

# PersonaPKT: Building Personalized Dialogue Agents via Parameter-efficient Knowledge Transfer

Xu Han<sup>1</sup>, Bin Guo<sup>2</sup>, Yoon Jung<sup>2</sup>, Benjamin Yao<sup>2</sup>, Yu Zhang<sup>2</sup>, Xiaohu Liu<sup>2</sup> and Chenlei Guo<sup>2</sup>

<sup>1</sup>University of Colorado Boulder

<sup>2</sup>Amazon Alexa

xuha2442@colorado.edu, {guobg, yoonjun, benjamy, yzzhan, derecliu, guochenl}@amazon.com

## Abstract

Personalized dialogue agents (DAs) powered by large pre-trained language models (PLMs) often rely on explicit persona descriptions to maintain personality consistency. However, such descriptions may not always be available or may pose privacy concerns. To tackle this bottleneck, we introduce PersonaPKT, a lightweight transfer learning approach that can build persona-consistent dialogue models without explicit persona descriptions. By representing each persona as a continuous vector, PersonaPKT learns implicit persona-specific features directly from a small number of dialogue samples produced by the same persona, adding less than 0.1% trainable parameters for each persona on top of the PLM backbone. Empirical results demonstrate that PersonaPKT effectively builds personalized DAs with high storage efficiency, outperforming various baselines in terms of persona consistency while maintaining good response generation quality. In addition, it enhances privacy protection by avoiding explicit persona descriptions. Overall, PersonaPKT is an effective solution for creating personalized DAs that respect user privacy.

## 1 Introduction

Recent advances in large-scale pre-trained language models (PLMs) greatly boost the performance of chat dialogue agents (DAs) in generating understandable and fluent responses. However, a PLM-powered DA can potentially suffer from the lack of a consistent personality (Zhang et al., 2018; Li et al., 2016; Lian et al., 2022) since it is typically trained on dialogues collected from many different personas (i.e., *persona cross-contamination*). To address the issue, many approaches have been proposed to build more persona-consistent models by conditioning on explicit persona descriptions (Zhang et al., 2018; Wolf et al., 2019). These descriptions can steer the response generation and are usually presented

in the form of several sentences like "*I love the beach.*", "*I am on a diet now.*". However, such explicit persona statements are rarely available in practice: They require hand-crafted feature designs (Madotto et al., 2019) and are intractable to be directly extracted from real-world conversations or speakers (Zhang et al., 2018; Madotto et al., 2019; Lee et al., 2021a). Moreover, explicit persona statements may contain sensitive user information, thereby raising privacy concerns.

In light of this, we introduce PersonaPKT: **Persona**-based **Parameter**-efficient **Knowledge** Transfer, a lightweight transfer learning approach to build persona-consistent dialogue models without explicit persona descriptions. Inspired by the recent success in lightweight PLM-tuning approaches such as prefix-tuning (Li and Liang, 2021) and prompt-tuning (Lester et al., 2021), each persona is represented as a continuous vector (i.e., a *personalized prefix*) in PersonaPKT, adding less than 0.1% trainable parameters compared to a full PLM. PersonaPKT then prepends these personalized prefixes to the PLM with frozen weights to steer the response generation. Instead of utilizing explicit persona descriptions, each personalized prefix learns *implicit* persona-specific features directly from a small number of dialogues produced by the same persona (i.e., low data scenarios). To further improve the response generation quality under such low data scenarios, PersonaPKT first trains one *source prefix* over multiple personas' data agnostically, then uses the *source prefix* to initialize the training of the *personalized prefix* for a target persona (as in Fig 1). Through such initialization, PersonaPKT is able to transfer the knowledge learned from various personas to the target *personalized prefix* training, preventing a significant drop in generation quality due to limited personalized training data (Madotto et al., 2019). Empirical results show that PersonaPKT is able to build personalized dialogue models with high storage efficiency, out-

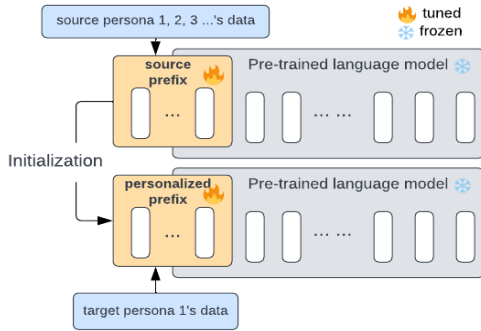


Figure 1: An overview of PersonaPKT.

performing various baselines in terms of persona consistency while maintaining good response generation quality. In addition, it enhances privacy protection by avoiding explicit persona descriptions.

In practice, PersonaPKT is advantageous in its modularity and extendability. PersonaPKT maintains a separate *personalized prefix* for each persona, making it easily scalable to batch process multiple users while ensuring privacy protection and eliminating the risk of cross-contamination among users’ data. Additionally, PersonaPKT’s two-stage transfer learning process allows the *source prefix* to be further optimized via different training strategies based on practical needs, showing its great extendability. In this work, we experimented with both data-level (temperature-scaled mixing) and model-level (meta-learning) optimization algorithms to train the *source prefix* to demonstrate the effectiveness of PersonaPKT. To the best of our knowledge, our work is the first on building personalized dialogue models via parameter-efficient knowledge transfer. As a result, our work makes three unique contributions:

- We introduce PersonaPKT, a lightweight transfer learning approach to build personalized dialogue models that respect user privacy.
- We show the great potential of PersonaPKT by further optimizing it via data-level and model-level optimization algorithms.
- We conduct large-scale empirical experiments to show that PersonaPKT builds better personalized dialogue models in terms of both persona consistency and storage efficiency while maintaining good response generation quality.

## 2 Related Work

### 2.1 Personalized Dialogue Generation

Previous studies have shown that when explicit persona descriptions are available, they can be encoded into memory networks (e.g., Zhang et al., 2018) or appended to dialogue histories to generate persona-consistent responses (e.g., Wolf et al., 2019). However, the utilization of explicit persona descriptions raises privacy concerns as it may require access to personal information. To address this issue, Madotto et al. (2019) introduced a persona-agnostic meta-learning (PAML) algorithm that enables the learning of persona-specific dialogue generation models without the need for explicit persona descriptions. Subsequently, several studies have explored this direction using various methodologies (Lee et al., 2021b; Yao et al., 2021; Wu et al., 2021). For example, Lee et al. (2021b) trained persona-specific models via multi-task meta-learning without any explicit persona descriptions. While the PAML algorithm and its follow-up work demonstrate the feasibility of training persona-consistent dialogue agents without explicit persona description, they still require modifying the entire language model parameters and storing a full copy of the pre-trained model for each persona. To address this limitation, our approach here provides a more storage-efficient solution for creating effective personalized DAs without the need for explicit persona descriptions.

### 2.2 Parameter-Efficient Knowledge Transfer

Instead of modifying all the language model parameters, efficient tuning approaches such as prompt-tuning (Lester et al., 2021) and prefix-tuning (Li and Liang, 2021) optimize a small number of trainable parameters (i.e., prompts/prefixes) for higher flexibility and storage efficiency. However, these parameter-efficient tuning methods occasionally suffer from low training efficiency (e.g., slower convergence rate (Su et al., 2022; Xie et al., 2022)) and limited model performance compared to full fine-tuning settings (Xie et al., 2022; Li and Liang, 2021). To tackle these issues, many studies have explored the extent to which these parameter-efficient approaches can perform knowledge transfer. Vu et al. (2022) pointed out that prompts/prefixes learned from one or more source tasks were transferrable and could be used to initialize the prompt/prefix for a target task. Su et al. (2022) further validated such findings and claimed

that knowledge transfer can accelerate parameter-efficient tuning methods and improve their performance. In this work, we adapt parameter-efficient knowledge transfer to personalized dialogue modeling, which to our knowledge is the first attempt in this area.

### 3 Methodology

#### 3.1 PersonaPKT

An overview of PersonaPKT is shown in Fig 1. Building on the parameter-efficient tuning approach (e.g., prefix-tuning<sup>1</sup> (Li and Liang, 2021)) which learns task-specific continuous vectors to condition a frozen pre-trained model, PersonaPKT considers learning different personas as different tasks. Specifically, PersonaPKT adds a trainable persona-specific vector  $p_\theta$  for each persona, which we call a *personalized prefix*, depicted by orange blocks in Fig 1. The *personalized prefix* is prepended to the task input and the backbone PLM can attend to it as if it were a sequence of "virtual tokens". During training, PersonaPKT only optimizes the personalized prefix while the backbone PLM remains frozen.

In PersonaPKT, a *personalized prefix* is trained on each persona’s dialogue data only. The limited dialogue data per persona will result in a low-data resource scenario, potentially leading to a significant performance drop in terms of dialogue generation quality. In light of this, PersonaPKT is novel in introducing *source prefix* tuning, an extra training stage before *personalized prefix* tuning (Fig 1). It first trains one *source prefix* over multiple personas’ data agnostically and then uses the *source prefix* to initialize the training of the *personalized prefix* for a target persona. Via such a two-stage transfer learning process, PersonaPKT is able to transfer the knowledge learned from various personas to the target prefix training, preventing the generation quality from dramatically dropping due to limited training data per persona.

#### 3.2 Optimizing the Training of Source Prefix

PersonaPKT’s two-stage learning process allows the *source prefix* to be further optimized via different training strategies. To validate and fully leverage the benefits of its two-stage learning process, we explored two specific optimization algorithms

<sup>1</sup>In our study, we utilized prefix-tuning. However, PersonaPKT can be adapted to any of the parameter-efficient tuning approaches, such as prompt-tuning.

in terms of data-level (temperature-scaled mixing) and model-level (meta-learning), to train the *source prefix*.

##### 3.2.1 Data-level optimization

PersonaPKT considers learning the *source prefix* as a multi-task learning process, which involves dialogue data from various personas. As pointed out by previous studies (Arivazhagan et al., 2019), it is important for multi-task learning to properly mix the data from each task the model should be trained on. We thus utilized temperature-scaled mixing, a common data mixing approach used by multilingual BERT (Devlin et al., 2018) and T5 (Raffel et al., 2020), ensuring that the *source prefix* was sufficiently trained on low-resource personas. We implemented temperature scaling with a temperature  $T$ , where the mixing rate of each persona’s data is raised to the power of  $\frac{1}{T}$  and then re-normalized so that they sum to 1.

##### 3.2.2 Model-level optimization

Madotto et al. (2019) proposed a PAML algorithm to learn personalized DA models without any explicit persona descriptions. Inspired by PAML, we employed meta-learning algorithms to further optimize the training of *source prefix*. Specifically, we adapted Reptile (Nichol and Schulman, 2018), a widely-used meta-learning algorithm, into our parameter-efficient-tuning-based setting. The adapted algorithm, which we refer to as **Parameter-efficient Persona-agnostic Reptile** (PPReptile), is described in Algorithm 1. We further present algorithm details as follows.

We define the persona dataset as  $\mathcal{D} = \{P_1, P_2, \dots, P_z\}$ , where  $z$  is the number of personas in  $\mathcal{D}$ .  $\mathcal{D}$  is further split into  $\mathcal{D}_{train}$ ,  $\mathcal{D}_{valid}$ ,  $\mathcal{D}_{test}$ . PPREptile first randomly samples  $n$  personas  $\rho_1, \rho_2, \dots, \rho_n$  from  $\mathcal{D}_{train}$ . With pre-trained parameters  $\theta_{pretrain}$  and randomly initialized weights  $\theta_{prefix}$ , PPREptile then updates the dialogue model  $f_{\theta=\theta_{pretrain} \cup \theta_{prefix}}$  with

$$\theta_{prefix} \leftarrow \theta_{prefix} + \beta \frac{1}{n} \sum_{i=1}^n (W_i - \theta_{prefix}) \quad (1)$$

, where the gradient  $W_i$

$$W_i = \text{SGD}(\mathcal{L}_{\rho_i}, \theta_{prefix}, \alpha, k) \quad (2)$$

. Here,  $\beta$  in Equation 1,  $\alpha$  in Equation 2 denotes the inner and outer learning rate respectively.  $k$  in Equation 2 denotes  $k$  steps of SGD

on  $\rho_i$ .  $\mathcal{L}_{\rho_i}$  is from  $L = \{\mathcal{L}_1, \mathcal{L}_1, \dots, \mathcal{L}_z\}$ , which is the set of loss functions corresponding to all  $\rho_i$  in  $\rho_1, \rho_2, \dots, \rho_n$ . Specifically, we use the cross-entropy loss for our response generation task.

The main difference between PPreptile and vanilla Reptile is that PPreptile updates prefix parameters only, even though the gradient computation still relies on both pre-trained and prefix parameters.

---

**Algorithm 1** Parameter-efficient Persona-agnostic Reptile

---

**Require:** model  $f_{\theta=\theta_{pretrain}\cup\theta_{prefix}}$   
 $\mathcal{D}_{train} = \{\rho_1, \rho_2, \dots, \rho_z\}$   
 $\alpha, \beta$ : learning rates  
 $L = \{\mathcal{L}_1, \mathcal{L}_1, \dots, \mathcal{L}_z\}$ : set of loss functions corresponding to all potential personas  
 $k$ : inner step number  
 $n$ : persona batch size  
1: **for** iteration in  $[1, 2, \dots]$  **do**  
2:   Sample  $n$  personas  $\rho_{\{1,2,\dots,n\}} \sim \mathcal{D}_{train}$ ,  
3:   **for**  $i$  in  $[1,2,\dots,n]$  **do**  
4:      $W_i = SGD(\mathcal{L}_{\rho_i}, \theta_{prefix}, \alpha, k)$   
5:   **end for**  
6:    $\theta_{prefix} \leftarrow \theta_{prefix} + \beta \frac{1}{n} \sum_{i=1}^n (W_i - \theta_{prefix})$   
7: **end for**

---

## 4 Experiment and Results

### 4.1 The task of personalized dialogue generation

The task of personalized dialogue generation aims to build dialogue models that are able to generate personalized utterances as response to the input utterance in the context of given dialogue histories. The generated response is expected to not only have good generation quality but also contain information that is consistent with the desired personas. Desired personas are usually provided in the form of several sentences as described in Section 1. In our study, we explore building dialogue models with PersonaPKT for both regular and few-shot personas (i.e., personas with less than 6 dialogues, more details in section 4.2.1).

### 4.2 Experiment setup

#### 4.2.1 Dataset

Our experiments are conducted using PERSONA-CHAT (Zhang et al., 2018), a widely-used conversational dataset that contains persona-based dia-

Table 1: Statistics of dataset

	# of personas	Number of dialogues		
		Train	Validation	Test
Part A <sup>1</sup>	754	5471	774	774
Part B <sup>2</sup>	300	2166	304	304
Part C <sup>3</sup>	239	538	239	239

<sup>1</sup> Part A: for *source prefix* training

<sup>2</sup> Part B: for *personalized prefixes* training with regular personas

<sup>3</sup> Part C: for *personalized prefixes* training with few-shot personas

logues. Following Madotto et al. (2019), we first match all dialogues in PERSONA-CHAT by their persona descriptions. After examining the distribution of the number of dialogues per persona (Fig 3 in Appendix A.1), we define a few-shot persona if the number of dialogues of that persona is smaller than 6. We had 239 few-shot personas and 1054 regular personas in total. We then randomly set aside 300 regular personas. The remaining 754 regular personas are used as the dataset for the *source prefix* training (Part A). The 300 regular personas (Part B) and 239 few-shot personas (Part C) are used to train target *personalized prefixes*. There were no overlapped personas among Part A, Part B, and Part C. Within each persona, train, valid and test set are randomly created by dialogue numbers following the ratio of 8:1:1. Table 1 are the basic statistics of our newly-split dataset.

#### 4.2.2 Evaluation Metrics

**Automated evaluation** We report F1 score of the generated responses against the human-generated target, which is the standard metric used for PERSONA-CHAT. F1 score can reflect the response quality (Madotto et al., 2019). For persona consistency, we report a widely used consistency metric called  $C$  score, which was defined by Madotto et al. (2019). They first trained a BERT-based Natural Language Inference (NLI) model to automatically generate NLI annotation between persona descriptions  $p_j$  and dialogue utterances  $u$  (as Formula (3)).

$$NLI(u, p_j) = \begin{cases} 1 & \text{if } u \text{ entails } p_j \\ 0 & \text{if } u \text{ is independent to } p_j \\ -1 & \text{if } u \text{ contradicts } p_j \end{cases} \quad (3)$$

. Based on  $NLI(u, p_j)$ , the persona consistency score  $C$  is then defined as below.

$$C(u) = \sum_j^m NLI(u, p_j) \quad (4)$$

. The BERT-based NLI model was trained on Dialog NLI (Welleck et al., 2019) dataset which

achieved a test set accuracy of 88.43%. In Formula (4),  $m$  is the number of sentences in the explicit persona descriptions.  $C$  score is shown to be aligned with human-evaluated consistency (Madotto et al., 2019) and a higher  $C$  score means having a more persona-consistent dialogue response. In addition, we report trainable parameter sizes to reflect the storage efficiency of each experiment setting.

**Human evaluation** We also conduct a human evaluation on 377 generated response examples from 50 randomly selected personas in Part B to complement the automatic evaluation results. In accordance with the guidelines provided by Madotto et al. (2019), we requested crowd-sourced workers to assess the fluency (response quality) and persona consistency of the generated response with respect to the dialogue histories and explicit persona descriptions. Specifically, the workers were instructed to rate the response fluency on a 5-point Likert scale ("1 (Not at all)" to "5 (To a great extent)"). For persona consistency, they were required to assign a score of -1, 0, or 1 for *contradicts*, *neutral*, or *consistent*, respectively. To ensure the annotation quality, we adopted the following strategies: (1) we only recruited crowd-sourced workers who had an approval rate greater than or equal to 99% while being located in the United States; (2) two additional annotators further validated the crowd-sourced annotations independently. Following their validation, they resolved any annotation conflicts through discussion or by taking the average score as the final decision.

#### 4.2.3 Implementation details

We implemented PersonaPKT using GPT2<sup>2</sup> (Radford et al., 2019) as the backbone PLM, which has around 345M parameters. A persona-agnostic *source prefix* was first trained with Part A’s data. Then we trained *personalized prefixes* for each persona on part B and C. In order to stabilize the training of prefixes, we followed Li and Liang (2021) and employed their parametrization strategy with  $k = 512$  (the number of persona-specific parameters is less than 0.1% of the total GPT2 parameters with  $k = 512$ ). When optimizing the training of the *source prefix*, we used  $T = 10$  for temperature-scaled mixing. For meta-learning-based optimization,

<sup>2</sup>We only used GPT2 as a testbed to explore the effectiveness of PersonaPKT. PersonaPKT can be adapted to different PLM backbones.

we used learning rates of  $\alpha = 10^{-4}$ ,  $\beta = 3 \times 10^{-5}$ , and batch sizes of  $b_{in} = 2$ ,  $b_{out} = 4$  for the inner, outer loops, respectively. When training target *personalized prefixes*, we tuned batch size and learning rate with early stopping on each persona’s validation set. During model training, we used the AdamW optimizer (Loshchilov and Hutter, 2019) and a linear learning rate scheduler for all the models. We also used beam search with a beam size of 5 when decoding.

### 4.3 Experiment Settings

We compare the following 8 training settings:

- **Persona + Fine-tuning:** A GPT2 model fine-tuned on Part A and evaluated on the test set of Part B and C. Explicit persona descriptions were appended to the input utterance during both training and inference. Although this setting utilized explicit persona descriptions, which is different from the underlying assumptions of PersonaPKT (i.e., without explicit persona descriptions), we still include this setting as a point of reference.
- **Fine-tuning:** Without any explicit persona descriptions as input, this setting fine-tuned a GPT2 model on Part A in a persona-agnostic manner and evaluated on the test set of Part B and C;
- **Persona\_id + Fine-tuning:** A GPT2 model fine-tuned on Part A and evaluated on the test set of Part B and C. Same as Persona + Fine-tuning while only the persona id was appended to the input utterance instead of the explicit persona descriptions;
- **Reptile + Fine-tuning:** Multiple persona-specific GPT2 models fine-tuned for each persona respectively in Part B and C. Each persona-specific GPT2 model was trained on dialogues produced by that persona only. Each personalized GPT2 model was initialized using a GPT2 model, which was fine-tuned in a persona-agnostic manner on Part A with the Reptile algorithm;
- **Rand init + Prefix-tuning:** *Personalized prefixes* were trained for each persona respectively in Part B and C. Each *personalized prefix* was trained on dialogues produced by that persona only. Prefix weights were randomly initialized;

Table 2: Results of automatic evaluation. Significantly underperforming settings are highlighted with  $\times$ .

		Automatic Metrics								Trainable Parameter Size $\downarrow$ <sup>2</sup>
		1-gram F1 $\uparrow$		2-gram F1 $\uparrow$		LCS F1 $\uparrow$ <sup>1</sup>		C Score $\uparrow$		
		Part B <sup>3</sup>	Part C <sup>4</sup>	Part B	Part C	Part B	Part C	Part B	Part C	
With descriptions	Persona + Fine-tuning <sup>5</sup>	<b>18.64</b>	<b>17.42</b>	<b>6.76</b>	5.68	<b>17.69</b>	<b>16.50</b>	0.20	0.23	1 * 100%
Without descriptions	Fine-tuning	18.16	17.18	6.27	<b>5.86</b>	17.30	16.22	<u>-0.0082</u> $\times$	<u>0.0076</u> $\times$	1 * 100%
	Persona_id + Fine-tuning	17.87	16.84	5.98	5.45	16.89	15.88	<u>0.00</u> $\times$	<u>0.00</u> $\times$	1 * 100%
	Reptile + Fine-tuning	16.30	15.40	4.44	3.78	15.20	14.22	0.27	0.21	(N+1) * 100% $\times$
	Rand init + Prefix-tuning	<u>9.36</u> $\times$	<u>8.12</u> $\times$	<u>1.34</u> $\times$	<u>1.21</u> $\times$	<u>6.74</u> $\times$	<u>4.98</u> $\times$	<b>0.28</b>	<b>0.26</b>	N * 0.1%
	PersonaPKT (base)	16.35	14.91	4.77	3.96	15.39	13.89	0.14	0.11	(N+1) * 0.1%
	PersonaPKT (temperature)	16.40	16.38	4.53	4.66	15.41	15.15	0.15	0.12	(N+1) * 0.1%
	PersonaPKT (PPReptile)	16.25	15.30	4.69	3.64	15.17	14.01	0.21	0.16	(N+1) * 0.1%

<sup>1</sup> LCS F1 denotes the longest common subsequence F1;

<sup>2</sup> N in this column represents the number of personas;

<sup>3</sup> Part B for regular personas while part C for few-shot personas;

<sup>5</sup> The model is trained with explicit persona descriptions while other models not.

- **PersonaPKT (base):** *Personalized prefixes* were trained for each persona respectively in Part B and C. Each *personalized prefix* was trained on dialogues produced by that persona only. Each personalized prefix was initialized using a *source prefix*, which was trained in a persona-agnostic manner on Part A;
- **PersonaPKT (temperature):** *Personalized prefixes* were trained for each persona respectively in Part B and C. Each *personalized prefix* was trained on dialogues produced by that persona only. Each personalized prefix was initialized using a *source prefix*, which was trained in a persona-agnostic manner on Part A’s data after temperature-scaled mixing;
- **PersonaPKT (PPReptile):** *Personalized prefixes* were trained for each persona respectively in Part B and C. Each *personalized prefix* was trained on dialogues produced by that persona only. Each personalized prefix was initialized using a *source prefix*, which was trained in a persona-agnostic manner with PPReptile in Algorithm 1 using Part A’s data.

#### 4.4 Results

Automatic and human evaluation results are presented in Table 2 and Table 3, respectively. Overall, PersonaPKT outperforms various baselines in terms of persona consistency in both automatic and human evaluation metrics. When explicit persona descriptions are not available, PersonaPKT is capable of achieving significantly higher persona consistency compared to fine-tuning baselines in both

Table 3: Results of human evaluation. Significantly underperforming settings are highlighted with  $\times$ .

		Human Metrics	
		Fluency	Persona Consistency
With descriptions	Persona + Fine-tuning	3.59	0.18
Without descriptions	Fine-tuning	3.68	<u>0.00080</u> $\times$
	Persona_id + Fine-tuning	3.69	<u>-0.024</u> $\times$
	Reptile + Fine-tuning	3.62	<b>0.22</b>
	Rand init + Prefix-tuning	<u>2.49</u> $\times$	0.12
	PersonaPKT (base)	<b>3.71</b>	0.17
	PersonaPKT (temperature)	3.46	0.20
	PersonaPKT (PPReptile)	3.42	0.20

automatic and human evaluation metrics, regardless of the optimization strategies (**PersonaPKT** vs. **Fine-tuning**, **Persona\_id+Fine-tuning**). Moreover, **PersonaPKT (PPReptile)** achieves even higher human-annotated persona consistency than the fine-tuning baseline in scenarios where explicit persona descriptions are available (**PersonaPKT** vs. **Persona+Fine-tuning**).

For response quality, the comparison between **PersonaPKT** and **Rand init + Prefix-tuning** further shows the effectiveness of using a *source prefix* to maintain good response generation quality. While PersonaPKT has lower F1 scores than other fine-tuning baselines, **PersonaPKT (base)** has achieved the highest human-annotated fluency compared to other baselines. This finding aligns with previous studies that indicate F1 measures are not highly correlated with human judgments (Madotto et al., 2019; Liu et al., 2016). For completeness, we show generated response examples from Per-

sonaPKT and baseline models in Appendix A.2.

In addition, PersonaPKT finds a good trade-off between storage efficiency and model performance. Although PersonaPKT performs slightly worse than the **Reptile + Fine-tuning** baseline in terms of both automatic and human-annotated persona consistency, its storage efficiency is considerably higher, resulting in more practical utility. As shown in Table 2, all our findings in terms of the automatic metrics can be generalized to few-shot personas as well (Part C).

In conclusion, although PersonaPKT doesn’t achieve the highest score in any of the evaluation metrics across the board except human-annotated fluency (Table 2, Table 3), it overcomes the limitations of all the baseline models, avoiding significantly poor performances in neither response quality, persona consistency nor storage-efficiency (significantly underperforming metrics under each setting were highlighted with **X**).

#### 4.5 Ablation Study

In the ablation study, we further evaluate the impact of different PersonaPKT optimization approaches. Specifically, we study how fast the *source prefix* can be adapted to a certain persona when training *personalized prefix*. As shown in Fig 2, we compared **PersonaPKT (base)**, **PersonaPKT (temperature)** and **PersonaPKT (PPReptile)** in terms of their *C* score with controlled numbers of epochs. The experiment was conducted on 100 randomly-selected personas from Part B.

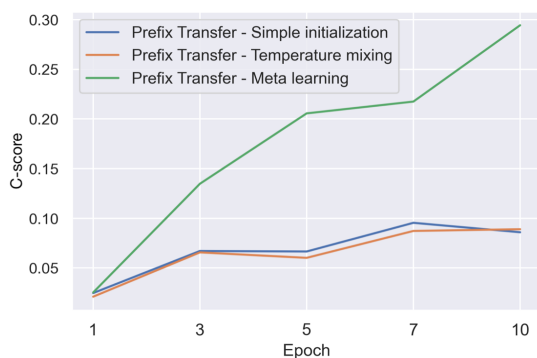


Figure 2: Ablation study results: how fast the *source prefix* can be adapted to a certain persona with different PersonaPKT optimization strategies.

As shown in Fig 2, **PersonaPKT (PPReptile)** has the best performance: it can achieve the highest *C* score with the fewest training epochs, demonstrating the effectiveness of utilizing meta-learning

to train the *source prefix*. In contrast, Table 2 and Table 3 reveal that **PersonaPKT (base)** and **PersonaPKT (temperature)** have better response quality than **PersonaPKT (PPReptile)**. These observations highlight the extendability of PersonaPKT’s two-stage transfer training process, which enables the source prefix to be further optimized via various training strategies. Such extendability is valuable in practice as it indicates that engineers can choose or even propose customized optimization strategies to train the source prefix based on their specific needs. For example, if their products prioritize persona consistency over F1 score, PPR reptile could be a suitable choice for optimization.

## 5 Limitations

### 5.1 Automatic metrics vs. Human annotation

**F1 vs. fluency.** As discussed in section 4.4, we have observed that the F1 score cannot be well-correlated with human annotations across almost all experiment settings. This is mostly due to the inflexibility of computing the F1 score solely based on the similarity between the generated responses and golden references. Such observations echo findings from previous studies (Madotto et al., 2019; Liu et al., 2016; Deutsch et al., 2022). In this work, we still utilize F1 score since it’s a standardized metric for PERSONA-CHAT evaluation. However, future work should aim to find a better evaluation metric that can more comprehensively reflect the quality of the generated responses.

***C* score vs. human-annotated consistency.** We observed a significant discrepancy between the *C* score and human-annotated consistency under the setting of Rand init + Prefix-tuning. Upon further analysis, we discovered that many responses generated from this setting contain repeated sentences with persona-consistent keywords (e.g., as shown in Table 4, Rand init + Prefix-tuning repeated "he is a preacher" multiple times). While human annotators tend to ignore these repeated sentences, the *C* score considers them as highly consistent. Our observations indicate that although the *C* score has been shown to be a good indicator of persona consistency (Madotto et al., 2019), it is still limited under certain experiment settings like Rand init + Prefix-tuning. Similar to the F1 score, a better evaluation metric is needed in the future to more comprehensively reflect the persona consistency of

generated responses.

## 5.2 Hyperparameter Selection

One limitation of PersonaPKT is its sensitivity to hyperparameters. Since PersonaPKT maintains persona-specific prefixes, this modularity of PersonaPKT allows flexibility in optimizing hyperparameters for each persona, such as learning rate, batch size, etc. Despite the advantage, our experiments show that the performance of PersonaPKT is more sensitive to hyperparameters compared to baseline models trained over multiple personas' data agnostically (Persona + Fine-tuning, Fine-tuning, Persona\_id + Fine-tuning). At the same time, persona-specific baseline models trained on a small number of dialogues produced by a single persona (Reptile + Fine-tuning, Rand init + Prefix-tuning) demonstrate similarly high sensitivity to hyperparameters as PersonaPKT. Such observations indicate that this high sensitivity is a common issue in persona-specific models like PersonaPKT. Strategies for improving their robustness to hyperparameters can be a potential study area in the future.

## 6 Discussion

In practice, PersonaPKT is advantageous in its modularity and extendability. Due to its high storage efficiency, PersonaPKT is advantageous when there are a large number of user-specific models that need to be maintained independently. PersonaPKT offers not only scalability to batch process multiple users, but also enhances user privacy by avoiding cross-contamination between different users' data. Moreover, PersonaPKT's great extendability allows engineers to adopt various source prefix optimization strategies, parameter-efficient tuning approaches or even PLM backbones based on practical needs. Lastly, it enhances privacy protection by avoiding the use of explicit persona descriptions. All of these advantages of PersonaPKT make it a valuable contribution to personalized DA training in the industry.

## 7 Conclusion

We present PersonaPKT, a lightweight transfer learning approach for building persona-consistent dialogue models without the need for explicit persona descriptions. By representing each persona as a continuous vector, PersonaPKT learns implicit persona-specific features directly from dialogue

samples produced by the same persona, with less than 0.1% trainable parameters added for each persona on top of the PLM backbone. Its two-stage learning process provides training flexibility, allowing for various optimization strategies to further enhance the training of the source prefix. PersonaPKT offers potential in terms of privacy protection and batch processing of multiple users. Future work will explore different optimization strategies and generalize PersonaPKT to additional applications.

## References

- Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.
- Daniel Deutsch, Rotem Dror, and Dan Roth. 2022. Re-examining system-level correlations of automatic summarization evaluation metrics. *arXiv preprint arXiv:2204.10216*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jing Yang Lee, Kong Aik Lee, and Woon Seng Gan. 2021a. Generating personalized dialogue via multi-task meta-learning. *arXiv preprint arXiv:2108.03377*.
- Jing Yang Lee, Kong Aik Lee, and Woon Seng Gan. 2021b. [Generating personalized dialogue via multi-task meta-learning](#). In *Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Potsdam, Germany. SEMDIAL.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. [A persona-based neural conversation model](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.
- 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 (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.



- Ruixue Lian, Che-Wei Huang, Yuqing Tang, Qilong Gu, Chengyuan Ma, and Chenlei Guo. 2022. Incremental user embedding modeling for personalized text classification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7832–7836. IEEE.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. [How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. [Personalizing dialogue agents via meta-learning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, Florence, Italy. Association for Computational Linguistics.
- Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2022. [On transferability of prompt tuning for natural language processing](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969, Seattle, United States. Association for Computational Linguistics.
- Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou’, and Daniel Cer. 2022. [SPoT: Better frozen model adaptation through soft prompt transfer](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.
- 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.
- Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. [Transfertransfo: A transfer learning approach for neural network based conversational agents](#). *CoRR*, abs/1901.08149.
- Yuwei Wu, Xuezhe Ma, and Diyi Yang. 2021. [Personalized response generation via generative split memory network](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1956–1970, Online. Association for Computational Linguistics.
- Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.
- Huaxiu Yao, Long-Kai Huang, Linjun Zhang, Ying Wei, Li Tian, James Zou, Junzhou Huang, et al. 2021. Improving generalization in meta-learning via task augmentation. In *International Conference on Machine Learning*, pages 11887–11897. PMLR.
- 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.

## A Appendix

### A.1 Distribution of the number of dialogues per persona (Fig 3)

### A.2 Generated response examples (Table 4)

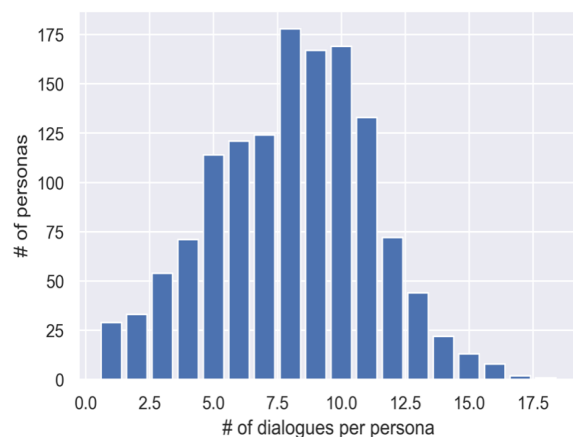


Figure 3: Distribution of the number of dialogues per persona in PERSONA-CHAT

Table 4: Generated examples

<b>Persona</b>	
my favorite color is purple i am a 1st grade teacher i have a 3 year old my father is a preacher i go to church on sunday	
<b>Dialogue history</b>	
Speaker 1: well i do not have any kids yet just four cats . Speaker 2: i would love to see a purple cat ! its my favorite color Speaker 1: oh my favorite color is blue . any hobbies ?	
<b>Target utterance</b>	
Speaker 2: not really . i really love church though , maybe because my dad is a preacher . you ?	
<b>Generated Responses</b>	
Persona + Fine-tuning:	i like to go to church on sunday.
Fine-tuning:	i like to go to the beach.
Persona_id + Fine-tuning:	i like to play video games.
Reptile + Fine-tuning:	i run an tumblr blog dedicated to purple things it is my favorite color.
Rand init + Prefix-tuning:	with my dad. he is a preacher. i am a preacher. i am a preacher. i am a preacher...
PersonaPT (base):	i go to church every sunday at sunday
PersonaPT (temperature):	i do not have any hobbies, i am a teacher. how about you?
PersonaPT (PPReptile):	i do not have any hobbies at all. i am a teacher