LAGIM: A Label-Aware Graph Interaction Model for Joint Multiple Intent Detection and Slot Filling

1st Penghua Li
Key Laboratory of Intelligent Computing for Big Data
College of Automation
Chongqing University of Posts and Telecommunications
Chongqing 400065, China
lipenghua88@163.com

2nd Ziheng Huang
Key Laboratory of Intelligent Computing for Big Data
College of Automation
Chongqing University of Posts and Telecommunications
Chongqing 400065, China
s200303078@stu.cqupt.edu.cn

***Abstract*—Multi-intent spoken language understanding joint model can handle multiple intents in an utterance and is closer to complicated real-world scenarios, attracting increasing attention. However, existing research (1) usually focuses on identifying implicit correlations between utterances and one-hot encoding while ignoring intuitive and explicit original label characteristics; (2) only considers the token-level intent-slot interaction, which results in the limitation of the performance. In this paper, we propose a Label-Aware Graph Interaction Model (LAGIM), which captures the correlation between utterances and explicit labels' semantics to deliver enriched priors. Then, a global graph interaction module is constructed to model the sentence-level interaction between intents and slots. Specifically, we propose a novel framework to model the global interactive graph based on the injection of the original label semantics, which can fuse explicit original label features and provide global optimization. Experimental results show that our model outperforms existing approaches, achieving a relative improvement of 11.9% and 2.1% overall accuracy over the previous state-of-the-art model on the MixATIS and MixSnips datasets, respectively.**

***Index Terms*—Spoken Language Understanding, Joint Model, Label Semantics, Global Graph Interaction**

I. Introduction

Spoken language understanding (SLU) is one of the critical components of the task-oriented dialogue system, aiming to understand users' queries [1]. It traditionally consists of two subtasks: intent detection (ID) and slot filling (SF) [2]. The former can be treated as a classification task, and the latter can be seen as a sequence labeling task.

Since intents and slots are closely related [3], many researchers have worked on the single-intent SLU system with the joint model, considering the correlation between the two subtasks and obtaining remarkable success. For example, Goo et al. [4] propose the Slot-Gated, focusing on learning the relationship between intent and slot features to obtain better semantic results. Their experiments show that the model significantly improves semantic accuracy with 4.2% and 1.9% relative improvement compared to the benchmark model on ATIS and Snips datasets, respectively.

This work is supported by Chongqing Outstanding Youth Fund Project (cstc2021jcyj-jqX0001); Science and Technology Research Project of Chongqing Education Commission (KJZD-K202100603); National Natural Science Foundation of China (52272388).

In recent years, researchers have gradually discovered that an utterance constantly contains multiple intents in complex real-world scenarios. To this end, many studies are progressively focusing on the more practical multi-intent SLU system. The work in [5] begins to explore modeling multi-intent scenarios in dialogue systems. However, the proposed model only considers the multiple ID task and ignores the SF task, which does not achieve true multi-intent SLU. To handle this, Gangadharaiah [6] first attempts to conduct the multi-task framework with an attention-based neural network, which provides a improvement of 0.2% on ATIS dataset and 55% intent accuracy improvement on an internal multi-intent dataset. The study in [7] further proposes an adaptive interaction framework, named AGIF, to achieve fine-grained multi-intent information integration for slot filling, obtaining 44.5% and 76.4% semantic accuracy on MixATIS and MixSnips datasets, respectively. The work in [8] proposes a framework based on BERT [9], called SLIM, to fully exploit the existing annotation data and capture the interactions between intents and slots, achieving the semantic accuracy of 47.6% and 84.0% on MixATIS and MixSnips datasets, respectively. Although the above research has shown promising performance, the existing approaches for building the SLU system require further discussion with the following concerns.

- Oversimplified label representation. These models usually focus on identifying implicit correlations between utterances and one-hot encoding, ignoring intuitive and explicit original label features. The label semantics may be helpful, which can improve the performance of both subtasks by evaluating the semantic similarity between words in utterances and words in labels.
- Insufficient interaction. These methods only consider the token-level intent-slot interaction while the sentence-level global optimization is missing, resulting in leaking contextual features.

In this paper, we propose the **L**abel-**A**ware **G**raph **I**nteraction **M**odel (LAGIM) for joint multiple ID and SF, shown in Fig. 1, aiming to handle the above issues and achieve high accuracy in SLU system. Concretely, motivated by the triumph of leveraging label characteristics to assist

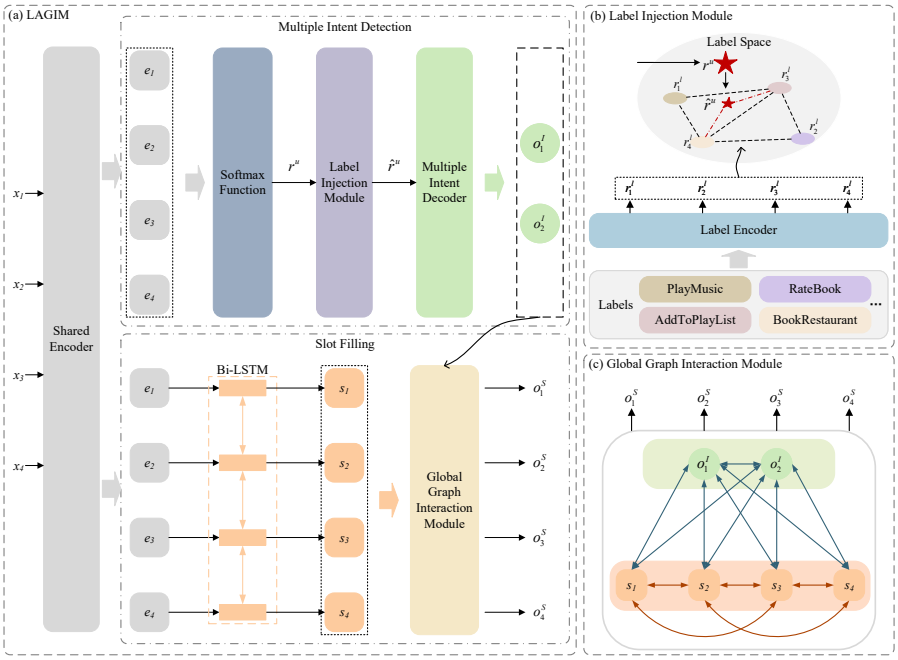

Fig. 1. The architecture of LAGIM (a), where (b) presents the label injection module and (c) indicates the global graph interaction module. For simplicity, we only draw one case with multiple labels.

model optimization [10], we construct the label space using words in intent labels to map label information into utterance representations. Then, we further introduce the global graph interaction module to perform sentence-level intent-slot interaction by using all tokens with multiple intents, as opposed to prior studies simply considering the token-level intent-slot interaction. Empirical results on two public datasets (MixATIS [11] and MixSnips [12]) demonstrate that our novel model outperforms competitive baselines. In addition, we explore the effect of the pre-trained model [13] in our framework.

To summarize, the contributions of our work are as follows: (1) We propose the novel label-aware graph interaction model, where the label injection module is used to merge initial label features while the global graph interaction module is introduced to model sentence-level intent-slot interaction; (2) We conduct experiments on two public benchmarks, and the results show that our model provides a small but statistically significant improvement. Moreover, we also explore the effect of the pre-trained model in our framework. With the pre-trained model, our model obtains a new state-of-the-art (SOTA) result.

II. Methodology

We first introduce the definition of the two subtasks, and then describe the proposed model, as shown in Fig. 1. It mainly consists of a shared encoder (II-B), a label injection module (II-C), a multiple ID decoder (II-D), and a global graph interaction module (II-E). We use a joint training scheme for the multiple ID and SF to optimize simultaneously.

A. Problem Definition

1) Multiple Intent Detection: Given input sequence $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)$ embedded by the utterance, multiple ID outputs a sequence intent label $o^I = \left(o_1^I, o_2^I, \cdots, o_m^I\right)$, where m is the number of intents in the utterance and n is the length of the utterance.

2) Slot Filling: Given the same sequence x, SF maps the x into a slot output sequence $o^S = \left(o_1^S, o_2^S, \cdots, o_n^S\right)$.

B. Shared Encoder

Following the work in [14], we utilize a shared encoder by stacking the bidirectional LSTM (Bi-LSTM) [15] and the self-attention mechanism [16] to obtain the representation, which can benefit from the temporal features in word order and contextual information.

1) Bi-LSTM: The Bi-LSTM consists of two LSTM layers and has been remarkably applied to the sequence labeling task. We utilize the Bi-LSTM to read the input sequence $\boldsymbol{X}$ forwardly and backwardly to produce context-sensitive hidden states:

$$\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n) \tag{1}$$

where $\boldsymbol{h}_i = Bi{-}LSTM\left(x_i, \boldsymbol{h}_{i-1}, \boldsymbol{h}_{i+1}\right)$.

2) Self-Attention: Firstly, we map the matrix of input vectors $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ (d is the mapped dimension) to $\boldsymbol{Q}$ (queries), $\boldsymbol{K}$ (keys), and $\boldsymbol{V}$ (values) matrices by using different linear projections $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$, respectively. Then, attention weight is computed by the dot product between $\boldsymbol{Q}$ and $\boldsymbol{K}$, and the self-attention output $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ is a weighted sum of values:

$$\boldsymbol{A} = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{2}$$

where d_k represents the dimension of $\boldsymbol{K}$. Finally, we concatenate the above two representations as the final encoding representation:

$$\boldsymbol{E} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n) = \boldsymbol{H}||\boldsymbol{A} \tag{3}$$

where $\boldsymbol{E} \in \mathbb{R}^{n \times 2d}$ and $||$ denotes the concatenation operation.

Following the study in [17], we employ the Softmax function over encoding representation $\boldsymbol{E}$ in the multiple ID task to capture the relevant context of utterance:

$$v_i = \textit{softmax}\left(\boldsymbol{W}_e \boldsymbol{e}_i + b\right), \boldsymbol{r}^u = \sum_i v_i \boldsymbol{e}_i \tag{4}$$

where $\boldsymbol{r}^u$ denotes the weighted sum of each encoding element e_i, v_i represents the corresponding normalized self-attention score, and $\boldsymbol{W}_e$ is the trainable matrix parameters. Likewise, we also involve a Bi-LSTM network to obtain slot-related hidden representation for the subsequent prediction.

$$\boldsymbol{S} = \left(\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n\right) \tag{5}$$

where $\boldsymbol{s}_t = Bi{-}LSTM\left(\boldsymbol{e}_t, \boldsymbol{s}_{t-1}, \boldsymbol{s}_{t+1}\right)$, t denotes the number of steps.

C. Label Injection Module

Inspired by the work in [10], instead of simply classifying utterance into a predefined set of intents, we leverage the best linear approximation idea [18] to assist us in specifying the intents of an utterance, where label features are adaptively fused into the utterance representation.

The best approximation problem specifies that Υ is a subspace of Hilbert space S. For a given vector $\boldsymbol{x} \in S$, we need to find the closest point $\hat{\boldsymbol{x}} \in \Upsilon$. Specifically, we first construct an embedded label subspace Υ with the basis sequence $\boldsymbol{R} = \left(r_1^l, r_2^l, \cdots, r_M^l\right)$, where M is the number of all intent labels. To obtain $\boldsymbol{R}$, we adopt a label encoder similar to section II-B to encode M intents. Then, given the utterance $\boldsymbol{r}^u$, we can inject it onto Υ to fetch its best linear approximation:

$$\hat{\boldsymbol{r}}^u = \sum\nolimits_{m=1}^{M} \boldsymbol{w}_m r_m^l \tag{6}$$

where $\boldsymbol{w} \in \mathbb{R}^M$ is computed as $\boldsymbol{w} = \boldsymbol{G}^{-1}\boldsymbol{b}$. The Gram matrix $\boldsymbol{G}$ and $\boldsymbol{b}$ are defined as follows.

$$\boldsymbol{G} = \begin{bmatrix} \left\langle r_1^l, r_1^l \right\rangle & \cdots & \left\langle r_M^l, r_1^l \right\rangle \\ \vdots & \ddots & \vdots \\ \left\langle r_1^l, r_M^l \right\rangle & \cdots & \left\langle r_M^l, r_M^l \right\rangle \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} \left\langle r^u, r_1^l \right\rangle \\ \vdots \\ \left\langle r^u, r_M^l \right\rangle \end{bmatrix} \tag{7}$$

Notably, we assume that $\left(r_1^l, r_2^l, \cdots, r_M^l\right)$ is linearly independent since each vector represents an intent concept. It should be a more balanced combination of other intent vectors. Therefore, $\boldsymbol{G}$ is guaranteed to be positive definite and has an inverse.

D. Multiple Intent Detection Decoder

Following [7], we predict multiple intents on each token, and the utterance results are obtained by voting for all tokens. After computing $\hat{\boldsymbol{r}}^u$, we can employ it for multiple ID decoding:

$$\boldsymbol{P}^I = \sigma\left(\boldsymbol{W}_i\left(\text{LeakyReLU}\left(\boldsymbol{W}_u \hat{\boldsymbol{r}}^u + \boldsymbol{b}_u\right)\right) + \boldsymbol{b}_i\right) \tag{8}$$

where $\boldsymbol{P}^I = \left(p_1^I, p_2^I, \cdots, p_M^I\right)$ denotes the distribution of intent probability, σ represents the Sigmoid activation function, $\boldsymbol{W}_*$ and $\boldsymbol{b}_*$ mean trainable matrix parameters and bias terms, respectively.

We obtain the utterance intent result $o^I = \left(o_1^I, o_2^I, \cdots, o_k^I\right)$ and o_i^I means p_i^I greater than the threshold t_u, where $t_u \in (0, 1)$ is a hyper-parameter adjusted using the validation subset [1]. For instance, if we have $\boldsymbol{P}^I = (0.2, 0.4, 0.6, 0.9, 0.3)$ and $t_u = 0.5$, the result $o^I = \left(o_3^I, o_4^I\right)$.

E. Global Graph Interaction Module

The proposed global graph interaction module, one of our model's main advantages, considers the sentence-level intent-slot interaction and achieves global optimization. In the following, we first represent the vanilla graph attention network (II-E1) and then demonstrate the mechanism of intent-slot interaction (II-E2) for decoding.

1) Vanilla Graph Attention Network: Graph attention network (GAT) [19] is a graph neural network variant that fuses graph structure information and node features in the model. Its masked self-attention layer lets nodes concentrate on neighborhood features and learn different attention weights, which can automatically decide the significance and correlation between the current node and its neighbors. Concretely, for a given graph with N nodes, one-layer GAT takes the initial node features $\tilde{\boldsymbol{H}} = \left(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, \cdots, \tilde{\boldsymbol{h}}_N\right), \tilde{\boldsymbol{h}}_n \in \mathbb{R}^F$ as input, aiming at producing a more abstract representation $\tilde{\boldsymbol{H}}' = \left(\tilde{\boldsymbol{h}}_1', \tilde{\boldsymbol{h}}_2', \cdots, \tilde{\boldsymbol{h}}_N'\right), \tilde{\boldsymbol{h}}_n' \in \mathbb{R}^{F'}$ as output. The attention mechanism of a typical GAT operated on the node representation can can be summarized as below:

$$\begin{aligned} \Psi\left(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_j\right) &= \text{LeakyReLU}\left(\boldsymbol{a}^T\left[\boldsymbol{W}_h \tilde{\boldsymbol{h}}_i || \boldsymbol{W}_h \tilde{\boldsymbol{h}}_j\right]\right) \\ \alpha_{ij} &= \frac{\exp\left(\Psi\left(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_j\right)\right)}{\sum_{j' \in \mho_i} \exp\left(\Psi\left(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_{j'}\right)\right)} \\ \tilde{\boldsymbol{h}}_i' &= ||_{k=1}^K \sigma\left(\sum\nolimits_{j' \in \mho_i} \alpha_{ij} \boldsymbol{W}_h \tilde{\boldsymbol{h}}_j\right) \end{aligned} \tag{9}$$

where $\boldsymbol{a} \in \mathbb{R}^{2F'}$ and $\boldsymbol{W}_h \in \mathbb{R}^{F \times F'}$ are the trainable matrix parameters, α_{ij} is the normalized attention weight denoting the significance of each $\boldsymbol{h}_j$ to $\boldsymbol{h}_i$, $\mho_i$ represents the first-order neighbors of node i (including itself) in the graph, K is the number of heads, and σ indicates the nonlinearity activation function.

2) Global Intent-Slot Interaction: To achieve sentence-level intent-slot interaction, given the slot-related hidden representation $\boldsymbol{S}$ and intent result o^I, we construct a global graph where all predicted multiple intents and sequence slots are connected. Mathematically, the graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertices refer to intents and slots, and edges refer to correlations between them.

Vertices We have $n + k$ nodes in the graph, where n is the utterance length and k is the number of o^I. The input of slot token feature is $\boldsymbol{G}^{[S,1]} = \boldsymbol{S}^{L+1} = \left\{\boldsymbol{s}_1^{L+1}, \boldsymbol{s}_2^{L+1}, \ldots, \boldsymbol{s}_n^{L+1}\right\}$, where $\boldsymbol{s}_i^{L+1} = \sigma\left(\sum_{j \in \mho_i} \alpha_{ij} \boldsymbol{W}_l \boldsymbol{s}_j^L\right)$, $\boldsymbol{W}_l$ is the trainable

[1] In our experiment, t_u is set to 0.5.

matrix parameter, L is the number of layer, and the initial states vector is $\boldsymbol{S}^1 = \boldsymbol{S}$. The input of intent feature is $\boldsymbol{G}^{[I,1]} = \left\{\phi^{emb}\left(\boldsymbol{o}_1^I\right), \phi^{emb}\left(\boldsymbol{o}_2^I\right), \ldots, \phi^{emb}\left(\boldsymbol{o}_k^I\right)\right\}$, where ϕ^{emb} is a trainable embedding matrix parameter for mapping o_*^I. The first layer states vector for two kind of nodes is $\boldsymbol{G}^1 = \left\{\boldsymbol{G}^{[I,1]}, \boldsymbol{G}^{[S,1]}\right\}$.

Edges There are three types of connections in this graph. (a) slot-slot connection: We assemble the slot-slot connection, where each slot node connects other slot nodes with the sliding window size [2], to model the slot dependency further and incorporate the bidirectional contextual information. (b) intent-intent connection: We connect all intent nodes to model the relationship between each intent node since they all materialize in the same utterance. (c) intent-slot connection: Since intents and slots are closely related, we form the intent-slot connection to model the global interaction between the two subtasks. Specifically, each slot node connects all predicted intent nodes to capture relevant intent information automatically.

Within the graph, the aggregation process at l-th layer can be described as:

$$\boldsymbol{g}_i^{[S,l+1]} = \sigma\left(\sum_{j\in\mathcal{G}^S}\alpha_{ij}\boldsymbol{W}_g\boldsymbol{g}_j^{[S,l]} + \sum_{j\in\mathcal{G}^I}\alpha_{ij}\boldsymbol{W}_g\boldsymbol{g}_j^{[I,l]}\right) \quad (10)$$

where $\mathcal{G}^S$ and $\mathcal{G}^I$ are vertices sets in connected slot nodes and intent nodes, respectively. After L layers' aggregation, we obtain the final representation for slot prediction as follows.

$$\boldsymbol{o}_t^S = \text{argmax}\left(\text{softmax}\left(\boldsymbol{W}_s\boldsymbol{g}_t^{[S,L+1]}\right)\right) \quad (11)$$

There, $\boldsymbol{W}_s$ is the trainable matrix parameter and o_t^S is the predicted slot of the t-th token in the utterance.

F. Joint Training

1) Loss Function: The loss functions for multiple ID is formulated as:

$$\begin{aligned}\vartheta\left(\bar{y}, y\right) &= \bar{y}\log\left(y\right) + \left(1-\bar{y}\right)\log\left(1-y\right)\\ \mathcal{L}_1 &\triangleq -\sum_{i=1}^{n}\sum_{j=1}^{M}\left(\vartheta\left(\bar{y}_i^{(I)}[j], y_i^{(I)}[j]\right)\right)\end{aligned} \quad (12)$$

Similarly, the loss function for SF is described as:

$$\mathcal{L}_2 \triangleq -\sum_{i=1}^{n}\sum_{j=1}^{Q}\left(\bar{y}_i^{(S)}[j]\log\left(y_i^{(S)}[j]\right)\right) \quad (13)$$

where Q is the number of slot labels, $\bar{y}^{(I)}$ and $\bar{y}^{(S)}$ are gold intent label and gold slot label, respectively.

2) Model Training: We adopt a joint training mode to consider the two subtasks and update parameters by global optimization. The final joint objection is combined from the two loss functions as follows.

$$\mathcal{L} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2 \quad (14)$$

Here, λ_* is the hyper-parameter.

[2]In our experiment, the sliding window size is 2/1 for MixATIS/MixSnips, respectively.

III. Experiments

A. Datasets and Metrics

We conduct experiments on two public multi-intent SLU datasets. One is MixATIS, which is expanded from ATIS and contains 13162/756/828 utterances for training/validation/testing, covering 17 intent categories such as `atis_flight_no`, `atis_flight_time`, and `atis_meal`. The other is MixSnips, which is expanded from Snips and contains 39776/2198/2199 utterances for training/validation/testing, covering 7 intent categories such as `GetWeather`, `BookRestaruant`, and `PlayMusic`. In addition, both datasets are the cleaned version, removing the repeated sentences.

Following previous work, we adopt the accuracy for multiple ID, the F1 score for SF, and the overall accuracy for the semantic frame parsing. The overall accuracy illustrates the ratio of an utterance whose intent labels and slot labels are all accurately predicted.

B. Experimental Settings

The dimensionality of the hidden unit in the shared encoder, word embedding, and label embedding are 256, 128, and 256 on MixATIS, while on MixSnips, they are 256, 64, and 128. We set the number of the attention head is 4/8 for MixATIS/MixSNIPS, respectively. All layer number of GAT is fixed to 2. We use Adam [25] for optimization with a learning rate of 1e-3 and a weight decay of 1e-6, and set the dropout rate as 0.4. As for the joint objection (Eq.(14)), the hyper-parameters λ_1 and λ_2 are initialed to 0.85 and 0.15, respectively. In all the experiments, we choose the model which performs the best on the validation subset and then evaluate it on the test subset for a fair comparison. All experiments are conducted at Nvidia GeForce GTX 1080Ti.

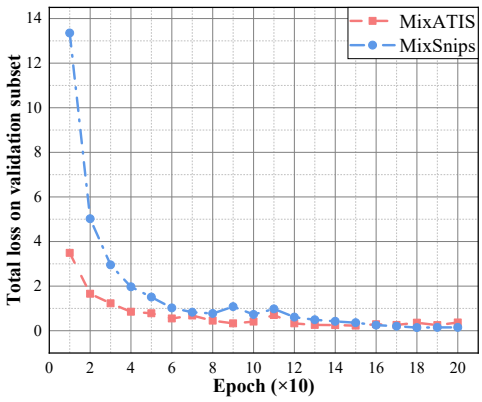

Fig. 2. The total loss on validation subset during training of LAGIM.

C. Main Results

The loss curve of LAGIM during training on the two datasets, from 10 epoch, is shown in Fig. 2. The performance comparison between LAGIM and other baselines is shown in Table I, from which we mainly have the following observations.

TABLE I
THE PERFORMANCE OF DIFFERENT MODELS ON TWO PUBLIC TARGET DATASETS.

Model	MixATIS			MixSnips		
	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)
Attention BiRNN [20]	39.1	86.4	74.6	59.5	89.4	95.4
Slot-Gated [4]	35.5	87.7	63.9	55.4	87.9	94.6
Bi-Model [21]	34.4	83.9	70.3	63.4	90.7	95.6
SF-ID [22]	34.9	87.4	66.2	59.9	90.6	95.0
Stack-Propagation(Concat) [23]	39.6	86.5	76.2	72.4	93.7	96.2
Joint Multiple ID-SF [6]	36.1	84.6	73.4	62.9	90.6	95.1
AGIF [7]	40.8	86.7	74.4	74.2	94.2	95.1
SDJN [24]	44.6	88.2	77.1	75.7	94.4	96.5
LAGIM (Ours)	**49.9***	**88.3***	**77.8***	**77.3***	**94.8***	**97.1***

* These results show that the advancement of LAGIM over all baselines is statistically momentous with $p < 0.05$ under t-test.

The LAGIM outperforms all baselines on all subtasks and datasets. Specifically, on the MixATIS dataset, LAGIM overpasses the previous SOTA model SDJN by 5.3%, 0.1%, and 0.7% on sentence-level semantic frame parsing, SF, and multiple ID, respectively. LAGIM surpasses SDJN by 1.6%, 0.4%, and 0.6% on sentence-level semantic frame parsing, SF, and multiple ID on the MixSNIPS dataset. We attribute the improvement to the fact that our proposed global graph interaction module can sufficiently capture the correlation between intents and slots, enhancing the performance of multi-intent SLU system.

LAGIM achieves a more considerable improvement on multiple ID than on SF. The reason is that except for the implicit correlations between utterances and one-hot encodings, our model injects label semantics into utterances to fuse explicit original label features and deliver enriched priors. In contrast, the previous models all ignore this. Based on semantic information injection, the label injection module catch the high-confidence resemblance between intent representations and label features through the best linear approximation, which brings significant and consistent improvements to multiple ID.

D. Model Analysis

1) Ablation Study: To verify the efficacy of our model from different perspectives, we conduct a set of ablation experimentations on the two datasets, which are shown in Table II.

TABLE II
EFFECT OF VARIANTS ON LAGIM

Model	MixATIS			MixSnips		
	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)
LAGIM	**49.9**	**88.3**	**77.8**	**77.3**	**94.8**	**97.1**
w/o LIM	42.3 (↓ 7.6)	87.1 (↓ 1.2)	76.4 (↓ 1.4)	72.2 (↓ 5.1)	93.8 (↓ 1.0)	95.5 (↓ 1.6)
w/o GGIM	43.1 (↓ 6.8)	85.7 (↓ 2.6)	77.2 (↓ 0.6)	74.0 (↓ 3.3)	92.5 (↓ 2.3)	96.4 (↓ 0.7)
+ More Para.	46.5 (↓ 3.4)	86.0 (↓ 2.3)	77.3 (↓ 0.5)	75.4 (↓ 1.9)	92.8 (↓ 2.0)	96.6 (↓ 0.5)

Efficacy of Label Injection Module We remove the label injection module (LIM), utilize the outcome of utterance/token representations and labels, and maintain other components unchanged. It is named as *w/o LIM* in Table II. We can observe that the absence of the LIM leads to 1.4% and 1.6% intent accuracy drops clearly on MixATIS and MixSnips, respectively. This phenomenon signifies that LIM can capture the correlation between utterances and explicit labels' semantics, which is advantageous for the semantic performance of the multiple ID subtask.

Efficacy of Global Graph Interaction Module We remove the global graph interaction module (GGIM), utilize the product of the slot-related Bi-SLTM encoder for SF, and maintain other components unchanged. We refer it to *w/o GGIM* in Table II. We can observe that the slot f1 score drops by 2.6% and 2.3% on MixATIS and MixSnips, respectively. It proves that the GGIM can model the sentence-level intent-slot interaction and navigate the prediction for SF through global optimization, which significantly boosts the performance of the SLU system.

Following [7], we replace the LSTM (2 layers) as the proposed GGIM to verify that the added parameters do not work. Table II (*More Para.*) shows the consequences. We celebrate that our model outperforms more parameters by 3.4% and 1.9% overall accuracy in MixATIS and MixSnips, respectively, indicating that the advancements come from the proposed GGIM.

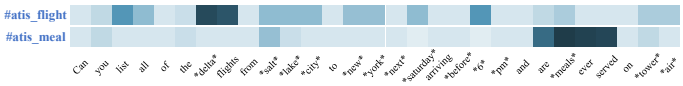

Fig. 3. The visualization result. The Y-axis is the predicted intents, and X-axis is the input utterance where all slot tokens are surrounded by *. For each block, the darker the color is, the more relevant they are.

TABLE III
LAGIM PERFORMANCE ON PRE-TRAINED-BASED MODEL ON DATASETS

Model	MixATIS			MixSnips		
	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)
LAGIM	49.9	88.3	77.8	77.3	94.8	97.1
Multiple ID (RoBERTa)	-	-	77.9	-	-	96.3
SF (RoBERTa)	-	88.6	-	-	85.1	-
LAGIM+RoBERTa	**60.3**	**91.2**	**80.1**	**83.5**	**97.5**	**98.7**

2) Visualization: To better comprehend what the global graph has learned and contributes to the final prediction, we visualize its attention weight shown in Fig. 3, using an utterance from the MixATIS dataset. Based on the utterance with the multiple intents `atis_flight` and `atis_meal`, we can see the attention weights successfully focus on the correct slot,

which means our LAGIM can learn to incorporate the correlated intent information at each slot. More detailly, LAGIM correctly aggregates the corresponding `atis_flight` intent information at slot `delta` and `atis_meal` intent information at slot `meals`.

3) Exploration of Pre-trained Model: Following the work in [23], we explore the pre-trained model in our model and replace the shared encoder as RoBERTa [13] with the fine-tuning approach. We keep the other components identified in our model and consider the first subword label if a word is broken into multiple subwords.

The result, which introduces the pre-trained model and is shown in Table III, exhibits that the RoBERTa-based model is incredibly well on two public datasets and reaches a new SOTA performance. We attribute this to the fact that pre-trained models can provide more rich semantic features, which is significant for predicting the multi-intent SLU system. Primarily, we also conduct multiple ID and SF experiments separately based on the RoBERTa model. For ID, we set the special `[CLS]` word embedding into a classification layer to classify the intent; For SF, we also feed the final hidden representation $\boldsymbol{h}_{BERT_i} \in \mathbb{R}^d$ for each token i into a classification layer. From the result, shown in Table III, we can see that the F1 and Acc are lower than our joint model based on RoBERTa, which again demonstrates the effectiveness of Pre-trained model manipulating the relationship between the two tasks.

IV. Conclusion

This paper proposes the LAGIM for joint multiple ID and SF. The novel model mainly consists of two modules, which are the proposed LIM and the introduced GGIM. The former can capture the correlation between utterances and explicit labels' semantics, and the latter can model the sentence-level intent-slot interaction for global optimization. Empirical results on two public datasets reveal that our novel model outperforms the prior research.

References

[1] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot filling models in natural language understanding," *ACM Computing Surveys (ACM Comput Surv)*, 2022.

[2] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[3] P. Zhou, Z. Huang, F. Liu, and Y. Zou, "Pin: A novel parallel interactive network for spoken language understanding," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2950–2957, 2021.

[4] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "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 (NAACL-HLT)*, pp. 753–757, 2018.

[5] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, and M. Henderson, "The fourth dialog state tracking challenge," in *Dialogues with Social Robots*, pp. 435–449, Springer, 2017.

[6] R. Gangadharaiah and B. Narayanaswamy, "Joint multiple intent detection and slot labeling for goal-oriented dialog," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 564–569, 2019.

[7] L. Qin, X. Xu, W. Che, and T. Liu, "Agif: An adaptive graph-interactive framework for joint multiple intent detection and slot filling," in *Findings of the Association for Computational Linguistics: EMNLP*, pp. 1807–1816, 2020.

[8] F. Cai, W. Zhou, F. Mi, and B. Faltings, "Slim: Explicit slot-intent mapping with bert for joint multi-intent detection and slot filling," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7607–7611, 2022.

[9] J. Devlin, M.-W. Chang, K. Lee, and 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)*, pp. 4171–4186, 2019.

[10] T.-W. Wu, R. Su, and B. Juang, "A label-aware bert attention network for zero-shot multi-intent detection in spoken language understanding," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4884–4896, 2021.

[11] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[12] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[14] L. Qin, W. Che, Y. Li, M. Ni, and T. Liu, "Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification," in *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, pp. 8665–8672, 2020.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems (NIPS)*, vol. 30, 2017.

[17] V. Zhong, C. Xiong, and R. Socher, "Global-locally self-attentive encoder for dialogue state tracking," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1458–1467, 2018.

[18] G. E. del Pino and H. Galaz, "Statistical applications of the inverse gram matrix: A revisitation," *Brazilian Journal of Probability and Statistics*, pp. 177–196, 1995.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[20] B. Liu and I. Lane, "Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling," in *Proc. Interspeech 2016*, pp. 685–689, 2016.

[21] Y. Wang, Y. Shen, and H. Jin, "A bi-model based RNN semantic frame parsing model for intent detection and slot filling," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 309–314, 2018.

[22] H. E, P. Niu, Z. Chen, and M. Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5467–5471, 2019.

[23] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2078–2087, 2019.

[24] L. Chen, P. Zhou, and Y. Zou, "Joint multiple intent detection and slot filling via self-distillation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7612–7616, 2022.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.