CGIM: A Cycle Guided Interactive Learning Model for Consistency Identification in Task-oriented Dialogue

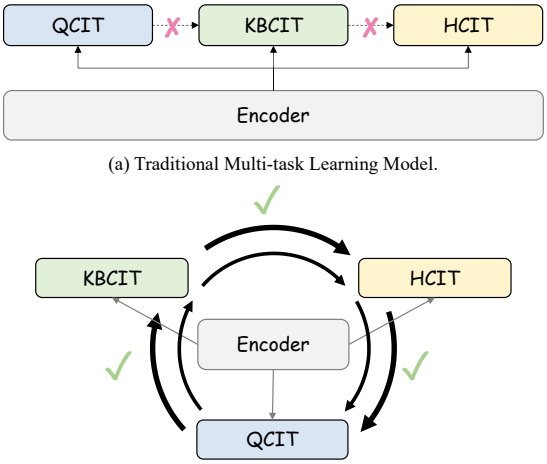
Libo Qin[1], **Qiguang Chen**[1], **Tianbao Xie**[1], **Qian Liu**[2],
Shijue Huang[1], **Wanxiang Che**[1]*, **Zhou Yu**[3]
[1]Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
[2] Beihang University, Beijing, China
[3]Columbia University
{lbqin,tianbaoxie,car}@ir.hit.edu.cn; qian.liu@buaa.edu.cn; zy2461@columbia.edu

Abstract

Consistency identification in task-oriented dialog (CI-ToD) usually consists of three subtasks, aiming to identify inconsistency between current system response and current user response, dialog history and the corresponding knowledge base. This work aims to solve CI-ToD task by introducing an explicit interaction paradigm, **C**ycle **G**uided **I**nteractive learning **M**odel (CGIM), which achieves to make information exchange explicitly from all the three tasks. Specifically, CGIM relies on two core insights, referred to as *guided multi-head attention module* and *cycle interactive mechanism*, that collaborate from each other. On the one hand, each two tasks are linked with the *guided multi-head attention module*, aiming to explicitly model the interaction across two related tasks. On the other hand, we further introduce *cycle interactive mechanism* that focuses on facilitating model to exchange information among the three correlated sub-tasks via a cycle interaction manner. Experimental results on CI-ToD benchmark show that our model achieves the state-of-the-art performance, pushing the overall score to 56.3% (5.0% point absolute improvement). In addition, we find that CGIM is robust to the initial task flow order.

1 Introduction

Consistency identification task in dialogue has the potential benefits of preventing inconsistent response generation (Welleck et al., 2019), which has attracted increasing attention. Recent years have witnessed two promising research directions in *consistency identification*. The first focuses on *consistency identification* in open-domain dialogue (Zhang et al., 2018; Zheng et al., 2019). The second direction consider *consistency identification* in task-oriented dialogue system (CI-ToD) (Qin et al., 2021b). In this work, we focus on the latter. Recently, Qin et al. (2021b) introduces a benchmark (CI-ToD) for consistency identification in task-oriented dialogue to facilitate the relevant research. CI-ToD introduces three subtasks including: (1) *dialogue **h**istory **c**onsistency **i**dentification **t**ask* (HCIT) to judge whether generated response is inconsistent with dialogue history; (2) *user **q**uery **c**onsistency **i**dentification **t**ask* (QCIT) to detect the consistency status between query and system response, and (3) ***k**nowledge **b**ase **c**onsistency **i**dentification **t**ask* (KBCIT) to determine system response is contradicted with the corresponding knowledge base.



(a) Traditional Multi-task Learning Model.

(b) Cycle Guided Interative Learning Model (CGIM).

Figure 1: (a) Traditional multi-task learning models learn mutual information across tasks via an implicit interaction manner vs. (b) Our proposed cycle interactive learning model explicitly consider cross-impact across three tasks via an explicit interaction manner.

Intuitively, the three tasks are closely related, indicating information of one task can be utilized in other related tasks. For example, if we first complete HCIT, the result of HCIT can assist the QCIT to determine whether the system response is contradicted with the user query, since the dialogue history and the user query tend to share similar topic (Chen et al., 2020). Similarly, KBCIT can also provide additional information for helping

*Email corresponding.

Proceedings of the 29th International Conference on Computational Linguistics, pages 461–470
October 12–17, 2022.

HCIT, because dialogue history can be regarded as an unstructured knowledge description for the corresponding KB. Above observations suggest that it is imperative to take cross-impact across three tasks into account. To this end, Qin et al. (2021b) explore a simple multi-task framework that consists of a shared encoder and different task decoders to jointly consider correlation, which is shown in Figure 1(a). Though achieving superior performance compared with single models, their approaches solely rely on shared latent representations to model the interaction in an implicit interaction manner, which limits their performance. Therefore, it is promising to consider an explicit joint modeling approach for CI-ToD.

While the idea seems promising, achieving this objective is challenging, since we need to jointly model the three sub-tasks simultaneously rather than simple two tasks setting. Recent work have shown explicit joint modeling is superior to the implicit joint modeling (Goo et al., 2018; Qin et al., 2021a). Nevertheless, their work still limits to modeling the relationship between two tasks, it remains clear if the explicit modeling paradigm can be applied in three tasks. To this end, as shown in Figure 1(b), we propose a novel **C**ycle **G**uided **I**nteractive learning **M**odel (CGIM) for CI-ToD, which achieves to perform the three sub-tasks jointly and interactively in an explicit interaction paradigm. Specifically, CGIM first consists of a ***g**uided **m**ulti-head **a**ttention module* (GMA) that is used for two related tasks, which aims to explicitly utilize information from another task. With the help of GMA, each task can not only rely on its own task information but also performed with the guidance of the corresponding correlated task explicitly. Furthermore, since GMA can only enable the single information flow from one task to another task, we further propose a novel *cycle interactive mechanism* to facilitate information flow across the three tasks in an cycle interaction fashion. With the use of cycle interactive mechanism, CGIM can be stacked to form a hierarchy, which can gradually capture interaction information and better transfer knowledge.

We conduct experiments on CI-ToD benchmark and results show that CGIM achieves the best performance, outperforming previous state-of-the-art methods by at least 5.0% (overall accuracy). Besides, extensive analysis further demonstrate the superior and robustness of our approach.

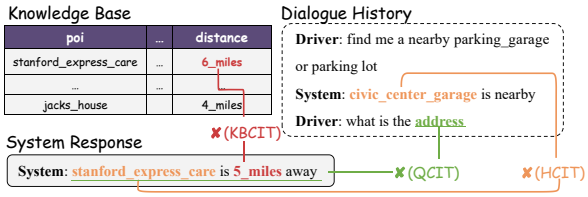


Figure 2: Example illustration in CI-ToD. Different colors denote inconsistency type of different tasks.

Main contributions are summarized as follows:

- To the best of our knowledge, we make the first attempt to explore an explicit interaction model for CI-ToD.
- We introduce a novel cycle interactive learning model for CI-ToD, which achieves establishing a triple-interaction across the three tasks simultaneously.
- Results on CI-ToD benchmark show that CGIM achieves state-of-the-art performance. Besides, we observe that CGIM is robust to the initial task flow order.

All codes in this work will be publicly available at https://github.com/LightChen233/CGIM.

2 Background

To make the paper self-complete, we present the definition of the task that follows Qin et al. (2021b) in this section.

2.1 Task Definition

Given a task-oriented dialogue between a user (u) and a system (s), the dialogue history is defined as $H = \{(u_1, s_1),(u_2, s_2), \ldots,(u_{n-1}, s_{n-1})\}$, the corresponding knowledge base KB is B, the user query is denoted by u_n and the system response is denoted by s_n.

Formally, the consistency identification in task-oriented dialogue contains three tasks: the dialogue **h**istory **c**onsistency **i**dentification **t**ask (HCIT), the user **q**uery **c**onsistency **i**dentification **t**ask (QCIT), and the **k**nowledge **b**ase **c**onsistency **i**dentification **t**ask (KBCIT) to judge whether system response is contradicted with the corresponding dialogue history, user query, and KB, respectively, which are defined as:

$$(y^{\mathrm{Q}}, y^{\mathrm{H}}, y^{\mathrm{B}}) = f_\theta([H, B, u_n], s_n), \quad (1)$$

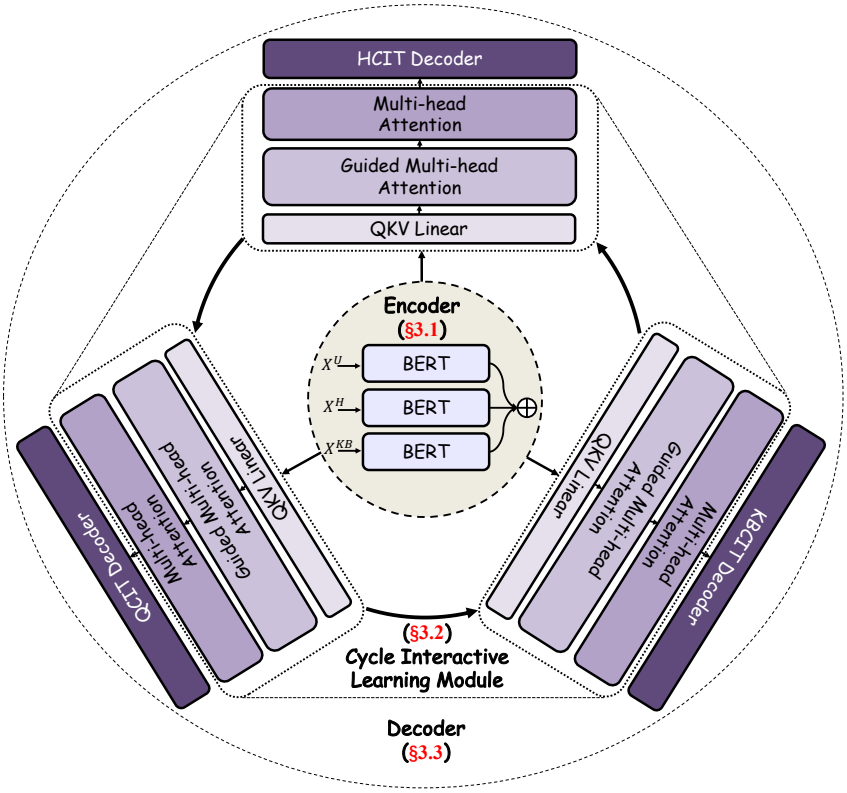


Figure 3: The illustration of the proposed cycle interactive learning model (CGIM), which consists of three components: encoder, cycle interactive learning module and decoder.

where f_θ denotes the parameters of model; $y^{\mathrm{Q}}, y^{\mathrm{H}}, y^{\mathrm{B}}$ represents the probabilities of inconsistent system response in QCIT, HCIT and KBCIT, respectively.

2.2 Example Illustration

To understand HCIT, QCIT, and KBCIT intutively, we provide some example cases, which are shown in the following:

QCIT QCIT aims at detecting the inconsistency between dialogue system response and current user query. As shown in Figure 2, user is intended to ask for the *address*. However, the system response provide the answer about *distence* to stanford_express_care, which results in inconsistency with user query.

HCIT HCIT aims at detecting the inconsistency between system response and dialogue history except the current query. Figure 2 shows the inconsistent dialogue, where the previous dialogue history talked about *civic_center_garage* and the user did not change the topic. However, the system responded by talking about *stanford_express_care*, which is contradicted with the dialogue history.

KBCIT KBCIT aims at detecting the inconsistency between dialogue system response and corresponding KB. As shown in Figure 2, *stanford_express_care* is located *6_miles* according to the corresponding KB. However, the system indicates that the *distance* to *stanford_express_care* is *5_miles*, which is inconsistent with the information provided in the KB.

3 Approach

The architecture of the cycle guided interactive learning model (CGIM) is depicted in Figure 3. It mainly consists of three components: three task-specific encoders for obtaining encoding representation for each task (§3.1); a cycle interactive learning module for explicitly establishing the interaction across the three tasks (§3.2); three separate decoders for HCIT, QCIT and KBCIT (§3.3), respectively. In the following sections, the details of our framework are given.

3.1 Encoder

Following Qin et al. (2021b), we employ the pre-trained model (i.e., BERT) (Devlin et al., 2019) as the encoder and use delimiter tokens `[SOK]`,

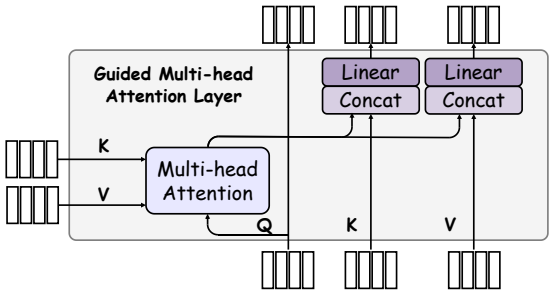


Figure 4: Guided Multi-head Attention Layer.

[EOK], [USR], [SYS] to capture the role feature of KB, user and system response.

Query Representation To consider the system response, we concatenate u_n and the last system response s_n to obtain the query encoding representation. Therefore, the input can be represented as X^U = ([CLS], u_n, [SEP], s_n, [SEP]), where [CLS] and [SEP] are special symbol, and BERT reads it to produce the representation:

$$\mathbf{h}^{\mathrm{Q}} = \mathrm{BERT}(X^{\mathrm{U}}), \tag{2}$$

where the last layer's hidden representation $\mathbf{h}^{\mathrm{Q}}$ of the [CLS] token is considered as the query representation.

Dialogue History Representation Similarly, we concatenate the dialog history H and the system response s_n as X^{H} = ([CLS], $\hat{H}$, [SEP], s_n, [SEP]), which is used for acquiring dialogue history representation $\mathbf{h}^{\mathrm{H}}$:

$$\mathbf{h}^{\mathrm{H}} = \mathrm{BERT}(X^{\mathrm{H}}), \tag{3}$$

where $\hat{H}$ is [USR] u_1 [SYS] $s_1 \ldots$ [USR] u_n.

Knowledge Base Representation For KB representation, we first linearize the KB and then concatenate the linearized KB and system response to obtain X^{B}= ([CLS], $\hat{B}$, [SEP], s_n, [SEP]). Feeding it into BERT, we obtain the knowledge base representation $\mathbf{h}^{\mathrm{B}}$:

$$\mathbf{h}^{\mathrm{B}} = \mathrm{BERT}(X^{\mathrm{B}}), \tag{4}$$

where $\hat{B}$ is [SOK] B [EOK].

3.2 Cycle Guided Interactive Learning Module

Traditional multi-task learning only depend on a set of shared parameters to implicitly consider correlation across different correlated tasks. In contrast, we present a cycle guided interactive learning model to explicit model the interaction, which consists of two parts: the ***g**uided **m**ulti-head **a**ttention module* (GMA) and the ***c**ycle **i**nteraction **m**echanism* (CIM).

3.2.1 Guided Multi-head Attention Module

GMA mainly consists of a guided multi-head attention layer and self multi-head attention layer, achieving to explicitly model interaction across related tasks.

Guided Multi-head Attention Layer. First, given $\mathbf{h}^{\mathrm{Q}}$, $\mathbf{h}^{\mathrm{H}}$ and $\mathbf{h}^{\mathrm{B}}$, we first directly perform a concatenation operation upon them and adopt different projection linear layers to obtain different updated representations for QCIT, HCIT and KBCIT, which are denoted as:

$$\mathbf{H} = \mathrm{Concat}(\mathbf{h}^{\mathrm{Q}}, \mathbf{h}^{\mathrm{H}}, \mathbf{h}^{\mathrm{B}}), \tag{5}$$

$$\mathbf{H}^{\mathrm{Q}}, \mathbf{H}^{\mathrm{H}}, \mathbf{H}^{\mathrm{B}} = \mathbf{W}^{\mathrm{Q}}\mathbf{H}, \mathbf{W}^{\mathrm{H}}\mathbf{H}, \mathbf{W}^{\mathrm{B}}\mathbf{H}, \tag{6}$$

where $\mathbf{H} \in \mathbb{R}^{3\times d}$ (d represents the encoding dimension); Concat is concatenation operation; $\mathbf{W}^{\mathrm{H}}, \mathbf{W}^{\mathrm{Q}}, \mathbf{W}^{\mathrm{B}}$ are the trainable matrix.

Then, to obtain updated QCIT representations with the guidance of HCIT explicitly, it is necessary to align query with its closely related dialogue history information. To be more specific, as shown in Figure 4, we first employ QKV Linear to map the dialog history and query representations $\mathbf{H}^{\mathrm{H}}$ and $\mathbf{H}^{\mathrm{Q}}$ to query ($\mathbf{Q}^{\mathrm{H}}$, $\mathbf{Q}^{\mathrm{Q}}$), keys ($\mathbf{K}^{\mathrm{H}}$, $\mathbf{K}^{\mathrm{Q}}$) and values ($\mathbf{V}^{\mathrm{H}}$, $\mathbf{V}^{\mathrm{Q}}$) matrices. We then treat $\mathbf{Q}^{\mathrm{Q}}$ as queries, $\mathbf{K}^{\mathrm{H}}$ as keys and $\mathbf{V}^{\mathrm{H}}$ as values to obtain the updated representation with explicitly considering the information from the HCIT task. The output is a weighted sum of values:

$$\hat{\mathbf{H}} = \mathrm{MultiHead}(\mathbf{Q}^{\mathrm{Q}}, \mathbf{K}^{\mathrm{H}}, \mathbf{V}^{\mathrm{H}}), \tag{7}$$

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{QK}}{\sqrt{d_k}}\right)\mathbf{V}, \tag{8}$$

$$\mathrm{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Concat}(\mathbf{H}_i, \ldots, \mathbf{H}_h)\mathbf{W}^0, \tag{9}$$

$$\text{where } \mathbf{H}_i = \mathrm{Attention}(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V), \tag{10}$$

where $\hat{\mathbf{H}}$ can be seen as the updated query information with the guidance of dialogue history information; $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$ denote the query, key, and value respectively. d_k is the dimension of the key; $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ denote the head projection matrix, respectively; h is the head number.

Then, we concatenate the updated representation $\mathbf{H}$ with the original $\mathbf{K}^{\mathrm{Q}}$ and $\mathbf{V}^{\mathrm{Q}}$ to construct new

keys $\hat{\mathbf{K}}^{\mathbf{Q}}$ and values $\hat{\mathbf{V}}^{\mathbf{Q}}$, calculating as:

$$\hat{\mathbf{K}}^{\mathbf{Q}} = \text{Linear}(\text{Concat}(\mathbf{K}^{\mathbf{Q}}, \hat{\mathbf{H}})), \quad (11)$$

$$\hat{\mathbf{V}}^{\mathbf{Q}} = \text{Linear}(\text{Concat}(\mathbf{V}^{\mathbf{Q}}, \hat{\mathbf{H}})). \quad (12)$$

Self Multi-head Attention Layer. Given the obtained updated keys, updated values, and queries, we further introduce a self multi-head attention layer to enhance the interaction information and obtain the final query guided query representation:

$$\widetilde{\mathbf{H}}^{\mathrm{Q}} = \text{MultiHead}(\mathbf{Q}^{\mathrm{Q}}, \hat{\mathbf{K}}^{\mathrm{Q}}, \hat{\mathbf{V}}^{\mathrm{Q}}), \quad (13)$$

where $\widetilde{\mathbf{H}}^{\mathrm{Q}}$ denotes the updated representations.

Similiar to transformer, we also employ a residual connection (He et al., 2016), layer normalization (Ba et al., 2016) and a fully connected feed-forward network.

3.2.2 Cycle Interactive Mechanism

With the help of the GMA, single information flow can be established. Formally, given features representation $\mathbf{H}^{\mathrm{Q}}$ and $\mathbf{H}^{\mathrm{H}}$, GMA aims to output the attended features $\widetilde{\mathbf{H}}^{\mathrm{Q}}$ for $\mathbf{H}^{\mathrm{Q}}$ guided by $\mathbf{H}^{\mathrm{H}}$, which can be formulated as:

$$\widetilde{\mathbf{H}}_{\mathbf{1}}^{\mathrm{Q}} = \text{GMA}_{\mathrm{H}\to\mathrm{Q}}(\mathbf{H}_0^{\mathrm{H}}, \mathbf{H}_0^{\mathrm{Q}}), \quad (14)$$

where $\mathbf{H}_0^{\mathrm{H}}$ and $\mathbf{H}_0^{\mathrm{Q}}$ are initialized as $\mathbf{H}^{\mathrm{H}}$ and $\mathbf{H}^{\mathrm{H}}$, respectively.

Similarly, the query guided dialogue history representation and dialogue history guided knowledge base representation can be obtained in the same manner, which are shown as:

$$\widetilde{\mathbf{H}}_{\mathbf{1}}^{\mathrm{B}} = \text{GMA}_{\mathrm{Q}\to\mathrm{B}}(\mathbf{H}_0^{\mathrm{Q}}, \mathbf{H}_0^{\mathrm{B}}), \quad (15)$$

$$\widetilde{\mathbf{H}}_{\mathbf{1}}^{\mathrm{H}} = \text{GMA}_{\mathrm{B}\to\mathrm{H}}(\mathbf{H}_0^{\mathrm{B}}, \mathbf{H}_0^{\mathrm{H}}). \quad (16)$$

To enable the shared knowledge flowed across the three subtasks, we further introduce a cycle interaction mechanism (CIM) with multiple layers to gradually and iteratively control knowledge transfer, which can be formulated as:

$$\widetilde{\mathbf{H}}^{\mathrm{Q}}, \widetilde{\mathbf{H}}^{\mathrm{H}}, \widetilde{\mathbf{H}}^{\mathrm{B}} = \text{CIM}_{(\mathrm{H}\to\mathrm{Q}\to\mathrm{B}\to\mathrm{Q})}(\mathbf{H}^{\mathrm{Q}}, \mathbf{H}^{\mathrm{H}}, \mathbf{H}^{\mathrm{B}}), \quad (17)$$

After stacking L layer, we obtain a final updated feature representation: $\widetilde{\mathbf{H}}_{\mathrm{L}}^{\mathrm{B}}$, $\widetilde{\mathbf{H}}_{\mathrm{L}}^{\mathrm{H}}$ and $\widetilde{\mathbf{H}}_{\mathrm{L}}^{\mathrm{Q}}$.

3.3 Decoder

Given the final updated representations $\widetilde{\mathbf{H}}_{\mathrm{L}}^{\mathrm{B}}$, $\widetilde{\mathbf{H}}_{\mathrm{L}}^{\mathrm{H}}$ and $\widetilde{\mathbf{H}}_{\mathrm{L}}^{\mathrm{Q}}$ for each task, we directly flatten them into a single vector $\mathbf{d}^{B}$, $\mathbf{d}^{\mathrm{H}}$ and $\mathbf{d}^{\mathrm{Q}}$, which are fed into separate decoders to perform QCIT, HCIT and KBCIT, which can be denoted as:

$$\mathbf{y}^{\mathrm{H}} = \text{softmax}(\mathbf{W}^{\mathrm{H}}\mathbf{d}^{\mathrm{H}} + \mathbf{b}_{\mathrm{H}}), \quad (18)$$

$$\mathbf{y}^{\mathrm{Q}} = \text{softmax}(\mathbf{W}^{Q}\mathbf{d}^{Q} + \mathbf{b}_{Q}), \quad (19)$$

$$\mathbf{y}^{\mathrm{B}} = \text{softmax}(\mathbf{W}^{B}\mathbf{d}^{B} + \mathbf{b}_{B}), \quad (20)$$

where $\mathbf{y}^{\mathrm{H}}$, $\mathbf{y}^{\mathrm{Q}}$ and $\mathbf{y}^{\mathrm{B}}$ are the predicted distribution result for three tasks, respectively; $\mathbf{W}^{\mathrm{H}}$, $\mathbf{W}^{\mathrm{Q}}$ and $\mathbf{W}^{\mathrm{B}}$ are learnable transformation matrices; $\mathbf{b}_{\mathrm{Q}}$, $\mathbf{b}_{\mathrm{H}}$ and $\mathbf{b}_{\mathrm{B}}$ are learnable bias vectors.

3.4 Joint Training

The training objective of each task is the binary cross-entropy loss. Specifically, the objective for QCIT is:

$$\mathcal{L}_Q = -\sum^{T} \left(\hat{\mathbf{y}}^{\mathrm{Q}} \log\left(\mathbf{y}^{\mathrm{Q}}\right)\right), \quad (21)$$

where $\hat{\mathbf{y}}$ is gold label and T is the training data size.

Similar, $\mathcal{L}_H$ and $\mathcal{L}_B$ can be obtained in a similar manner. Following Bai et al. (2021) and Bao et al. (2021), the final joint loss function is as:

$$\mathcal{L}_\theta = \alpha_Q\mathcal{L}_Q + \alpha_H\mathcal{L}_H + \alpha_B\mathcal{L}_B, \quad (22)$$

where α_Q, α_H and α_B are hyper-parameter[1]

4 Experiments

4.1 Experimental Settings

To evaluate the effectiveness of CGIM, we conduct experiments on the CI-ToD benchmark (Qin et al., 2021b). Specifically, CI-ToD consists of 2,553 dialogues for training, 319 dialogues for validation, and 318 dialogues for testing.

In our experimental setting, we adopt `BERT-base` and the dimension of all hidden units is 768. The batch size we use is selected from $\{4, 8, 16\}$ and learning rate is selected from $\{1e^{-5}, 2e^{-5}, 5e^{-5}\}$. We use AdamW (Loshchilov and Hutter, 2019) to optimize the parameters in our model. We select all hyper-parameters from the validation set. All experiments are conducted at Tesla P100 and Tesla V100.

4.2 Baselines

Following Qin et al. (2021b), we compare our model with the following state-of-the-art multi-task

[1]In our experiment, we set them as 1.

Model	QI F1	HI F1	KBI F1	Overall Acc
BART-separate (Lewis et al., 2020)	0.695	0.496	0.721	0.450
BERT-multi-task (Devlin et al., 2019)	0.691	0.555	0.740	0.500
RoBERTa-multi-task (Liu et al., 2019)	0.715	0.472	0.715	0.500
XLNet-multi-task (Yang et al., 2020)	0.725	0.487	0.736	0.509
Longformer-multi-task (Beltagy et al., 2020)	0.717	0.500	0.710	0.497
BART-multi-task (Lewis et al., 2020)	0.744	0.510	0.761	0.513
CGIM	**0.764**	**0.567**	**0.772**	**0.563**

Table 1: Main results. The bolded number indicates the best performance. All baselines results are taken from Qin et al. (2021b).

learning models based on the strong pre-trained models:

(1) `BERT` (Devlin et al., 2019): the model pre-trains bidirectional representations from a large-scale text corpus; (2) `RoBERTa` (Liu et al., 2019): the model improves the training procedure of BERT to make it perform better; (3) `XLNet` (Yang et al., 2020): the model combines the advantage of autoregressive and autoencoding approaches by performing a permutation language objective; (4) `Longformer` (Beltagy et al., 2020): the model employs an attention pattern that combines local and global information while also scaling linearly with the sequence length, making it easy to process long documents; (5) `BART` (Lewis et al., 2020): the model uses a pre-training approach to map corrupted documents to the original document, which works well on both various generation tasks and understanding tasks.

We refer to the model with multi-task learning for three tasks as `model-multi-task`. In addition, we also compare CGIM with the state-of-the-art separate model for each task, which is referred as `model-separate`.

4.3 Main Results

Following Qin et al. (2021b), we adopt query inconsistency (QI) F1 scores, dialogue history inconsistency (HI) F1 scores, knowledge base inconsistency (KBI) F1 scores to evaluate QCIT, HCIT, and KBCIT respectively. Besides, we also use overall accuracy, a strict metric that requires all tasks are predicted correctly.

From the results shown in Table 1. We have the following observations:

(1) `CGIM` yields better performance compared with `BART-separate` on all metrics, which verifies that QCIT, HCIT and KBCIT tasks are correlated where joint model can be benefited from capturing shared knowledge across tasks, supporting our motivation;

(2) `CGIM` achieves the best performance on three tasks compared with all baselines. Compared with `BART-multi-task`, our framework obtains 2.0%, 5.7% and 1.1% improvements on three tasks, respectively. This indicates that the proposed explicit interaction paradigm is better than the implicit interaction paradigm that is insufficient to grasp knowledge transfer, which is consistent to the observation on other explicit joint modeling work on two tasks (Goo et al., 2018; Qin et al., 2021a);

(3) `CGIM` attains the best results on Overall Acc. and beats the best model `BART-multi-task` by a large margin of 5.0%. This suggests that all three tasks are highly correlated and explicit modeling mechanism can help to improve the whole dialogue understanding ability than the implicit modeling. It is worth noticing that the backbone of `CGIM` is BERT and it still outperforms `BART-multi-task` by a large margin, which further verifies the effectiveness of explicit modeling paradigm.

4.4 Analysis

This section answer the following research questions to understand CGIM in depth:

(1) Does each guided multi-head attention (GMA) module improve performance?

(2) Does a deeper layer of guided multi-head attention module bring a better performance?

(3) Is CGIM robust to the initial task flow order?

(4) Does explicit interaction modeling gain the performance improvement rather than the involved parameters?

Model	QI F1	HI F1	KBI F1	Overall Acc
CGIM	**0.764**	**0.567**	**0.772**	**0.563**
w/o QCIT → KBCIT	0.712	0.539	0.749	0.512
w/o KBCIT → HCIT	0.731	0.506	0.752	0.494
w/o HCIT → QCIT	0.710	0.507	0.764	0.521
w/MLP	0.725	0.515	0.686	0.507

Table 2: Ablation Study. The bolded number indicates the best performance in the first block.

Model	QI F1	HI F1	KBI F1	Overall Acc
BART	0.744	0.510	0.761	0.513
CGIM (QCIT → KBCIT → HCIT)	0.764	0.567	0.772	0.562
CGIM (QCIT → HCIT → KBCIT)	0.787	0.599	0.778	0.560

Table 3: Robust Test. The performance of BART and different information flow model.

(5) Is the explicit modeling method still effective in low-resource scenario?

(6) How CGIM is useful in CI-ToD?

4.4.1 Answer 1: GMA boosts performance across the related tasks

We devise three variations for exploring the effect of guided multi-head attention layer. In particular, QCIT → KBCIT is the variation by removing guided multi-head attention layer from QCIT to KBCIT and all the other components keep unchanged. Similarly, KBCIT → HCIT and HCIT → QCIT variation denotes that remove the corresponding guided multi-head attention layer for HCIT and QCIT, respectively.

Results are presented in Table 2 (row 2,3,4), we observe that without guided multi-head attention layer leads to a drop in the corresponding tasks. We attribute it to the fact that all the two sub-tasks are highly correlated, it hinders the information transfer and thus hurts the performance without the corresponding guided multi-head attention layer.

4.4.2 Answer 2: More layers may not be better

To investigate the influence of layers of guided multi-head attention module, we conduct experiments on the different layers of our framework. Figure 5 presents the results. We can observe: (1) The performance of CGIM with two or three layers is better than the model with one layer, which indicates that a deeper layer can achieve better interactions across three tasks. (2) Another interesting observation is that when the number of layers is five, we can observe the performance on overall accuracy drops a lot, even underperforming the model with one layer. We speculate that there may

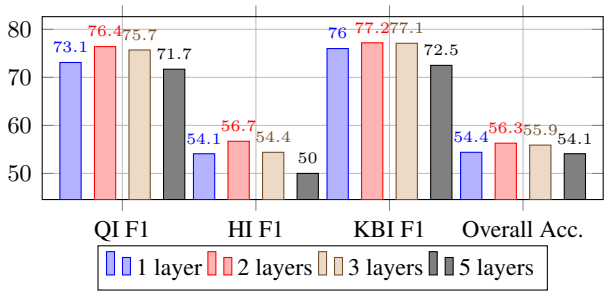


Figure 5: Influence of Layers.

be gradient vanishing or over-fitting problem when the layer of network exceeds five, which is consistent with prior observation (Feng et al., 2017; Qin et al., 2020).

4.4.3 Answer 3: CGIM is robust

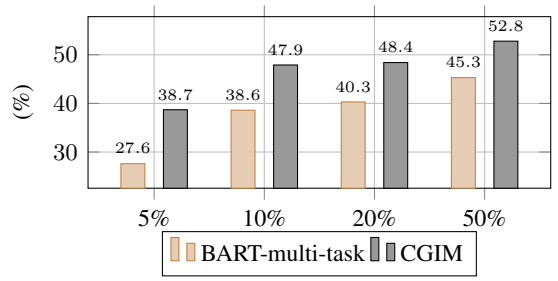
Another interesting research question is whether CGIM is robust to the initial task information flow. To answer this question, we conduct experiments with another initial information flow order QCIT → HCIT → KBCIT and the results are presented in Table 3. We witness two observations: (1) the performance CGIM (QCIT → KBCIT → HCIT) is comparable with the original CGIM; (2) it also outperforms BART by a large margin. Above observations verifies the robustness of our method to the initial task flow order.

4.4.4 Answer 4: Explicit interaction modeling boosts performance

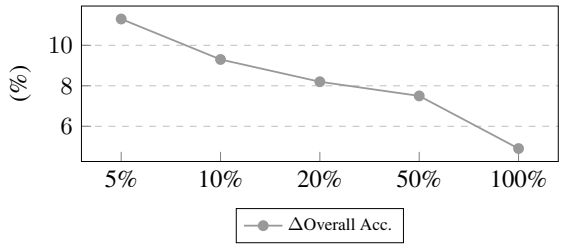
We replace three guided multi-head attention layers with three MLP modules. We refer to it as "w/MLP" and the results are reported in Table 2. As seen, CGIM yields better results than the "w/MLP" model with the same amount of parameters in all three tasks (over 4% drops on all tasks), which demonstrates that the improvements come from the proposed explicit interaction mechanism across the three tasks rather than the extra parameters.

4.4.5 Answer 5: CGIM works in few-shot scenario

We further investigate the effectiveness of CGIM in a low-resource setting. We randomly extract different proportions of datasets from the entire dataset to simulate the low-resource setting, i.e., [5%, 10%, 20%, 50%]. The results are shown in the Figure 6 (a). We observe that our framework outperforms the BART on all low-resource settings. We attribute it to the fact the proposed explicit interaction can make the most limited data and



(a) Performance of BART-multi-task and CGIM.



(b) ΔOverall Accuracy on Low-Resource Setting.

Figure 6: Low-Resource Performance.

effectively share the knowledge between the three tasks compared to implicit modeling method.

In addition, we also analyze the performance gap on different datasets. The results are shown in Figure 6 (b), we find that the less data we have, the higher the performance improvement of our model compared to `BART-multi-task`, which indicates that our framework is more practical and scalable in a low resource setting.

4.4.6 Answer 6: Qualitative analysis

This section provides a case study for better understanding of our model. Figure 7 shows one case made by baseline model `BART-multi-task` and `CGIM`. In this case, user query and dialogue history talks about the same topic (*the_clement_hotel*), which demonstrates that the QCIT and HCIT are highly correlated.

However, `BART-multi-task` predict the HCIT correctly but QCIT incorrectly, which demonstrates original implicit interaction paradigm does not effectively model the correlation across the tasks. In contrast, `CGIM` predicts both QCIT and HCIT correctly. We think that the proposed explicit interaction paradigm successfully grasps correlation and thus enhance each task.

5 Related Work

Increasing attention has been witnessed in consistency identification in dialogue. To this end, PersonaChat (Zhang et al., 2018) and PersonalDialog (Zheng et al., 2019) are introduced to implicitly

	QI	HI	KBI	
BART-m	0,	1,	1	**Driver**:where is the nearest hotel
CGIM	1,	1,	1	**System**:the nearest hotal is the_clement_hotel
GOLD	1,	1,	1	**Driver**:what is the address (for the_clement_hotel)
				System: hotel_keen is at 347_alta_mesa_ave

Figure 7: Prediction made by `BART-m` and `CGIM`. `BART-m` denotes `BART-multi-task`.

consider the consistency in dialogue generation. Welleck et al. (2019) model the consistency of dialogue systems by introducing a new natural language inference dataset called DialogueNLI. Dziri et al. (2019) propose to use state-of-the-art entailment techniques for evaluating the coherence of dialogue systems. Nie et al. (2021) propose a DialoguE COntradiction DEtection task (DECODE) to evaluate the ability to detect contradictory in dialogue. However, their work mainly focuses on consistency in open-domain direction. In contrast, our framework mainly considers improving consistency in task-oriented dialogues.

In recent years, Qin et al. (2021b) make the first step towards consistency identification in task-oriented dialogues and propose three sub-tasks to detect whether the system response is contradicted with the corresponding dialogue history, user query, and knowledge base. In addition, they also introduce a public benchmark CI-ToD and provide some state-of-the-art pre-trained models to facilitate the research. Unfortunately, their models only jointly consider the correlated three tasks in an implicit manner. Compared with their model, we propose a cycle guided interactive learning model (CGIM), which can explicitly model interaction across the three tasks in a cycled interaction manner. To our knowledge, we are the first to explore an explicit interaction paradigm for CI-ToD.

6 Conclusion

We studied how to explicitly model the interaction across three sub-tasks for consistency identification in task-oriented dialogue (CI-ToD). To this end, we introduced a cycle interactive learning model (CGIM), which facilitates the knowledge transfer across the three correlated tasks. Experiments show CGIM achieves state-of-the-art performance. In addition, CGIM is robust to the initial task flow order and works better in a low-resource setting, which is scalable in a real-world system deployment.

Acknowledgements

We thank all anonymous reviewers for their constructive comments. This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 62176078.

References

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

Jiaqi Bai, Long Zhou, Ambrosio Blanco, Shujie Liu, Furu Wei, Ming Zhou, and Zhoujun Li. 2021. Jointly learning to repair code and generate commit message. *arXiv preprint arXiv:2109.12296*.

Jianzhu Bao, Bin Liang, Jingyi Sun, Yice Zhang, Min Yang, and Ruifeng Xu. 2021. Argument pair extraction with mutual guidance and inter-sentence relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3923–3934.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Xiuying Chen, Zhi Cui, Jiayi Zhang, Chen Wei, Jianwei Cui, Bin Wang, Dongyan Zhao, and Rui Yan. 2020. Reasoning in dialog: Improving response generation by context reading comprehension. *arXiv preprint arXiv:2012.07410*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3806–3812, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaocheng Feng, Jiang Guo, Bing Qin, Ting Liu, and Yongjie Liu. 2017. Effective deep memory networks for distant supervised relation extraction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4002–4008.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2021. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1699–1713, Online. Association for Computational Linguistics.

Libo Qin, Wanxiang Che, Yangming Li, Mingheng Ni, and Ting Liu. 2020. Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8665–8672.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021a. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197. IEEE.

Libo Qin, Tianbao Xie, Shijue Huang, Qiguang Chen, Xiao Xu, and Wanxiang Che. 2021b. Don't be contradicted with anything! ci-tod: Towards bench-

marking consistency for task-oriented dialogue system.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. 2019. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672*.