# Enhancing LLM's Dialogue State Tracking Performance via a Novel LoRA Finetuning Method

**Anonymous ACL submission**

## Abstract

The rapid development of large language models (LLMs) has increasingly positioned them as crucial components in task oriented dialog (TOD), enabling more flexible task completion. However, the substantial size of LLMs incurs significant resource consumption during full-parameter fine-tuning. Against this backdrop, parameter-efficient fine-tuning methods have garnered attention, with LoRA being particularly noteworthy. However, LoRA is not without limitations; it overlooks the varying importance of different weight parameters. Inspired by LoRA, we introduce a novel importance assessment method, Sensitivity Under Cooperative Game (SUCG), which is applied to the Dialogue State Tracking (DST) module within TOD for task evaluation. Extensive experiments have validated that our innovation effectively enhances model performance and efficiency in natural language processing. This work provides new insights for the future development of the DST module.

## 1 Introduction

Advancements in natural language processing (NLP) and the enhancement of computing resources have driven remarkable progress in task oriented dialog (TOD) systems, such as in the realms of intelligent virtual assistants, customer service, hotel reservations, and so on. With the continuous emergence of various large language models (LLMs) and the growing capabilities of them, the application of LLMs in TOD has become increasingly in depth (Zhang et al., 2023b; Chung et al., 2023; Xu et al., 2024; Kazi et al., 2024).

A typical pipeline architecture TOD system comprises four key modules: Natural Language Understanding (NLU), Dialogue State Tracking (DST), Dialogue Policy (DP), and Natural Language Generation (NLG) (Zhang et al., 2020; Qin et al., 2023). Among these, the DST module is of particular significance as it tracks the dialogue state through interactions between the user and the system (Yang et al., 2023), ensuring stable and reliable operation of the TOD system.

However, when a full-parameter fine-tuning of LLMs was performed in TOD, it was found that the large size of the LLMs consumed a significant amount of computational resources, resulting in substantial deployment costs. After complete fine-tuning, LLMs can only adapt to a single task or some specific tasks, leading to lack of flexibility and exponential growth in deployment costs. For example, a V100 GPU has 16GB graphics memory, which can only deploy one instance of a 7B model. Moreover, this deployment is only applicable for inference purposes and is insufficient to perform full-parameter fine-tuning operations.

To address these issues, parameter-efficient fine-tuning methods have been proposed. One of the most notable is the Low-Rank Adaptation (LoRA) technique (Hu et al., 2022). LoRA freezes the pre-trained model weights and injects trainable rank-decomposition matrices into each layer of the transformer architecture, effectively reducing the number of trainable parameters for downstream tasks, and theoretically, it does not increase the inference latency. Despite its advantages, LoRA has limitations in practical applications. It evenly allocates the budget for incremental updates, overlooking the varying importance of different weight parameters. This not only results in suboptimal fine-tuning performance but also causes problems such as poor adaptability in complex tasks, limited applicability to certain model architectures, instability in low-resource scenarios, and bottlenecks in performance improvement.

Against this backdrop, this paper focuses on improving the fine-tuning of LLMs for the DST module in TOD systems. Based on the DST research paradigm in Zhu et al. (2022), we introduce an improved method. We innovately improve the way of evaluating the importance of LoRA

parameters. Using the frameworks of AdaLoRA (Zhang et al., 2023a) and AutoLoRA (Zhang et al., 2024b), which are improvements to LoRA, we incorporate the concept of the Shapley value to innovate calculate the expected gradient. This leads to the development of a new importance evaluation method named Sensitivity Under Cooperative Game (SUCG). Our method takes into account different combinations of single-rank LoRAs during gradient computation and integrates this approach into the AdaLoRA framework, aiming to enhance the accuracy and efficiency of the LoRA parameter importance evaluation.

In summary, our research presents innovative methods for optimizing LoRA in the context of the DST module in TOD. Through fine-tuning models like Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct, and DeepSeek-R1-Distill-Qwen-1.5B, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct in ablation studies, significant progress has been achieved. Experimental results demonstrate that the SUCG method effectively enhances the performance and efficiency of models in natural language processing tasks. Specifically, the experimental success rate has increased from [X]% to [X]%, and the [Specific Metric] rate has improved from [X]% to [X]%. This indicates that the SUCG method designed for the DST module in TOD is highly efficient and brings about remarkable improvements, offering a more effective solution for relevant DST module research and applications.

## 2 Related Work

### 2.1 LLM Research on TOD

TOD systems aim to assist users in completing specific tasks within certain domains, such as restaurant reservation, car rentals, and flight bookings. This makes them highly valuable for real-world business applications (Zhang et al., 2020). Early TOD systems were largely based on sequence-to-sequence models (Gao et al., 2022). However, these sequence-to-sequence models rely heavily on large amounts of training data, and their performance drops significantly when data is scarce.

With breakthroughs in LLM technology, such as GPT-4 (OpenAI et al., 2024), o3-mini (Arrieta et al., 2025), QWen2.5 (Yang et al., 2024), DeepSeek V3 (DeepSeek-AI et al., 2024) and R1 (DeepSeek-AI et al., 2025), researchers have begun to explore the potential of LLM in TOD systems (Yi et al., 2024). A typical TOD system consists of four modules: Natural Language Understanding (NLU), Dialogue State Tracking (DST), Dialogue Policy (DP), and Natural Language Generation (NLG) (Ohashi and Higashinaka, 2022; Yoshimaru et al., 2023; Xu et al., 2024). DST plays a crucial role in the TOD system by tracking the dialogue state through user-system interactions, providing stable and reliable operation for the entire system, allowing accurate understanding of user intents, and ensuring smooth progress of task completion processes (Yang et al., 2023). The emergence of LLMs has brought new opportunities to DST, as they can reduce the dependence on annotated data and infer undefined slots based on common sense.

### 2.2 Fine-Tuning LLMs

In the ongoing evolution of the LLM field, model fine-tuning techniques have emerged as a significant subdiscipline within research and application. Parameter-Efficient Fine-Tuning (PEFT) represents an empirical approach that uses prompts to guide the model in performing specific tasks (Han et al., 2024). This methodology encompasses a variety of techniques, including BitFit, Prefix Tuning, Prompt Tuning, P-Tuning, P-Tuning v2 and LoRA, each designed to efficiently adapt large language models to diverse tasks with minimal additional parameterization (Ben Zaken et al., 2022; Li and Liang, 2021; Lester et al., 2021; Liu et al., 2024, 2022; Hu et al., 2022). LoRA and its derivatives, such as AdaLoRA and AutoLoRA, have made significant advancements in the field of parameter fine-tuning (Zhang et al., 2023a, 2024b). However, these methods have limitations in accurately reflecting the importance of LoRA with the same rank. To address this issue, we have innovatively proposed an importance assessment method based on these two LoRA variants, effectively overcoming the limitations mentioned above.

### 2.3 PEFT on TOD

Research on PEFT methods has already been conducted in TOD systems and has achieved some progress. Jung et al. (2023) enhanced the understanding of TOD contexts by fine-tuning the Flan-T5-XL model and fine-tuning the DeBERTa model to more accurately select relevant knowledge fragments based on the dialogue history and extracted entities. Zhang et al. (2024a) conducted full-parameter fine-tuning and LoRA fine-tuning on the Baichuan2-7B-Base model to allow the model to learn dialogue patterns in different scenarios

and further enhanced the model's adaptability and generalization ability in various scenarios through a secondary LoRA fine-tuning approach. However, existing PEFT methods in TOD still face challenges such as inaccurate assessment of the importance of LoRA parameters, which this document aims to address.

## 3 Preliminaries

### 3.1 LoRA: Low-Rank Adaptation

LoRA is a crucial model fine-tuning technique, widely used in the fields of natural language processing and the field of computer vision. The LoRA framework diagram is shown in Figure 1. LoRA performs low-rank decomposition on pre-trained models during the fine-tuning process. This effectively reduces the number of training parameters, thus achieving the goal of reducing computational costs while improving training efficiency. For example, in the traditional fine-tuning method, updating a weight matrix $W$ with a dimension of $d \times k$ requires updating the entire matrix, which involves a large number of parameters. In contrast, LoRA introduces two low-rank matrices $A$ ($A \in \mathbb{R}^{r \times k}$) and $B$ ($B \in \mathbb{R}^{d \times r}$), with dimensions $d \times r$ and $r \times k$, respectively, where $r$ is the rank number and $r < d$ and $r < k$. The weight matrix $W$ is adjusted indirectly through these two low-rank matrices, and the number of training parameters is $d \times r + r \times k$, which significantly reduces the number of parameters to be trained. Mathematically, the adjustment of the weight matrix $W$ by LoRA can be expressed as:

$$W' = W + \Delta W = W + BA. \quad (1)$$

Let $W'$ be the adjusted weight matrix and $\Delta W$ be the adjustment amount introduced by the low-rank matrices $A$ and $B$.

AdaLoRA is derived from further improvements in LoRA and introduces three key matrices $P$, $Q$, and $\wedge$. $P \in \mathbb{R}^{d_1 \times r}$ is the set of singular vectors left, $Q \in \mathbb{R}^{r \times d_2}$ is the set of singular vectors right, and $\wedge \in \mathbb{R}^{r \times r}$ is the singular value matrix. Therefore, the core formula of AdaLoRA can be expressed as:

$$W' = W + \Delta W = W + BA = W + P\Lambda Q. \quad (2)$$

For the $i$ -th singular value of $\Delta W$ and its corresponding left and right singular vectors, we represent them as a triple $G_i = \{P_{*i}, \lambda_i, Q_{i*}\}$. For a complete model, assume that it contains $n$ single-rank triples to be computed, where the $K$-th triple is denoted as $G_{k,i} = \{P_{k,*i}, \lambda_{k,i}, Q_{k,i*}\}$. To ensure the orthogonality of $P$ and $Q$, AdaLoRA introduces a regularization term:

$$R(P, Q) = \|P^T P - I\|_F^2 + \|QQ^T - I\|_F^2. \quad (3)$$

### 3.2 Measuring Importance of LoRA

During the pruning process of AdaLoRA, the sensitivity of a single parameter is defined as the absolute value of the product of the gradient and the weight (Molchanov et al., 2019):

$$I(w_{ij}) = |w_{ij} \times \nabla_{w_{ij}} L|. \quad (4)$$

Here, $w_{ij}$ represents any trainable weight parameter and $\nabla_{w_{ij}} L$ denotes the gradient corresponding to this weight. If removing a parameter has a significant impact, then the model is sensitive to it. In Stochastic Gradient Descent (SGD), this importance reflects the importance of a single batch of samples. To reduce the evaluation error caused by a single batch of samples, the idea of moving average can be adopted to mitigate the evaluation error of importance caused by a single batch of samples. The expression is:

$$\bar{I}^{(t)}(w_{ij}) = \beta_1 \bar{I}^{(t-1)}(w_{ij}) + (1 - \beta_1) I^{(t)}(w_{ij}). \quad (5)$$

Where $t$ represents the training step and $0 < \beta_1 < 1$ is a hyperparameter in the moving average, which is used to adjust the weight ratio between historical records and the current batch of samples in the calculation. Based on the importance measure, the uncertainty of the sensitivity can be further calculated. This uncertainty characterizes the change in sensitivity on a local time scale (Zhang et al., 2022). For the assessment of uncertainty, it is also recommended to use the moving average method for smoothing. The definition formula of the importance measure is:

$$U^{(t)}(w_{ij}) = |I^{(t)}(w_{ij}) - \bar{I}^{(t)}(w_{ij})|. \quad (6)$$

The definition formula of the unimportance measure is:

$$\bar{U}^{(t)}(w_{ij}) = \beta_2 \bar{U}^{(t-1)}(w_{ij}) + (1 - \beta_2) U^{(t)}(w_{ij}) \quad (7)$$

The importance of a feature can be represented by the product of the sensitivity $\bar{I}^{(t)}(w_{ij})$ and the uncertainty $\bar{U}^{(t)}(w_{ij})$:

$$s^{(t)}(w_{ij}) = \bar{I}^{(t)}(w_{ij}) \times \bar{U}^{(t)}(w_{ij}). \quad (8)$$

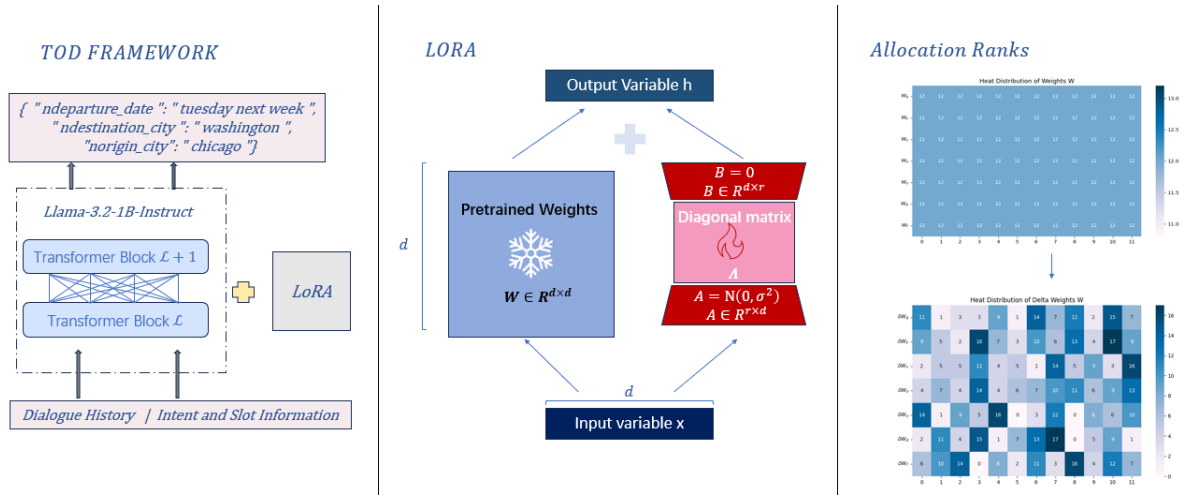


Figure 1: The figure on the left illustrates our model architecture based on the TOD framework, utilizing the Llama-3.2-1B-Instruct/Llama-3.2-3B-Instruct model. The middle figure depicts the structure of LoRA low-rank decomposition. The figure on the right shows the heatmap of the rank allocation obtained through a series of training using our SUCG method.

For the triple $G_{k,i}$, its importance is defined as the weighted sum of its three elements, and the weights are determined by $d_1$ and $d_2$:

$$S_{k,i} = s(\lambda_{k,i}) + \frac{1}{d_1}\sum_{j=1}^{d_1} s(P_{k,ji}) + \frac{1}{d_2}\sum_{j=1}^{d_2} s(Q_{k,ji}). \tag{9}$$

In the AdaLoRA sensitivity analysis framework, only the impact of the changes in the parameters themselves is considered on the model, while the potential impacts of other participants are not taken into account. Therefore, there is a bias in its importance scoring. This bias leads to the fact that the pruning of LoRA is not the most reasonable and fails to fully reflect the true importance of the model parameters in the overall interaction.

AutoLoRA mainly determines the rank of the matrix by selecting variables. Unlike the derivation idea of AdaLoRA's importance-score formula, it conducts analysis from the perspective of optimizing the selection variables. Developed further on the basis of AdaLoRA, it automatically determines the optimal rank for each LoRA layer through meta-learning techniques. This framework associates each rank 1 matrix with a selection variable $\alpha$, and this variable decides whether to retain the corresponding rank 1 matrix. Through meta-learning methods, AutoLoRA learns these selection variables and automatically adjusts the rank of each update matrix. Therefore, the formula for the update matrix of AutoLoRA can be expressed as:

$$\Delta = \sum_{j=1}^{k} \alpha_j \Delta_j \tag{10}$$

In the AutoLoRA framework, each rank 1 matrix is associated with a continuous trainable selection variable $\alpha_j$, where $\alpha_j \in [0, 1]$. This variable determines whether to retain the corresponding rank 1 matrix $\Delta_j$.

Although AutoLoRA has made improvements based on AdaLoRA, its sensitivity analysis still does not fully consider the impacts of other players, resulting in a bias in its importance scoring.

## 4 A novel LoRA Fine-Tuning Method

### 4.1 Motivation

Based on the analysis of the two typical works in 3.1 and 3.2, we can clearly observe the deficiencies present in these two works. In the sensitivity-based method of AdaLoRA, the main defect lies in the locality of its measurement approach. It only focuses on the sensitivity of individual self-parameters and modules, without fully considering the impact of changes in other modules on the overall model performance. The AutoLoRA method also has its drawbacks, which determines the importance of the parameters based on the magnitude of the architectural parameters, but this approach cannot accurately reflect the actual importance of a given module within the model.

To address the deficiencies mentioned above, the introduction of the Shapley value method can be considered. The Shapley value ensures that it provides a comprehensive, fair, and consistent way to allocate the total payoff among players in a cooperative game by considering all possible coalitions and their marginal contributions, and it has a well-defined mathematical foundation for accurate quantification. Its core idea lies in comprehensively evaluating the contribution of each player (which can be analogized to a module or parameter in the model) to the whole by considering various situations where the player participates and does not participate, so as to accurately measure its importance.

However, the Shapley value algorithm faces the challenge of insufficient computational resources. Take an example of a model with an initial setting of $r = 16$, 32 layers and each layer containing 7 linear layers. Calculate roughly the number of computations required. The number of players is $32 \times 7 \times 16$. Taking into account the combination of players, the number of computations is as high as $2^{(32\times7\times16)} - 1$. Such a huge amount of computation makes this method face serious efficiency bottlenecks in practical applications. Therefore, Held and Yang (2023) adopts a Monte Carlo simulation for an approximate calculation, replacing all possible permutations with randomly constructed permutations. Although the amount of computation is greatly reduced, it still takes several days to calculate on a single GPU, which limits its use as a tool for rapid iteration.

To tackle this challenge, we can draw on the idea of the Shapley value, that is, when conducting an importance evaluation, place the current player in different combinations of players. Meanwhile, to improve computational efficiency, we still use the gradient based method. In this case, the gradients of all parameters can be obtained through a single backpropagation, providing a feasible approach for applying the Shapley value-like idea in practical model optimization.

### 4.2 Sensitivity Under Cooperative Game

Therefore, under the AdaLoRA and AutoLoRA frameworks, based on the above content, we have made improvements to the expected gradient calculation. In the parameter set $\lambda_{k,i} = \{\lambda_{k,1}, \lambda_{k,2}, \ldots, \lambda_{k,i}\}$, we randomly select some parameters and set their values to 0. To this end, a random variable $X_i$ is defined, which represents whether the parameter $\lambda_{k,i}$ is set to 0. $X_i$ follows a Bernoulli distribution:

$$X_i \sim \text{Bernoulli}(0.5). \tag{11}$$

This means that the random variable $X_i$ has a $50\%$ probability of taking the value 1, in which case the corresponding output is the parameter $\lambda_{k,i}$; and $X_i$ also has a $50\%$ probability of taking the value 0, in which case the corresponding output value is 0. Furthermore, we define a new random variable $\lambda'_{k,i}$ to represent the parameter value after the random setting to the operation 0.

$$\lambda'_{k,i} = X_i \lambda_{k,i}. \tag{12}$$

Similarly, the random variable $\lambda'_{k,i}$ has a $50\%$ probability of being equal to $\lambda_{k,i}$ and a $50\%$ probability of being equal to 0. In this context, since $\lambda'_{k,i}$ is a random variable, we need to calculate the expected gradient of $L(\lambda'_{k,i})$ with respect to $\lambda_{k,i}$. Meanwhile, the random variable $X_i$ must be related to the gradient of the parameter $\lambda_{k,i}$. Given the randomness of $X_i$, we need to calculate the expected value $E$ of the gradient to obtain a stable gradient estimate. Based on the chain rule and the linearity property of expectation, we derive the following expressions:

$$E\left[\nabla_{\lambda_{k,i}} L(\lambda'_{k,i})\right] = 0.5 \times E\left[\nabla_{\lambda'_{k,i}} L(\lambda'_{k,i})\right]. \tag{13}$$

Finally, we repeat the above mentioned process 5 times, and each repetition is an independent event, which means that different parameters will be randomly set to 0 each time. Repetition experiments improve the robustness and accuracy of the results.

## 5 Experimental Setup

### 5.1 Datasets

We selected the Schema Guided Dialogue (SGD) dataset (Rastogi et al., 2020) for experiments, which is under the CC BY-SA 4.0 license. This dataset contains more than 16k multi-domain dialogues across 16 domains. In terms of scale, it exceeds the existing TOD corpora. We partitioned the data in the dataset into multiple service units for separate calculations and introduced slot names in the dataset to facilitate the extraction of values corresponding to the slots. The statistical information of the data is presented in the Appendix Table 2.

### 5.2 Evaluation Metrics

We use six evaluation metrics to measure the experimental results, which are Acc(JGA), Acc(SL), Acc(VAS), GORT, GMU and NFTP:

**Acc(JGA) - widely adopted Joint Goal Accuracy** represents the accuracy of Joint Goal Accuracy (JGA) (Wu et al., 2019), which is used to evaluate the performance of DST. The larger the Acc(JGA), the better the accuracy of the model in predicting the dialogue state.

**Acc(SL) - Slot-level Accuracy** is used to measure whether the slot values predicted by the model are correct. The larger the Acc(SL), the more accurate the prediction model is when predicting slot values, and the more effectively it can extract and understand the important information in the user input.

**Acc(VAS) - Accuracy of Volatility Between Services** is used to measure the accuracy fluctuations of the model between different services or scenarios. By analyzing the volatility, the weak links of the model can be identified and optimized accordingly, thereby improving the generalization ability and stability of the model. The smaller the Acc(VAS), the smaller the fluctuations, indicating that the model performs more stably.

**GORT - GPU Occupancy Rate during Training** is used to measure the utilization rate of the GPU. A higher utilization rate generally implies that more GPU resources are being utilized, leading to a faster training time.

**GMU - GPU Memory Usage** is used to measure the usage of GPU memory during the generation task. If the GMU is too high, approaching or exceeding the upper limit of the GPU memory, it may lead to performance degradation and out-of-memory issues. If the GMU is too low, it means that the GPU memory is not fully utilized, resulting in reduced efficiency. Therefore, it is of great importance to control the value of GMU within a reasonable range.

**NFTP - Number of Fine-Tunable Parameters** is used to measure the number of finetunable parameters in the model. NFTP is neither better when it is larger nor better when it is smaller. Instead, a trade-off needs to be made according to specific application scenarios and requirements. When higher flexibility and expressive ability are required, a larger number of fine-tunable parameters may be more advantageous. When pursuing higher efficiency, better generalization ability, and a lower risk of overfitting, a smaller number of finetunable parameters may be more appropriate.

### 5.3 Baselines

We conducted experiments on the 1B model and the 3B model, respectively, and compared the experimental results with three other groups of baselines. The three groups of baselines are Full scale Finetuning, LoRA Finetuning, and AdaLoRA. Full-scale fine-tuning means that all parameters participate in the training and the weights of the parameters are adjusted. LoRA Fine-Tuning introduces low-rank matrices based on the pre-trained model and only fine-tuning a small number of newly added parameters. AdaLoRA can adaptively adjust the rank and better adapt to tasks of different complexities while reducing resource waste. Compared with the three baselines, the effectiveness and rationality of the experiments are verified.

### 5.4 Experiment Setting

The experiments are carried out on an Ubuntu desktop with 64 GB memory, Ultra9 CPU (24 cores), NVIDIA A6000 GPUs (48 GB).

During the training process, the pre-trained model we selected is Llama-3.2-1B-Instruct. The seed is set to 100 to ensure the reproducibility of the experiments. The size of the block is set to 1024, which defines the size of the input data blocks.

In the training process, the batch parameter is set to 1, indicating that the number of batch samples is 1. In the evaluation phase, the number of batch samples in each device is set to 16. The gradient accumulation steps are set to 8, which means that the gradients are accumulated in every 8 step. The number of training epochs is set to 10, indicating that the model traverses and learns from the training dataset 10 times. The warm-up steps are set to 100. The evaluation steps are set to 100, which means that the model is evaluated every 100 training step.

The learning rate is set to $1.0 \times 10^{-3}$. The maximum patience is set to 10, which indicates that when the performance of the model in the validation set does not improve in 10 consecutive evaluations, the training will be stopped. The LoRA rank is set to 8.

## 6 Experiments and Analysis

### 6.1 Main results

We analyze the main experimental results from two aspects, namely Automatic Evaluation and Human

| Model | Acc(JGA) | Acc (SL) | Acc (VAS) | GORT | GMU | NFTP |
|---|---|---|---|---|---|---|
| ***1B Model*** | | | | | | |
| Full scale Fine-tuning | 0.45 | 0.90 | 0.11 | 1.24 h | 18 G | 1,235,815,408 |
| LoRA Fine-tuning | 0.49 | 0.84 | 0.32 | 1.35 h | 7 G | 5,636,992 |
| AdaLoRA | 0.58 | 0.89 | 0.51 | 2.58 h | 8 G | 1,704,448 |
| **SUCG*** | 0.59 | 0.87 | 0.53 | 2.52 h | 28 G | 11,273,984 |
| ***3B Model*** | | | | | | |
| Full scale Fine-tuning | 0.48 | 0.92 | 0.28 | 3.10 h | 40 G | 3,212,751,588 |
| LoRA Fine-tuning | 0.54 | 0.87 | 0.69 | 2.56 h | 13 G | 12,158,496 |
| AdaLoRA | 0.59 | 0.89 | 0.73 | 2.63 h | 20 G | 4,588,416 |
| **SUCG*** | 0.61 | 0.91 | 0.76 | 2.04 h | 27 G | 24,316,992 |

Table 1: Experimental results of SUCG are compared with those of three other groups of baselines on the 1B Model and 3B Model respectively. To conduct a more comprehensive and effective evaluation, a total of six evaluation metrics are used.

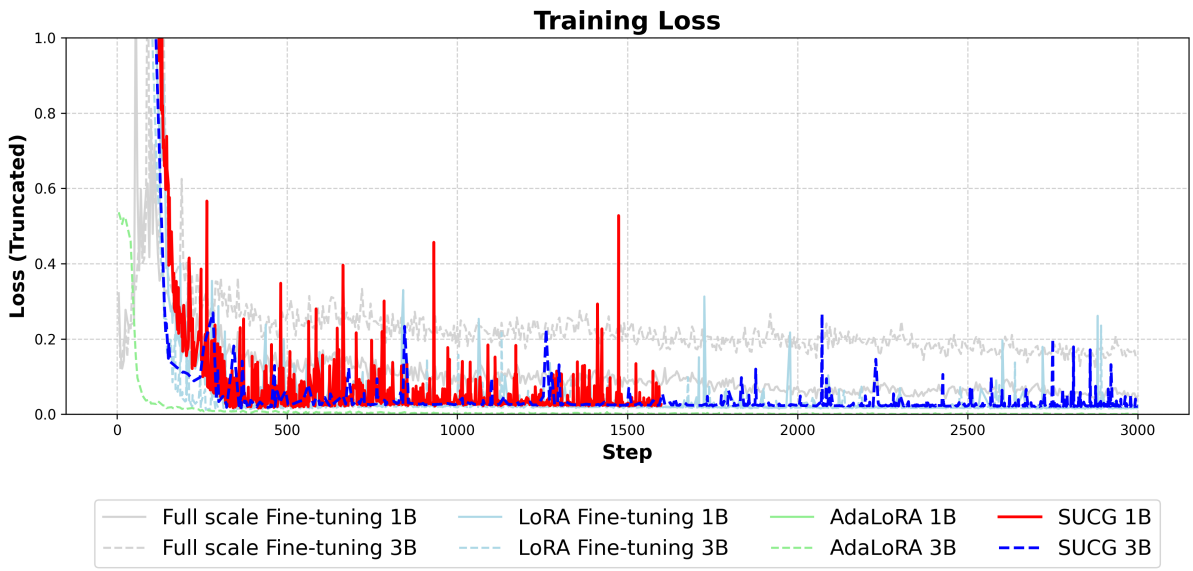


Figure 2: Line Chart of Loss during Training for SUCG and Other Baselines.

Evaluation.

### 6.1.1 Automatic Evaluation

The experimental results with Llama-3.2 1B and 3B models are presented in Table 1. Si

To comprehensively explore the Llama-3.2-1B-Instruct model, we conducted ablation studies, comparing it with Qwen2.5-1.5B-Instruct and Deepseek Distill 3B using six metrics: Acc(JGA), Acc(SL), Acc(VAS), GORT, GMU, and NFTP.

As shown in Table 1, for the 1B model, SUCG had competitive Acc(JGA) and Acc(SL) scores. Its Acc(VAS) was average, GORT moderate, GMU relatively high, and NFTP distinct. For the 3B Model, SUCG outperformed Acc(JGA) and Acc(SL), had a decent Acc(VAS), a fast GORT, a reasonable GMU, and a specific NFTP.

From the training loss line graph in Figure 2, the SUCG 1B model performed worse than the SUCG 3B model. The SUCG 3B model's loss was lower than that of Full scale Fine tuning and LoRA Fine-tuning models as training progressed.

Evaluating the six metrics under different ranks, we found that Acc(JGA) and Acc(SL) first improved, then leveled off or declined with increasing rank. Lower ranks led to a more stable Acc(VAS). Higher ranks slightly increased GORT, GMU was stable, and NFTP grew linearly. These findings are useful for model optimization and hyperparameter selection.

### 6.1.2 Human Evaluation

To compensate for the limitations of Automatic Evaluation in measuring the dimension of natural

language, this study performed a human evaluation. The aim was to conduct a more comprehensive analysis of the eight groups of experimental results in 6.1.1 from a human perspective. We invited 20 volunteers, including researchers and ordinary users, who were selected. During the evaluation process, we mainly considered three dimensions: Accuracy, Fluency, and Completion. Accuracy assesses whether the information provided by the dialogue system is correct. Fluency evaluates whether the natural language provided by the dialogue system is natural and coherent. Completion measures whether the dialogue system successfully completes the tasks set by the user.

From Appendix 3, in the human evaluation, the superiority of the SUCG model was evident. In terms of precision, with an average score of [Accuracy score for SUCG in human evaluation] (compared to [Baseline Accuracy scores]), it frequently surpassed some of the baselines, offering more accurate information. Regarding Fluency, it obtained an average score of [Fluency score for SUCG in human evaluation], achieving high marks and thus generating more natural and coherent language. When it came to Completion, the SUCG model had an average score of [Completion score for SUCG in human evaluation], demonstrating greater success in fulfilling user-set tasks. In general, these results not only provided additional evidence for the effectiveness of the SUCG method, supplementing the automatic evaluation findings, but also indicated its potential to enhance the state tracking of LLM-based dialogues.

### 6.2 Ablation Studies and Further Analysis

To thoroughly explore the technical aspects of the Llama-3.2-1B Instruct model used in the experiment, we performed the ablation experiments detailed in the Appendix Table 3. In these experiments, we juxtaposed the experimental outcomes of Llama-3.2-1B-Instruct with those of Qwen2.5-1.5B-Instruct and DeepSeek-R1-Distill-Qwen-1.5B.

As revealed by the evaluation metrics, when optimized with our proposed method, the Llama-3.2-1B-Instruct model showcases competitive performance in Acc(JGA), Acc(SL), and Acc(VAS). For instance, in Acc(JGA), it achieved a score of 0.59, close to the leading values among the compared models. In Acc(SL), it reached 0.87, outperforming some baselines. And in Acc(VAS), its score of 0.53 indicated a stable performance across different services or scenarios.

Regarding GORT, the Llama-3.2-1B-Instruct model took only 2.52h, which is significantly less than the 8.49h of DeepseekDistill-3B. This indicates more efficient GPU utilization and thus faster training. In terms of GMU, it used 28G of GPU memory, a reasonable amount that ensures stable operation without overconsuming resources.

For NFTP, Llama-3.2-1B-Instruct had 11,273(k) fine-tunable parameters, differing from the 18,467(k) of Qwen2.5-1.5B-Instruct and Deepseek Distill 3B. This difference implies a unique balance between flexibility and generalizability.

In general, these results emphasize the efficacy of our method in optimizing Llama-3.2-1B-Instruct. They also offer practical guidance for model selection and parameter tuning in related research.

## 7 Conclusion

In this paper, to enhance the performance of leveraging LLMs in TOD, we propose a novel importance evaluation method, Sensitivity under cooperative game. Specifically, when calculating the gradients, we consider different combinations of single-rank LoRAs and apply this approach within the AdaLoR framework. As a result, the proposed model exhibits a strong ability for efficient fine-tuning. Experimental results demonstrate that in automatic and human evaluations, the proposed model achieves significant performance improvements compared to previous state of the art models.

## Limitations

We've proven that our SUCG method can remarkably boost parameter-efficient tuning performance across diverse tasks and pre-trained models (e.g., Llama-3.2 series, Qwen2.5 series). However, our study has limitations. Limited by computational resources, we could not test on larger-scale models like Llama-3 30B or 70B. Also, we did not consider tasks like information extraction. However, our SUCG framework is likely to be adaptable to other models and tasks. Whether it remains superior in such scenarios is worth exploring. We will focus on this in future research.

## Ethical Considerations

Our work does not involve risk issues, including: (1) No privacy concerns; (2) No potential to misguide humans. Moreover, no new risks are introduced and all risks are inherent in the LLM itself.

Our work is mainly for academic research rather than commercial use, so it will not pose risks to users.

## References


Aitor Arrieta, Miriam Ugarte, Pablo Valle, José Antonio Parejo, and Sergio Segura. 2025. o3-mini vs deepseek-r1: Which one is safer?

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Willy Chung, Samuel Cahyawijaya, Bryan Wilie, Holy Lovenia, and Pascale Fung. 2023. InstructTODS: Large language models for end-to-end task-oriented dialogue systems. In *Proceedings of the Second Workshop on Natural Language Interfaces*, pages 1–21, Bali, Indonesia. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhi-

gang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. Deepseek-v3 technical report.

Silin Gao, Ryuichi Takanobu, Antoine Bosselut, and Minlie Huang. 2022. End-to-end task-oriented dialog modeling with semi-structured knowledge management. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2173–2187.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *ArXiv*, abs/2403.14608.

William Held and Diyi Yang. 2023. Shapley head pruning: Identifying and removing interference in multilingual transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2416–2427, Dubrovnik, Croatia. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR 2022*.

Haein Jung, Heuiyeen Yeen, Jeehyun Lee, Minju Kim, Namo Bang, and Myoung-Wan Koo. 2023. Enhancing task-oriented dialog system with subjective knowledge: A large language model-based data augmentation framework. In *Proceedings of The Eleventh Dialog System Technology Challenge*, pages 150–165.

Taaha Kazi, Ruiliang Lyu, Sizhe Zhou, Dilek Hakkani-Tür, and Gokhan Tur. 2024. Large language models as user-agents for evaluating task-oriented-dialogue systems. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 913–920. IEEE.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. 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.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2024. Gpt understands, too. *AI Open*, 5:208–215.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272.

Atsumoto Ohashi and Ryuichiro Higashinaka. 2022. Post-processing networks: Method for optimizing pipeline task-oriented dialogue systems using reinforcement learning. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–13, Edinburgh, UK. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue,

Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Libo Qin, Wenbo Pan, Qiguang Chen, Lizi Liao, Zhou Yu, Yue Zhang, Wanxiang Che, and Min Li. 2023. End-to-end task-oriented dialogue: A survey of tasks, methods, and future directions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5925–5941, Singapore. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8689–8696.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Weijie Xu, Zicheng Huang, Wenxiang Hu, Xi Fang, Rajesh Cherukuri, Naumaan Nayyar, Lorenzo Malandri, and Srinivasan Sengamedu. 2024. HR-MultiWOZ: A task oriented dialogue (TOD) dataset for HR LLM agent. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 59–72, St. Julian's, Malta. Association for Computational Linguistics.

Longfei Yang, Jiyi Li, Sheng Li, and Takahiro Shinozaki. 2023. Multi-domain dialogue state tracking with disentangled domain-slot attention. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4928–4938.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. 2024. Qwen2.5 technical report. *ArXiv*.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Naoki Yoshimaru, Motoharu Okuma, Takamasa Iio, and Kenji Hatano. 2023. Asyncmld: Asynchronous multi-llm framework for dialogue recommendation system. *arXiv preprint arXiv:2312.13925*.

Ming Zhang, Caishuang Huang, Yilong Wu, Shichun Liu, Huiyuan Zheng, Yurui Dong, Yujiong Shen, Shihan Dou, Jun Zhao, Junjie Ye, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024a. TransferTOD: A generalizable Chinese multi-domain task-oriented dialogue system with transfer capabilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12750–12771, Miami, Florida, USA. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023a. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning.

Qingru Zhang, Simiao Zuo, Chen Liang, Alexander Bukharin, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2022. Platon: Pruning large transformer models with upper confidence bound of weight importance. In *International conference on machine learning*, pages 26809–26823. PMLR.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. 2024b. AutoLoRA: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5048–5060, Mexico City, Mexico. Association for Computational Linguistics.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023b. SGP-TOD: Building task bots effortlessly via schema-guided LLM prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13348–13369, Singapore. Association for Computational Linguistics.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.

# A Dataset Distribution Display Table

# B Chart of Human Evaluation

# C Other Ablation Experiments

| Task ID | Service | # Slots | # Dialogs | | | # Samples | | | Avg. tokens | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | *Train* | *Dev* | *Test* | *Train* | *Dev* | *Test* | *Context* | *Query* |
| 1 | events_3 | 5 | 53 | 7 | 16 | 312 | 40 | 105 | 121 | 47 |
| 2 | banks_2 | 4 | 29 | 4 | 9 | 220 | 31 | 72 | 111 | 49 |
| 3 | banks_1 | 4 | 144 | 21 | 42 | 1138 | 169 | 335 | 114 | 57 |
| 4 | calendar_1 | 4 | 118 | 17 | 34 | 773 | 110 | 234 | 112 | 33 |
| 5 | movies_3 | 3 | 33 | 5 | 10 | 112 | 18 | 37 | 72 | 26 |
| 6 | movies_2 | 5 | 231 | 33 | 67 | 1593 | 221 | 469 | 117 | 52 |
| 7 | services_2 | 5 | 129 | 19 | 37 | 917 | 148 | 253 | 131 | 54 |
| 8 | payment_1 | 4 | 25 | 3 | 8 | 233 | 33 | 89 | 171 | 52 |
| 9 | media_1 | 4 | 196 | 28 | 57 | 1207 | 182 | 360 | 99 | 48 |
| 10 | weather_1 | 2 | 58 | 8 | 17 | 259 | 39 | 66 | 77 | 16 |
| 11 | hotels_2 | 6 | 202 | 29 | 58 | 1424 | 195 | 400 | 132 | 64 |
| 12 | flights_4 | 7 | 60 | 9 | 18 | 290 | 41 | 87 | 90 | 77 |
| 13 | travel_1 | 4 | 48 | 7 | 14 | 231 | 28 | 63 | 87 | 59 |
| 14 | buses_2 | 6 | 111 | 16 | 32 | 857 | 120 | 234 | 137 | 54 |
| 15 | events_1 | 4 | 400 | 57 | 115 | 3537 | 521 | 1067 | 159 | 59 |
| 16 | alarm_1 | 2 | 58 | 9 | 17 | 367 | 49 | 107 | 101 | 22 |
| 17 | buses_3 | 1 | 61 | 9 | 18 | 405 | 60 | 114 | 123 | 69 |
| 18 | services_1 | 5 | 185 | 27 | 53 | 1241 | 180 | 352 | 129 | 58 |
| 19 | buses_1 | 5 | 136 | 20 | 39 | 1054 | 143 | 313 | 138 | 49 |
| 20 | restaurant_2 | 9 | 87 | 13 | 28 | 807 | 113 | 240 | 154 | 97 |
| 21 | hotels_2 | 6 | 212 | 31 | 61 | 1569 | 234 | 460 | 152 | 73 |
| 22 | ridesharing_2 | 3 | 64 | 9 | 19 | 380 | 49 | 108 | 106 | 34 |
| 23 | rentalcars_1 | 6 | 100 | 14 | 29 | 840 | 120 | 242 | 161 | 70 |
| 24 | movies_1 | 8 | 263 | 37 | 76 | 1873 | 250 | 556 | 122 | 59 |
| 25 | ridesharing_1 | 3 | 74 | 10 | 22 | 412 | 57 | 125 | 103 | 36 |
| 26 | media_2 | 4 | 56 | 8 | 16 | 327 | 42 | 89 | 95 | 36 |
| 27 | music_3 | 1 | 17 | 3 | 5 | 112 | 19 | 32 | 114 | 60 |
| 28 | movies_2 | 6 | 32 | 5 | 10 | 118 | 20 | 38 | 70 | 30 |
| 29 | flights_2 | 7 | 129 | 19 | 37 | 822 | 115 | 251 | 127 | 75 |
| 30 | services_4 | 6 | 86 | 13 | 25 | 680 | 97 | 208 | 154 | 49 |
| 31 | flights_1 | 10 | 560 | 80 | 160 | 4680 | 667 | 1379 | 168 | 10 |
| 32 | services_3 | 5 | 131 | 19 | 38 | 959 | 143 | 290 | 143 | 54 |
| 33 | flights_3 | 8 | 65 | 10 | 19 | 420 | 75 | 116 | 133 | 76 |
| 34 | trains_1 | 7 | 58 | 9 | 17 | 415 | 67 | 117 | 131 | 76 |
| 35 | homes_2 | 8 | 62 | 11 | 18 | 424 | 56 | 139 | 140 | 89 |
| 36 | rentalcars_2 | 6 | 77 | 11 | 23 | 631 | 91 | 185 | 157 | 61 |
| 37 | restaurant_1 | 9 | 256 | 37 | 74 | 2098 | 297 | 581 | 153 | 10 |
| 38 | hotels_4 | 6 | 68 | 10 | 20 | 468 | 73 | 142 | 118 | 61 |
| 39 | hotels_4 | 5 | 80 | 12 | 23 | 559 | 99 | 141 | 134 | 72 |
| 40 | media_2 | 7 | 32 | 4 | 10 | 215 | 29 | 71 | 112 | 59 |
| 41 | hotels_3 | 6 | 90 | 13 | 26 | 737 | 100 | 193 | 157 | 64 |
| 42 | rentalcars_3 | 7 | 44 | 7 | 13 | 332 | 55 | 99 | 148 | 72 |
| 43 | hotels_1 | 7 | 99 | 14 | 29 | 868 | 105 | 250 | 161 | 71 |
| 44 | homes_1 | 7 | 244 | 35 | 70 | 1829 | 282 | 540 | 159 | 81 |

Table 2: Dataset Distribution Display Table

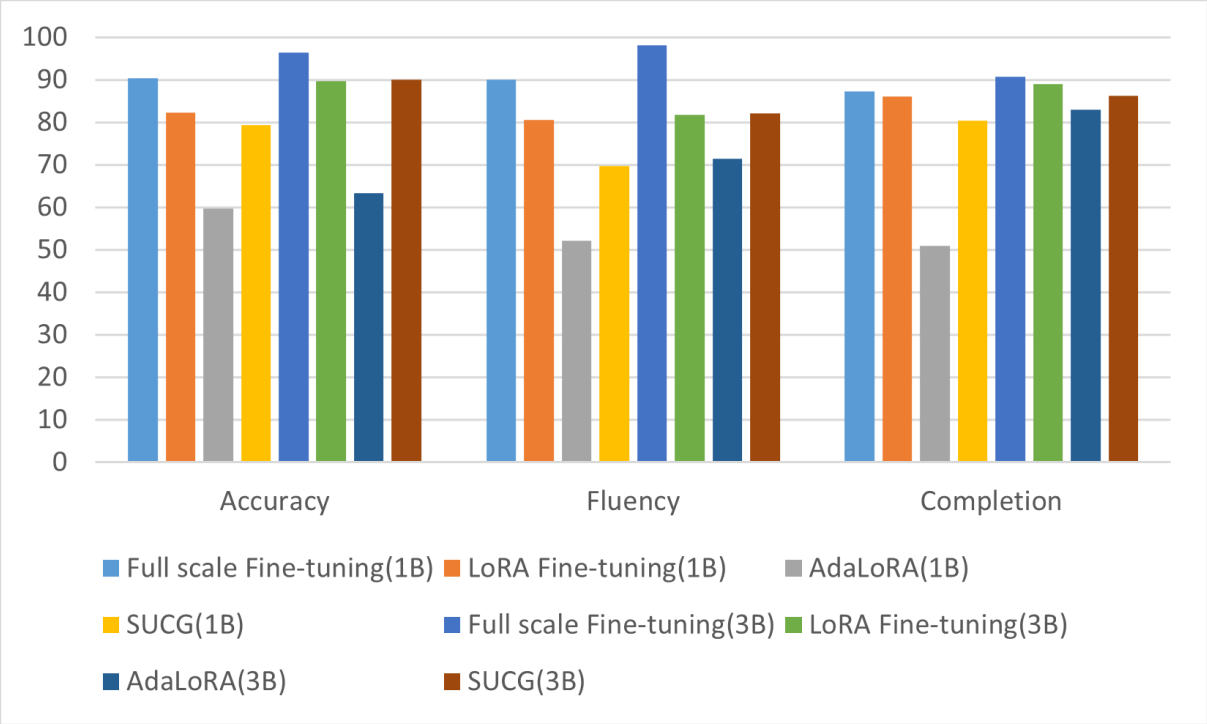


Figure 3: This human evaluation chart requires scoring for eight groups of experiments. The horizontal axis represents three types of evaluation criteria, and the vertical axis represents the average scores given by 20 human evaluators.

| **Indicators** | **LI3.2*** | **Q2.5** | **DS1B** |
|---|---|---|---|
| Acc(JGA) | 0.59 | 0.51 | 0.49 |
| Acc (SL) | 0.87 | 0.87 | 0.79 |
| Acc (VAS) | 0.53 | 0.51 | 0.49 |
| GORT | 2.52 h | 4.39 h | 8.49 h |
| GMU | 28 G | 18 G | 40 G |
| NFTP | 11,273(k) | 18,467(k) | 18,467(k) |

Table 3: LI3.2* represents Llama-3.2-1B-Instruct, Q2.5 represents Qwen2.5-1.5B-Instruct, and DS1B represents DeepSeek-R1-Distill-Qwen-1.5B. In NFTP, the (k) denotes the order of magnitude of thousand.