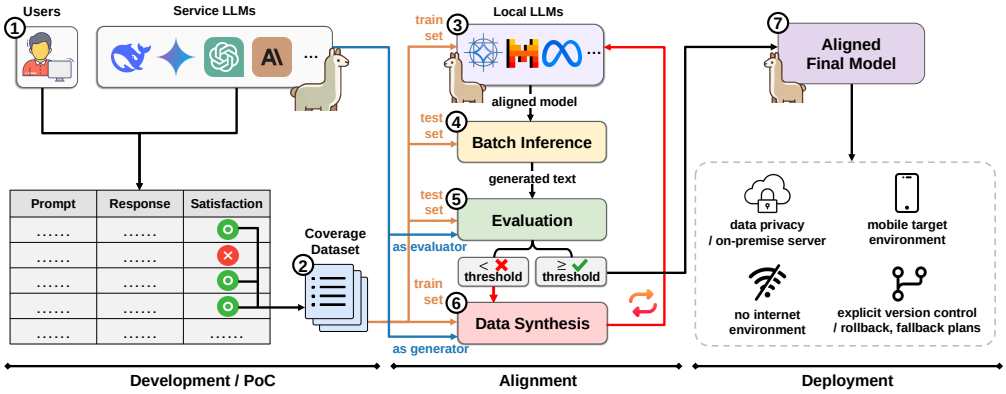

Figure 1: The LLMOps pipeline namely LlamaDuo for migrating from service LLMs to small-scale local LLMs involves three phases. In the Development/PoC phase, ① users manually engineer prompts to interact with service LLMs and ② collect satisfying (prompt, response) pairs into train and test datasets. In the Alignment phase, ③ local LLMs are aligned with the train dataset, ④ tested on the test dataset, and ⑤ evaluated by service LLMs. ⑥ Synthetic data is generated iteratively until the performance of the aligned model meets a threshold. In the Deployment phase, ⑦ the satisfactory model is deployed in constrained environments.

LLMs as evaluators, the evaluation metrics can be more flexibly adapted to specific tasks, along with a thorough evaluation guide. In this paper, we measure the similarity between $\hat{\mathcal{R}}$ and $\mathcal{R}^{(0)}$, and how precise $(\mathcal{I}^{(0)}, \hat{\mathcal{R}})$ the responses generated by the local LLM answer the given instructions. These two metrics are provided simultaneously through a prompt, as shown in Figure 5 of Appendix A. Therefore, $\{(\mathcal{I}_i^{(0)}, \hat{\mathcal{R}}_i, \mathcal{R}_i^{(0)})\}_{i=1}^{(1-\Phi)\cdot N\cdot K}$ invokes service LLMs to perform evaluation by $(1-\Phi)\cdot N\cdot K\cdot M$ times. Subsequently, the evaluation results can be leveraged according to the intention of the operator performing this LLMOps pipeline. For example, actions can be taken to increase the reliability of service LLM as an evaluator by calculating the mean or median. In this study, we adopt the mean score $V_{\pi^{(t)}}$ and coverage percentage $C_{\pi^{(t)}}$ with ζ score as evaluation results. Here, the coverage percentage $C_{\pi^{(t)}}$ indicates the proportion of responses that have met or exceeded the quality benchmark. Formally,

$$V_{\pi^{(t)}} = \frac{1}{(1-\Phi)\cdot N\cdot K} \sum_{j=1}^{(1-\Phi)\cdot N\cdot K} V_{\pi^{(t)}}^{j}, \quad (3)$$

$$C_{\pi^{(t)}} = \frac{1}{(1-\Phi)\cdot N\cdot K} \sum_{j=1}^{(1-\Phi)\cdot N\cdot K} \mathbb{1}(V_{\pi^{(t)}}^{j} \geq \zeta), \quad (4)$$

$$V_{\pi^{(t)}}^{j} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{E}_{\text{LLM}}(\text{prompt}^{(eval)}, d_j), \quad (5)$$

$$d_j \sim \{(\mathcal{I}_i^{(0)}, \hat{\mathcal{R}}_i, \mathcal{R}_i^{(0)})\}_{i=1}^{(1-\Phi)\cdot N\cdot K}, \quad (6)$$

where $V_{\pi^{(t)}}$ and $C_{\pi^{(t)}}$ denote the performance of local LLM at t-th cyclicality, $\mathbb{1}(\cdot)$ is an indicator function, ζ denotes a threshold score of $C_{\pi^{(t)}}$, $\text{prompt}^{(eval)}$ is the system prompt used for LLM-as-a-Judge.

3.5 Data Synthesis

If the performance of fine-tuned local LLM $V_{\pi^{(t)}}$ or $C_{\pi^{(t)}}$ fails to reach or surpass the predetermined evaluation threshold ε of specific tasks, it indicates that fine-tuned local LLM's capabilities are insufficient for the tasks at hand. Thus, the local LLM cannot yet serve as a substitute for the service LLM and necessitates further refinement. To achieve this, we utilize service LLMs to generate additional synthetic datasets for fine-tuning local LLM in the next cyclicality. To maintain the consistency of data distribution of coverage dataset $\mathcal{D}^{(0)}$ constructed from real-world scenarios, we employ the train subsets $\mathcal{D}_{train}^{(0)}$ as seeds and apply the same framework (Wang et al., 2023b; Taori et al., 2023a) for synthetic dataset generation. During synthetic dataset generation, we perform data deduplication to exclude identical samples from $\mathcal{D}' = \{\mathcal{D}_{train}^{(0)}, \{\mathcal{D}_{synth}^{(1)}, \mathcal{D}_{synth}^{(2)}, \ldots, \mathcal{D}_{synth}^{(t-1)}\}\}$ and filter out low-quality samples based on carefully designed rules. Finally, we conduct rigorous data decontamination for the synthetic dataset to remove samples that closely resemble those in the test subset $\mathcal{D}_{test}^{(0)}$ of the coverage dataset. Formally, the data synthesis stage can be formulated as

$$\mathcal{D}_{synth}^{(t)} \leftarrow \bigcup \psi(\mathcal{D}_{synth}^{(t)}, \mathcal{D}', \mathcal{D}_{test}^{(0)}), \quad (7)$$

$$\mathcal{D}_{synth}^{(t)} \sim \mathcal{S}_{\text{LLM}}(\text{prompt}^{(synth)}, seed), \quad (8)$$

$$seed \sim \mathcal{D}_{train}^{(0)}, \text{for } V_{\pi^{(t)}} < \varepsilon \text{ or } C_{\pi^{(t)}} < \varepsilon, \quad (9)$$