Continual Few-shot Intent Detection

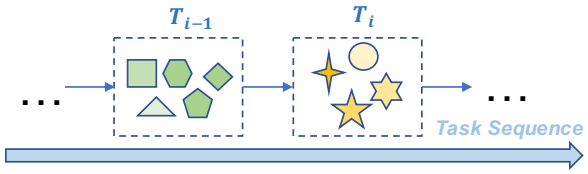
Guodun Li[1*], **Yuchen Zhai**[1], **QiangLong Chen**[1],
Xing Gao[2], **Ji Zhang**[2], **Yin Zhang**[1†]
[1]College of Computer Science and Technology, Zhejiang University
[2]DAMO Academy, Alibaba Group
{guodun.li,zhaiyuchen,chenqianglong,zhangyin98}@zju.edu.cn
{gaoxing.gx,zj122146}@alibaba-inc.com

Abstract

Intent detection is at the core of task-oriented dialogue systems. Existing intent detection systems are typically trained with a large amount of data over a predefined set of intent classes. However, newly emerged intents in multiple domains are commonplace in the real world. And it is time-consuming and impractical for dialogue systems to re-collect enough annotated data and re-train the model. These limitations call for an intent detection system that could continually recognize new intents with very few labeled examples. In this work, we study the Continual Few-shot Intent Detection (CFID) problem and construct a benchmark consisting of nine tasks with multiple domains and imbalanced classes. To address the key challenges of (a) catastrophic forgetting during continuous learning and (b) negative knowledge transfer across tasks, we propose the Prefix-guided Lightweight Encoder (PLE) with three auxiliary strategies, namely Pseudo Samples Replay (PSR), Teacher Knowledge Transfer (TKT) and Dynamic Weighting Replay (DWR). Extensive experiments demonstrate the effectiveness and efficiency of our method in preventing catastrophic forgetting and encouraging positive knowledge transfer across tasks.

1 Introduction

Intent Detection (ID) is at the core of task-oriented dialogue systems. It aims at understanding the goals underlying user utterances and classifying them into different intents accurately (Zhang et al., 2020; Qin et al., 2021). Traditionally, the ID system is trained with plenty of labeled data to identify a predefined set of intent classes (Larson et al., 2019). However, newly emerged intents in multiple domains are commonplace in the real scenario. A naive approach to detecting new intents is to re-collect annotated data and re-train the model, which is time-consuming and impractical for dialogue systems in deployment. Thus, an intent detection system that could continually recognize new intents with very few labeled examples is called for.



Setting	Few-shot	Continual	Multi-Domain	#Classes
FSID (Zhang et al., 2020)	✓	✗	✓	-
CID (Liu et al., 2021)	✗	✓	✓	-
FSCIL-ID (Xia et al., 2021)	✓	✓	✗	Balanced
CFID (Ours)	✓	✓	✓	Imblanced

Figure 1: Illustration of Continual Few-shot Intent Detection (CFID). Compared with existing works, CFID aims to recognize continually new intents from multiple domains with very few labeled examples and imbalanced number of classes across tasks.

Many efforts have been made in existing works to achieve this goal. Zhang et al. (2020) propose a discriminative nearest neighbor classification method to solve the Few-Shot Intent Detection (FSID) problem. They mainly focus on the data scarcity and ignore the ability to learn consecutive tasks. Liu et al. (2021) and Wang et al. (2021) propose promising solutions for the Continual Intent Detection (CID) problem. Nevertheless, they do not consider the few-shot setting, which is more realistic due to the scarcity of labeled data for new intents. The most recent work (Xia et al., 2021) provides the first study on few-shot class-incremental learning for intent detection (FSCIL-ID). However, it assumes (1) all tasks belong to the same domain and (2) the number of classes across few-shot tasks is balanced, which is not practical in the real world.

In this work, we define a more realistic problem as Continual Few-shot Intent Detection (CFID) and construct a benchmark consisting of nine tasks with multiple domains and imbalanced classes in 5-shot and 10-shot settings. As shown in Figure 1, the

[*] Work done during internship at Alibaba.
[†] Corresponding author: Yin Zhang.

Proceedings of the 29th International Conference on Computational Linguistics, pages 333–343
October 12–17, 2022.

system is provided with a sequence of tasks with limited labeled data and expected to continually learn on new intents while performing accurate classification on all previously seen tasks. Compared with existing works, CFID is more aligned with real scenario where the number of classes is highly imbalanced and task domains vary widely.

We consider addressing the problem from the intersection perspective of few-shot and lifelong learning. A strong baseline is to construct prototypical networks with a pre-trained language model (PrLM) and sequentially update all the weights on each task. However, there are two issues: (i) over-parameterization of PrLMs makes them prone to overfit the current task and cause *catastrophic forgetting* of previous knowledge (Ke et al., 2021a; Yuan et al., 2021). (ii) due to the domain gaps and imbalanced classes, the knowledge inherited from the past task may degrade the performance of the current, namely *negative knowledge transfer*.

To address the above issues, we propose a novel Prefix-guided Lightweight Encoder (PLE) with three auxiliary strategies. In detail, PLE adopts a parameter-efficient tuning paradigm to alleviate forgetting caused by over-parameterization, consisting of a lightweight Continual Adapter module to interact with a frozen PrLM, and a Prefix-guided Attention mechanism to guide the frozen PrLM. To further alleviate forgetting, we propose the Pseudo Samples Replay (PSR) strategy, which consolidates previous knowledge by replaying two essential samples that best approximate the previous tasks. To alleviate negative knowledge transfer, we propose the Teacher Knowledge Transfer (TKT) strategy, which transfers the task-specific knowledge into the current model via distillation to compensate for the performance drop of new tasks. Moreover, due to the variability of tasks, it is hard to identify whether a past task transfers positive or negative knowledge to the current. Thus, we propose the Dynamic Weighting Replay (DWR) strategy to balance learning new tasks and replaying old ones, which dynamically determines the learning weight of the old task in each iteration.

Our main contributions are as follows: 1) To the best of our knowledge, we are the first to formulate the Continual Few-shot Intent Detection (CFID) problem and construct a benchmark for it. 2) We propose a novel method PLE with three strategies for CFID to alleviate forgetting and negative transfer. 3) Extensive experiments show the effectiveness of our method in preventing forgetting and encouraging positive knowledge transfer across tasks.

2 Related Work

Traditional Intent Detection aims to classify intent in the utterance, which can be defined as a sentence classification task. Popular approaches such as Goo et al. (2018); Qin et al. (2019); Mehri et al. (2020) have achieved promising performance. However, such methods heavily rely on large amounts of labeled data.

Few-shot Intent Detection aims to classify accurately identify intents in few-shot settings. Zhang et al. (2020) solves it as a textual entailment problem and uses large-scale entailment datasets for pre-training. However, it is time-consuming and expensive to train with hundreds of intents. Mehri and Eric (2021) proposes an example-driven strategy to tackle this task, which learns to classify utterances by comparing them to examples. Luo et al. (2021) and Dopierre et al. (2021) solve the data scarcity by leveraging the label names or augmented samples. More recently, Zhang et al. (2021a,b) show the effectiveness of pre-training and contrastive fine-tuning on this task.

Continual Learning aims to learn a sequence of tasks incrementally. Most works in NLP domains focus on text classification tasks in continual settings (Ke et al., 2021c,a,b; Geng et al., 2021; Qin and Joty, 2022). The main problem for continual text classification is *catastrophic forgetting* and replay-based methods (Han et al., 2020; Cui et al., 2021) have been proven promising to alleviate the problem, which retain a few examples in previous tasks and continually replay them with new tasks.

Continual Intent Detection. Recently, Liu et al. (2021) and Wang et al. (2021) have made some efforts on the Continual Intent Detection task (CID). However, they did not further investigate with the few-shot setting, which is more challenging and crucial for the low-resource dialogue systems. The most similar to our work is (Xia et al., 2021), which firstly proposes the Few-shot Class-Incremental Learning for Intent Detection (FSCIL-ID). However, it is not aligned with the real scenario for the following reasons: (i) All tasks belong to the same domain without considering the domain gaps of different intents. (ii) The number of classes in emerging new tasks is fixed without considering the imbalance of classes across tasks in real systems.

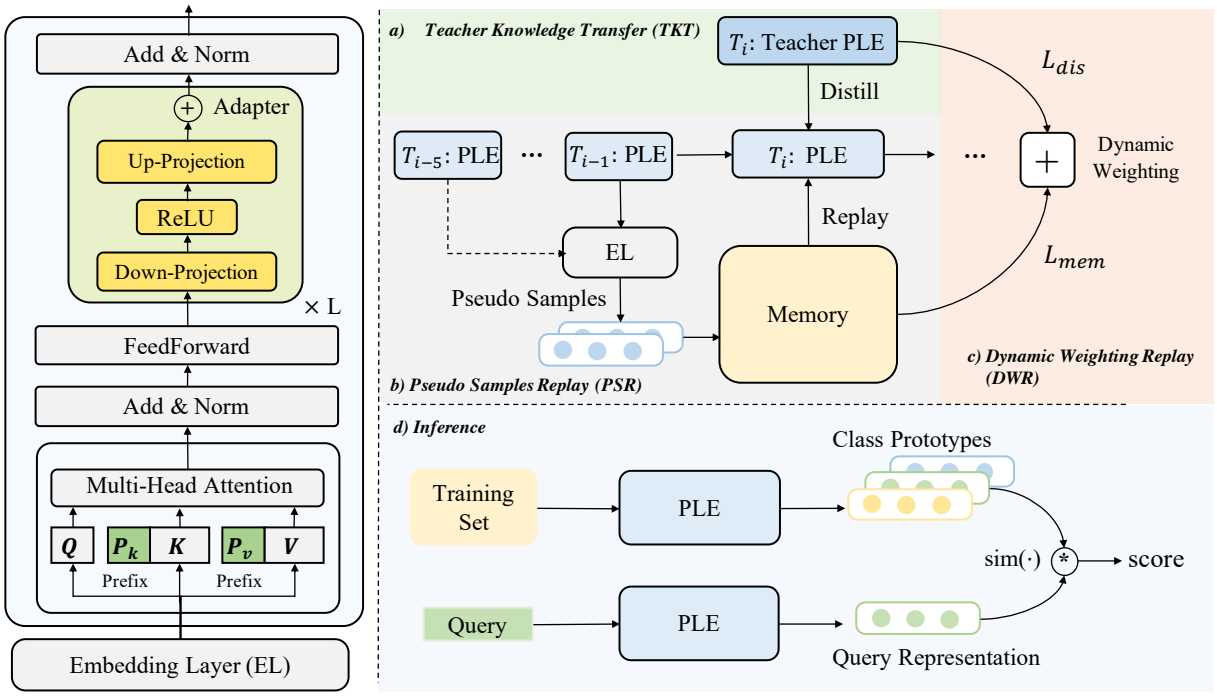


Figure 2: Overview of the proposed framework. The left side shows the structure of our PLE, which consists of a frozen PrLM and the Prefix (P_k, P_v) and Continual Adapter inserted into each layer. The trainable parameters are in green. The top right shows the process of learning for a new task $\mathcal{T}_i$ with three strategies: Teacher Knowledge Transfer (TKT), Pseudo Samples Replay (PSR), and Dynamic Weighting Replay (DWR). The lower right shows the distance-based classification pipeline at inference.

Summary. Existing works in few-shot intent detection mainly focus on the data scarcity and ignore the ability to learn consecutive tasks, which is essential for the online dialogue systems. The works in continual intent detection do not consider the data scarcity of emerging new intents. The newly proposed FSCIL-ID setting is also not aligned with the online dialogue systems. In contrast to those works, our work aims to recognize continually emerging new intents from multiple domains with very few labeled examples.

3 Methodology

3.1 Problem Formulation

In the CFID setting, given a sequence of n tasks $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n\}$, each task $\mathcal{T}_i$ contains its own training set D^i_{train}, development set D^i_{dev}, and test set D^i_{test}. Each dataset D contains a series of samples $\{(x_i, y_i)\}|_{i=1}^{|D|}$, where y_i is the ground-truth intent class of the input utterance x_i. In particular, we describe its few-shot nature that there are only $K \in \{5, 10\}$ samples for each class in the training set. There are also a few (e.g., 10) samples for each class in the development set. This is because using a larger development set brings significant advantages and defeats the goal of few-shot learning (Gao et al., 2021). After learning $\mathcal{T}_i$, the model is evaluated separately on the test set of seen tasks. The setup is aligned with the real scenario, where the data privacy of different users is protected while the task information is available.

3.2 Overall Framework

As shown in Figure 2, the framework of the proposed method consists of one main module and three strategies of continual learning: 1) The lightweight PLE is responsible for extracting semantic features of the input utterances. 2) The TKT aims to transfer task-specific knowledge to the current model. 3) The PSR first selects two key samples per class and encodes them through the frozen embedding layer (EL) to generate pseudo samples and save them into the Memory. 4) The DWR is responsible for balancing the learning of new tasks and replaying past tasks.

3.3 Prefix-guided Lightweight Encoder (PLE)

PLE serves as the main module to alleviate catastrophic forgetting caused by over-parameterization. As a sub-module of PLE, the **Continual Adapter** is a full continual learning lightweight module

designed to capture knowledge across tasks and mitigate over-fitting by only tuning a small number of parameters, inspired by adapter-based tuning (Houlsby et al., 2019). More specifically, it consists of a down-projection with $W_{\text{down}} \in \mathbb{R}^{d\times r}$ to project the input hidden states $h \in \mathbb{R}^d$, followed by a nonlinear activation function $\text{ReLU}(\cdot)$, and an up-projection with $W_{\text{up}} \in \mathbb{R}^{r\times d}$, formally:

$$h \leftarrow h + \text{ReLU}(hW_{\text{down}})W_{\text{up}}. \tag{1}$$

Following (He et al., 2021), the adapter is inserted only after the feed forward layer of the transformer block. Note that the parameters of Continual Adapter are shared by each task and continually updated in the continual learning process while the PrLM is kept frozen.

To guide the frozen backbone in capturing task-specific knowledge dynamically, we further propose the **Prefix-guided Attention** mechanism inspired by prefix tuning (Li and Liang, 2021). It incorporates continuous prompts into the self-attention layer to guide the final self-attention flow. More specifically, we concatenate two sets of l tunable prefix vectors $P_k, P_v \in \mathbb{R}^{l\times d}$ to the keys and values of the multi-head attention at every layer. In this way, the computation of head_i is modified as:

$$\text{head}_i = \text{Attn}(XW_i^Q, [P_k^i, XW_i^K], [P_v^i, XW_i^V]), \tag{2}$$

where $X \in \mathbb{R}^{m\times d}$ is the input sequence representation and W_i^Q, W_i^K, and $W_i^V \in \mathbb{R}^{d\times d_h}$ are the parameter matrices. With the guidance of the prefix, the distribution of attention can be re-modulated dynamically in the continual learning process.

3.4 Teacher Knowledge Transfer (TKT)

TKT is to alleviate negative knowledge transfer across tasks, a phenomenon that impairs model performance on the current task. While most works (Ke et al., 2021a) design complicated dynamic architecture to encourage positive knowledge transfer, TKT can simply and explicitly distill task-specific knowledge into the model to compensate for the performance.

Concretely, we first train a teacher PLE individually. The parameters of prefix and adapters are randomly initialized to avoid transferring knowledge from past tasks and gain more task-specific knowledge from the current. Then, we transfer the task-specific knowledge into the continually learning model through knowledge distillation.

As for the teacher PLE f_θ^T, in each iteration, N classes are randomly selected from the label space, and then K samples are selected for the encoder to extract features. The obtained features are averaged for each class prototype: $\hat{y}_j = \frac{1}{K}\sum_{k=1}^{K} f_\theta^T(x_k)$. The teacher is optimized by minimizing the cross entropy loss L_{sim}, formally:

$$L_{\text{sim}} = -\sum_{i=1}^{N\times K}\sum_{j=1}^{N}\mathbb{I}(y_i = y_j)\times \log\frac{\exp(\text{sim}(f_\theta^T(x_i), \hat{y}_j)/\tau)}{\sum_{l=1}^{N}\exp(\text{sim}(f_\theta^T(x_i), \hat{y}_l)/\tau)}. \tag{3}$$

where $\text{sim}(\cdot)$ is the cosine similarity function and τ is a temperature hyper-parameter and $\mathbb{I}(\cdot)$ is the indicator function. To fully make use of $N \times K$ samples, we select one sample at a time from a class as a query and the rest of the samples as support samples to compute the prototype so that there are $N \times K$ times of nearest neighbor classification in parallel at each iteration.

As for the student PLE f_θ^S, it firstly inherits the previous knowledge by reusing the parameters of the last learned model. Then, it gains task-specific knowledge by training on the current task with knowledge distillation, formally:

$$L_{\text{dis}} = \sum_{i=1}^{N\times K} \|f_\theta^S(x_i) - f_\theta^T(x_i)\|. \tag{4}$$

3.5 Pseudo Samples Replay (PSR)

PSR is to consolidates previous knowledge in replaying-based ways. Concretely, after learning for new tasks, we first obtain the prototype feature of each class by averaging the features of all samples labeled as this class: $\hat{y}_j = \frac{1}{K}\sum_{k=1}^{K} f_\theta^S(x_k)$. Then we select the instance closest to the prototype of class as the most representative sample, and select the instance farthest to the prototype of class as the hardest sample. To avoid direct access to the raw texts for privacy, these two samples are encoded with the frozen PrLM to generate pseudo samples, whose embedding space is always not distorted during continual learning. Finally, we store the two samples in the memory for each class.

In this way, the goal of replaying can be achieved by storing a minimum number of samples (i.e, two samples per class). During replaying the pseudo samples, we randomly select N classes from the previous task and adopt the cross-entropy loss

L_{mem} to ensure intra-class compactness while increasing inter-class distances, formally:

$$L_{\text{mem}} = -\sum_{i=1}^{N\times 2}\sum_{j=1}^{N}\mathbb{I}(y_i = y_j)\times \log\frac{\exp(\text{sim}(f_\theta^S(x_i), \hat{y}_j)/\tau)}{\sum_{l=1}^{N}\exp(\text{sim}(f_\theta^S(x_i), \hat{y}_l)/\tau)}. \quad (5)$$

3.6 Dynamic Weighting Replay (DWR)

DWR is to find a good trade-off between learning new tasks and replaying. Due to the domain variety, it is hard to determine whether to replay more on old tasks (i.e., PSR) or distill more on new tasks (i.e., TKT). They can be regarded as two contradictory optimization objectives. It drives us to design DWR to dynamically decide the weights of the two objectives and get a Pareto optimal solution.

Concretely, we first randomly sample one previous task to replay at each iteration rather than all the previous tasks. Then, we adopt a Pareto-optimal weighting strategy (Sener and Koltun, 2018) inspired by multi-task learning. The learning weight of the sampled old task can be determined dynamically in each iteration. The total loss is defined as follows:

$$\begin{aligned} L &= \lambda_{\text{dis}} L_{\text{dis}} + \lambda_{\text{mem}} L_{\text{mem}}, \\ \lambda_{\text{dis}}, \lambda_{\text{mem}} &= \text{Pareto\_Solver}(L_{\text{dis}}, L_{\text{mem}}). \end{aligned} \quad (6)$$

The details of Pareto_Solver can be referred in Sener and Koltun (2018).

3.7 Inference

For a given utterance x in D_{test}^t, we calculate the similarity between the extracted feature of x and all class prototypes $\{\hat{y}_i\}$ in the t-th task and pick the one with the highest cosine similarity:

$$y^* = \arg\max_{\hat{y}_i \in \{\hat{y}_i\}} \text{Sim}(f_\theta^S(x), \hat{y}_i). \quad (7)$$

The prototype of class $\hat{y}_i$ can be obtained by averaging the features of training samples labeled as y_i through the current trained PLE.

4 Experiments

4.1 CFID Benchmark

As for the first work in CFID, we first collect nine popular intent detection datasets and arrange them in a fixed random order to construct the benchmark: CLINC150, ATIS, HWU64, BANKING77, MTOP, SNIPS, LEYZER, MSLU, and TOP. For each dataset, we randomly select $K = 5$ or 10 samples per class as a 5-shot or 10-shot training set and select 10 samples per class as a development set. Details of nine datasets are reported in Table 1.

Dataset	#Domain	#Class	#Train	#Dev	#Test
CLINC150	10	150	750/1500	1500	4500
ATIS	1	14	70/140	121	827
HWU64	18	64	320/640	640	1076
BANKING77	1	77	385/770	770	3080
MTOP	11	85	425/860	850	4354
SNIPS	7	7	35/70	70	1429
LEYZER	15	57	285/570	469	381
MSLU	3	12	60/120	120	7799
TOP	2	11	55/110	110	8196

Table 1: The statistics of datasets (5-shot/10-shot).

4.2 Evaluation Protocol

Following (Geng et al., 2021), we run all methods with the same task ordering during training. The test accuracy of each task is reported after all tasks are visited.

At time step t, following (Mehta et al., 2021), we employ the average accuracy A_t, average forgetting F_t and learning accuracy LA_t metrics after learning on the t-th task. Let $a_{t,i}$ denote the test accuracy on the task i after learning task t, those metrics are defined as follows:

$$\begin{aligned} A_t &= \frac{1}{t}\sum_{i=1}^{t} a_{t,i} \quad LA_t = \frac{1}{t}\sum_{i=1}^{t} a_{i,i} \\ F_t &= \frac{1}{t-1}\sum_{i=1}^{t-1}\max_{j\in\{1,\ldots,t-1\}}(a_{j,i} - a_{t,i}). \end{aligned} \quad (8)$$

A_t measures the average performance over all previously seen tasks. F_t measures how much the model has forgotten about all previously seen tasks after learning task t. LA_t measures the learning capability when the model sees the new task.

To measure the parameter efficiency, we also employ the following metrics: trainable parameters and storage parameters after learning n tasks.

4.3 Compared Methods

Since this is the first work in CFID, there is no prior method that solves exactly the same task. We extend the typical methods in the few-shot ID setting to the CFID setting to construct the following strong baselines.

- **Lifelong Classifier (LC)** consists of a pre-trained backbone and a task-specific classification layer. Each task shares a backbone and owns its specific layer.

Task ID	0	1	2	3	4	5	6	7	8	Avg.
Method	CLINC150	ATIS	HWU64	BANKING77	MTOP	SNIPS	LEYZER	MSLU	TOP	
LC	10.32/10.55	38.09/43.65	33.18/31.78	45.13/45.47	56.51/57.16	78.61/72.92	90.81/92.21	94.13/94.97	82.83/86.49	58.85/59.47
L-DNNC	83.85/85.82	71.46/81.74	74.38/78.13	63.91/71.40	**80.91/85.24**	**93.52**/93.52	92.65/95.28	95.17/**97.27**	88.14/**90.93**	82.67/86.59
L-PN	77.93/85.67	73.28/89.68	71.90/79.18	58.97/71.67	80.90/84.55	91.07/93.00	**93.00**/95.36	95.93/96.72	**88.98**/90.92	81.33/87.42
PN-AGEM	79.73/86.24	79.60/88.51	72.83/78.81	61.21/72.82	80.79/84.24	90.97/94.29	92.65/**95.63**	**96.15**/96.62	88.14/90.69	82.45/87.54
PLE (Ours)	**88.70***/**91.20***	**87.91***/**91.29***	**76.46***/**80.36**	**74.90***/**79.09***	76.14/80.64	93.40/**94.56**	89.68/91.16	95.06/96.18	88.27/88.91	**85.61/88.16**
PN-Joint	89.04/93.19	84.52/90.08	76.58/84.08	76.61/84.19	78.72/86.28	92.93/95.89	92.56/95.10	91.52/96.30	87.15/89.25	85.52/90.49

Table 2: Test accuracy (%) evaluated on the final model in 5-shot/10-shot regime after all 9 tasks are visited. We use Avg. to represent the average accuracy of all tasks for each method. * indicate statistically significant ($p < .05$) improvements over the best baseline.

Method	Avg. A_t	Avg. F_t	Avg. LA_t
LC	70.00/70.54	20.91/26.09	85.21/89.62
L-DNNC	84.22/87.32	4.91/4.54	**87.69**/90.54
L-PN	82.26/88.95	3.16/3.48	83.87/**91.44**
PN-AGEM	83.76/**89.03**	3.11/3.05	85.97/91.21
PLE (Ours)	**84.73**/88.10	**1.16/1.03**	85.49/88.79
SC	86.62/90.69	0.00/0.00	86.62/90.69
S-PN	86.90/90.76	0.00/0.00	86.90/90.76
S-PLE	86.93/90.48	0.00/0.00	86.93/90.48
PN-Joint	85.05/90.27	0.80/0.65	85.41/90.52

Table 3: Performance of different methods in 5-shot/10-shot regime. We use Avg. to All metrics are averaged over all time steps in three trials.

- **Lifelong DNNC (L-DNNC)**. DNNC (Zhang et al., 2020) is one of the state-of-the-art methods in the few-shot ID task, which solves it as a textual entailment problem and uses large-scale entailment datasets for pre-training. L-DNNC tunes the whole DNNC model in a sequential manner when a new task arrives.

- **Lifelong Prototypical Network (L-PN)**. Prototypical Network (PN) (Snell et al., 2017) is also a strong distance-based baseline for few-shot ID tasks. Lifelong PN (LPN) tunes the whole PN model during lifelong learning.

- **PN-AGEM**. We also compared with a strong replay-based lifelong learning method called AGEM (Chaudhry et al., 2019). It needs to maintain a memory for storing selected samples from previous tasks. We apply it to the prototypical network and get a variant referred to as PN-AGEM.

- **PN-Joint** stores all data from all seen previous tasks and trains the whole prototypical network with all data when learning the new task. It serves as an upper bound of the prototypical network.

We also test those baselines in a single-task setting to measure the knowledge transfer ability.

- **Single Classifier (SC)** trains one classifier for each task. Obviously, it suffer from serious parameter explosion problem when the number of tasks increasing.

- **Single Prototypical Network (S-PN)** trains one prototypical network for each task. It also suffers from the parameter explosion problem.

- **Single PLE (S-PLE)** is an extension of our PLE model, which trains one adapter with one prefix individually for each task.

4.4 Implementation Details.

We use a pre-trained model $\mathrm{SimCSE}_{\mathrm{base}}$ as the backbone for all experiments, because of its powerful text representation capabilities. For classifier-based experiments, the batch size is 4 and 8 in the 5/10-shot setting respectively. For all experiments except those using the PLE, the learning rate is 2e-5. For PLE, it is 1e-4. For the replay-based baseline, the memory size is the same as ours. For experiments with episode training, we chose N and K for each task based on the maximum memory capacity and ensured that the same values were used for each experiment. For DNNC, we follow the settings in Zhang et al. (2020).

4.5 Main Results

In this part, we report the test accuracy of each task, referred to as "Overall Performance" and provide more insights into the catastrophic forgetting and average performance at each time step, referred to as "Middle States Performance".

Overall Performance As shown in Table 2, we report the experimental results of our approach and baselines. From the results, we can observe that: **1)** Our proposed PLE outperforms previous baselines concerning the average accuracy of all tasks (85.61% and 88.16% for 5-shot and 10-shot settings), which demonstrates the effectiveness of our

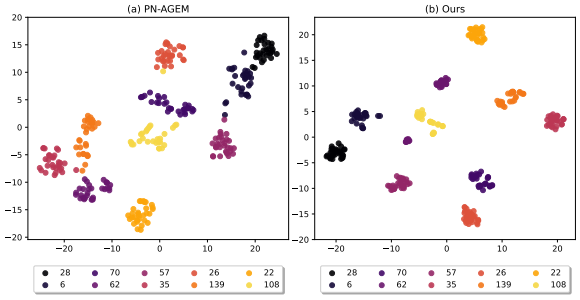


Figure 3: t-SNE visualization of PN-AGEM and Ours on the final model with test data of CLINC150. We randomly choose ten classes of the task to visualize.

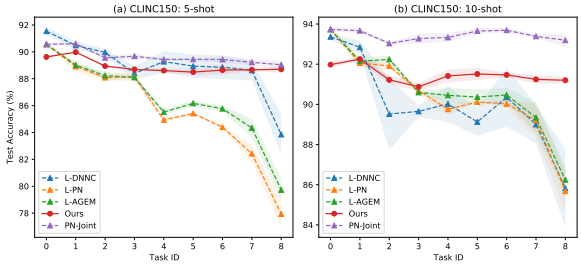


Figure 4: Test accuracy (%) of different methods on the CLINC150 dataset in 5-shot/10-shot CFID setting. Each curve denotes a kind of method. Shaded regions indicate standard deviation over three trials.

Method	#Trainable Params.	#Storage Params.
LC	125M+$\Delta\mathcal{T}_i$	125M+$\sum_{i=1}^{n}\Delta\mathcal{T}_i$
L-DNNC	125M	125M
L-PN	125M	125M
PN-AGEW	125M	125M
PLE (Ours)	**15M+15M=30M**	125M+15M=140M
SC	125M+$\Delta\mathcal{T}_i$	125M$\times n$ + $\sum_{i=1}^{n}\Delta\mathcal{T}_i$
S-PN	125M	125M$\times n$
S-PLE	15M	125M+15M$\times n$
PN-Joint	125M	125M

Table 4: Number of trainable and storage parameters in different methods. n denotes the number of tasks and $\Delta\mathcal{T}_i$ denotes the number of parameters of the task-specific layer. Here, the number of parameters of the PrLM and additional parameters of ours are 125M and 15M, respectively.

method. **2)** Our method still achieve better performance in the earlier accessed tasks, which demonstrates the superiority of our model in avoiding catastrophic forgetting.

In comparison, simply fine-tuning the backbone and the new involved classifier inevitably suffers from catastrophic forgetting. For example, the accuracy of LC on the first task is only 10.32%. We attribute it to the mismatch between the updated backbone and the classifier of the old task. For L-DNNC and L-PN, since they only have a shared encoder across tasks, catastrophic forgetting can be avoided. Thus they achieve 83.85% and 77.93% accuracy on the first task. Compared with L-PN, PN-AGEM is better on the early accessed tasks as it replays some samples of past tasks. As shown in Figure 3, compared with PN-AGEM, our PLE shows better intra-class compactness and larger inter-class distances.

For PN-Joint, it uses training data of all previously seen tasks at each step, which is more likely to be affected by negative knowledge transfer in a few cases. For tasks with very different domains, e.g., ATIS with flight domain, other tasks may transfer more negative knowledge to it. Thus, we can observe a worse performance than our method in this case (Accuracy of 90.08% vs. 91.29% on ATIS). It shows our effectiveness in alleviating this problem.

Middle States Performance As shown in Table 3, we report the average accuracy, forgetting, and learning accuracy of our method and baselines. All metrics are averaged over all time steps. From the results, we can observe that: Our proposed PLE outperforms previous baselines in the 5-shot CFID setting concerning Avg. A_t and Avg. F_t. It also has competitive performance in the 10-shot CFID setting and less forgetting than other baselines. For baselines in the single-task setting (i.e., SC, S-PN, and S-PLE), although they perform well, when the number of tasks is large, they inevitably suffer from parameter explosion.

However, there is a slight drop in Avg. LA_t in our method compared to others. A similar phenomenon can be observed in Table 2, i.e., for tasks newly visited, the test accuracy is not as good as other methods. We attribute it to a trade-off between learning about new tasks and preventing forgetting of past tasks. Freezing the backbone in our method damages the expressiveness but guarantees the overall performance of all tasks. Specifically, taking the earliest visited task CLINC150 as an example, Figure 4 shows the accuracy curves of the different methods throughout the continual learning. Compared with other methods, the performance of our method is relatively stable in the whole process, although the performance is not the best at the beginning.

Overall, our proposed PLE is a promising solution for the CFID problem with less forgetting and comparable performance.

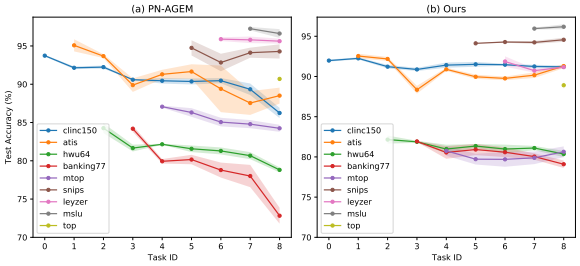


Figure 5: Test accuracy (%) of different tasks between PN-AGEM and Ours. Each curve in the sub-figure denotes a kind of task in the 10-shot setting. Shaded regions indicate standard deviation over three trials.

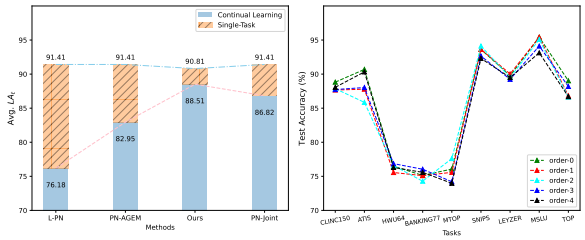


Figure 6: Left side: Avg. LA_t on ATIS in the 5-shot CFID and single-task settings, respectively. Right side: Test accuracy (%) of different tasks in the 5-shot setting with five different task orderings.

4.6 Efficiency & Robustness of PLE

Parameter Efficiency As shown in Table 4, we list the number of trainable and stored parameters in different methods. As the number of tasks increases, the stored parameters of the baselines in the single-task setting (i.e., SC, S-PN, and S-PLE) also increase, eventually leading to the explosion problem. Compared to them and other baselines, PLE achieves competitive performance and parameter efficiency with 76% less trainable parameters (from 125M to 30M) and only 15M additional parameter storage. As a result, it is possible to employ a larger pre-trained language model to achieve better performance.

Training and Inference Efficiency We observe that incorporating continuous prompts into PLE does not suffer from too slower training than other prototypical-based baselines (i.e., L-PN and PN-AGEM). However, for L-DNNC, despite its high performance in Avg. LA_t, it makes predictions by enumerating all the labels to decide whether a query and a label match or not, which is so time-consuming during training and inference.

Robustness for Task Ordering To analyze the effect of task ordering when PLE is learning different tasks, we randomly sample five different task orderings in the 5-shot setting. After all tasks are learned, we report the test accuracy over different orderings. As shown in the the right side of Figure 6, we can see our method is insensitive to different task orderings.

4.7 Knowledge Transfer Assessment

Assessing Backward Knowledge Transfer To assess the influence of learning new tasks on the performance of previous tasks (**backward transfer**), we visualize the curve of the test accuracy of different tasks between our method and the baseline with the lowest Avg. F_t of 3.05%, i.e., PN-AGEM in the 10-shot setting. As seen in the left side of Figure 5, the curve of PN-AGEM has a clear downward trend, while the curve of our method remains stable overall. In particular, we observe that on ATIS, the performance of PN-AGEM continues to decline, while our method goes through a phase of slight decline followed by an increase. It shows a promising ability to backward knowledge transfer.

Assessing Forward Knowledge Transfer To further assess the capability to learn new tasks with the help of knowledge from past tasks (**forward transfer**), we compare the results of Avg. LA_t in the single-task setting and continual learning setting. From Table 3, we observed that there is a significant drop in the extremely few-shot (i.e., 5-shot) regime. For example, there is a drop from 86.90 to 83.87 comparing S-PN and L-PN. In particular, there is still a drop of 1.49% compared to S-PN for the upper-bound baseline PN-Joint. It confirms the existence of negative knowledge transfer across tasks. We select the task ATIS to assess forward knowledge transfer, which is most affected by negative knowledge transfer (4% performance drop comparing PN-Joint with S-PN). From the left-top side of Figure 6, we can see our method has the highest Avg. LA_t and is closest to the performance in the single-task setting. It shows that our approach effectively reduces the effect of the negative knowledge transfer and enhances the effect of the forward knowledge transfer.

4.8 Analysis of Domain Variety

To simulate a realistic continual setting, we collect as many public datasets as possible, a few of which inevitably overlap in intent classes and domains, such as MSLU and MTOP. However, we count the number of similar domains in any two datasets and

Method	Avg. A_t	Avg. F_t	Avg. LA_t
Ours	**85.15/88.22**	0.99/0.83	85.79/88.75
w/o prefix	84.74/87.81	2.55/2.25	86.49/89.37
w/o memory	83.22/87.45	4.86/3.39	**86.55/89.83**
w/o TKT	82.35/87.25	**0.84/0.81**	82.84/87.82
w/o PSR	84.06/87.86	1.85/1.35	85.33/88.83
w/o DWR	84.63/87.88	1.11/1.43	85.63/88.59

Table 5: Ablation results in 5-shot/10-shot regime. All metrics are averaged over all time steps.

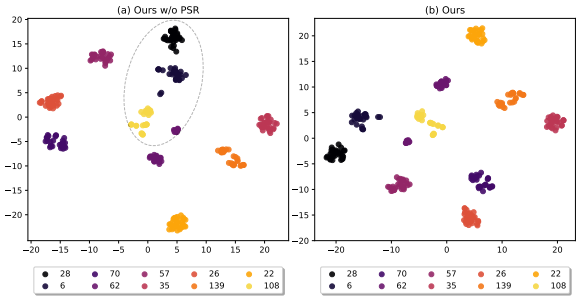


Figure 7: t-SNE visualization of ablation: w/o PSR and Ours on the final model with test data of CLINC150. The circle region shows our PLE has larger inter-class distances than the ablation variant.

find less than 1 domain overlap on average. There are also domain differences in some overlapping classes, such as the class "exchange_rate" in the "banking" and "travel" domain. It reflects the domain variety between tasks to some extent.

In general, our approach works well in very different domains. For example, the task ATIS belongs to the "flight" domain, which is different from all domains in another task, CLINC150. When continually learning ATIS after CLINC150, we observed there is a huge performance drop using L-PN baseline compared to S-PN in a single-task setting (accuracy of ATIS from 91% to 76%). This is mostly due to negative knowledge transfer caused by the domain gap between CLINC150 and ATIS. With the distillation strategy, our method alleviates this problem and achieves an accuracy of 89% when continually learning ATIS.

4.9 Ablation Study

As reported in Table 5, we conduct ablation studies to investigate the impact of different components of PLE in the 5-shot and 10-shot setting.

Specifically, we analyze the following variants: a) *w/o prefix* removes the prefix from the PLE. b) *w/o memory* removes the memory and merely adopting the TKT strategy. c) *w/o TKT* discards the TKT strategy and merely adopting the L_{sim} with the memory. d) *w/o PSR* randomly selects the same number of saved samples instead of using the PSR strategy. e) *w/o DWR* sets fixed weight hyper-parameters heuristically ($\lambda_{\text{dis}} = 0.9, \lambda_{\text{mem}} = 0.1$) instead of dynamically weighting.

From the results in Table 5, we can make the following observations. First, the introduction of prefix improves the performance of our method. Second, the variant without the TKT has a significant drop in performance on the Avg. A_t and Avg. LA_t. It confirms the existence of negative knowledge transfer across tasks. Using the TKT to gain more task-specific knowledge can effectively alleviate this problem. Also, the memory can effectively alleviate the catastrophic forgetting problem. Figure 7 shows that the PSR strategy is an efficient way to select saved samples.

Moreover, we observe that the Avg. LA_t and Avg. F_t are the highest in the "w/o memory" and "TKT" settings, respectively. It confirms a trade-off between learning new tasks and replaying past tasks. From the results in the "w/o DWR" setting, we can see the DWR strategy can effectively balance them and significantly improve the performance of our method.

5 Conclusion

In this paper, we define a more challenging yet practical problem as Continual Few-shot Intent Detection (CFID), where the system needs to handle continually emerging new intents with very few labeled data. To deal with the problem, we propose a novel prefix-guided lightweight encoder with three auxiliary strategies. Extensive experiments demonstrate the effectiveness and efficiency of our method in preventing catastrophic forgetting and encouraging positive knowledge transfer across tasks.

Acknowledgments

This work was supported by the NSFC projects (No. 62072399, No. U19B2042, No. 61402403), Chinese Knowledge Center for Engineering Sciences and Technology, MoE Engineering Research Center of Digital Library, Alibaba Group, Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, and the Fundamental Research Funds for the Central Universities (No. 226-2022-00070)

References

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with a-gem. In *ICLR*.

Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi, and Yanghua Xiao. 2021. Refining sample embeddings with relation prototypes to enhance continual relation extraction. 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*, pages 232–243. Association for Computational Linguistics.

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. Protaugment: Intent detection meta-learning through unsupervised diverse paraphrasing. 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*, pages 2454–2466. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Binzong Geng, Min Yang, Fajie Yuan, Shupeng Wang, Xiang Ao, and Ruifeng Xu. 2021. Iterative network pruning with uncertainty regularization for lifelong sentiment classification. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1229–1238. ACM.

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.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6429–6440. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *CoRR*, abs/2110.04366.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021a. Achieving forgetting prevention and knowledge transfer in continual learning. pages 22443–22456.

Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021b. CLASSIC: continual and contrastive learning of aspect sentiment classification tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6871–6883. Association for Computational Linguistics.

Zixuan Ke, Hu Xu, and Bing Liu. 2021c. Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks. 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*, pages 4746–4755. Association for Computational Linguistics.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. 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)*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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*, pages 4582–4597. Association for Computational Linguistics.

Qingbin Liu, Xiaoyan Yu, Shizhu He, Kang Liu, and Jun Zhao. 2021. Lifelong intent detection via multi-strategy rebalancing. *CoRR*, abs/2108.04445.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of

Findings of ACL, pages 2773–2782. Association for Computational Linguistics.

Shikib Mehri and Mihail Eric. 2021. Example-driven intent prediction with observers. 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*, pages 2979–2992. Association for Computational Linguistics.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tür. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *CoRR*, abs/2009.13570.

Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. 2021. An empirical investigation of the role of pre-training in lifelong learning. *CoRR*, abs/2112.09153.

Chengwei Qin and Shafiq Joty. 2022. Continual few-shot relation learning via embedding space regularization and data augmentation. *CoRR*, abs/2203.02135.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2078–2087. Association for Computational Linguistics.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021. A survey on spoken language understanding: Recent advances and new frontiers. In *IJCAI*.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 525–536. Curran Associates, Inc.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Chengyu Wang, Haojie Pan, Yuan Liu, Kehan Chen, Minghui Qiu, Wei Zhou, Jun Huang, Haiqing Chen, Wei Lin, and Deng Cai. 2021. Mell: Large-scale extensible user intent classification for dialogue systems with meta lifelong learning. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3649–3659. ACM.

Congying Xia, Wenpeng Yin, Yihao Feng, and Philip S. Yu. 2021. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. 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*, pages 1351–1360. Association for Computational Linguistics.

Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon M. Jose, Beibei Kong, and Yudong Li. 2021. One person, one model, one world: Learning continual user representation without forgetting. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 696–705. ACM.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y. S. Lam. 2021a. Effectiveness of pre-training for few-shot intent classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1114–1120. Association for Computational Linguistics.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip S. Yu. 2021b. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1906–1912. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5064–5082. Association for Computational Linguistics.