanding module addresses a user's inputs to classify dialogue intents and also is responsible for the slot filling task. Then, the dialogue states tracking module updates the dialogue state and makes an API call to obtain the knowledge base (KB) results that match the user's goal. Furthermore, the dialogue policy learning module decides which dialogue act (including dialogue act and slot type) to choose for the next turn. Finally, the natural language generation module maps the system's act to a natural dialogue response.

Traditional pipeline solutions [3,5] model three modules independently, which requires considerable human effort to design and suffers from adaptation to new domains. Recent works usually adopt an end-to-end framework [6–9] that inputs the dialogue history and the related knowledge base, and directly carries out the responses to alleviate the above problems. Although an end-to-end approach achieves promising results, it often models a TOD in a single task domain, which is insufficient in real life. Thus, a more challenging problem appears, a multi-domain task-oriented dialogue. An example is shown in Fig. 1.

A primary challenge of a multi-domain task-oriented dialogue system is how to deal with cross-domain slot sharing and dialogue act temporal planning, which requires a model with the

[*] Correspondence to: School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, 710072, PR China.
E-mail address: claud@nwpu.edu.cn (Z. Jiang).

https://doi.org/10.1016/j.knosys.2022.110069
0950-7051/© 2022 Elsevier B.V. All rights reserved.



User: Okay , do you have any hotels with a 3 star rating instead?
System: There's one with 3 stars - Bridge guest house . It's in the south and is moderately priced, and they have free Wi-Fi and parking. Would you like to book a room ?
User: I guess that's fine, book it for 6 people for 2 nights, we 'll be there on Thursday.
System: I was able to book your party of 6 into Bridge guest house for 2 nights starting Thursday! Reference : gaaqjaqq. Can i help more today?
User: Yes, i 'm also looking for info on Wandlebury Country Park.
System: Wandlebury Country Park is located at Wandlebury Ring, Gog Magog hills, Babraham. Is there any other information you need today?
User: Can you give me their phone number please?
System: The phone number for Wandlebury Country Park is 01223243830. Do you any further assistance today?
User: Thanks. I want to leave the hotel by 24:45.
System: Do you mean that you want to book a taxi? If so, what will your destination be?
User: Yes i do. I want to leave the hotel by 24:45 to go to the park.
System: Your taxi will be available and has been booked.

taxi-dep.: Bridge guest house
taxi-des.: Wandlebury Country Park
taxi-leaveAt: 24:45

Knowledge Base (KB)

HOTEL	
Name	bridge guest house
Area	south
Address	151 hills road
Internet	yes
Parking	yes
Phone	01223247942
Postcode	cb28rj
Pricerange	moderate
Stars	3
Type	guesthouse

ATTRACTION	
Name	wandlebury country park
Area	south
Type	park
Address	wandlebury ring, ...

Fig. 1. An example of multi-domain task-oriented dialogue is shown in which a user arranges the trip schedule by interacting with the dialogue system. The left block records the dialogue history. The green chunk *Bridge guest house* in history is associated with the *hotel*. Also, the blue chunk *Wandlebury Country Park* is related to the *park* where is the taxi's destination. The right block is the knowledge base (KB) entries related to the dialogue, where the italic token *HOTEl* and *ATTRACTION* is the specific domain name.

capability of reasoning. Here, we regard the process of dialogue act selection limited on the temporal line of a dialogue as a reflection of the temporal planning problem. As shown in Fig. 1, this dialogue session occurs while a user arranges a trip plan. It discusses hotel reservations, attraction planning and taxi booking. In the first several turns, the user interacts with the system to reserve the desired hotel, the *Bridge guest house*, located in the *south*. In the following several turns, the user asks for information about *Wandlebury Country Park*, also situated in the *south*. As the dialogue flow proceeds, the user needs to book a taxi to drive from the hotel to the park. However, the user does not explicitly give the departure and destination in the dialogue. Intuitively, the system should at least have two abilities to disentangle this involved dialogue context. First, the system needs to be aware that the user has completed booking a hotel and an attraction; then, a taxi may be needed to travel between the two places. Second, the system should reason out the departure and destination of the taxi booking based on the context, where they correspond to the *Bridge guest house* and *Wandlebury Country Park*, respectively.

However, existing studies seldom consider this issue. A small number of relevant studies [10,11] have modelled the TOD in the multi-domain and suffered by exploring the dialogue response generation task as generating problems in the condition of the previously generated dialogue act. Consequently, they explicitly adopt the ground-truth of the dialogue states, which creates a gap between the real-life dialogue setting and the developed dialogue system. In addition, some recent works [12–14] have regarded TODs as end-to-end pipelined[1] frameworks by jointly optimizing three subtasks, i.e., dialogue state tracking, dialogue act prediction and response generation, to perform multi-task learning. Although these works mitigate the gap between the desired and the developed dialogue systems, they still pay little attention to the model's reasoning ability. Furthermore, they ignore the role of the structure information of the ontology schema in dialogue state reasoning.

To alleviate the above problems, we propose a **M**ulti-task learning framework equipped with graph **A**ttention **N**etworks (GATs) [16] to create **TOD**s with reasoning capabilities (**MAN-TOD**). The intuition is that when we tackle information-rich reading comprehension tasks, we always mark the helpful information (e.g., entities that represent person, place, or time, etc.) to assist us in understanding the documents. Meanwhile, tagging such helpful information can facilitate us to refocus on them more easily in later problem processing. On the other hand, tagged information allows us to extract the pulse of the historical information, thereby improving the ability to handle complex problems correctly. The graph structure is robust and solid for dealing with reasoning problems [17]. Our work embeds helpful information into the graphical nodes to implicitly mark them, making complex and changeable natural language reasonable and intuitive. In our method, we opt GAT as our backbone to aggregate the feature information; it adopts an attention mechanism that has been proven effective in solving natural language processing issues [3,18–20] to learn the relative weights between two connected nodes rather than explicitly assigning a non-parametric weight such as a graph convolutional network (GCN) [16,17,21,22].

Specifically, to address the cross-domain slot sharing issue, we propose a multi-granularity graph attention network to represent the dialogue history and dataset ontology schema in one graph, including a dialogue context subgraph and an ontology schema subgraph. The dialogue history carries the crucial semantic information and determines the direction of dialogue, and the ontology schema guides the information needed for a specific goal of the user. In the constructed graph, there are four kinds of nodes, i.e., dialogue token-level nodes, dialogue turn-level nodes, domain-level nodes, and slot-level nodes. In addition, we create eight types of edges to build connections between the two subgraphs and construct connections among different nodes in each subgraph. In this way, our model can capture the representations of the present slots (i.e., nodes in the ontology schema subgraph) with the corresponding values and share the cross-domain slot information through multi-layers updates. To implicitly obtain dialogue states to assist the response generation process, we use the updated nodes in the ontology schema subgraph to build a GAT-enhanced memory network (MN) [23,24] to filter out the irrelevant nodes. Therefore, we can acquire the necessary dialogue state representation. To enhance the model's ability to represent dialogue states, we employ a domain classification task to guide the semantic parsing.

For the dialogue act selection problem, we use a graph construction strategy consistent with the dialogue state. The only

[1] To the best of our knowledge, this term is the first to emerge in the literature [15]. This means that a model is organized following a traditional dialogue management pipeline while trained in an end-to-end fashion. Note that there are other names, such as "fully end-to-end" or is abbreviated to "end-to-end". We consider that the term "end-to-end pipelined" reflects the characteristics of this framework, and distinguishes it from the previous end-to-end training mode, which had no intermediate supervisions. Therefore, we use it here.

difference is that the dialogue act nodes are added to the graph. Then, we use the updated representations of dialogue act nodes to build a GAT-enhanced memory network. After multiple hops, the acquired outputs serve as selected dialogue act representations to initialize the response generation process. During the decoding stage, we utilize a gate mechanism to select words from the vocabulary or KB, step by step. To strengthen the performance of the soft gate mechanism, we design an auxiliary task by detecting whether the token in the response is an entity from the KB or not. Within the training time, four training tasks are combined to optimize simultaneously, the domain classification, dialogue act prediction, response generation and response entity detection.

In contrast to other representation learning methods [25–29], ours is essentially for information interaction and fusion. However, our approach is suitable for cases where the graph structure is stable (or partially stable) since the proposed method focuses on fusing information from one variable subgraph to another fixed one. The experimental results on the two public task-oriented dialogue datasets, MultiWOZ 2.0 [30] and Multi-WOZ 2.1 [31], demonstrate that the proposed method outperforms the state-of-the-art methods in terms of automatic evaluation and human evaluation by a big margin. In addition, the ablation experiments and further analyses show the effectiveness of the components of the proposed model.

The main research contributions of this paper are as follows:

- We propose two multi-granularity graph attention networks that combine the dialogue history and ontology schema for dialogue state parsing and act prediction, respectively, to explore the response generation of multi-domain task-oriented dialogue systems.
- We propose a GAT-enhanced MN module, where the GAT is adopted to aggregate helpful dialogue history information into the ontology subgraph and the MN is employed to filter out irrelevant nodes of it. To the best of our knowledge, we are the first to present the GAT-enhanced MN to handle the response generation of multi-domain TODs.
- To enhance the power of the soft gate mechanism, we design an auxiliary task named the response entity detection task and combine it with the three tasks of domain classification, dialogue act prediction, and response generation as a joint optimization to facilitate the response generation process.
- Experimental results show that our method outperforms the previous state-of-the-art methods. Moreover, ablation experiments and further analyses demonstrate the effectiveness of the proposed method.

The remainder of this paper is organized as follows. In Section 2, the previous works performed on TOD, multi-task learning and graph neural networks are examined and discussed. In Section 3, the GAT architecture and task definition that we have utilized is briefly presented. In Section 4, the proposed approach is introduced, step by step. Then, in Section 5, the datasets, experimental settings, the results, and a baseline comparison of our work are described. In Section 6, further analyses and discussions are presented. Finally, conclusions and potential research directions are presented in Section 7.

2. Related work

In this section, we provide a summary of the related work from three aspects, task-oriented dialogue systems, multi-task learning and graph neural networks.

2.1. Task-oriented dialogue system

With the development of deep learning in natural language processing (NLP) research, especially in machine translation issues, the end-to-end framework has achieved promising performance. Inspired by this methodology, many studies have introduced this data-driven method to build a task-oriented dialogue system [1,3]. The response generation process of a TOD needs to incorporate a KB into a learning framework, and subsequent works directly integrate a KB into a memory network [23,24]. They adopt an end-to-end framework without intermediate outputs to generate a response, achieving promising results [6–9]. The main drawback of MN is that it is short of sequential information of dialogue and it lacks scalability with the increasing numbers of KBs. Although some work [6] merges the hidden states of the dialogue history with an external KB to provide sequential and contextualized information, they suffer from disposing of the scalability problem.

With more complex and more domains' TOD datasets released [30–32], the capacity of KB becomes increasingly larger. Using an MN to store a KB will be an unwise choice. Some studies [12–14,33–36] use an end-to-end pipelined method to explore multi-domain TOD issues. They regard the KB as an external source of information, interacting with it after getting the dialogue states. In this way, the models eliminate the scalability problem from the KB. Compared with an encoder or decoder using recurrent neural networks (RNN), the Transformer [18] is a more potent choice. A few studies [9,14,37] adopt Transformer in the models and achieve an impressive performance. Furthermore, powered by Transformer, several generative pre-trained language models have been developed [38,39]. A list of research [40–42] demonstrates that they can handle TODs with their solid and robust generation capability. As expected, these methods perform better than the other models.

Previous studies either focused on end-to-end modelling in a single domain or multi-task learning in multi-domain but seldom considered the model's reasoning ability and ignored the structure information of the ontology schema. In contrast, the proposed method follows the above direction of joint training in an end-to-end pipelined way with a multi-task learning framework. We mainly consider leveraging GAT to handle cross-domain slot sharing and dialogue act temporal planning to facilitate the natural response generation of the multi-domain.

2.2. Multi-task learning

Multi-task learning is a learning paradigm in machine learning. It aims to leverage helpful information contained in multiple related tasks to help improve the generalization performance of all the tasks [43,44]. It has been widely used in various fields [45,46], such as computer vision and NLP [47], to achieve better performance. In this section, we mainly present the advances in the NLP fields. We strongly recommend the reader to refer to these two review articles [43,48] for a more comprehensive overview of the progress. The introduction of RNNs for NLP yields a family of sequence-to-sequence (seq2seq) models for multi-task learning, such as text classification [19], language translation [20] and machine reading comprehension [26]. In all of the NLP architectures we have presented thus far, supervision for the task-specific features of each task occurs at the same depth. Subsequently, several works [25,49] propose supervising lower-level tasks at the earlier layers so that the features learned for these tasks may be used by higher-level tasks. Note that the proposed multi-task architecture belongs to the latter category. An impressive work using multi-task learning is BERT [50], and many studies [51,52] have evolved from, or have been inspired by, BERT.

Our work follows the above procession of models where part of the auxiliary tasks are learned in the early phase. Nevertheless, the above works dispose of the task unbalance problem using hyper-parameters. In contrast, we employ uncertainty loss [53] to address this issue.

2.3. Graph neural network

Graph data are one kind of data that are widely available in a variety of works. Recently, there has been increasing interest in extending deep learning approaches for graph data. We will allocate space to introduce the advances of GAT and then give a brief review of GCN [21] and the graph isomorphism network (GIN) [54] to assist readers in comprehending this paper. We recommend that readers to refer to the literature [17] for a more detailed and comprehensive introduction. GAT [16] is a combination of a graph neural network (GNN) [55] and an attention mechanism that gives different attention to neighbouring nodes on a graph. Since there are multiple relations in one graph, Busbridge et al. [56] proposed relational graph attention networks to explore the attention mechanism between neighbouring nodes under the same relation. Wang et al. [22] presented a heterogeneous graph neural network based on node-level and semantic-level attentions to jointly consider node and meta-path information. Zeng et al. [57] also proposed a heterogeneous graph neural network. They used a word-level GAT to aggregate salient word representations, and a sentence-level GAT to capture inter- and intra-relations, respectively. Liang et al. [58] proposed a gated GAT, which unifies a graph neural network and the celebrated seq2seq to encode complete graph-structured information. Chairatanakul et al. [59] proposed a projected graph relation-feature attention network, which considers the node features and the compatibility between the connected and target relations to learn salient neighbours. GCN [21] is a semi-supervised graph convolutional network that is designed for homogeneous graphs. Recently, Song et al. [29,60] adopted a GCN to tackle the knowledge base tracing problem and achieved promising results. GIN [54] is a powerful graph neural network for graph classification, which can be considered an extension of the GCN as a first order approximation of the spectral GNN using the unnormalized graph Laplacian [61].

Our work follows the above procession of GATs and considers the different weights of the various relations. The proposed multi-relational GAT can jointly fuse the information from neighbours with various relation features.

3. Preliminary

In this section, we first introduce the task definition we attempt to address. Furthermore, we present a brief description of GAT.

3.1. Task definition

Given the dialogue history between a user and a system, we represent N turn dialogue utterances as $X = \{(U_1, S_1), (U_2, S_2), \dots, (U_{n-1}, S_{n-1}), U_n\}$, where U denotes the utterance from the *User* and S denotes the utterance from the *System*. The whole dialogue history of X can be tokenized as $(x_1, x_2, \dots, x_q)$, where q represents the lengths of the tokens in the entire dialogue history. The current dialogue turn related KB is represented as $D = (d_1, d_2, \dots, d_l)$, where l is the length of KB. Thus, given the dialogue history X and related knowledge base D, the objective of the dialogue system is to generate a natural language response $S_n = (y_1, y_2, \dots, y_j)$, step by step. Formally, the probability of a response is defined as

$$\boldsymbol{p}(y \mid X, D) = \prod_{t=1}^{j} \boldsymbol{p}(y_t \mid y_1, \dots, y_{t-1}, X, D) \tag{1}$$

where y_t represents an output token, and j denotes the length of the response.

3.2. Vanilla graph attention network

GAT is a variant of a graph neural network. It pays different attention to the neighbours and aggregates the weighted neighbours to update the current node. We denote a graph as $G = (V, E)$, where V and E are the set of nodes and edges in the graph, respectively. Specifically, given an N node graph, the initial node features of a single layer of GAT can be denoted as $\boldsymbol{H} = (\vec{\boldsymbol{h}}_1, \vec{\boldsymbol{h}}_2, \dots, \vec{\boldsymbol{h}}_N)$, $\vec{\boldsymbol{h}}_i \in \mathbb{R}^F$, where F is the number of features in each node. After an aggregation operation, the layer produces the updated node features, $\boldsymbol{H}' = (\vec{\boldsymbol{h}}'_1, \vec{\boldsymbol{h}}'_2, \dots, \vec{\boldsymbol{h}}'_N)$, $\vec{\boldsymbol{h}}'_i \in \mathbb{R}^{F'}$. The aggregation operation is conducted on its neighbourhood, which can be described as:

$$\vec{\boldsymbol{h}}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij} \boldsymbol{W}_h \vec{\boldsymbol{h}}_j\right) \tag{2}$$

where $\mathcal{N}_i$ is the first-order neighbour of node i in the graph; $\boldsymbol{W}_h \in \mathbb{R}^{F'\times F}$ is a trainable weight matrix; σ denotes the nonlinearity activation function; and α_{ij} is the attention coefficient, which measures the relevance of node i for node j. The weight α_{ij} is formulated as:

$$\alpha_{ij} = \frac{\exp\left(\mathcal{F}(\vec{\boldsymbol{h}}_i, \vec{\boldsymbol{h}}_j)\right)}{\sum_{k\in\mathcal{N}_i} \exp\left(\mathcal{F}(\vec{\boldsymbol{h}}_i, \vec{\boldsymbol{h}}_k)\right)} \tag{3}$$

where $\mathcal{F}$ is an attention function. It is denoted as follows.

$$\mathcal{F}(\vec{\boldsymbol{h}}_i, \vec{\boldsymbol{h}}_j) = \text{LeakyReLU}\left(\mathbf{a}^{\mathrm{T}}[\boldsymbol{W}_h\vec{\boldsymbol{h}}_i \parallel \boldsymbol{W}_h\vec{\boldsymbol{h}}_j]\right) \tag{4}$$

where $\mathbf{a} \in \mathbb{R}^{2F'}$ is a trainable weight matrix, T represents the transpose operation, and $\parallel$ is the concatenation operation.

In addition, to stabilize the self-attention learning process, GAT extends the above mechanism to employ *multi-head attention* following Transformer [18]:

$$\vec{\boldsymbol{h}}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \boldsymbol{W}_h^k \vec{\boldsymbol{h}}_j\right) \tag{5}$$

where K is the number of heads, and α_{ij}^k is the normalized attention weight at the kth head.

Finally, the final output is aggregated using averaging instead of concatenation on the final layer of the network. Thus, $\vec{h}'_i$ is rewritten as:

$$\vec{\boldsymbol{h}}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \boldsymbol{W}_h^k \vec{\boldsymbol{h}}_j\right) \tag{6}$$

4. Approach

In this section, we will explain our MAN-TOD model. The overall architecture of the proposed model is shown in Fig. 2. MAN-TOD mainly consists of four modules, a context encoder transforms dialogue history into token-level and turn-level semantic representations; a dialogue state reasoning module is responsible for guiding semantic parsing and filtering out unnecessary information; a dialogue act prediction module selects dialogue acts and provides aggregated dialogue act representations; a dialogue response decoder presents a soft gate mechanism to direct the generating process, determining the token either from the vocabulary or the KB. In the following sections, the details of these four modules are given.

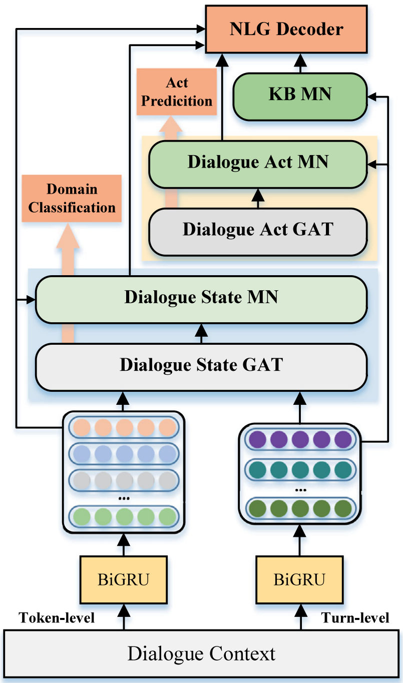


Fig. 2. The overall architecture of our proposed method. First, dialogue history is encoded into token-level and turn-level representations by two independent encoders. Then the two levels' representations and ontology schema are constructed into dialogue state GAT and dialogue act GAT. After multi-hop reasoning on these two graphs, the obtained node representations are used to build corresponding MN to filter out unnecessary information. At last, four outputs are combined to initialize decoder to generate a response token by token.

4.1. Context encoder

Our framework adopts two independent encoders to obtain token-level and turn-level dialogue semantic representations. The token-level semantic representation provides fine-grained contextual features, and the turn-level semantic representation gives coarse-grained and the turns' logical structure information. Additionally, the acquired two levels' representations initialize the dialogue state graph and the act graph.

Token-level Encoder: We adopt a bidirectional GRU [62] (BiGRU) to encode dialogue context. Given the dialogue context $X = \{T_1, T_2, \ldots, T_{n-1}, U_n\}$, $T_i = (U_i, S_i)$ is a one-turn dialogue. Note that the nth turn only appears in the user's utterance U_n; the corresponding utterance S_n is the target response generated. For the ith turn, dialogue $T_i = \{x_1, x_2, \ldots, x_k\}$, which consists of k words. The entire dialogue history represented as $x = (x_1, x_2, \ldots, x_q)$, which consists of q words. The token-level dialogue representation $\boldsymbol{H}^t = \{\boldsymbol{h}^t_1, \boldsymbol{h}^t_2, \ldots, \boldsymbol{h}^t_q\}$ can be achieved by using:

$$\boldsymbol{h}^t_i = \text{BiGRU}\left(\phi^{emb}(x_i), \boldsymbol{h}^t_{i-1}\right) \tag{7}$$

where ϕ^{emb} is the embedding function. Note that we use $\boldsymbol{h}^t_q$ as the final dialogue encoding representation to query dialogue state MN (see Section 4.2.2) to capture the information relevant to the dialogue history.

Turn-level Encoder: After obtaining the token-level dialogue representation, we utilize another BiGRU to transform every turn dialogue $T_i = \{x_1, x_2, \ldots, x_k\}$ to acquire turn-level semantic representation $\boldsymbol{H}^T_i = \{\boldsymbol{h}^T_{i,1}, \boldsymbol{h}^T_{i,2}, \ldots, \boldsymbol{h}^T_{i,k}\}$. For the jth word in T_i,

$$\boldsymbol{h}^T_{i,j} = \text{BiGRU}\left(\phi^{emb}(x_{i,j}), \boldsymbol{h}^T_{i,j-1}\right) \tag{8}$$

Normally, the last hidden state $\boldsymbol{h}^T_{i,k}$ is used as the corresponding turn's representation $\boldsymbol{H}^T_i$. Thus, the encoding representation of all the turns can be expressed as $\boldsymbol{H}^T = \{\boldsymbol{H}^T_1, \boldsymbol{H}^T_2, \ldots, \boldsymbol{H}^T_n\}$. Inspired by Qin's [63] work, we apply the concatenation of the max-pooling and mean pooling [64] operation over $\boldsymbol{H}^T_i = \{\boldsymbol{h}^T_{i,1}, \boldsymbol{h}^T_{i,2}, \ldots, \boldsymbol{h}^T_{i,k}\}$ to obtain the final turn-level dialogue representation $\hat{\boldsymbol{H}}^T_i$. Therefore, $\boldsymbol{H}^T$ is updated as $\hat{\boldsymbol{H}}^T = \{\hat{\boldsymbol{H}}^T_1, \hat{\boldsymbol{H}}^T_2, \ldots, \hat{\boldsymbol{H}}^T_n\}$.

Note that we apply the current dialogue's turn (i.e., nth turn) representation $\hat{\boldsymbol{H}}^T_n$ to query the dialogue act MN (see Section 4.3) and knowledge base MN (see Section 4.4) to filter out unnecessary information.

4.2. Dialogue state reasoning

This section presents the dialogue state reasoning module, consisting of a dialogue state graph and a memory network. We use the ontology schema, including the domain and corresponding slots, and the token-level and turn-level dialogue representations to jointly construct a multi-granularity graph. This graph aggregates helpful dialogue history information and transfer slot sharing information in the multi-domain TOD. Furthermore, the updated domain and slot nodes carrying dialogue semantic features are incorporated into an MN. Then, we utilize the dialogue history representation $\boldsymbol{h}^t_q$ to query MN to capture the critical and relevant dialogue states.

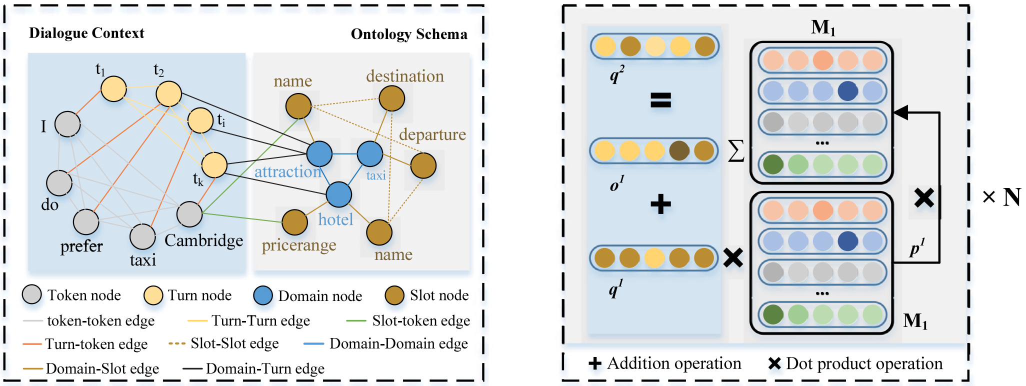
4.2.1. Dialogue state graph

Dialogue states are essential to a dialogue flow procedure. To obtain the current dialogue state representation, we leverage a graph attention network, stacking multiple layers to capture the relevant semantic features. In the graph, there are two sources of nodes separate from the ontology and dialogue history. The former node source guides the dialogue flow with required slots during the interactions. The later node source provides contextual semantic information. After combining both sources of nodes into one graph and performing multi-layers updates, the relevant nodes from the ontology will carry the corresponding semantic information.

Vertices: There are two kinds of nodes in the ontology subgraph, i.e., domain nodes N_D and corresponding slots N_S. To provide contextual semantic information at different levels, we propose two granularities of nodes of dialogue semantic representation, i.e., token-level N_t and turn-level N_T nodes. They are initialized by representations of the token-level and turn-level dialogue encoding. The token-level nodes offer fine-grained semantic features that are mainly responsible for the domain and slot nodes to identify specific entities in the dialogue history. The turn-level nodes provide utterance-level semantic features and the turns' logical structure information, which help the domain and slot nodes to capture the relevance of in-turn and across-turn dialogue semantics. Therefore, a total of four types of nodes are presented in the graph, i.e., $N = \{N_D, N_S, N_t, N_T\}$ (see Fig. 3(a)).

Edges: In the ontology schema subgraph, we attach an edge, E_{DD}, between the different domain nodes to facilitate the conveying of information in the various domains. We build an edge, E_{DS}, between each domain node and the slot nodes belonging to it. By doing this, the nodes in a subgraph belonging to the same domain can transfer information to each other. To overcome entities sharing in the multi-domain TOD, we also construct an edge, E_{SS}, between the slot nodes with the same slot type, e.g., as shown in Fig. 1, the slot of the name of the hotel and the attraction has the same slot type. Therefore there is an edge between them.

In the dialogue context subgraph, we create an edge, E_{tt}, between a token and its front tokens. This consideration is the same as in a real-world setting, where each token can only access historical information. This rule is also applied to make an edge E_{TT}



(a)

(b)

Fig. 3. The left block (a) is an illustration of the GAT we constructed. For brevity, we do not show all the edges in this graph. The right block (b) is a brief sketch of the memory network. **M** represents a memory network. **N** means the number of hops.

between a turn and its front turns in the graph. Furthermore, we also make an edge, E_{Tt}, between a token node and its corresponding turn node. Since we aim to transfer semantic information into an ontology schema, we finally make an edge, E_{DT}, between domain nodes and turn-level dialogue representation nodes and build an edge, E_{St}, between slot nodes and entity nodes.[2] Thus, there are eight kinds of edges in total, which can be represented as $E = \{E_{DD}, E_{DS}, E_{SS}, E_{tt}, E_{TT}, E_{Tt}, E_{DT}, E_{St}\}$.

Graph Update with Weighted Token-level Nodes: Through the above design of nodes and edges, the token-level nodes (i.e., special token nodes) establish connections with all the slot nodes. However, most of these connections are superfluous since one special token is always associated with one slot at its first appearance. To prune the useless connections, we compute a weight between the current node and the connected token-level nodes as follows.

$$\beta_j = \mathrm{Sigmoid}\,(\vec{\boldsymbol{h}}_d\, \vec{\boldsymbol{h}}_j) \tag{9}$$

where $\vec{h}_d$ is the domain node representation corresponding to the current update node, and $\vec{h}_j$ is the token-level node representation. Therefore, the update process in Eq. (2) can be rewritten as:

$$\vec{\boldsymbol{h}}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i} \beta_j\, \alpha_{ij} \boldsymbol{W}_h \vec{\boldsymbol{h}}_j\right) \tag{10}$$

Graph Update with Weighted Relations: As mentioned in Section 3.2, a vanilla GAT typically treats all relations in the graph as equally important, making it challenging to capture crucial relation information from many relations. Inspired by the neighbours giving different attention to the current node, we propose a multi-relational GAT that considers a weighted relation aggregation for fine-grained fusion. To be specific, we gradually calculate the neighbours' aggregation under the same relation over the entire relation. We design a hierarchical fusion mechanism that contains node-level and relation-level attention mechanisms. The relation-level attention mechanism follows [8]. Therefore, the update process of GAT will be rewritten as:

$$\vec{\boldsymbol{h}}_i' = \sigma\left(\sum_{r\in\mathcal{R}} \boldsymbol{c}^r\, \sigma\,\Big(\sum_{j\in\mathcal{N}_i} \beta_j\, \alpha_{ij} \boldsymbol{W}_h \vec{\boldsymbol{h}}_j\Big)\right) \tag{11}$$

where $\mathcal{R}$ is a set of different relations, a.k.a. edges. c^r is a trainable parameter to measure the importance of the rth relation, which can be calculated as:

$$c^r = \frac{\exp\,(o^r)}{\sum_{r'\in\mathcal{R}} \exp\,(o^{r'})} \tag{12}$$

where o^r is a non-normalized parameter for the relation r. By doing this, we can assign different weights to the different relations and perform hierarchical information fusion.

After stacking multiple layers of GAT, the node representations of the domains and slots are utilized to build a dialogue state MN to filter out the irrelevant nodes to acquire a dialogue state representation.

4.2.2. Dialogue state memory network

A memory network [23,24] is always employed to filter out the redundant information with multi-hop updates. Through the dialogue state graph, the dialogue-related domain and slot nodes capture helpful information. Since we conduct interactions and updates on the entire ontology schema subgraph, some invalid nodes, that are not associated with the dialogue history, appear. To identify and remove these invalid nodes, we utilize the updated ontology nodes (including the domain and slot nodes) to construct a dialogue state MN and adopt the dialogue history representation $\boldsymbol{h}_q^t$ to query the MN to filter out the irrelevant nodes.

Given the updated domain nodes $\{\vec{\boldsymbol{D}}_1', \vec{\boldsymbol{D}}_2', \dots, \vec{\boldsymbol{D}}_m'\}$ and slot nodes $\{\vec{\boldsymbol{S}}_1', \vec{\boldsymbol{S}}_2', \dots, \vec{\boldsymbol{S}}_n'\}$, we build a dialogue state MN $\boldsymbol{M} = \{\vec{\boldsymbol{D}}_1', \vec{\boldsymbol{D}}_2', \dots, \vec{\boldsymbol{D}}_m', \vec{\boldsymbol{S}}_1', \vec{\boldsymbol{S}}_2', \dots, \vec{\boldsymbol{S}}_n'\}$. The interaction process over MN at the kth hop can be described as follows.

First, the query vector $\boldsymbol{q}^k$ interacts with each node in MN to compute the various weights:

$$\boldsymbol{p}_i^k = \frac{\exp\,((\boldsymbol{q}^k)^{\mathrm{T}}\boldsymbol{m}_i^k)}{\sum_{j\in\mathcal{N}} \exp\,((\boldsymbol{q}^k)^{\mathrm{T}}\boldsymbol{m}_j^k)} \tag{13}$$

where $\boldsymbol{m}_i$ is the representation of the ith item in MN, $\boldsymbol{m}_i \in \{\vec{\boldsymbol{D}}', \vec{\boldsymbol{S}}'\}$, and $\mathcal{N}$ denotes all the nodes in MN. Note that $\boldsymbol{q}^0$ is $\boldsymbol{h}_q^t$.

Second, the attention $\boldsymbol{p}_i^k$ multiplies corresponding items from the adjacent MN $\boldsymbol{M}^{k+1}$ to obtain the aggregated output.

$$\boldsymbol{o}^k = \sum_{i\in\mathcal{N}} \boldsymbol{p}_i^k \boldsymbol{m}_i^{k+1} \tag{14}$$

Finally, the output plus the current query vector obtains the query vector for the next hop:

$$\boldsymbol{q}^{k+1} = \boldsymbol{o}^k + \boldsymbol{q}^k \tag{15}$$

After N hops are updated, we acquire the final output $\boldsymbol{o}^N$ as one source to initialize the input of the response decoder.

[2] We use Stanford NER toolkit to tokenize entities in the dialogue history. The link is here: https://stanfordnlp.github.io/CoreNLP/

4.3. Dialogue act prediction

This section presents the dialogue act prediction module, which consists of a dialogue act graph and a memory network. The dialogue act graph captures dialogue act representations (including act, domain, and slot) through interactions between the dialogue act ontology subgraph and dialogue context subgraph. The updated nodes in the dialogue act ontology subgraph are subsequently build a dialogue act MN to get a fine-grained dialogue act representation which is one initial input of the response generation decoder. Since the construction methods of the graph and MN are similar to those of the dialogue state reasoning module, we will briefly introduce this section and mainly introduce the different parts.

Dialogue Act Prediction Graph: Similar to the dialogue state graph, the dialogue act prediction graph is built by a dialogue act ontology subgraph and a dialogue context subgraph. There are three kinds of nodes in the ontology subgraph, domain nodes N_D, slot nodes N_S and act nodes N_A. This subgraph is adopted to guide the framework of the entire dialogue flow. The setting of the dialogue context subgraph is the same as the dialogue state graph, which consists of token-level nodes N_t and turn-level nodes N_T. The ontology nodes can capture the relevant semantic features by combining these two subgraphs. Therefore, there are five types of nodes in the dialogue act prediction graph, i.e., $N = \{N_D, N_S, N_A, N_t, N_T\}$. Note that only one kind of act node, N_A, is added compared with the dialogue state graph.

For dialogue act prediction, it is more important to give the dialogue acts in addition to providing the domains and slots. To this end, we add a new edge, E_{DA}, between the domain and the corresponding acts, and E_{TA} to connect a turn-level node and an act node. Typically, one domain is talked about in one single dialogue response. Thus, we use the softmax operation to accomplish the domain classification. A total of ten edges exist in the dialogue act prediction graph, i.e., $E = \{E_{DD}, E_{DS}, E_{SS}, E_{DA}, E_{TA}, E_{tt}, E_{TT}, E_{Tt}, E_{DT}, E_{St}\}$.

Dialogue Act Memory Network: The dialogue act prediction guides the subsequent response generation, where the domain is first needed to facilitate the act and slot production. To better predict, we separately build three MNs with corresponding nodes. Then we use the initial query vector (i.e., the current dialogue turn's representation $\hat{\boldsymbol{H}}_n^T$) to obtain the respective outputs.

First, we obtain the output $\boldsymbol{o}_N^1$ and every domain's attention weights $\boldsymbol{p}_N$ after the N-hops update. Furthermore, we utilize the attention $\boldsymbol{p}_N$ multiplied by the corresponding slot nodes and act nodes in act MN and slot MN, respectively. By doing this, we can obtain more fine-grained features. After multiple hops, we obtain the output $\boldsymbol{o}_N^2$ and $\boldsymbol{o}_N^3$, respectively. Finally, we concatenate three outputs as one initial input to the decoder.

4.4. Dialogue response decoder

In this section, we describe the details of the decoder, which generates a token-by-token response. To obtain a better generation performance, we combine the last hidden state $\boldsymbol{h}_q^t$ of the dialogue history, the output of dialogue state MN, dialogue act MN and the related KB representation to initialize the input of the decoder. Note that we adopt an MN to store the relevant KB information in the form of triplets [6] and query it using the current utterance representation $\hat{\boldsymbol{H}}_n^T$ to obtain the desired KB entries.

At the tth step, the generated word is either from the vocabulary or copied from the KB. To improve the model dealing with the special tokens in the knowledge base and dialogue history, we leverage a soft gate to determine whether the output token comes from the vocabulary or KB. We further design an auxiliary task to enhance the power of the soft gate by checking whether the generated token is a KB entry. Note that the special tokens in the dialogue history also appear in the related knowledge base. We adopt GRU as our decoder (see Fig. 4).

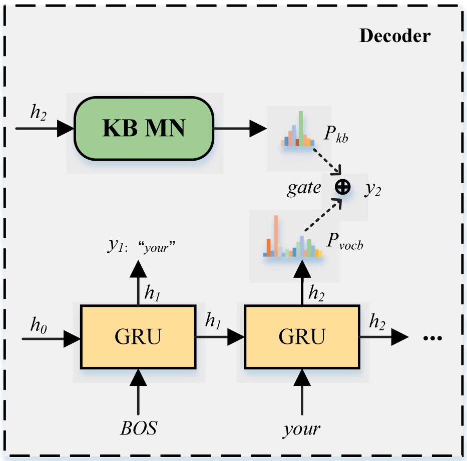


Fig. 4. The decoder architecture. At the tth decoding step, the soft gate generates a probability, determining whether the response token is generated from vocabulary or KB.

Generating Words from the Vocabulary: At the tth decoding step, given the last output word y_{t-1} and hidden state $\boldsymbol{h}_{t-1}$, the decoding process is formally described as:

$$\boldsymbol{h}_t^d = \mathrm{GRU}\,(\phi^{emb}(y_{t-1}), \boldsymbol{h}_{t-1}^d) \tag{16}$$

$$\boldsymbol{P}^{vocb}(y_t) = \mathrm{Softmax}(\boldsymbol{W}_g \boldsymbol{h}_t^d) \tag{17}$$

where $\boldsymbol{W}_g$ is a trainable matrix.

Copying Entries from the KB: When the generated word is a special token that needs to copy from the KB, the current hidden state $\boldsymbol{h}_t$ is employed to query the KB memory network. After N hops are updated, we obtain the attention distribution $\boldsymbol{p}_t^d$ as our pointer $\boldsymbol{P}^{kb}(y_t)$ to point to the copied word. The operations of a memory network are the same as Eqs. (13)–(15).

The generated token at the tth step is determined by a soft gate g_s, which can be described as:

$$g_s = \mathrm{Sigmoid}\,(\boldsymbol{W}_c[\boldsymbol{o}_c \parallel \boldsymbol{h}_t] + \boldsymbol{b}_c) \tag{18}$$

$$\boldsymbol{P}(y_t) = g_s \boldsymbol{P}^{vocb}(y_t) + (1 - g_s)\boldsymbol{P}^{kb}(y_t) \tag{19}$$

where $\boldsymbol{W}_c$ and $\boldsymbol{b}_c$ are learnable parameters, and $\boldsymbol{o}_c$ is the output of the KB memory network after N hops. We use a cross-entropy loss $\mathcal{L}_{gen}$ to train our response generation process. Formally:

$$\mathcal{L}_{gen} = -\sum_{t=1}^{j} \log\,(\boldsymbol{P}(y_t)) \tag{20}$$

Auxiliary Task: Since the ability to generate KB entries determines the model's performance, we design an auxiliary task to enhance the soft gate. We use a label l_a to mark whether the token in the response is an entity from the KB. If so, the label is set to 1; otherwise, it is 0. We use a binary cross-entropy loss $\mathcal{L}_{aux}$ to train this task. In the formula:

$$\mathcal{L}_{aux} = -\sum_{t=1}^{j} [g_s^t \log\,(l_a^t) + (1 - g_s^t) \log\,(1 - l_a^t)] \tag{21}$$

As these two tasks are trained at the same phase (i.e., response generation phase), we optimize these two tasks together.

$$\mathcal{L}_{res} = \mathcal{L}_{gen} + \alpha \mathcal{L}_{aux} \tag{22}$$

where the relative weight α is a hyper-parameter.

4.5. Multi-task learning

In our approach, there are four tasks that must be optimized simultaneously. They are domain classification, dialogue act prediction, dialogue response generation and an auxiliary task for detecting whether the generated tokens are from the vocabulary or knowledge base.

Domain Classification: To improve the performance of dialogue state reasoning, we utilize a domain classification task to capture the mentioned domains. Since there may exist more than one domain in the dialogue history, which can be considered multiple binary classification tasks, we use a binary cross-entropy loss $\mathcal{L}_{dom}$ to train the task. The classification task is described as follows.

$$g_d^i = \text{Sigmoid}\,(\boldsymbol{W}_d \vec{\boldsymbol{D}}_i + \boldsymbol{b}_d) \tag{23}$$

$$\mathcal{L}_{dom} = -\sum_{i=1}^{m} [g_d^i \log(l_d^i) + (1 - g_d^i)\log(1 - l_d^i)] \tag{24}$$

where $\vec{\boldsymbol{D}}'_i$ is the updated embedding of the ith domain in the dialogue state graph, l_d^i is the label of the ith domain, and $\boldsymbol{W}_d$ and $\boldsymbol{b}_d$ are trainable matrices.

Dialogue Act Prediction: For the dialogue act prediction task, we treat the domain prediction as a multi-class classification task, and the act and slot prediction as a multiple binary classification task.

We observe that only one domain is discussed in one system's response. Therefore, we utilize a softmax operation to accomplish the domain classification and formulate it as:

$$\boldsymbol{p}^d = \text{Softmax}\,(\boldsymbol{W}_e\, \boldsymbol{o}_N^1 + \boldsymbol{b}_e) \tag{25}$$

where $\boldsymbol{o}_N^1$ is the output vector after N hops updating in domain MN. A cross-entropy loss is employed to optimize this objective.

After recognizing the domain, we can further predict the dialogue act and slot belonging to this domain. More concretely, in the training stage, we use the ground-truth label of the domain to guide the space of the dialogue act nodes and slot prediction nodes. In the inference stage, we select the max number (i.e., target domain) of softmax, rescale it to one, and rescale the others to zero to conduct the prediction task. A sigmoid function is directly adopted over the updated act nodes and slot nodes (i.e., the nodes belonging to the target domain) in the dialogue act ontology subgraph to conduct the respective binary classifications. In the inference stage, the sigmoid thresholds for the two tasks are set to 0.55 and 0.5, respectively. A binary cross-entropy loss is selected for the act and slot prediction tasks. The loss of the three tasks is represented as $\mathcal{L}_{act}$.

Uncertainty Loss: Typically, all the parameters of the multi-task learning models are jointly trained by minimizing the weighted losses, which is defined as:

$$\mathcal{L} = \beta\mathcal{L}_{res} + \gamma\mathcal{L}_{dom} + \delta\mathcal{L}_{act} \tag{26}$$

where the relative weights β, γ and δ are hyper-parameters. However, these three supervised tasks occur at different depths, making the relative weights unstable to tune [65]. Instead, we adopt an uncertainty loss [53] to adjust them adaptively:

$$\mathcal{L}(\sigma_1, \sigma_2, \sigma_3) = \frac{1}{{\sigma_1}^2}\mathcal{L}_{res} + \frac{1}{{\sigma_2}^2}\mathcal{L}_{dom} + \frac{1}{{\sigma_3}^2}\mathcal{L}_{act} + \log(\sigma_1\sigma_2\sigma_3) \tag{27}$$

where σ_1, σ_2, and σ_3 are three learnable parameters. The advantage of this uncertainty loss is that it models the homoscedastic uncertainty of each task and provides task-dependent weight for multi-task learning [53,66].

In general, the complete loss function of the proposed method can be written as

$$\mathcal{L}(\sigma_1, \sigma_2, \sigma_3, \alpha) = \frac{1}{{\sigma_1}^2}(\mathcal{L}_{gen} + \alpha\mathcal{L}_{aux}) + \frac{1}{{\sigma_2}^2}\mathcal{L}_{dom} + \frac{1}{{\sigma_3}^2}\mathcal{L}_{act} + \log(\sigma_1\sigma_2\sigma_3) \tag{28}$$

where the relative weight α is a hyper-parameter as mentioned in Section 4.4.

5. Experiments

In this section, we conduct a series of experiments to demonstrate the effectiveness of our proposed method. We first introduce the datasets, automatic evaluation metrics and the experimental settings. Then, we present several baselines to compare with our method. Finally, we show the results of an automatic evaluation and a human evaluation.

5.1. Datasets and evaluation metrics

To better evaluate the performance of the proposed model, two popular datasets are used to conduct the experiments. They are the MultiWOZ 2.0 [30] dataset and its extended version the MultiWOZ 2.1 [31] dataset. MultiWOZ 2.0 is a large-scale multi-domain Wizard-of-Oz dataset, a fully labelled collection of human–human written conversions spanning seven domains, including *Attraction, Hospital, Police, Hotel, Restaurant, Taxi and Train*. Each dialogue is annotated with a sequence of dialogue states and corresponding system dialogue acts. There are 3406 single-domain dialogues and 7032 multi-domain dialogues, and each multi-domain dialogue contains at least two to five domains. The corpus is randomly split into training, development, and test sets, containing 8438, 1000 and 1000 dialogues, respectively. Note that *Hospital* and *Police* are not present in the development and test sets. The MultiWOZ 2.1 dataset is the corrected and consolidated MultiWOZ 2.0 dataset, which corrected the dialogue state annotations, dialogue utterances, and augmented dialogue act information.

To evaluate the proposed model, we opt for four widely used automatic metrics: **Inform** measures whether a system provides appropriate entities, **Success** assesses whether it answers all requested information, and **BLEU** is used to measure the fluency of a generated response. **Combined** score: (($Inform$+$Success$)×0.5+ $BLEU$) is as an overall quality measure as before [12,34,36].

5.2. Implementation details

We use the PyTorch library to implement our experiments. The dimensions of the word embeddings and hidden sizes are 100 and 200, respectively. The Adam [67] optimizer is adopted. The learning rate annealing starts from 5e-5 with a decay rate of 0.5. The dropout ratio equals 0.1, and the batch size is 16. We set α to 0.5, which is selected via a grid search from {0.2, 0.4, 0.5, 0.6, 0.8, 1.0, 1.1}. The layers of the two GATs are set to 3 and 2, respectively. The number N of multiple hops of the memory network is set to 3. We use the early stop training strategy, in which training is stopped if no improvement is observed, which lasts for 8 epochs on the development set. We select the best models based on validation loss. A simple greedy strategy is employed during the decoding phase. We utilize HuggingFace's Transformer [68] to implement the proposed MAN-TOD+ model. The multi-head of self-attention is 5. The stacked layer is set to 3. In addition, since our approach directly generates natural text responses, we apply delexicalization processing for the final outputs in MultiWOZ 2.0 and 2.1, making it comparable to the baselines. We use the ground-truth to build dialogue state MN and do not perform the domain classification task in the context-to-text setting. We report the average performance of each metric over 5 runs in the test phase.

Table 1
Main results of the experiments on MultiWOZ 2.0 and MultiWOZ 2.1 in a context-to-text setting. The bold numbers in the table represent the best results in the corresponding task, and the underlined numbers represent the second-best results. In a context-to-text setting, the model uses the ground-truth belief states.

Model	MultiWOZ 2.0				MultiWOZ 2.1			
	Inform	Success	BLEU	Combined	Inform	Success	BLEU	Combined
SFN+RL [34]	82.7	72.1	16.3	93.7	–	–	–	–
HDSA [10]	82.9	68.9	**23.6**	99.5	86.3	70.6	**22.4**	100.8
DAMD [12]	89.2	77.9	18.6	102.2	–	–	–	–
Marco [11]	90.3	75.2	19.5	102.2	91.5	76.1	18.5	102.3
PARG [13]	91.1	78.9	18.8	103.8	–	–	–	–
Marco(BERT)[11]	92.3	78.6	20.0	105.5	92.5	77.8	19.5	104.7
MAN-TOD	91.6	81.5	19.5	106.1	91.8	81.6	19.2	105.9
MAN-TOD+	**93.2**	**82.7**	20.7	**108.7**	**93.6**	**82.8**	19.9	**108.1**

5.3. Baselines

We choose several strong models of the same magnitude as ours for comparison.

- **TSCP** [33]: This seq2seq model is designed based on the pipeline method. The belief span and response are generated sequentially. *L* means the maximum dialogue state span.
- **SFN+RL** [34]: This model uses independently pre-trained neural dialogue modules and reinforcement learning for end-to-end dialogue generation.
- **HDSA** [10]: This model proposes a multi-layer hierarchical graph to represent dialogue acts as a root-to-leaf route and uses hierarchical disentangled self-attention to model designated nodes on the dialogue act graph.
- **DAMD** [12]: This model is a pipelined-based modelling method that proposes a multi-action augmentation framework to generate diverse dialogue responses.
- **MARCO** [11]: This model proposes a neural co-generation framework that generates dialogue acts and responses concurrently.
- **PARG** [13]: This model proposes a paraphrase augmented response generation framework.
- **HRED** [35]: This model proposes a teacher–student framework to transfer well-learned knowledge into an inference phrase to avert error propagation of the external state tracker.
- **UniConv** [14]: This model proposes a unified neural architecture for end-to-end conversational systems, including a bi-level state tracker and a joint dialogue act and response generator.
- **LABES-S2S** [36]: This model proposes a probabilistic dialogue model where belief states are represented as discrete latent variables and are jointly modelled with system responses, given the user inputs.
- **HIER** [37]: This model is a generalized framework for hierarchical encoders in transformer-based models.
- **JOUST** [69]: This model proposes a novel learning framework for developing dialogue systems that perform joint optimization with a user simulator.

5.4. Automatic evaluation

We report the main results of the proposed method on both datasets in tables. Table 1 shows the results in the context-to-text setting, in which the model uses the ground-truth belief states and the generated dialogue acts to carry out responses. Table 2 shows the results in the end-to-end setting, in which the model adopts the generated belief states and dialogue acts to develop responses.

Context-to-Text: From Table 1, we notice that our proposed model, MAN-TOD, achieves comparable results compared with the baseline models, where the metrics *SUCCESS* and *Combined* obtain the best performance. More importantly, the proposed model MAN-TOD+ achieves superior performance on almost all metrics (except *BLEU*) on both datasets, outperforming the state-of-the-art (SOTA) model MARCO(BERT) by a big margin. Furthermore, MAN-TOD+ gets the second-best performance on the *BLEU* metric for both datasets. These observations demonstrate that the proposed method is effective. On the other hand, we also find that the BLEU performance of the proposed MAN-TOD model is lower than that of the HDSA and MARCO models, both of which use BERT as the dialogue act predictor. We argue that this may be due to the BERT providing a more appropriate dialogue act. Similar to the baseline models, there is an overall improvement when our model is equipped with a Transformer. This indicates that the Transformer holds solid capabilities in semantic parsing.

End-to-End: In this setting, our MAN-TOD and MAN-TOD+ models achieve the best performance on most metrics on both datasets. More importantly, MAN-TOD+ outperforms the SOTA model JOUST on all metrics. Specifically, MAN-TOD+ delivers 0.7%, 1.6%, 8.5% and 2.5% improvements over the JOUST model on MultiWOZ 2.0 in terms of *Inform*, *Success*, *BLEU*, and *Combined* respectively. Moreover, MAN-TOD+ attains a similar trend improvement to the best results of the baseline model on MultiWOZ 2.1 on four metrics. The experimental results show that the proposed method is effective. In addition, we observe that the scores of the end-to-end setting on all the metrics show a pronounced decrease compared to the context-to-text setting. This is consistent with the general perception. On the other hand, we find that HIER-Joint performs better on the BLEU metric than our model. This is due to the baseline model integrating a hierarchical Transformer to obtain a more powerful generation capability.

5.5. Human evaluation

To more thoroughly evaluate the proposed model, we randomly select 200 responses generated by MAN-TOD+ on the MultiWOZ 2.0 dataset, and hire two human experts to distribute a score for each response from 1 to 5 in three aspects, i.e., correctness, fluency and human-likeness [7]. Note that the higher score is, the better the performance. The results are given in Table 3. We can see that the proposed model achieves better a performance compared with the two baseline models, which is consistent with the automatic evaluation.

6. Analysis and discussions

6.1. Ablation study

To verify the effectiveness of the different components of the proposed model, we assess the value on both datasets by removing them from our framework in an end-to-end dialogue setting.

Table 2
Main results of the experiments on MultiWOZ 2.0 and MultiWOZ 2.1 in an end-to-end setting.

Model	MultiWOZ 2.0				MultiWOZ 2.1			
	Inform	Success	BLEU	Combined	Inform	Success	BLEU	Combined
TSCP(L=20)[33][a]	–	–	–	–	66.4	45.3	15.5	71.4
HRED [35]	66.0	53.3	17.5	77.2	–	–	–	–
HRED-TS [35]	70.0	58.0	17.5	81.5	–	–	–	–
SFN+RL [34]	73.8	58.6	16.9	83.0	–	–	–	–
DAMD [12]	76.3	60.4	16.6	85.0	–	–	–	–
UniConv [14]	–	–	–	–	72.6	62.9	**19.8**	87.6
LABES-S2S [36][b]	–	–	–	–	76.9	63.3	17.9	88.0
HIER-Joint [37]	80.5	71.7	**19.7**	95.8	–	–	–	–
JOUST [69]	83.2	73.5	17.6	96.0	–	–	–	–
MAN-TOD	82.5	73.8	18.4	96.6	82.7	73.7	18.3	96.5
MAN-TOD+	**83.8**	**74.7**	19.1	**98.4**	**84.0**	**74.8**	18.8	**98.2**

[a] Means the results are extracted from [14].
[b] Indicates that the results are reported as the mean of 5 runs from their paper.

Table 3
Human evaluation of responses on the randomly selected dialogues from the MultiWOZ 2.0 dataset in an end-to-end setting. "Average Agreement" is the inter-annotator agreement.

Model	Correct	Fluent	Human-like
UniConv	3.82	3.98	4.08
JOUST	3.89	4.08	4.13
MAN-TOD	3.95	4.14	4.21
Average agreement	70.2%	61.5%	63.0%

We conduct a series of experiments from four perspectives, and the ablation test results are reported in Table 4.

First, we remove the token-level encoder and turn-level encoder in turn to investigate the effect of the hierarchical encoders. It can be seen that the performance of both datasets displays a decrease in all the metrics. This demonstrates that both encoders play a practical role. Moreover, compared to removing the turn-level encoder, we observe a sharper drop by removing the token-level encoder. The reason for this is that the token-level encoder provides fine-grained semantic information.

Second, we find a significant performance drop by detaching the domain classification task, which suggests that this task is critical to our framework. We also observe that there is a decrease when deleting the auxiliary task. This indicates that it is helpful to point out the KB entities during the training phase. In addition, instead of using uncertainty loss, we adopt the classic hyper-parameter setting to train multi-task learning (β, γ, and δ in Eq. (26) are all set to 1). To compare the discrepancy between the manually set hyper-parameters and the learnt weights, we select $\sigma_1, \sigma_2, \sigma_3$ (in Eq. (27)) of the best performance model in five runs on the MultiWOZ 2.0 dataset, where they are 0.8011, 0.8309 and 0.8247, respectively. Because the losses are weighted by the inverse of the uncertainty estimates (i.e., σ^2), this results in task weighting ratios of approximately $1, 0.927$ and 0.941 between $\mathcal{L}_{res}$, $\mathcal{L}_{dom}$ and $\mathcal{L}_{act}$, respectively. We observe that there exists a certain gap between these two types of weights. More importantly, it can be found that there is a significant reduction in the performance of the proposed model when using the manually set hyper-parameters, demonstrating that the uncertainty loss helps to optimize multi-task learning automatically.

Third, we remove the weighted token-level nodes and update the current node with old graph nodes to verify the validity of the pruning strategy. The results of the proposed model are greatly reduced, which proves the importance of pruning the redundant edges. In addition, we delete the weighted relation in both graph neural networks and replace it with the same weight relation, resulting in a slight reduction in performance. This proves the proposed hypothesis that different relations play different roles in the graph. On the other hand, we replace the unidirectional edges of E_{tt} and E_{TT} with bidirectional edges, which leads to a slight reduction in performance. This demonstrates that bidirectional representation between nodes is unnecessary for the dialogue context subgraph.

Fourth, we vary the query vectors of three MNs to verify the effect of the different initial query vectors. Note that $\boldsymbol{h}_q^t$ and $\hat{\boldsymbol{H}}_n^T$ are the last hidden states of the token-level encoder and turn-level encoder (see Section 4.1). The proposed model's performance on the dialogue state MN achieves a slight decline, while there is almost no change on the remaining two MNs. This indicates that the dialogue state MN is more sensitive to the initial query vector.

6.2. Effectiveness of various nodes and relations

In this section, we discuss the effectiveness of the different relations and nodes in both neural graphs in an end-to-end setting. Specifically, we do not explore the relations (i.e., E_{DT}, E_{St}, E_{TA}) between the ontology nodes and dialogue representation nodes because these relations are indispensable for transferring semantic information from the dialogue context subgraph to the ontology subgraph. So is the relation E_{DA} with a similar reason.

From the results in Table 5, we find an overall degradation in all the metrics by removing the corresponding relations. More specifically, we observe that removing relations belonging to the ontology subgraph has a larger drop than removing relations belonging to the dialogue context subgraph. This demonstrates that the relations in the ontology subgraph are more important. Meanwhile, there is a slight decrease when removing the relation E_{Tt}, which is connected between turn-level dialogue context representation nodes and token-level dialogue context representation nodes. This is our view because the turn-level nodes also contain the corresponding tokens' information. In addition, we observe that a larger drop occurs when the relation E_{DD}, representing the domain–domain edge, is removed. This indicates that cross-domain information sharing is essential for the dialogue state updating process.

We remove the token-level and turn-level dialogue representation nodes of both neural graphs separately to investigate the impact of different grains of semantic information. The results are shown at the bottom of Table 5. Specifically, we find that our model performs poorly on both operations, suggesting that both fine-grained and coarse-grained semantics can work well in our model. However, we observe that there is a greater drop by cutting the turn-level nodes than the turn-level nodes, indicating that fine-grained semantic information plays a more important role. This is in line with what we know.

Table 4
Results of ablation study on both datasets in an end-to-end setting. w/o mean the proposed model without the corresponding component. w means the opposite.

Model	MultiWOZ 2.0			MultiWOZ 2.1		
	Inform	Success	BLEU	Inform	Success	BLEU
MAN-TOD	82.5	73.8	18.4	82.7	73.7	18.3
w/o token-level Encoder	81.2(−1.3)	73.1(−0.7)	17.8(−0.6)	81.2(−1.5)	73.1(−0.6)	17.6(−0.7)
w/o turn-level Encoder	81.5(−1.0)	73.4(−0.4)	18.2(−0.2)	81.8(−0.9)	73.3(−0.4)	17.9(−0.4)
w/o domain classification	75.7(−6.8)	68.3(−5.5)	16.8(−1.6)	76.2(−6.5)	68.1(−5.6)	16.8(−1.5)
w/o auxiliary task	81.7(−0.8)	73.4(−0.4)	18.0(−0.4)	82.0(−0.7)	73.3(−0.4)	17.8(−0.5)
w/o uncertainty loss	79.8(−2.7)	72.0(−1.8)	17.2(−1.2)	80.1(−2.6)	71.9(−1.8)	17.0(−1.3)
w/o weighted token-level nodes	78.3(−4.2)	70.1(−3.7)	17.2(−1.2)	78.7(−4.0)	70.2(−3.5)	17.2(−1.1)
w/o weighted relations	81.9(−0.6)	73.5(−0.3)	18.2(−0.2)	82.2(−0.5)	73.3(−0.4)	18.1(−0.2)
w E_{tt} and E_{TT} with bidirectional edge	82.6(+0.1)	73.7(−0.1)	18.3(−0.1)	82.7(0.0)	73.6(−0.1)	18.3(0.0)
w q^0 in dialogue state MN using $\hat{H}_n^T$	82.3(−0.2)	73.7(−0.1)	18.3(−0.1)	82.6(−0.1)	73.6(−0.1)	18.3(0.0)
w q^0 in dialogue act MN using h_q^t	82.4(−0.1)	73.7(−0.1)	18.4(0.0)	82.6(−0.1)	73.7(0.0)	18.3(0.0)
w q^0 in knowledge base MN using h_q^t	82.6(+0.1)	73.8(0.0)	18.3(−0.0)	82.6(−0.1)	73.6(−0.1)	18.3(0.0)

Table 5
Results of removing different nodes and relations in both neural graphs in an end-to-end dialogue setting.

Model	MultiWOZ 2.0			MultiWOZ 2.1		
	Inform	Success	BLEU	Inform	Success	BLEU
MAN-TOD	82.5	73.8	18.4	82.7	73.7	18.3
w/o E_{tt}	81.9(−0.6)	73.5(−0.3)	18.3(−0.1)	82.3(−0.4)	73.5(−0.2)	18.1(−0.2)
w/o E_{TT}	81.8(−0.7)	73.3(−0.5)	18.2(−0.2)	82.0(−0.7)	73.3(−0.4)	18.1(−0.2)
w/o E_{Tt}	82.1(−0.4)	73.7(−0.1)	18.3(−0.1)	82.3(−0.4)	73.6(−0.1)	18.2(−0.1)
w/o E_{DD}	80.8(−1.7)	72.9(−0.9)	17.8(−0.6)	81.2(−1.5)	72.7(−1.0)	17.6(−0.7)
w/o E_{DS}	81.0(−1.5)	73.0(−0.8)	17.8(−0.6)	81.3(−1.4)	72.9(−0.8)	17.8(−0.5)
w/o E_{SS}	81.4(−1.1)	73.2(−0.6)	18.0(−0.4)	81.7(−1.0)	73.1(−0.6)	17.9(−0.4)
w/o token nodes	78.9(−3.6)	71.0(−2.8)	17.2(−1.2)	79.3(−3.4)	71.2(−2.5)	17.0(−1.3)
w/o turn nodes	81.4(−1.1)	73.2(−0.6)	18.1(−0.3)	81.5(−1.2)	73.2(−0.5)	18.1(−0.2)

Table 6
Performance achieved by different GNN variants as backbone in an end-to-end setting. *WR* represents weighted relations.

Backbone	MultiWOZ 2.0			MultiWOZ 2.1		
	Inform	Success	BLEU	Inform	Success	BLEU
MAN-TOD	82.5	73.8	18.4	82.7	73.7	18.3
GCN	80.8(−1.7)	72.8(−1.0)	17.7(−0.7)	81.1(−1.6)	72.7(−1.0)	17.5(−0.8)
GCN with *WR*	81.3(−1.2)	73.0(−0.8)	17.9(−0.5)	81.3(−1.4)	72.8(−0.9)	17.7(−0.6)
GIN-0	81.0(−1.5)	73.0(−0.8)	17.8(−0.6)	81.2(−1.5)	72.9(−0.8)	17.7(−0.6)
GIN-0 with *WR*	81.4(−1.1)	73.2(−0.6)	18.0(−0.4)	81.6(−1.1)	73.1(−0.6)	17.9(−0.4)
GIN-ϵ	81.1(−1.4)	73.1(−0.7)	17.8(−0.6)	81.4(−1.3)	73.0(−0.7)	17.8(−0.5)
GIN-ϵ with *WR*	81.4(−1.1)	73.3(−0.5)	18.0(−0.4)	81.5(−1.2)	73.1(−0.6)	17.8(−0.5)

6.3. Effect of different GNN variants

The popularity of convolutional GNNs has experienced rapid growth over the past few years due to the more efficient and convenient feature extractions in combination with graph convolutions and other neural networks. As a consequence, many variants have emerged that can be categorized into two groups, spectral-based and spatial-based. To verify the validity of the GAT employed in our model, we opt for another well-known spectral-based GNN, namely, GCN [21], and a spatial-based GNN, namely, GIN [54] for comparison experiments. Note that, ϵ is a learnable parameter. The results can be found in Table 6.

It can be seen that both variants perform poorly compared with the GAT (i.e., the backbone in our method). This may indicate that GAT is more solid in fusing useful information from neighbours, especially in the proposed method. More importantly, the scores of adopting GCN as the backbone are lower than those of using GIN, which illustrates that GIN is more capable of aggregating feature information. On the other hand, we notice that all the variants achieve better performance when considering the various weights of each relation, which proves that updating the graph with weighted relations is adequate. Enlightened by this cascading update thought, it may be a productive endeavour to consider asynchronously updating [70] the nodes of a hierarchical graph since the nodes at the different levels may carry various types of information.

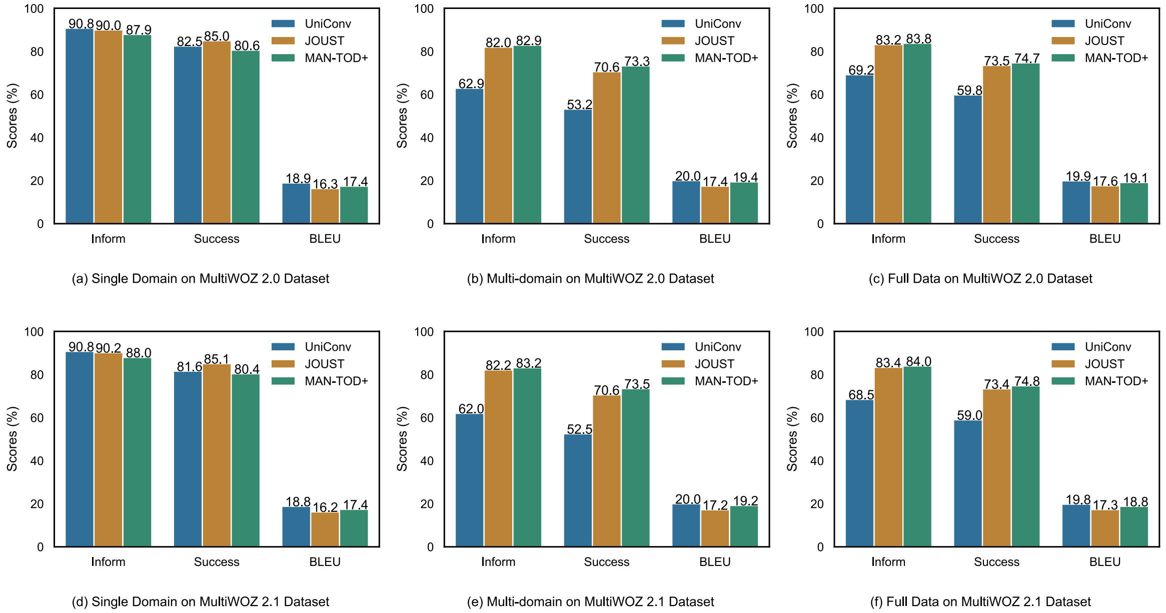
6.4. Effect of different settings in turn-level dialogue representation

As described in Section 4.1, we select max-mean pooling to acquire each dialogue turn's turn-level representation. To explore the impacts of the different representations, we conduct a series of experiments to verify the differences of the four methods, namely, h, which represents the last hidden state of each dialogue turn, max pooling, mean pooling, and max-mean pooling. The results are reported in Table 7.

It is noticed that there is a different degradation in the performance of our model when max-mean pooling is not used. This demonstrates that max-mean pooling can capture more comprehensive semantic features. More importantly, we find that the model performs worst when choosing h as the representation of the turn-level dialogue. This may be because h contains less information as the length of the turn-level dialogue increases.

6.5. Single domain vs. multi-domain

To prove the effectiveness of the proposed method in a multi-domain dialogue setting, we split the test set into two parts, one for a single-domain dialogue and the other for a multi-domain dialogue, and then we design two kinds of experiments. In the first experimental setting, all models are trained on the original training set. In other words, these models are trained on both the



(a) Single Domain on MultiWOZ 2.0 Dataset

(b) Multi-domain on MultiWOZ 2.0 Dataset

(c) Full Data on MultiWOZ 2.0 Dataset

(d) Single Domain on MultiWOZ 2.1 Dataset

(e) Multi-domain on MultiWOZ 2.1 Dataset

(f) Full Data on MultiWOZ 2.1 Dataset

Fig. 5. The performance of the baseline model and the proposed model are trained on the original train set and tested on single-domain or multi-domain dialogue sets, respectively. The upper figure shows the performance on MultiWOZ 2.0 dataset, and the lower figure presents the performance on MultiWOZ 2.1 dataset. All experiments are accomplished in an end-to-end setting.

Table 7
Effect of different choices of turn-level dialogue representation in an end-to-end setting.

Operations	MultiWOZ 2.0			MultiWOZ 2.1		
	Inform	Success	BLEU	Inform	Success	BLEU
h	82.2(−0.3)	73.6(−0.2)	18.3(−0.1)	82.4(−0.3)	73.6(−0.1)	18.1(−0.2)
Max	82.3(−0.2)	73.7(−0.1)	18.3(−0.1)	82.5(−0.2)	73.5(−0.2)	18.2(−0.1)
Mean	82.1(−0.4)	73.5(−0.3)	18.2(−0.2)	82.3(−0.4)	73.5(−0.2)	18.1(−0.2)
Max-Mean	82.5	73.8	18.4	82.7	73.7	18.3

Table 8
Results of the baseline model and the proposed model when trained and tested in an end-to-end setting on the same type of sets.

Different Setting	Model	MultiWOZ 2.0			MultiWOZ 2.1		
		Inform	Success	BLEU	Inform	Success	BLEU
Single Domain	JOUST	79.5	65.0	14.7	79.8	65.1	14.7
	MAN-TOD+	74.1	63.7	16.1	74.5	63.9	16.0
Multi-domain	JOUST	80.3	70.3	19.3	80.5	70.4	19.1
	MAN-TOD+	81.5	72.4	20.1	81.8	72.5	19.8

single and multi-domain dialogues. Then, the well trained models are used to test the two test sets. The results are shown in Fig. 5. In another experimental setting, we split the training set into two parts (i.e., single-domain and multi-domain), and all the models are trained and tested on the same type of training and test sets. The results are reported in Table 8.

In Fig. 5, it can be seen that the proposed model achieves comparable results on single domain dialogues and performs best on multi-domain dialogues on both datasets. On the other hand, when the baseline JOUST model and our model are trained on single domain or multi-domain dialogues and tested on the corresponding setting, we observe the same trends on both datasets as shown in Table 8. These observations suggest that the proposed model is effective in a multi-domain dialogue setting.

6.6. Visualization for GAT

To more visually demonstrate the proposed framework's effectiveness, we visualize the attention weights of some nodes (mainly from the ontology subgraph) in the dialogue state graph during the multi-hop reasoning phase. Note that we modified edge E_{St} that connects slot nodes and entity nodes to make it a fully connected edge. The new edge indicates that the slot node builds connections with all the token nodes in a dialogue history, one by one. By doing so, we can obtain a more intuitive view of GAT's fusion mechanism and verify the validity of the original edge design. This example is from the dialogue in Fig. 1 and is completed in an end-to-end setting. As shown in Fig. 6, the Y-axis represents a small part of the dialogue history, and the X-axis denotes a part of the nodes in the dialogue state graph related to the dialogue history.

Specifically, nodes directly related to the context get a deep colour at the first hop, such as the slot *departure* and *leaveAt*. Meanwhile, the nodes related to the context but with an indirect relation gradually acquire a deeper colour (i.e., higher attention) at the following hops. For instance, the token *hotel, park* and entity *24:45*, corresponding to the departure, destination and leaving time of the taxi, progressively attain a higher attention weight to the *Taxi* domain node at the second and third hops. These observations suggest that most of the tokens in a dialogue history are useless for the slot nodes in an ontology subgraph, and the ontology subgraph can capture helpful information through the multi-hop interactions with the dialogue context subgraph.

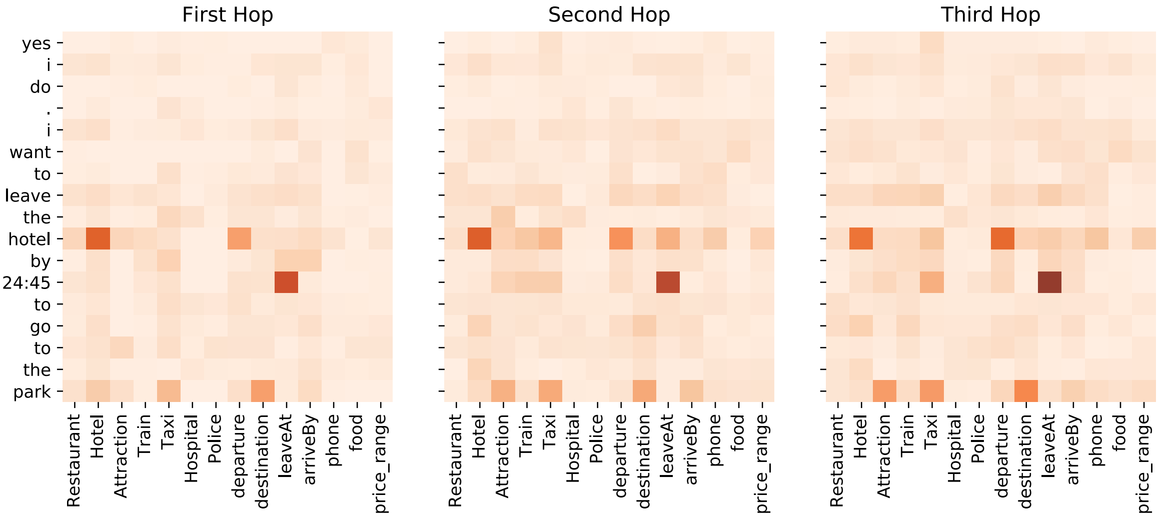


Fig. 6. Attention visualization of partial dialogue history related nodes in the dialogue state graph. The left panel shows the attention at the first hop, the middle panel displays the attention at the second hop, and the right panel presents the attention at the third hop. The darker colour appears in the figure, the higher score achieves.

Table 9
The performance of the proposed model (MAN-TOD) with a different number of hops of two memory networks on MultiWOZ 2.0 dataset. N means the number of hops.

N	Dialogue state MN			Dialogue act MN		
	Inform	Success	BLEU	Inform	Success	BLEU
1	81.7	73.2	18.1	82.1	73.3	18.2
2	82.3	73.6	18.3	82.4	73.8	18.3
3	82.5	73.8	18.4	82.5	73.8	18.4
4	82.1	73.5	18.2	82.2	73.6	18.2

It is noteworthy that the explainability of graph neural networks is important and meaningful. We adopt a post hoc approach that observes the effectiveness of each operation by removing it to explore the explainability of the proposed model to some extent. Thus, the experiments and analyses in Sections 6.1 and 6.2 can also be this type of work. We refer the reader to an interesting work [71] for an alternative method of quantification. For a comprehensive understanding of this direction, we recommend that the reader refer to the literature [72].

6.7. *Effect of different numbers of hops in the memory network*

In this section, we explore the impact of a different number of hops in the memory networks on the performance of the response generation. We verify this by setting different hops in dialogue state MN and dialogue act MN individually. Note that the number of hops in one MN is set to 3 when the other MN is set to a different number of hops. All the experiments are performed in an end-to-end setting on the MultiWOZ 2.0 dataset. The results are presented in Table 9.

It can be observed that the proposed model performs poorly when the memory network is set to be too small or too large on both tasks. Specifically, when the number of hops is set to 1, the model gets the worst score; as the number of hops increases, the performance improves. However, in regard to four, the model shows a decline. This indicates that the appropriate number of hops of a memory network has a significant impact on the performance of the model. On the other hand, there is a shallow variation in performance for dialogue act MN compared to dialogue state MN, which implies that the dialogue state reasoning task is more challenging.

6.8. *Case study*

To more qualitatively compare the performance of the proposed model with other baselines, we randomly select several generated responses from the test set of MultiWOZ 2.0, as shown in Table 10. The utterance spoken by **User** is the current dialogue history, and the utterance spoken by **System** is the ground-truth response. The remaining responses are generated by the baseline models and the proposed model.

As we can see in the first example, the ground-truth gives a negative response, indicating that none of the choices are eligible. However, both baseline models produce positive responses, which are the opposite to the actual results, and the proposed model gives the correct response. In the second example, we observe that all the generated responses are positive, the same as the ground-truth, while our generated response shows more of the desired information. In the third example, it can be noted that the user asks for details about an attraction. The ground-truth only provides information about the location and price. UniConv presents the attraction's type, location and phone number, and JOUST produces more information, such as the postcode and entrance fees. Compared with the responses generated by the baseline models, our generated response delivers the necessary information and gives a request utterance to promote the dialogue process. The above examples demonstrate that the proposed method outperforms the existing baseline models. Meanwhile, it can be noticed that all the generated responses are rigid and dull, which is distinct from reality. We will leave empathetic dialogue that consider affection and profiles as a future research direction.

7. Conclusion

This paper presents a graph attention network equipped with multi-task learning framework for task-oriented dialogue systems. In the framework, dialogue contexts and ontology schemas are adopted to construct two independent GATs for dialogue state parsing and act prediction. After that, to obtain fine-grained dialogue state representations and dialogue act representations, the updated ontology nodes are used to build two respective memory networks to filter out the irrelevant information. Eventually, the outputs of these two memory networks, the dialogue history representation, and the related KB, are concatenated as inputs

Table 10
Cases generated by baseline models and the proposed model in an end-to-end setting. To save space, we only present the current utterance spoken by user.

Dialogue Number: MUL0071, Turn Number: 6	
User	That sounds nice, yes please book a room for 3 guests staying for 5 nights starting on Sunday.
System	I'm sorry, those particular dates are filled, if you'd like i could try to make the trip shorter or maybe change days?
UniConv	booking was successful . reference number is [hotel_reference] . is there anything else i can help you with ?
JOUST	booking was successful . reference number is [hotel_reference] .
MAN-TOD+	i am sorry , i do not have any matches . would you like to try something else ?
Dialogue Number: PMUL4011, Turn Number: 10	
User	I'd like a taxi to pick me up at 23:15 at my hotel, then take me to the attraction. Please give me the car type and contact number.
System	Booking completed! Booked car type: black skoda, Contact number: 07804688108. Is there anything else I can help you with?
UniConv	i have booked a taxi for you . it will be a blue tesla and the contact number .
JOUST	i have booked your taxi . be expecting a [taxi_type] . please call [taxi_phone] should you need to reach them .
MAN-TOD+	booking compeleted ! booked car type black_skoda contact number 07804688108 . is there anything else i can help you with ?
Dialogue Number: PMUL1484, Turn Number: 4	
User	Thanks. Can you tell me anything about the castle galleries attraction?
System	Sure it is a lovely museum located in the centre, and the best part is it is free to get in!
UniConv	[attraction_name] is a [attraction_type] located in the [attraction_area] of town . their phone number is [attraction_phone] .
JOUST	it is located in the [attraction_area] . it s located at [attraction_address] , postcode [attraction_postcode] . phone number is [attraction_phone] . the entrance fee is free .
MAN-TOD+	yes , castle_galleries is located in the centre of town . the phone number is 01223307402 . it is free. is there anything else i can help you with ?

to initialize the decoder. To better handle KB entities, an entity detection task is proposed to enhance the soft gate in the training phase. The proposed model achieves superior performance in two dialogue settings on two open datasets. Moreover, an ablation study and further analysis have demonstrated that the components of the proposed model are valid. Finally, human evaluation and a case study qualitatively prove that the proposed model is practical. Based on this paper's analysis, our future work will focus on how to infuse emotional and personalized information into the dialogue procedure to facilitate more human-like dialogues.

CRediT authorship contribution statement

Meng Zhao: Investigation, Conceptualization, Methodology, Writing – review & editing, Writing – original draft. **Lifang Wang:** Supervision, Funding acquisition, Writing – review & editing. **Zejun Jiang:** Supervision, Methodology, Writing – review & editing. **Ronghan Li:** Validation, Writing – review & editing. **Xinyu Lu:** Writing – review & editing. **Zhongtian Hu:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This research is supported by the Open Project Foundation of Henan key Laboratory for Big Data Processing & Analytics of Electronic Commerce, China [2022-KF-2].

References

[1] A. Bordes, Y. Boureau, J. Weston, Learning end-to-end goal-oriented dialog, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[2] H.-y. Shum, X.-d. He, D. Li, From Eliza to XiaoIce: challenges and opportunities with social chatbots, Front. Inf. Technol. Electron. Eng. 1 (19) (2018) 10–26.

[3] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gašić, L.M. Rojas-Barahona, P.-H. Su, S. Ultes, S. Young, A network-based end-to-end trainable task-oriented dialogue system, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 438–449.

[4] Z. Zhang, R. Takanobu, Q. Zhu, M. Huang, X. Zhu, Recent advances and challenges in task-oriented dialog systems, Sci. China Technol. Sci. 63 (10) (2020) 2011–2027.

[5] W. Liang, Y. Tian, C. Chen, Z. Yu, MOSS: End-to-end dialog system framework with modular supervision, in: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, (05) 2020, pp. 8327–8335.

[6] C. Wu, R. Socher, C. Xiong, Global-to-local memory pointer networks for task-oriented dialogue, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019, OpenReview.net, 2019.

[7] L. Qin, X. Xu, W. Che, Y. Zhang, T. Liu, Dynamic fusion network for multi-domain end-to-end task-oriented dialog, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 6344–6354.

[8] Q. Liu, G. Bai, S. He, C. Liu, K. Liu, J. Zhao, Heterogeneous relational graph neural networks with adaptive objective for end-to-end task-oriented dialogue, Knowl.-Based Syst. (2021) 107186.

[9] Y. Gou, Y. Lei, L. Liu, Y. Dai, C. Shen, Contextualize knowledge bases with transformer for end-to-end task-oriented dialogue systems, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2021, pp. 4300–4310, Online and Punta Cana, Dominican Republic.

[10] W. Chen, J. Chen, P. Qin, X. Yan, W.Y. Wang, Semantically conditioned dialog response generation via hierarchical disentangled self-attention, in: A. Korhonen, D.R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 3696–3709.

[11] K. Wang, J. Tian, R. Wang, X. Quan, J. Yu, Multi-domain dialogue acts and response co-generation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 7125–7134.

[12] Y. Zhang, Z. Ou, Z. Yu, Task-oriented dialog systems that consider multiple appropriate responses under the same context, in: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, (05) 2020, pp. 9604–9611.

[13] S. Gao, Y. Zhang, Z. Ou, Z. Yu, Paraphrase augmented task-oriented dialog generation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 639–649.

[14] H. Le, D. Sahoo, C. Liu, N.F. Chen, S.C.H. Hoi, UniConv: A unified conversational neural architecture for multi-domain task-oriented dialogues, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 1860–1877.

[15] A. Madotto, S. Cahyawijaya, G.I. Winata, Y. Xu, Z. Liu, Z. Lin, P. Fung, Learning knowledge bases with parameters for task-oriented dialogue systems, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, in: Findings of ACL. EMNLP 2020, Association for Computational Linguistics, 2020, pp. 2372–2394.

[16] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018.

[17] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2021) 4–24.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008.

[19] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, IJCAI/AAAI Press, 2016, pp. 2873–2879.

[20] D. Dong, H. Wu, W. He, D. Yu, H. Wang, Multi-task learning for multiple language translation, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, The Association for Computer Linguistics, 2015, pp. 1723–1732.

[21] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[22] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, ACM, 2019, pp. 2022–2032.

[23] J. Weston, S. Chopra, A. Bordes, Memory networks, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[24] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, End-to-end memory networks, in: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 2440–2448.

[25] V. Sanh, T. Wolf, S. Ruder, A hierarchical multi-task approach for learning embeddings from semantic tasks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, AAAI Press, 2019, pp. 6949–6956.

[26] Q. Ren, X. Cheng, S. Su, Multi-task learning with generative adversarial training for multi-passage machine reading comprehension, (05) 2020, pp. 8705–8712.

[27] H. Yin, S. Yang, X. Song, W. Liu, J. Li, Deep fusion of multimodal features for social media retweet time prediction, World Wide Web 24 (4) (2021) 1027–1044.

[28] S. Yang, S. Verma, B. Cai, J. Jiang, K. Yu, F. Chen, S. Yu, Variational co-embedding learning for attributed network clustering, 2021, CoRR, arXiv:2104.07295.

[29] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, A. Mian, Bi-CLKT: Bi-graph contrastive learning based knowledge tracing, Knowl.-Based Syst. 241 (2022) 108274.

[30] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, M. Gašić, MultiWOZ - A large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 5016–5026.

[31] M. Eric, R. Goel, S. Paul, A. Sethi, S. Agarwal, S. Gao, A. Kumar, A. Goyal, P. Ku, D. Hakkani-Tur, MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines, in: Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 422–428.

[32] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, P. Khaitan, Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 8689–8696.

[33] W. Lei, X. Jin, M. Kan, Z. Ren, X. He, D. Yin, Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, Association for Computational Linguistics, 2018, pp. 1437–1447.

[34] S. Mehri, T. Srinivasan, M. Eskénazi, Structured fusion networks for dialog, in: Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019, Association for Computational Linguistics, 2019, pp. 165–177.

[35] S. Peng, X. Huang, Z. Lin, F. Ji, H. Chen, Y. Zhang, Teacher-student framework enhanced multi-domain dialogue generation, 2019, CoRR, arXiv:1908.07137.

[36] Y. Zhang, Z. Ou, M. Hu, J. Feng, A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 9207–9219.

[37] B. Santra, P. Anusha, P. Goyal, Hierarchical transformer for task oriented dialog systems, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, Association for Computational Linguistics, 2021, pp. 5649–5658.

[38] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019.

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67.

[40] D. Ham, J. Lee, Y. Jang, K. Kim, End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 583–592.

[41] E. Hosseini-Asl, B. McCann, C. Wu, S. Yavuz, R. Socher, A simple language model for task-oriented dialogue, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual, 2020.

[42] Y. Yang, Y. Li, X. Quan, UBAR: towards fully end-to-end task-oriented dialog system with GPT-2, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 14230–14238.

[43] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE Trans. Knowl. Data Eng. (2021) 1.

[44] C. Xu, Z. Guan, W. Zhao, H. Wu, Y. Niu, B. Ling, Adversarial incomplete multi-view clustering, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 3933–3939.

[45] C. Xu, H. Liu, Z. Guan, X. Wu, J. Tan, B. Ling, Adversarial incomplete multiview subspace clustering networks, IEEE Trans. Cybern. 52 (10) (2022) 10490–10503.

[46] C. Xu, Z. Guan, W. Zhao, Q. Wu, M. Yan, L. Chen, Q. Miao, Recommendation by users' multimodal preferences for smart city applications, IEEE Trans. Ind. Inform. 17 (6) (2021) 4197–4205.

[47] L. Wang, R. Li, Y. Wu, Z. Jiang, A multiturn complementary generative framework for conversational emotion recognition, Int. J. Intell. Syst. 37 (9) (2022) 5643–5671.

[48] M. Crawshaw, Multi-task learning with deep neural networks: A survey, 2020, CoRR, arXiv:2009.09796.

[49] C. Xu, Z. Guan, W. Zhao, Y. Niu, Q. Wang, Z. Wang, Deep multi-view concept learning, in: J. Lang (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 2898–2904.

[50] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association

for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.

[51] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, CoRR, arXiv:1907.11692.

[52] S. Bao, H. He, F. Wang, H. Wu, H. Wang, PLATO: pre-trained dialogue generation model with discrete latent variable, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 85–96.

[53] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018.

[54] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks? in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019, OpenReview.net, 2019.

[55] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80.

[56] D. Busbridge, D. Sherburn, P. Cavallo, N.Y. Hammerla, Relational graph attention networks, 2019, CoRR, arXiv:1904.05811.

[57] J. Zeng, T. Liu, W. Jia, J. Zhou, Fine-grained question-answer sentiment classification with hierarchical graph attention network, Neurocomputing 457 (2021) 214–224.

[58] Z. Liang, J. Du, Y. Shao, H. Ji, Gated graph neural attention networks for abstractive summarization, Neurocomputing 431 (2021) 128–136.

[59] N. Chairatanakul, X. Liu, T. Murata, PGRA: Projected graph relation-feature attention network for heterogeneous information network embedding, Inform. Sci. 570 (2021) 769–794.

[60] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, Z. Guan, JKT: A joint graph convolutional network based deep knowledge tracing, Inform. Sci. 580 (2021) 510–523.

[61] B.-H. Kim, J.C. Ye, Understanding graph isomorphism network for rs-fMRI functional connectivity analysis, Front. Neurosci. (2020) 630.

[62] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, CoRR, arXiv:1412.3555.

[63] L. Qin, W. Che, M. Ni, Y. Li, T. Liu, Knowing where to leverage: Context-aware graph convolutional network with an adaptive fusion layer for contextual spoken language understanding, IEEE ACM Trans. Audio Speech Lang. Process. 29 (2021) 1280–1289.

[64] A. Giusti, D.C. Ciresan, J. Masci, L.M. Gambardella, J. Schmidhuber, Fast image scanning with deep max-pooling convolutional neural networks, in: IEEE International Conference on Image Processing, ICIP 2013, Melbourne, Australia, September 15-18, 2013, IEEE, 2013, pp. 4034–4038.

[65] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, S. Savarese, Which tasks should be learned together in multi-task learning? in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 9120–9132.

[66] C. Xu, W. Zhao, J. Zhao, Z. Guan, X. Song, J. Li, Uncertainty-aware multi-view deep learning for internet of things applications, IEEE Trans. Ind. Inform. (2022).

[67] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[68] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, HuggingFace's transformers: State-of-the-art natural language processing, 2019, CoRR, arXiv:1910.03771.

[69] B. Tseng, Y. Dai, F. Kreyssig, B. Byrne, Transferable dialogue systems and user simulators, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, Association for Computational Linguistics, 2021, pp. 152–166.

[70] R. Li, L. Wang, S. Wang, Z. Jiang, Asynchronous multi-grained graph network for interpretable multi-hop reading comprehension, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 3857–3863.

[71] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, Q. Wang, Interpretable and efficient heterogeneous graph convolutional network, IEEE Trans. Knowl. Data Eng. (2021).

[72] H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in graph neural networks: A taxonomic survey, IEEE Trans. Pattern Anal. Mach. Intell. (2022).