Liu et al., 2024). Given these constraints, recent studies probe into utilizing LLMs to automatically generate synthetic instruction data (Whitehouse et al., 2023; Dai et al., 2023; Taori et al., 2023b). Specifically, these approaches involve prompting powerful LLMs with limited seed data to generate additional synthetic data. These data are subsequently employed to fine-tune smaller models, aiming to transfer knowledge to small LLMs and enhance their performance (Wang et al., 2023a). Leveraging LLMs to generate data can significantly reduce the costs and time for data curation (Liu et al., 2024), while simultaneously improving the efficacy of the fine-tuned models for designated downstream tasks (Yang et al., 2020; Puri et al., 2020; Guo et al., 2023; Samuel et al., 2023; Schlegel et al., 2023).

3 LLMOps Pipeline: LlamaDuo

In this section, we elaborate on the details of the proposed LlamaDuo, which are depicted in Figure 1. This LLMOps pipeline aims to ensure service LLMs continuity by transitioning knowledge and abilities from service-oriented LLMs to smaller, locally manageable LLMs without the need for human intervention.

3.1 Coverage Dataset

Users interact with service LLMs through prompt engineering efforts. The historical trials composed of the user input prompt and the responses of service LLMs, and potential errors will be recorded and saved in local storage. Subsequently, users annotate and collect the most satisfied prompt and response pairs conformed with their real-world use cases. The resulting instruction dataset is termed as coverage dataset, denoted as $\mathcal{D}^{(0)} := \{\mathcal{I}_i^{(0)}, \mathcal{R}_i^{(0)}\}_{i=1}^{N}$, and split as train and test subsets by ratio Φ. Here, $\mathcal{I}_i^{(0)}$ denotes the i-th instruction (prompt) in $\mathcal{D}^{(0)}$, $\mathcal{R}_i^{(0)}$ is the corresponding response for the i-th instruction, and N is the number of samples in $\mathcal{D}^{(0)}$. Since coverage dataset is of high quality and satisfying the user's intent in real-world context, the train subsets $\mid \mathcal{D}_{train}^{(0)} \mid = \Phi \cdot N$ will be served as seeds for synthetic datasets generation, while the test subset $\mid \mathcal{D}_{test}^{(0)} \mid = (1-\Phi) \cdot N$ is reserved for performance evaluation of the fine-tuned local LLMs.

3.2 Fine-tuning

To efficiently and effectively adapt the local model to specific downstream task(s), we finetune the local LLM with the supervised learning paradigm on high-quality instruction data. At the initial cyclicality of the pipeline, the selected local LLM is fine-tuned on the train subsets $\mathcal{D}_{train}^{(0)}$ of the coverage dataset, obtaining the fine-tuned model $\pi^{(0)}$. At subsequent cyclicality t, if the performance of fine-tuned model does not reach or surpass the predetermined evaluation threshold ε of specific tasks, the local LLM $\pi^{(t)}$ will be continuously fine-tuned on the increasing number of synthetic data $\{\mathcal{D}_{synth}^{(1)}, \mathcal{D}_{synth}^{(2)}, \ldots, \mathcal{D}_{synth}^{(t-1)}\}$ generated from service LLMs with $\mathcal{D}_{train}^{(0)}$ as seed dataset. Consequently, when $t \geq 1$, the objective of the fine-tuning phase can be formulated as

$$\mathcal{L}_{\text{SFT}}(\pi^{(t)}, \mathcal{D}^{(t)}) = -\mathbb{E}\left[\log P_{\pi^{(t-1)}}(\mathcal{R}^{(t)} \mid \mathcal{I}^{(t)})\right], \tag{1}$$

where $\mathcal{R}^{(t)} \sim \{\mathcal{D}_{train}^{(0)}, \{\mathcal{D}_{synth}^{(\tau)}\}_{\tau=1}^{t-1}\}$ and $\mathcal{I}^{(t)} \sim \mathcal{D}_{train}^{(0)}$.

3.3 Batch Inference

After the fine-tuning stage, the fine-tuned local model is prompted with prompts $\mathcal{I}^{(0)}$ sampled from the test subsets $\mathcal{D}_{test}^{(0)}$ of the coverage dataset to produce corresponding response $\hat{\mathcal{R}} \sim \pi^{(t)}(\mathcal{R}^{(0)} \mid \mathcal{I}^{(0)})$. To improve the diversity and robustness of responses, the local model generates a batch of K responses $\{\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, \ldots, \hat{\mathcal{R}}_K\}$ for each given prompt $\mathcal{I}^{(0)}$. Totally, it will construct prompt and responses pairs $\{(\mathcal{I}_i^{(0)}, \hat{\mathcal{R}}_i)\}_{i=1}^{(1-\Phi)\cdot N \cdot K}$. Formally,

$$\hat{\mathcal{R}}_k \sim \pi^{(t)}(\mathcal{R}^{(0)} \mid \mathcal{I}^{(0)}), \tag{2}$$

where $k \in \{1, 2, \ldots, K\}, \mathcal{I}^{(0)} \sim \mathcal{D}_{test}^{(0)}$.

3.4 Evaluation

In the evaluation stage, we employ "service LLMs-as-judge", denoted as $\mathcal{E}_{\text{LLM}}(\cdot)$, to conduct performance evaluation of local model on $\{(\mathcal{I}_i^{(0)}, \hat{\mathcal{R}}_i)\}_{i=1}^{(1-\Phi)\cdot N \cdot K}$. Following the works (Zheng et al., 2024; Yuan et al., 2024), the service LLMs evaluate each response triple $(\mathcal{I}^{(0)}, \hat{\mathcal{R}}, \mathcal{R}^{(0)})$, comprising prompt, the corresponding generated response, and the ground truth, by M times with pairwise comparison and single answer grading strategies. This evaluation process guarantees the trustworthy and reduces the inherent bias of results. Moreover, when employing