The Technical Report of DuIVRS-2.

1 EXPERIMENTAL IMPLEMENTATIONS

1.1 Test Data

Table 1 illustrates the detailed statistics of constructed test dataset, which provides a comprehensive evaluation of fine-tuned LLMs from three distinct dimensions.

Table 1. The statistics of test data.

Dataset	$D_{Effectiveness}$	$D_{Generalization}$	$D_{Robustness}$
Number of Samples	626	635	656
Avg. Reply Length	8.52	9.21	21.40
Avg. Reply Frequency	385	193	67

1.2 Training Details

The model training was conducted on Baidu's PaddleCloud, utilizing eight NVIDIA A100-80G GPUs for fine-tuning. We utilized the AdamW [16] optimizer with the parameters set to $\beta_1 = 0.9, \beta_2 = 0.95, eps = 1e-5$. The training process was configured with a batch size of 128 and a sequence length limited to 1024 tokens. A linear learning rate schedule was applied, incorporating a warm-up phase covering 3% of the total training steps. The maximum learning rates were established at 2×10^{-5} for the EB-turbo model and 1×10^{-4} for the EB-tiny model. For the EB-tiny model, we employed bf16 16-bit (mixed) precision for full-parameter fine-tuning. For the EB-turbo model, we use Lora [9] for parameter-efficient fine-tuning. Finally, we fine-tuned the two models for 2 epochs.

1.3 Power Consumption

Building on prior research [29, 33] and power consumption data for GPU devices, we aim to estimate the financial costs and carbon emissions associated with our training process. Along with previous work, our analysis excludes additional power requirements, such as those from interconnects or ancillary non-GPU energy expenditures. At each iterative stage, the training duration for EB-tiny is about 1 hour, and EB-turbo is 14 hours, amounting to a cumulative $(1+14)\times 8\times 6 = 720$ GPU hours on A100-80G units with a TDP of 400W. Considering GPUs' actual power use (typically under 400W) and an electricity rate of 1.2 RMB/kWh, the maximum training expense is roughly 400 RMB, with carbon emissions approximating 122kgCO2eq. Furthermore, utilizing the ERNIE 4.0 API service adds to the cost. With a rate of 0.15 RMB per 1k tokens and across five iterations totaling 72,000 requests at 0.5k tokens each, the API costs about $0.15 \times 72 \times 10^3 = 5400$ RMB. Overall, the complete training costs are projected to stay below 10,000 RMB, with carbon emissions under 1tkgCO2eq.

2 COOPERATIVE ITERATIVE LEARNING FRAMEWORK

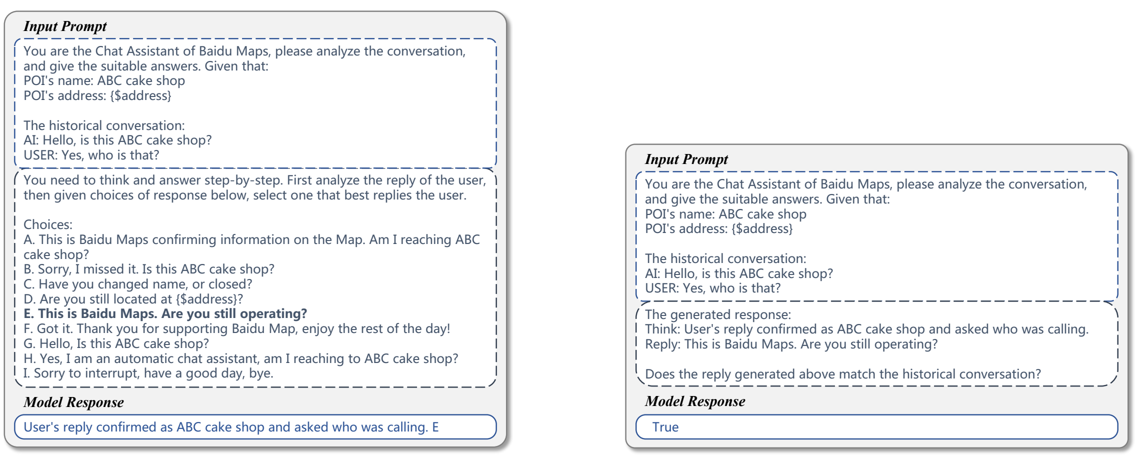
Algorithm 1 delineates the complete cooperative iterative learning framework encompassing data growth and policy improvement phases.

Algorithm 1 Cooperative Iterative Learning framework

Input: $\mathcal{D}_g^0$: the original dataset derived by DuIVR-1, $\mathcal{F}(\Theta)$: the parameter of LLM-S, $\mathcal{G}(\Theta)$: the parameter of LLM-L, T: number of iterations.
Train $\mathcal{F}(\Theta)$ using Equation 9 on $\mathcal{D}_g^0$.
for iteration t in $\{1, \ldots, T\}$ **do**
  // Data Growth Phase
  Generate dataset $\mathcal{D}_g^t$ according to Equation 10.
  for each sample in $\mathcal{D}_g^t$ **do**
    Use LLM-L for evaluation according to Equation 15.
    Use Black-box LLM for evaluation by prompt engineering.
    Vote and refine according to cooperative voting scheme.
  end for
  Construct the refined dataset $\bar{\mathcal{D}}_g^t$, and evaluation dataset $\bar{\mathcal{D}}_e^t$.
  // Policy Improvement Phase
  Improve LLM-S by fine-tuning the $\mathcal{F}(\Theta)$ on $\bar{\mathcal{D}}_g^t$.
  Improve LLM-L by fine-tuning the $\mathcal{G}(\Theta)$ on $\bar{\mathcal{D}}_g^t$ and $\bar{\mathcal{D}}_e^t$.
  Improve Black-box LLM by optimizing evaluation prompt.
end for
Output: Policy $\mathcal{F}(\Theta)$

3 PROMPT EXAMPLES

In this section, we showcase various prompts devised for both the training and evaluation phases of DuIVRS-2. Figure 1a provides an example for training the inference model, which adopts a selective generation strategy with CoT mechanism to ensure both safety and stability upon deployment. The model response, "User's reply confirmed as ABC cake shop" serves as an instance of the chain of thought process, with option "E" indicating the subsequent query "This is Baidu Maps. Are you still operating?". Figure 1b offers insight into the training evaluation ability of EB-turbo, where it functions to assess the accuracy of EB-tiny's outputs by yielding a "True/False" verdict.



(a) Example for training the inference model. (b) Example for training the evaluation model.

Fig. 1. Training example for fine-tuning LLMs.

Furthermore, the prompt iteration process for ERNIE 4.0 during the evaluation phase is illustrated in Figure 2. Iteration-V1 enhances the prompt by integrating the criterion "expresses a wish to hang up", and iteration-v2 further refines it by adding “indirectly expresses a wish for the conversation to end”, thus extending coverage to more domain-specific edge cases and enhancing the discernment precision of ERNIE 4.0.

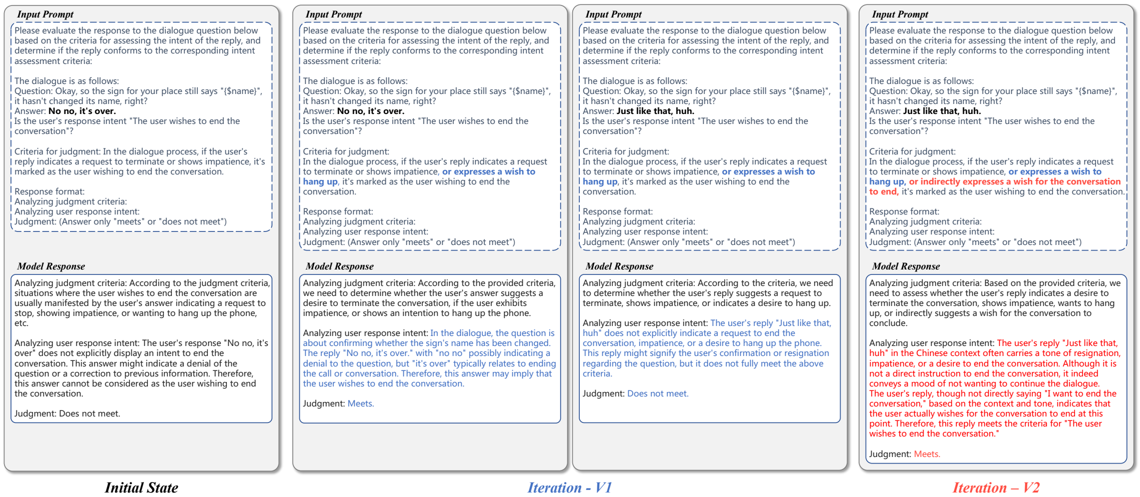


Fig. 2. The prompt iteration of ERNIE 4.0 in evaluation stage.