arXiv:2408.13467v1 [cs.LG] 24 Aug 2024

LlamaDuo: LLMOps Pipeline for Seamless Migration from Service LLMs to Small-Scale Local LLMs

Chansung Park[*♢] **Fan Wang**[*♡] **Juyong Jiang**[*♡] **Sayak Paul**[♠]

Jing Tang[†♡♣] **Sunghun Kim**[♡♣]

Electronics and Telecommunications Research Institute[♢]
The Hong Kong University of Science and Technology (Guangzhou)[♡]
The Hong Kong University of Science and Technology[♣]
Hugging Face[♠]
{deep.diver.csp,csjuyongjiang,csfanwang,spsayakpaul}@gmail.com
{jingtang,hunkim}@ust.hk

Abstract

The widespread adoption of cloud-based proprietary large language models (LLMs) has introduced significant challenges, including operational dependencies, privacy concerns, and the necessity of continuous internet connectivity. In this work, we introduce an LLMOps pipeline, "LlamaDuo", for the seamless migration of knowledge and abilities from service-oriented LLMs to smaller, locally manageable models. This pipeline is crucial for ensuring service continuity in the presence of operational failures, strict privacy policies, or offline requirements. Our LlamaDuo involves fine-tuning a small language model against the service LLM using a synthetic dataset generated by the latter. If the performance of the fine-tuned model falls short of expectations, it is enhanced by further fine-tuning with additional similar data created by the service LLM. This iterative process guarantees that the smaller model can eventually match or even surpass the service LLM's capabilities in specific downstream tasks, offering a practical and scalable solution for managing AI deployments in constrained environments. Extensive experiments with leading-edge LLMs are conducted to demonstrate the effectiveness, adaptability, and affordability of LlamaDuo across various downstream tasks. Our pipeline implementation is available at https://github.com/deep-diver/llamaduo.

[*]Equal contribution.
[†]Corresponding author.

Preprint. Under review.

1 Introduction

The emergence of LLMs has significantly transformed a myriad of tasks and domains [7, 11, 3, 35, 47, 17, 18]. In particular, cloud-based proprietary LLMs, referred to as service models, such as GPT-4 [3], Gemini 1.5 [11], and Claude 3 [4], have exhibited exceptional capabilities when compared to their smaller, open-source counterparts [5]. A notable survey involving 70 AI industry leaders from diverse enterprises reveals that approximately 80% of the enterprise market share is dominated by closed-source platforms, with a significant portion of this share attributed to OpenAI [39].

However, the increasing reliance on cloud-based service models presents significant challenges in terms of operational dependencies [3], privacy concerns [42], and accessibility challenges [30]. These challenges manifest in various ways, including potential service disruptions, heightened risks to data privacy due to the transmission of sensitive information to external providers, mandatory internet connectivity for utilization, and inconsistencies stemming from updates to service providers' LLMs [14, 47]. Additionally, the transition from proof-of-concept (PoC) development utilizing service LLMs to deployment with local models frequently leads to diminished prompt effectiveness owing to differences between models, subsequently resulting in a suboptimal experience for end-users [26, 23]. To address these concerns and ensure consistent service delivery, it is imperative to develop smaller, locally manageable LLMs that can operate independently of cloud-based infrastructures.

Recent studies have demonstrated that the strategic fine-tuning of smaller and open-source LLMs with high-quality synthetic data [40, 43] generated by service LLMs can achieve performances that are on par with, or even surpass, those of proprietary LLMs in specific downstream tasks [6, 33, 22, 2, 49]. Motivated by these findings, we introduce an LLMOps pipeline namely LlamaDuo designed to automatically facilitate the seamless migration from service-oriented LLMs to smaller, locally manageable models without the need for human intervention. Our pipeline begins with utilizing a task-specific initial dataset, referred to as the coverage dataset, to fine-tune a smaller open-source LLM. The performance of fine-tuned local LLMs is evaluated using a service LLMs-as-a-Judge strategy [48]. If the performance of the fine-tuned model falls short of expectations, we improve it by iteratively fine-tuning on additional synthetic data generated by the service LLM. LlamaDuo ensures that the smaller model is capable of eventually matching or even surpassing the service LLM's performance in specific downstream tasks, offering superior long-term economic advantages. Therefore, it presents a practical and scalable solution for managing AI deployments in environments where resources are limited.

We conduct extensive experiments and analysis across a range of tasks, including summarization, classification, coding, and closed QA, with most popular service LLMs such as GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash, as well as local LLMs, including Gemma 2B and 7B, Mistral 7B, and LLaMA3 8B, to demonstrate that our LlamaDuo guarantees the smaller local LLMs possesses the potential to eventually match or even exceed the performance of service LLMs in specific downstream tasks. We open-source our codebase, enabling users to seamlessly migrate knowledge and abilities from service LLMs to small-scale local LLMs in constrained environments. We release all synthetic datasets and model checkpoints on Hugging Face [3], empowering the community to enrich the capabilities of small LLMs and facilitating future research. To summarize, our key contributions are:

- We introduce the LlamaDuo, an efficient and affordable LLMOps pipeline designed to facilitate seamless migration from service-oriented LLMs to smaller, locally manageable models without the need for human intervention, ensuring service continuity in constrained environments.
- We iteratively employ task-specific synthetic data produced by service LLMs to guarantee that LlamaDuo enables the smaller model to eventually match or even surpass the performance of service LLM in specific downstream tasks, thus meeting specific task requirements.
- We substantiate the pipeline's robust performance and adaptability in real-world context through comprehensive experiments across a range of typical tasks, employing most popular service LLMs as synthetic data generators and judges for well-know small-scale local LLMs.
- We emphasize the significant economic advantages of LlamaDuo for investing in smaller, locally manageable LLMs and their deployment for sustained use, as opposed to the transient benefits derived from the token-based API usage of service LLMs.

[3] https://huggingface.co/llama-duo

2 Related Work

2.1 Alignment with Instruction Tuning

LLMs pretrained on massive corpora demonstrate remarkable capabilities across a wide range of tasks [47]. Despite their capabilities, a notable challenge with LLMs is their misalignment with user instructions, which limits their practical application in real-world scenarios [43, 40]. The misalignment stems from the initial pretraining objective of LLMs, which focuses on minimizing generation errors rather than adhering to human instructions [27, 8]. To solve the mismatch, instruction tuning is proposed, which enables LLMs to complete diverse tasks from instructions without significant computational resources or alterations to the model's architecture[21, 25, 34]. Specifically, instruction tuning involves supplementary training of pretrained LLMs with datasets structured as instruction-output pairs [46]. The efficacy of instruction tuning is largely contingent upon the quality and diversity of the instruction datasets employed [37]. However, the process of curating high-quality, diversified data is fraught with challenges, including the extensive time required for creation, privacy concerns, high costs, and the need for substantial human labor [43, 20]. In response to these challenges, recent studies have explored innovative methods for constructing instruction datasets, notably the utilization of LLMs for data synthesis [20].

2.2 LLM-synthetic Instruction Data

LLMs have demonstrated an unprecedented ability to comprehend and execute natural language instructions [27, 8, 35]. This ability is attributed to the process of training LLMs using substantial instruction datasets [40]. However, acquiring massive instruction datasets is challenging due to data scarcity, privacy issues, low data quality, and prohibitive costs associated with manual data curation [1, 43, 20]. Given these constraints, recent studies probe into utilizing LLMs to automatically generate synthetic instruction data [41, 9, 34]. 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 [38]. Leveraging LLMs to generate data can significantly reduce the costs and time for data curation [20], while simultaneously improving the efficacy of the fine-tuned models for designated downstream tasks [44, 28, 13, 31, 32].

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

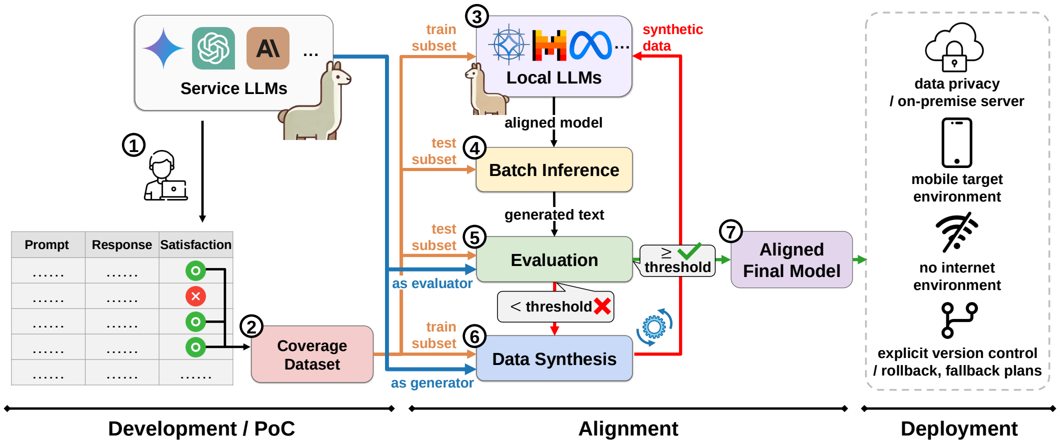

Figure 1: The LLMOps pipeline namely LlamaDuo for migrating from service LLMs to small-scale local LLMs involves three phases. In the Development/PoC phase, ① users manually engineer prompts to interact with service LLMs and ② collect satisfying (prompt, response) pairs into train and test datasets. 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.

initial cyclicality of the pipeline, the selected local LLM is fine-tuned on the train subsets $\mathcal{D}^{(0)}_{train}$ 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 $\{\mathcal{D}^{(1)}_{synth}, \mathcal{D}^{(2)}_{synth}, \ldots, \mathcal{D}^{(t-1)}_{synth}\}$ generated from service LLMs with $\mathcal{D}^{(0)}_{train}$ as seed dataset. Consequently, when $t \geq 1$, the objective of the fine-tuning phase can be formulated as

$$\mathcal{L}_{\text{SFT}}(\pi^{(t)}, \mathcal{D}^{(t)}) = -\mathbb{E}_{\mathcal{I}^{(t)} \sim \mathcal{D}^{(0)}_{train}, \mathcal{R}^{(t)} \sim \{\mathcal{D}^{(0)}_{train}, \{\mathcal{D}^{(\tau)}_{synth}\}^{t-1}_{\tau=1}\}} \left[\log P_{\pi^{(t-1)}}(\mathcal{R}^{(t)} \mid \mathcal{I}^{(t)}) \right], \quad (1)$$

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 $\mathcal{D}^{(0)}_{test}$ 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 $\{(\mathcal{I}^{(0)}_i, \hat{\mathcal{R}}_i)\}^{(1-\Phi)\cdot N \cdot K}_{i=1}$. Formally,

$$\hat{\mathcal{R}}_k \sim \pi^{(t)}(\mathcal{R}^{(0)} \mid \mathcal{I}^{(0)}), \text{for } \ k \in \{1, 2, \ldots, K\}, \quad (2)$$

$$\mathcal{I}^{(0)} \sim \mathcal{D}^{(0)}_{test}. \quad (3)$$

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 $\{(\mathcal{I}^{(0)}_i, \hat{\mathcal{R}}_i)\}^{(1-\Phi)\cdot N \cdot K}_{i=1}$. Following the works [48, 45], 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 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,

$$V_{\pi^{(t)}} = \frac{1}{(1-\Phi)\cdot N\cdot K} \sum_{j=1}^{(1-\Phi)\cdot N\cdot K} V_{\pi^{(t)}}^{j}, \tag{4}$$

$$C_{\pi^{(t)}} = \frac{1}{(1-\Phi)\cdot N\cdot K} \sum_{j=1}^{(1-\Phi)\cdot N\cdot K} \mathbb{1}(V_{\pi^{(t)}}^{j} \geq \zeta), \tag{5}$$

$$V_{\pi^{(t)}}^{j} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{E}_{\text{LLM}}(\text{prompt}^{(eval)}, d_j), \tag{6}$$

$$d_j \sim \{(\mathcal{I}_i^{(0)}, \hat{\mathcal{R}}_i, \mathcal{R}_i^{(0)})\}_{i=1}^{(1-\Phi)\cdot N\cdot K}, \tag{7}$$

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 [40, 33] for synthetic dataset generation. During synthetic dataset generation, we perform data deduplication to exclude identical samples from $\{\mathcal{D}_{train}^{(0)}, \{\mathcal{D}_{synth}^{(1)}, \mathcal{D}_{synth}^{(2)}, \ldots, \mathcal{D}_{synth}^{(t-1)}\}\}$ 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

$$\mathcal{D}_{synth}^{(t)} \leftarrow \bigcup \psi(\mathcal{D}_{synth}^{(t)}, \{\mathcal{D}_{train}^{(0)}, \{\mathcal{D}_{synth}^{(\tau)}\}_{\tau=1}^{t-1}\}, \mathcal{D}_{test}^{(0)}), \tag{8}$$

$$\mathcal{D}_{synth}^{(t)} \sim \mathcal{S}_{\text{LLM}}(\text{prompt}^{(synth)}, seed), \tag{9}$$

$$seed \sim \mathcal{D}_{train}^{(0)}, \text{for } \; V_{\pi^{(t)}} < \varepsilon \; \text{ or } \; C_{\pi^{(t)}} < \varepsilon, \tag{10}$$

where $\bigcup \psi(\cdot,\cdot,\cdot)$ represent a series of data post-processing operations, $\mathcal{D}_{synth}^{(t)}$ denotes synthetic data generated from service LLMs at t-th cyclicality, $\mathcal{S}_{\text{LLM}}$ and $\text{prompt}^{(synth)}$ are the service LLM and system prompt used for the data synthesis, respectively.

4 Experiments

In this section, we present a comprehensive evaluation of our LlamaDuo across a series of settings, demonstrating its robust performance, adaptability, and affordability in real-world scenarios. We first evaluate the performance of our pipeline across summarization, classification, coding, and closed QA tasks to underscore its efficacy and versatility. We further examine the impact of synthetic dataset volume and the choice of service model as data generator and judge on the performance of fine-tuned models, thereby showing the robustness of the LlamaDuo. Lastly, we analyze the cost-effectiveness associated with deploying a fine-tuned LLM via our pipeline, emphasizing its long-term economic advantages.

4.1 Experimental Settings

Tasks and coverage dataset. We select four categories of downstream tasks-summarization, classification, coding, and closed QA-based on their prevalent use and relevance to the operational scope of service LLMs. We utilize the open-source "No Robots" [29] dataset as the coverage dataset. This coverage dataset consists of 10K high-quality prompt and response pairs across 10 categories, crafted by expert annotators. Specifically, we utilize four subsets of the coverage dataset, each corresponding to our targeted tasks. These subsets serve as seeds for generating synthetic data that can closely align with user expectations for LLM interactions.

Service and local LLMs. Considering the API cost effectiveness, rate limit, and model utility, we select most popular service LLMs including GPT4o by OpenAI [4], Claude 3 Sonnet by Anthropic [5], and Gemini 1.5 Flash by Google [6] to serve as synthetic data generator and judges. As for the small-scale local LLMs to be fine-tuned, we opt for the open-source Gemma 2B and 7B [12], Mistral 7B [16], and LLaMA3 8B [24] as the base models. This selection is motivated by our aim to rigorously evaluate the efficacy and adaptability of our proposed pipeline across diverse settings. The varying scales of base models facilitate a nuanced comparison, allowing us to assess the impact of model scale on performance improvements. However, as a model-agnostic LLMOps pipeline, our LlamaDuo can be generalized to various forms of service and local LLMs beyond the aforementioned models.

4.2 Implementation Details

We implement LlamaDuo using PyTorch and conduct experiments on $8 \times$ A100 GPUs, with detailed hyper-parameter configurations presented in Appendix B.

Synthetic dataset by service LLMs. To generate high-quality synthetic datasets, we leverage the capabilities of advanced service LLMs, including GPT-4o, Claude 3 Sonnet, and Gemini 1.5 Flash. We utilize the seeds selected from the train subset of the coverage dataset to prompt these service LLMs to generate datasets, each comprising 300k samples. The specific prompt for data generation is presented in Figure 6 of Appendix A. Subsequently, we employ Locality-Sensitive Hashing (LSH) with MinHash and Rouge scoring mechanisms for data deduplication. Specifically, the LSH MinHash can efficiently identify and remove duplicate data samples, while the Rouge scoring mechanism ensures that the curated data exhibits high-quality and meaningful variations. After that, we acquire 256k samples for summarization tasks and 128k for other tasks.

Fine-tuning Local LLMs. We proceed to fine-tune the small local LLMs with 2^nk, $n \in \{0, 1, \dots, 8\}$ volumes of the synthetic dataset. To efficiently customize local LLM for a specific downstream task within constrained environments, we leverage QLoRA [10] for parameter-efficient fine-tuning (PEFT) with superior cost-effectiveness. The detailed configurations, which are tailored according to dataset sizes and tasks, can be found in Appendix B.

Batch inference. Each fine-tuned local model is prompted to generate $K = 4$ distinct responses, with each prompt sampled from the test subsets of the coverage dataset. To ensure fair comparisons, we maintain a consistent batch inference configuration across all fine-tuned models. The detailed configuration is depicted in Appendix B.

Service LLMs as judges. Following [48], we employ three service LLMs as judges and use pairwise comparison and single answer grading strategies to evaluate the response quality of the fine-tuned local LLMs. The corresponding prompts used for evaluation are given in Figure 5 of Appendix A. We utilize similarity and precision metrics for the assessment of the fine-tuned LLMs' performance. The similarity metric assesses the degree of correspondence between the generated responses and the ground truth, while the precision metric evaluates the accuracy of the match between the input prompts and their corresponding responses. To ensure reliability and mitigate inherent biases in the results, both metrics are quantified on a 0 to 100 scale, with each sample undergoing evaluation $M = 10$ times. The score of coverage percentage is set to $\zeta \in \{50, 70\}$. The mean scores and coverage percentage with score of $\geq \zeta$ are taken as the final result of the model performance.

[4] https://openai.com/index/hello-gpt-4o

[5] https://www.anthropic.com/news/claude-3-5-sonnet

[6] https://deepmind.google/technologies/gemini/flash

Table 1: Performance of the service LLMs and local LLMs fine-tuned on 128K synthetic dataset produced by GPT4o, evaluated by GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash as judges on test subsets of coverage dataset. Each entry is presented as mean score / coverage percentage (%) with 50 score. Scores in **Red** are the best results from service LLMs, while the scores in **Blue** are the best results from local LLMs. Perf. Matching represents performance matching which is defined as the best performance of the local LLM divided by the service LLM, where the best results are in **Pink**.

Task	Model	GPT4o		Claude 3 Sonnet		Gemini 1.5 Flash	
		Precision↑	Similarity↑	Precision↑	Similarity↑	Precision↑	Similarity↑
Summarization	GPT4o	**90.71** / **97 %**	**82.00** / **95%**	93.25 / **100%**	**86.60** / **100%**	**87.10** / **100%**	**67.45** / 85%
	Claude 3 Sonnet	88.04 / **97%**	78.18 / **95%**	**93.39** / **100%**	85.55 / **100%**	86.70 / **100%**	64.10 / 80%
	Gemini 1.5 Flash	87.90 / 96%	79.14 / **95%**	91.95 / **100%**	85.05 / **100%**	85.65 / 98%	66.45 / **89%**
	Gemma 2B	57.60 / 64%	54.49 / 61%	74.89 / 86%	64.09 / 73%	61.90 / 78%	42.15 / 38%
	Gemma 7B	73.54 / 85%	68.58 / 85%	86.19 / **99%**	77.41 / 94%	**74.59** / **95%**	**53.92** / **65%**
	Mistral 7B	**76.38** / **93%**	69.65 / **88%**	86.20 / **99%**	**78.44** / **96%**	72.74 / **95%**	50.15 / 54%
	LLaMA3 8B	75.67 / 88%	**70.54** / 86%	**87.02** / **99%**	78.42 / 93%	72.74 / 91%	52.23 / 64%
	Perf. Matching↑	84.20% / 95.88%	86.02% / 92.63%	**93.18%** / **99%**	**90.58%** / **96%**	85.64% / 95%	79.94% / 73.03%
Classification	GPT4o	83.62 / **94%**	**74.45** / 80%	87.50 / 92%	72.28 / 72%	82.68 / 94%	63.06 / 67%
	Claude 3 Sonnet	82.32 / 92%	71.53 / **81%**	**92.89** / **100%**	75.07 / **81%**	**87.34** / **97%**	**67.18** / **80%**
	Gemini 1.5 Flash	**85.43** / **94%**	72.73 / **81%**	89.03 / 94%	**77.96** / **81%**	83.35 / 94%	64.25 / 78%
	Gemma 2B	58.47 / 58%	52.76 / 50%	69.98 / 73%	56.31 / 58%	62.17 / 62%	48.54 / 50%
	Gemma 7B	70.73 / 69%	64.67 / 62%	78.78 / 81%	67.76 / 69%	70.73 / 75%	59.77 / 59%
	Mistral 7B	67.53 / 70%	61.65 / 67%	76.01 / 80%	64.43 / 70%	67.90 / 73%	54.27 / 53%
	LLaMA3 8B	**81.64** / **88%**	**78.02** / **77%**	**89.20** / **94%**	**82.18** / **88%**	**83.63** / **94%**	**72.54** / **73%**
	Perf. Matching↑	95.56% / 93.62%	104.80% / 95.06%	**96.03%** / 94%	105.41% / **108.64%**	95.75% / **96.91%**	**107.98%** / 91.25%
Coding	GPT4o	**90.31** / **100%**	75.18 / 92%	**94.57** / **100%**	86.32 / **100%**	**90.78** / **100%**	58.43 / 62%
	Claude 3 Sonnet	88.76 / **100%**	75.23 / **94%**	93.82 / **100%**	**87.42** / **100%**	89.84 / **100%**	60.46 / 69%
	Gemini 1.5 Flash	88.51 / 98%	**75.62** / 91%	93.59 / **100%**	82.92 / 97%	90.62 / **100%**	**64.21** / **84%**
	Gemma 2B	62.31 / 70%	56.48 / 66%	80.92 / 89%	67.24 / 78%	72.98 / 89%	44.08 / 50%
	Gemma 7B	**80.56** / **92%**	**71.92** / **89%**	**90.47** / **100%**	**80.26** / **92%**	**84.66** / **100%**	**61.23** / **72%**
	Mistral 7B	68.32 / 77%	61.01 / 69%	81.25 / 92%	69.10 / 83%	72.39 / 86%	45.25 / 50%
	LLaMA3 8B	77.47 / 88%	69.46 / 88%	83.97 / 94%	73.51 / 88%	75.55 / 89%	51.10 / 58%
	Perf. Matching↑	89.20% / 92%	95.11% / **94.68%**	**95.66%** / **100%**	91.81% / 92%	93.26% / **100%**	**95.36%** / 85.71%
Closed QA	GPT4o	**95.45** / **100%**	84.23 / 93%	97.21 / **100%**	92.56 / **100%**	93.58 / **100%**	75.58 / 85%
	Claude 3 Sonnet	94.03 / **100%**	85.28 / **100%**	97.60 / **100%**	93.95 / **100%**	93.66 / **100%**	76.33 / 92%
	Gemini 1.5 Flash	94.63 / **100%**	**87.43** / 95%	**98.25** / **100%**	**97.41** / **100%**	**95.00** / **100%**	**85.66** / **97%**
	Gemma 2B	67.25 / 65%	67.41 / 67%	80.22 / 85%	70.20 / 73%	70.33 / 73%	59.68 / 62%
	Gemma 7B	81.85 / **88%**	79.02 / **85%**	**88.83** / **93%**	83.95 / 87%	**82.51** / **93%**	72.24 / 75%
	Mistral 7B	**83.63** / 87%	**81.36** / **85%**	88.25 / **93%**	**84.77** / **88%**	82.04 / 85%	**73.95** / **78%**
	LLaMA3 8B	75.55 / 78%	72.62 / 77%	86.03 / 88%	77.64 / 80%	79.09 / 85%	68.78 / 75%
	Perf. Matching↑	87.62% / 88%	**93.06%** / 85%	**90.41%** / **93%**	87.02% / **88%**	86.85% / **93%**	86.33% / 80.41%

4.3 Experimental Results

This section delves into the effectiveness and adaptability of the LlamaDuo pipeline, spanning different tasks with varying degrees of complexity, including summarization, classification, coding, and closed QA. We utilize GPT-4o, Claude 3 Sonnet, and Gemini 1.5 Flash as judges to evaluate the fine-tuned model performance on test subsets of the coverage dataset. As demonstrated in Table 1, the fine-tuned local LLMs, despite their significantly smaller scale, achieve comparable performance on diverse tasks compared to much larger service LLMs. For instance, in the summarization task, LLaMA3 8B achieved a comparable precision score of 87.02 / 99%, compared to GPT4o's score of 93.25 / 100%, Claude 3 Sonnet's score of 93.39 / 100%, and Gemini 1.5 Flash's score of 91.95 / 100%, with Claude 3 Sonnet serving as judge. These results underscore the efficacy of LlamaDuo in seamlessly transferring knowledge and capabilities from service LLMs to smaller local LLMs without a substantial decrease in performance.

In Table 1, we observe distinct performance across four fine-tuned models when applied to different tasks. Specifically, Mistral 7B stands out in summarization tasks, achieving the best performance in 7 out of 12 cases. Moreover, LLaMA3 8B consistently outperforms competitors across all metrics and evaluators in the classification task. Conversely, in coding tasks, Gemma 7B is identified as the leading model, excelling across all metrics and evaluations. Mistral 7B shows superior performance in the closed QA task, leading in 8 out of 12 cases. Within the realm of service LLMs, Claude 3 Sonnet and Gemini 1.5 Flash demonstrates exceptional performance in classification and closed QA tasks, securing the best results in 8 and 10 out of 12 cases, respectively. Lastly, GPT4o emerges as the leading model in summarization and coding tasks, achieving the best performance in 10 and 7 out of 12 cases, respectively.

Notably, although Gemma 2B exhibits inferior performance compared to larger 7B models overall, the disparity in results is not markedly substantial, with Gemma 2B attaining closely comparable performance in certain tasks. For example, in closed QA tasks, Gemma 2B secures a mean precision score of 80.22, while Gemma 7B achieves 88.83, Mistral 7B reaches 88.25, and LLaMA3 8B obtains 86.03, as evaluated by Claude 3 Sonnet. This observation lends further support to the notion that

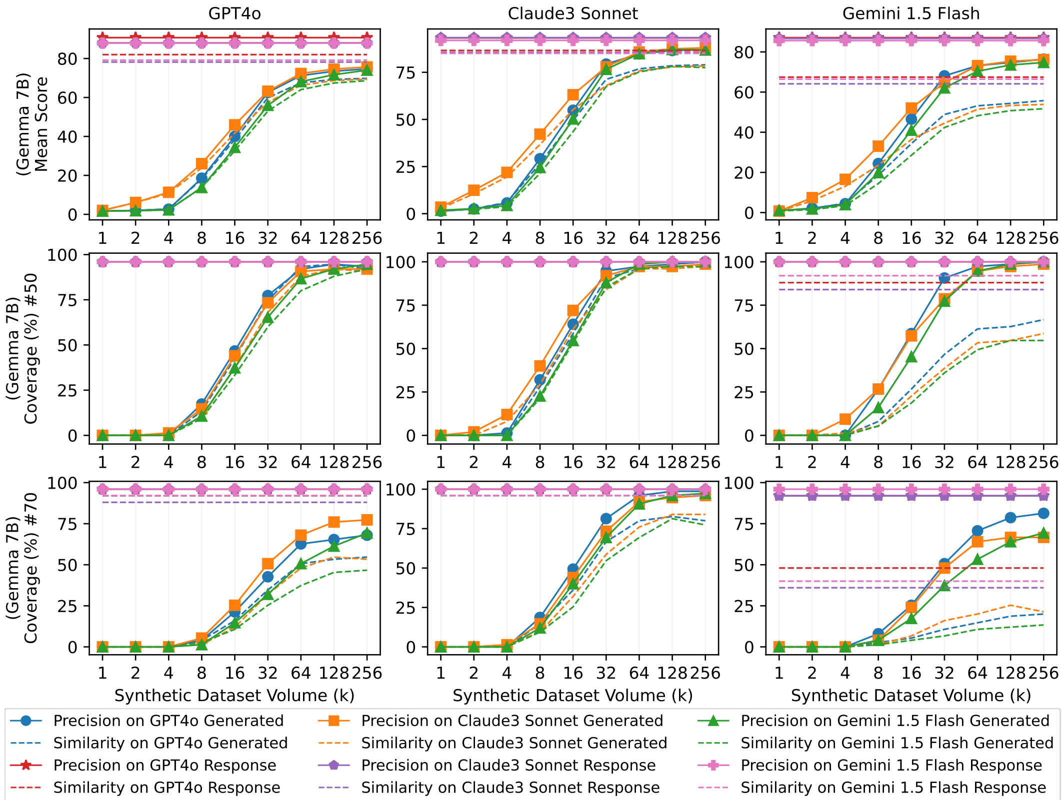

Figure 2: Performance of Gemma 7B fine-tuned on varied volumes of synthetic dataset producted by various service LLMs including GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash. The first to third columns represent the performance of the model evaluated by GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash as judges, respectively. The first row show mean scores, while the second and third rows show the coverage percentage with scores of 50 and 70, respectively.

through the strategic fine-tuning of smaller local LLMs on synthetic datasets via the LlamaDuo, it is possible to closely approximate the performance of their larger counterparts. Consequently, it offers increased flexibility and solutions for users and scenarios with budgetary considerations. Additional experimental results are presented in Appendix C.

4.4 In-depth LLMOps Pipeline Analysis

In this section, we conduct an in-depth analysis of LlamaDuo through summarization task. Notably, the experimental findings exhibit a consistent pattern across various tasks, underscoring the generalizability of LlamaDuo.

Impact of synthetic dataset volume. We explore how the volume of synthetic dataset influences the performances of fine-tuned local LLMs, aiming to elucidate a scaling law [19, 15] on how the performance of fine-tuned models changes as the number of synthetic dataset samples increases. Overall, the Gemma 7B model exhibit consistent performance improvements and come closer to the performance of service LLMs with increasing volumes of synthetic data, as assessed through precision and similarity metrics by diverse evaluators, as depicted in Figure 2. This suggests that fine-tuning local LLMs with synthetic data, which mimics the characteristics and patterns of real-world data, can bring the same effect as actual data. Thus, it paves a new way to eliminate the challenges of data scarcity, privacy concerns, and high costs associated with crafting data [20]. Notably, we observe that the synthetic data generated by Claude 3 Sonnet results in the highest-performing models, outperforming those fine-tuned with data produced by GPT4o and Gemini 1.5 Flash, in descending order. Moreover, when the synthetic dataset volume ranges from 64k to 256k, the Gemma 7B model reach the performance saturation point and achieve performance that is much closer to, or equal to,

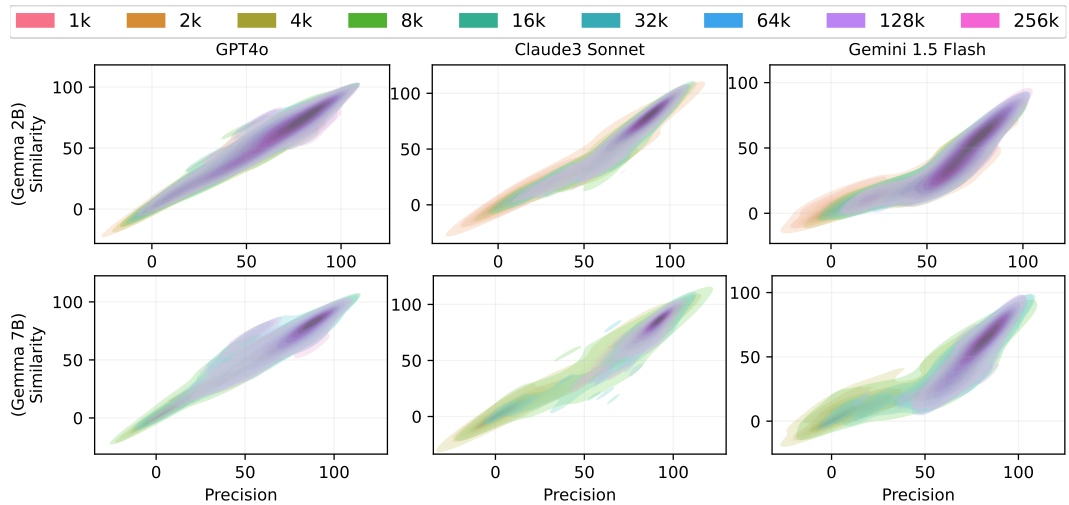

Figure 3: The KDE Plots of Precision v.s. Similarity by varied synthetic dataset volumes with 2^nk, $n \in \{0, 1, \ldots, 8\}$ and various evaluators with GPT4o, Claude 3 Sonnet, Gemini 1.5 Flash as judges from first to third columns, while the first and second rows represent the results of Gemma 2B (first row) and Gemma 7B (second row), respectively.

that of service LLMs. This demonstrates the efficacy of our LlamaDuo in enabling smaller models to replicate or even surpass the performance of service LLMs in specific downstream tasks.

Impact of service LLMs as data generator and judge. As shown in Figure 2, we observe that the choice of service LLM for data generation does not significantly impact the performance of the fine-tuned models. Specifically, (1) a consistent trend of performance enhancement is observed with the increased volume of synthetic data, irrespective of the service LLM employed for data generation; (2) the local LLMs fine-tuned on synthetic data generated by GPT4o and Claude3 Sonnet typically lead to slightly better performance than those by Gemini 1.5 Flash. On the other hand, employing different service LLMs as judges manifests a more pronounced impact on the performance of the fine-tuned local LLMs. Overall, GPT4o and Gemini 1.5 Flash emerge as more rigorous judges compared to Claude 3 Sonnet, with Gemini 1.5 Flash assigning notably lower similarity scores. Moreover, we observe that in data sparsity scenarios (1k to 4k), the type of evaluators has minimal influence on the performance of the Gemma 7B model, suggesting that larger local LLMs exhibit diminished sensitivity to the choice of service LLM as a judge. To qualitatively demonstrate the differences when using various types of service LLMs as evaluators, Figure 3 presents the results as KDE plots, characterized by the dataset volume. We observe that GPT-4 maintains consistency in its evaluations across both similarity and precision metrics. In contrast, Claude 3 Sonnet is found to be more lenient in scoring, while Gemini 1.5 Flash assigns higher precision scores but significantly lower similarity scores. This underscores the importance of strategically aligning the selection of service LLMs with specific task requirements.

4.5 Cost Effectiveness of Long-term Deployment

In this section, we elucidate the cost-effectiveness of our proposed LlamaDuo pipeline, by conducting a long-term operational cost comparison between the fine-tuning of the small LLMs (Gemma 7B) and the token-based API usage of service LLMs (GPT4o).

In the context of local LLM deployment, the QLoRA fine-tuning process of Gemma 7B, utilizing a dataset containing 256K samples, necessitates approximately one hour to complete a single experiment on $8 \times$ A100 GPUs. This process incurs an estimated cost of \$50, based on the price provided by Google Cloud Platform [7]. Accounting for multiple iterations of hyperparameter optimization, we estimate that the total fine-tuning cost remains below \$800, which is deemed to be negligible. Deploying a single instance of the Gemma 7B model with support for a 1024 context length necessitates 24GB of GPU memory, making the L4 GPU an appropriate choice. Depending on the projected

[7] https://cloud.google.com/compute/all-pricing#accelerator-optimized

Table 2: Monthly operational cost comparison between Gemma 7B and GPT-4o under different workloads. For GPT4o, input and output token counts are represented in the format input/output.

	Light Workload		Heavy Workload	
	Gemma 7B	GPT-4o	Gemma 7B	GPT-4o
Fine-tuning	Cloud $800	- -	Cloud $800	- -
Serving Specs.	1 x L4 $2,539	300M/30M $1,950	8 x L4 $20,312	1500M/150M $9,750
Serving Elec.	165 kWh $30	- -	1319 kWh $240	- -
2 Months	$3,369	$3,900	$21,592	$19,500
12 Months	$3,699	$23,400	$23,992	$117,000

workload, the Gemma 7B model can be deployed either on a single server equipped with one L4 GPU ($2,539) or across eight servers, each with one L4 GPU, with each server hosting a replica of the model instance ($20,312). In addition, the power consumption for each server is approximately $30 per month.

For GPT4o, as of August 2024, the pricing is $5 and $15 per million tokens for input and output, respectively. We estimate that a light workload, utilizing 10 million input tokens and 1 million output tokens per day, incurs a daily cost of $65. Conversely, a heavy workload, consuming 50 million input tokens and 10 million output tokens per day, is estimated to cost $325 daily. The monthly operational cost comparison between Gemma 7B and GPT-4o under different workloads is summarized in Table 2, demonstrating a significant advantage in fine-tuning and deploying a local LLM. Moreover, as depicted in Figure 4, after the first two months, the cost of using GPT-4 under both light and heavy workloads exceeds that of setting up and running a local model deployed on $1 \times$ L4 GPU and $8 \times$ L4 GPU. After one year, GPT-4's costs surpass those of deploying a local model in all scenarios. These findings highlight the substantial economic benefits of investing in local LLM fine-tuning and deployment for long-term use. Meanwhile, avoiding recurring token-based charges and maintaining control over model customization further enhances the appeal of the LlamaDuo for cost-conscious users and scenarios.

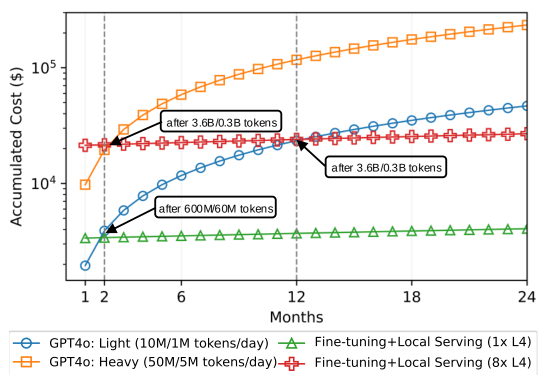

Figure 4: Long-term operational cost comparison between fine-tuning a local LLM and API-based token usage of GPT4o.

5 Conclusion

In this work, we introduce LlamaDuo, the first automatic LLMOps pipeline designed to facilitate the seamless migration from service-oriented LLMs to smaller, locally manageable models without the need for human intervention. We conduct extensive experiments and analysis across a range of tasks with most popular service and local LLMs to substantiate that our LlamaDuo guarantees the smaller local LLMs possesses the potential to match or even exceed the performance of service LLM in specific downstream tasks. Therefore, this work provides a promising research direction to maintain cloud-based LLMs' service continuity in constrained environments.

Acknowledgments

This work was supported by IITP grant funded by the Korea government(MSIT)[RS-2023-00215959, Development of Access Agnostic wired and wireless integrated optical access technology].

References

[1] Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. Privacy preserving synthetic data release using deep learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 510–526. Springer, 2019.

[2] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024.

[5] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[8] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[9] Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. Auggpt: Leveraging chatgpt for text data augmentation, 2023.

[10] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[12] Gemma Team. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[13] Z Guo, P Wang, Y Wang, and S Yu. Improving small language models on pubmedqa via generative data augmentation. *arXiv, Jul*, 12, 2023.

[14] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 2023.

[15] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[16] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[17] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[18] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*, 2024.

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[20] Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*, 2024.

[21] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR, 2023.

[22] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *The Twelfth International Conference on Learning Representations*, 2023.

[23] Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *arXiv preprint arXiv:2402.18540*, 2024.

[24] AI Meta. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2024.

[25] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, et al. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, 2023.

[26] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[27] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[28] Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*, 2020.

[29] Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No robots. https://huggingface.co/datasets/HuggingFaceH4/no_robots, 2023.

[30] Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154, 2023.

[31] Vinay Samuel, Houda Aynaou, Arijit Ghosh Chowdhury, Karthik Venkat Ramanan, and Aman Chadha. Can llms augment low-resource reading comprehension datasets? opportunities and challenges. *arXiv preprint arXiv:2309.12426*, 2023.

[32] Viktor Schlegel, Hao Li, Yuping Wu, Anand Subramanian, Thanh-Tung Nguyen, Abhinav Ramesh Kashyap, Daniel Beck, Xiaojun Zeng, Riza Theresa Batista-Navarro, Stefan Winkler, et al. Pulsar at mediqa-sum 2023: Large language models augmented by synthetic dialogue convert patient dialogues to medical records. *arXiv preprint arXiv:2307.02006*, 2023.

[33] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[34] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[35] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[36] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alexander M. Rush, and Thomas Wolf. The alignment handbook. https://github.com/huggingface/alignment-handbook, 2023. NA.

[37] Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. A survey on data selection for llm instruction tuning. *arXiv preprint arXiv:2402.05123*, 2024.

[38] Ruida Wang, Wangchunshu Zhou, and Mrinmaya Sachan. Let's synthesize step by step: Iterative dataset synthesis with large language models by extrapolating errors from small models. *arXiv preprint arXiv:2310.13671*, 2023.

[39] Sarah Wang and Shangda Xu. 16 changes to the way enterprises are building and buying generative ai. *URL: https://a16z.com/generative-ai-enterprise-2024/*, 2024.

[40] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, 2023.

[41] Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. LLM-powered data augmentation for enhanced cross-lingual performance. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[42] Xiaodong Wu, Ran Duan, and Jianbing Ni. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of Information and Intelligence*, 2(2):102–115, 2024.

[43] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

[44] Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546*, 2020.

[45] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[46] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

[47] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[48] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[49] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

A Prompt Templates

In the LlamaDuo pipeline, we employ two prompt templates that serve different purposes: one for the generation of synthetic datasets and another for the evaluation of the outputs produced by the fine-tuned LLMs.

Figure 5 illustrates the prompt template used to assess the precision and similarity of the response `$lm_response` generated by fine-tuned small-scale LLMs, based on the prompt `$instruction` and response `$human_response` from the test subset of the coverage dataset. It is important to note that the $ symbol indicates a placeholder, designed to be substituted with actual data during the runtime.

````
Given an instruction and two responses—one generated by a human and the other by a
language model—I'm seeking to evaluate how closely the language model's response
mirrors the human-generated one. Additionally, I want to assess the accuracy and
relevance of the language model's response to the original instruction.

Instruction:
```
$instruction
```

Human Response:
```
$human_response
```

Language Model Response:
```
$lm_response
```
You are quality assessor who analyzes the similarity between the Human Response and
the Language Model Response on a scale of 1 to 100, where 1 indicates no similarity and
100 indicates identical responses.

Also you analyze the Language Model Response how it accurately answers the given
Instruction on a scale of 1 to 100. Analysis MUST be rigorous and thorough. Provide the
assessment in the following JSON format:

{
  "similarity_assessment": {
    "score": [Insert similarity score here]
  },
  "precision_assessment": {
    "score": [Insert precision score here]
  }
}
````

Figure 5: Prompt template to evaluate the fine-tuned model's response.

Figure 6 shows the prompt template designed for the generation of synthetic data tailored to the summarization task while Figure 7 shows the prompt template for other tasks. Specifically, we use a prompt `$instruction` alongside its corresponding response `$response`, both sourced from the train subset of the coverage dataset, serving as an example pair. This example pair is utilized to instruct service LLMs to generate analogous data samples. In addition, our template is designed to generate multiple synthetic data samples through a singular request, thereby enhancing the efficiency of API utilization. Due to the unique features of different downstream tasks, there is no optimal prompt template that universally applies. The actual content of the prompt template is adjusted to align with the specific requirements of the task for which the synthetic dataset is being generated.

```
Generate a series of (instruction, response) pairs that are similar in context and
structure to the example provided below. Each pair should consist of a concise
instruction followed by an appropriate, detailed response. The instruction should pose a
clear task or question, while the response should provide a comprehensive answer or
solution that could be understood by someone with a basic understanding of the subject.

Example pair:
Instruction: $instruction
Response: $response

Your task is to generate more pairs that maintain this level of clarity and detail. The
topic is $topic. Write a long text of instruction by yourself, then summarize the given
instruction in a response. Ensure that the responses are informative and accurate,
suitable for an educational context.

Store the generated pairs in JSON format, with each pair as an object within an array.
Each object should have two key-value pairs: "instruction" and "response". For instance:

{
 "contents":
  [
   {"instruction": "text", "response": "text"},
   {"instruction": "text", "response": "text"},
   ...
  ]
}

Remember to maintain consistency in the format and ensure the generated pairs are
diverse and cover a broad range of subjects. You must return the response in the asked
format and you must not add any additional text in your response.
```

Figure 6: Prompt template of data synthesis for summarization tasks.

B Implementation Configuration

This section delineates the statistical information of the coverage dataset and synthetic dataset generated by service LLMs. In addition, we present the details of the training configurations of our experiments. The detailed pipeline implementation of LlamaDuo is available at https://github.com/deep-diver/llamaduo.

B.1 Coverage Datasets

The foundational coverage dataset employed in our study is the "No Robots" dataset [29]. We leverage four subsets of the coverage dataset, namely summarization, classification, coding, and closed QA, for synthetic data generation. Table 3 illustrates the initial composition of the task-specific subsets, with each initially containing approximately 300 original data points. These subsets are subsequently expanded to encompass more data points using the LlamaDuo framework. To perform an in-depth analysis of the behavior of different service LLMs, we create synthetic datasets for the summarization task by utilizing GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash. For all other tasks, we exclusively use GPT-4o, owing to budget constraints.

Table 4 presents the statistical information of the token count across each dataset. We only use data from the coverage train set for data synthesis and alignment tasks. We observe a reduction in both the average number of tokens and the standard deviation across the synthetic datasets compared to the original dataset. This is due to that the data synthesis process generates multiple synthetic data samples within a single API request.

```
Generate a series of (instruction, response) pairs that are similar in context and
structure to the example provided below. Each pair should consist of a concise
instruction followed by an appropriate, detailed response. The instruction should pose a
clear task or question, while the response should provide a comprehensive answer or
solution that could be understood by someone with a basic understanding of the subject.

Example pair:
Instruction: $instruction
Response: $response

Your task is to generate more pairs that maintain this level of clarity and detail. The
topic is $topic. Ensure that the responses are informative and accurate, suitable for an
educational context.

Store the generated pairs in JSON format, with each pair as an object within an array.
Each object should have two key-value pairs: "instruction" and "response". For instance:

{
 "contents":
  [
   {"instruction": "text", "response": "text"},
   {"instruction": "text", "response": "text"},
   ...
  ]
}

Remember to maintain consistency in the format and ensure the generated pairs are
diverse and cover a broad range of subjects. You must return the response in the asked
format and you must not add any additional text in your response.
```

Figure 7: Prompt template of data synthesis for classification, coding, and closed QA tasks.

Table 3: Volume of coverage dataset before and after LlamaDuo pipeline.

Task	Split	Before	After
Summarization(GPT4o)	train	395	256K
	test	25	100
Summarization(Claude 3 Sonnet)	train	395	256K
	test	25	100
Summarization(Gemini 1.5 Flash)	train	395	256K
	test	25	100
Classification(GPT4o)	train	334	128K
	test	16	64
Coding(GPT4o)	train	334	128K
	test	16	64
Closed QA(GPT4o)	train	245	128K
	test	15	60

B.2 Training Configurations

We utilize Hugging Face's "Alignment Handbook" [36] and the alignment recipes tailored for the Gemma models to streamline the fine-tuning process.

As outlined in Table 5, we employ QLoRA [10] to align the Gemma 2B and 7B, Mistral 7B, and LLaMA3 8B models efficiently. The QLoRA method leverages the advantages of low-rank adaptation, reducing the computational resources required for training. Throughout the alignment procedure, we incrementally adjust the rank and alpha values of LoRA, aiming to optimize the adaptation layer's capacity to match the increasing complexity of the datasets.

Table 4: Token-level statistics of the coverage and synthetic datasets.

Task	Min	Max	Avg.	Std.
Summarization (Coverage-Train)	85	2386	389	256
Summarization (Coverage-Test)	148	1150	426	245
Summarization (GPT4o)	10	2386	95	53
Summarization (Claude 3 Sonnet)	10	2386	118	64
Summarization (Gemini 1.5 Flash)	10	2386	108	62
Classification (Coverage-Train)	18	2159	207	244
Classification (Coverage-Test)	46	520	119	109
Classification (GPT4o)	6	2159	67	37
Coding (Coverage-Train)	38	6518	350	502
Coding (Coverage-Test)	49	821	317	189
Coding (GPT4o)	9	6518	151	84
Closed QA (Coverage-Train)	58	1497	320	241
Closed QA (Coverage-Test)	126	1578	411	378
Closed QA (GPT4o)	12	1701	135	59

Table 5: Detailed configurations used in the experiments.

	Configuration	Value
Common	Data Type	bfloat16
	Learning Rate Scheduler	cosine
	Max Number of Tokens	1024
	LoRA Type	QLoRA
	LoRA Dropout	0.05
1K$\sim$16K	LoRA Rank	8
	LoRA Alpha	16
32K	LoRA Rank	16
	LoRA Alpha	32
64K$\sim$256K	LoRA Rank	32
	LoRA Alpha	64

We set the maximum token as 1024 for the training phase, notwithstanding the presence of data samples exceeding this threshold. This decision is made based on a comprehensive analysis of the dataset, which indicates that data samples surpassing the token limit constitute a negligible portion of the total dataset. By imposing this limitation, we can concentrate our computational efforts on the majority of the data, thereby enhancing the efficiency of training without significantly compromising the models' ability to generalize to real-world scenarios.

The 1024-token limit, though seemingly restrictive, does not impede the performance of the aligned fine-tuned small-scale models. All fine-tuned models exhibit robust performances across the experiments, as they are trained and evaluated on data predominantly falling within the 1024-token boundary. This outcome corroborates our analysis of the data and demonstrates the efficacy of QLoRA, even within the constraints of our allocated computational budget.

C More Experimental Results

The performance of Gemma 2B fine-tuned on varied volumes of synthetic dataset producted by various service LLMs including GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash is shown in Figure 8. Table 6 showcases the performance of the service LLMs and local LLMs fine-tuned on 128K synthetic dataset produced by GPT4o, evaluated by GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash as judges on test subsets of coverage dataset with a coverage percentage (%) (70 score).

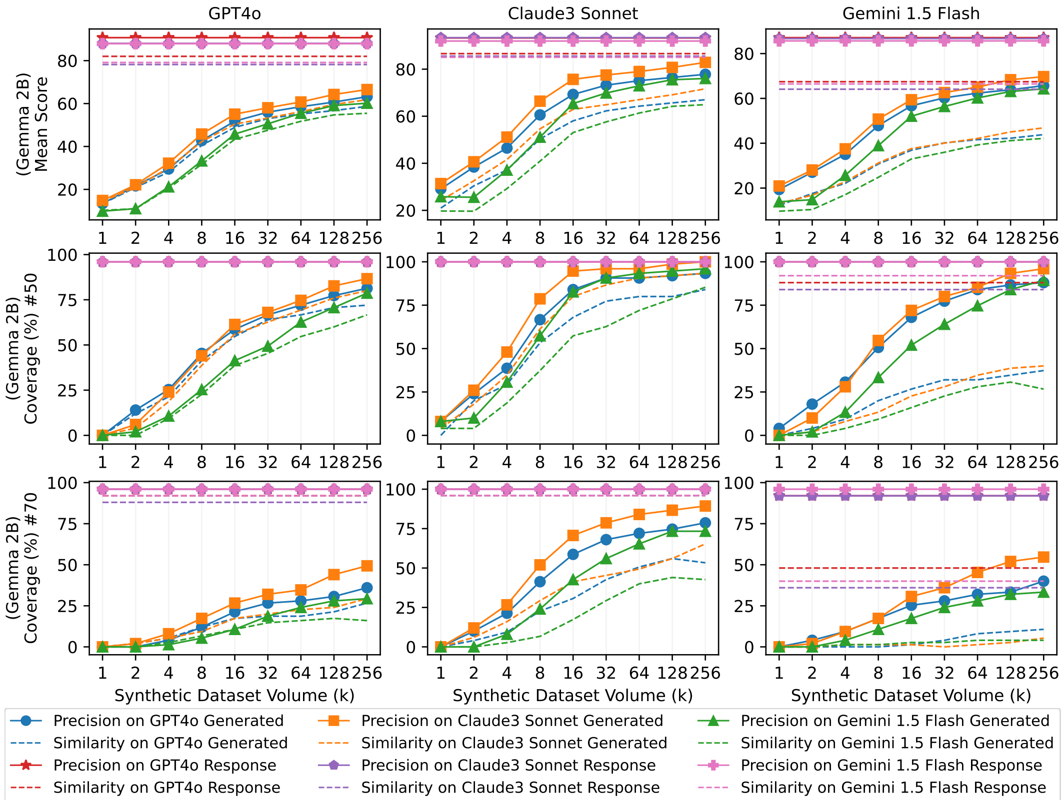

Figure 8: Performance of Gemma 2B fine-tuned on varied volumes of synthetic dataset producted by various service LLMs including GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash. The first to third columns represent the performance of the model evaluated by GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash as judges, respectively. The first row show mean scores, while the second and third rows show the coverage percentage with scores of 50 and 70, respectively.

D Case Study

This section delves into detailed case studies showcasing the enhanced capabilities of the aligned small-scale local LLMs. We use Gemma 2B and 7B models as examples to illustrate.

The cases (Figure 9-15) illustrate the performances of the aligned models across summarization, classification, coding, and closed QA tasks. Specifically, these models are tuned on distinct 128K datasets generated by GPT4o for each corresponding task. Each case provides evaluations by GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash, offering a comprehensive assessment of the precision and similarity of the models' responses.

To expand the scope of our analysis, we include two additional cases (Figure 17 and 18) to explore the summarization capabilities of the Gemma 2B and 7B models tuned with 256K synthetic datasets. These datasets are generated by GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash respectively, providing valuable insights into the models' adaptability to different training data sources.

The cases presented above demonstrate the capability of the aligned Gemma 2B and 7B models to produce high-quality responses. Additionally, the cases offer insight into how different service LLMs evaluate text. Through this comparative lens, we reveal discernible variances in judgment and assessment criteria, enriching our understanding of the models' operational dynamics.

Table 6: Performance of the service LLMs and local LLMs fine-tuned on 128K synthetic dataset produced by GPT4o, evaluated by GPT4o, Claude 3 Sonnet, and Gemini 1.5 Flash as judges on test subsets of coverage dataset. Each entry is presented as mean score / coverage percentage (%) with 70 score. Scores in **Red** are the best results from service LLMs, while the scores in **Blue** are the best results from local LLMs. Perf. Matching represents performance matching which is defined as the best performance of the local LLM divided by the service LLM, where the best results are in **Pink**

Task	Model	GPT4o		Claude 3 Sonnet		Gemini 1.5 Flash	
		Precision↑	Similarity↑	Precision↑	Similarity↑	Precision↑	Similarity↑
Summarization	GPT4o	**90.71 / 96%**	**82.00 / 89%**	93.25 / **100%**	**86.60 / 95%**	**87.10** / 92%	**67.45 / 48%**
	Claude 3 Sonnet	88.04 / 92%	78.18 / 78%	**93.39** / 99%	85.55 / **95%**	86.70 / 92%	64.10 / 36%
	Gemini 1.5 Flash	87.90 / **96%**	79.14 / 88%	91.95 / 98%	85.05 / **95%**	85.65 / **96%**	66.45 / 40%
	Gemma 2B	57.60 / 35%	54.49 / 35%	74.89 / 69%	64.09 / 50%	61.90 / 40%	42.15 / 12%
	Gemma 7B	73.54 / 65%	68.58 / 59%	86.19 / 93%	77.41 / 77%	**74.59 / 69%**	**53.92** / 22%
	Mistral 7B	**76.38** / 70%	69.65 / 56%	86.20 / 92%	**78.44** / 80%	72.74 / 62%	50.14 / 14%
	LLaMA3 8B	75.67 / **75%**	**70.54 / 69%**	**87.02 / 94%**	78.42 / **86%**	72.74 / 64%	52.23 / **25%**
	Perf. Matching↑	84.20% / 78.13%	86.02% / 77.53%	**93.18% / 94%**	**90.58% / 90.53%**	85.64% / 71.88%	79.94% / 52.08%
Classification	GPT4o	83.62 / 81%	**74.45** / 66%	87.50 / 92%	72.28 / 66%	82.68 / 80%	63.06 / 44%
	Claude 3 Sonnet	82.32 / 78%	71.53 / 70%	**92.89 / 100%**	75.07 / 73%	**87.34 / 97%**	**67.18** / 45%
	Gemini 1.5 Flash	**85.43 / 91%**	72.73 / **75%**	89.03 / 89%	**77.96 / 81%**	83.35 / 84%	64.25 / **47%**
	Gemma 2B	58.47 / 42%	52.76 / 39%	69.98 / 62%	56.31 / 47%	62.17 / 48%	48.54 / 39%
	Gemma 7B	70.73 / 55%	64.67 / 53%	78.78 / 75%	67.76 / 62%	70.73 / 61%	59.77 / 52%
	Mistral 7B	67.53 / 53%	61.65 / 47%	76.01 / 72%	64.43 / 52%	67.90 / 53%	54.27 / 45%
	LLaMA3 8B	**81.64 / 73%**	**78.02 / 67%**	**89.20 / 94%**	**82.18 / 75%**	**83.63 / 77%**	**72.54 / 64%**
	Perf. Matching↑	95.56% / 80.22%	104.80% / 89.33%	**96.03% / 94%**	105.41% / 92.59%	95.75% / 79.38%	**107.98% / 136.17%**
Coding	GPT4o	**90.31 / 98%**	75.18 / 70%	**94.57 / 100%**	86.32 / 91%	**90.78 / 100%**	58.43 / 25%
	Claude 3 Sonnet	88.76 / 92%	75.23 / 67%	93.82 / **100%**	**87.42 / 100%**	89.84 / **100%**	60.46 / 31%
	Gemini 1.5 Flash	88.51 / 94%	**75.62 / 73%**	93.59 / **100%**	82.92 / 84%	90.62 / 98%	**64.21 / 41%**
	Gemma 2B	62.31 / 44%	56.48 / 41%	80.92 / 84%	67.24 / 48%	72.98 / 66%	44.08 / 8%
	Gemma 7B	**80.56 / 80%**	**71.92 / 70%**	**90.47 / 98%**	**80.26 / 84%**	**84.66 / 88%**	**61.23 / 36%**
	Mistral 7B	68.32 / 56%	61.01 / 45%	81.25 / 81%	69.10 / 55%	72.39 / 69%	45.25 / 8%
	LLaMA3 8B	77.47 / 72%	69.46 / 61%	83.97 / 83%	73.51 / 67%	75.55 / 73%	51.10 / 17%
	Perf. Matching↑	89.20% / 81.63%	95.11% / 95.89%	**95.66% / 98%**	91.81% / 84%	93.26% / 88%	**95.36% / 97.80%**
Closed QA	GPT4o	**95.45 / 100%**	84.23 / 80%	97.21 / **100%**	92.56 / 97%	93.58 / **100%**	75.58 / 63%
	Claude 3 Sonnet	94.03 / 98%	85.28 / 82%	97.60 / **100%**	93.95 / **100%**	93.66 / **100%**	76.33 / 65%
	Gemini 1.5 Flash	94.63 / 97%	**87.43 / 87%**	**98.25 / 100%**	**97.41 / 100%**	**95.00 / 100%**	**85.66 / 83%**
	Gemma 2B	67.25 / 53%	67.41 / 57%	80.22 / 78%	70.20 / 72%	70.33 / 60%	59.68 / 50%
	Gemma 7B	81.85 / **83%**	79.02 / 78%	**88.83 / 93%**	83.95 / **83%**	**82.51 / 80%**	72.24 / 67%
	Mistral 7B	**83.63** / 82%	**81.36 / 83%**	88.25 / 85%	**84.77 / 83%**	82.04 / 78%	**73.95 / 70%**
	LLaMA3 8B	75.55 / 75%	72.62 / 72%	86.03 / 85%	77.64 / 80%	79.09 / 77%	68.78 / 65%
	Perf. Matching↑	87.62% / 83%	**93.06% / 95.40%**	**90.41% / 93%**	87.02% / 83%	86.85% / 80%	86.33% / 84.34%

Prompt

Read this article and give me a description of Eagleman.

Eagleman is thirty-nine now and an assistant professor of neuroscience at Baylor College of Medicine, in Houston. Physically, he seems no worse for the fall. He did a belly flop on the bricks, he says, and his nose took most of the impact. “He made a one-point landing,” as his father puts it. The cartilage was so badly smashed that an emergency-room surgeon had to remove it all, leaving Eagleman with a rubbery proboscis that he could bend in any direction. But it stiffened up eventually, and it’s hard to tell that it was ever injured. Eagleman has puckish, neatly carved features, with a lantern jaw and modish sideburns. In Baylor’s lab-coated corridors, he wears designer jeans and square-toed ankle boots, and walks with a bounce in his step that’s suspiciously close to a strut, like Pinocchio heading off to Pleasure Island.

If Eagleman’s body bears no marks of his childhood accident, his mind has been deeply imprinted by it. He is a man obsessed by time. As the head of a lab at Baylor, Eagleman has spent the past decade tracing the neural and psychological circuitry of the brain’s biological clocks. He has had the good fortune to arrive in his field at the same time as fMRI scanners, which allow neuroscientists to observe the brain at work, in the act of thinking. But his best results have often come through more inventive means: video games, optical illusions, physical challenges. Eagleman has a talent for testing the untestable, for taking seemingly sophomoric notions and using them to nail down the slippery stuff of consciousness. “There are an infinite number of boring things to do in science,” he told me. “But we live these short life spans. Why not do the thing that’s the coolest thing in the world to do?”

Target response

Eagleman is 39 with puckish features including a lantern jaw, styled sideburns, flashy jeans and ankle boots. Physically, he has completely recovered from his childhood fall that broke his nose. He is obsessed with time and is an assistant professor of neuroscience at Baylor College of Medicine.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Gemma 2B (aligned on GPT4o generated 128k dataset)			
Eagleman is a 39-year-old neuro-sci professor with a rubber nose, who used to be a professional scuba diver. He has excellent recall of marine life, which is useful in his research. His goal is to understand the biological clocks (the brain's internal clock) and how they were modified during evolution. He has also worked on physical challenges and video games to learn more about consciousness. His work aims to make the brain accessible to scientists who might not have the same training as neuro-sci professionals.			
Precision scores	35.5	42.5	53.0
Similarity scores	29.5	26.5	20.0
Gemma 7B (aligned on GPT4o generated 128k dataset)			
Dr. Eagleman is a thirty-nine-year-old professor of neuroscience known for his innovative research on the brain's biological clocks. Despite a severe nose injury as a child, Eagleman has made a remarkable recovery. His sharp features and interest in 'cool' experiments reflect his personality.			
Precision scores	82.5	90.5	74.0
Similarity scores	71.7	82.5	49.5

Figure 9: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for summarization task.

Prompt

Read this article and give me a description of Eagleman.

Eagleman is thirty-nine now and an assistant professor of neuroscience at Baylor College of Medicine, in Houston. Physically, he seems no worse for the fall. He did a belly flop on the bricks, he says, and his nose took most of the impact. "He made a one-point landing," as his father puts it. The cartilage was so badly smashed that an emergency-room surgeon had to remove it all, leaving Eagleman with a rubbery proboscis that he could bend in any direction. But it stiffened up eventually, and it's hard to tell that it was ever injured. Eagleman has puckish, neatly carved features, with a lantern jaw and modish sideburns. In Baylor's lab-coated corridors, he wears designer jeans and square-toed ankle boots, and walks with a bounce in his step that's suspiciously close to a strut, like Pinocchio heading off to Pleasure Island.

If Eagleman's body bears no marks of his childhood accident, his mind has been deeply imprinted by it. He is a man obsessed by time. As the head of a lab at Baylor, Eagleman has spent the past decade tracing the neural and psychological circuitry of the brain's biological clocks. He has had the good fortune to arrive in his field at the same time as fMRI scanners, which allow neuroscientists to observe the brain at work, in the act of thinking. But his best results have often come through more inventive means: video games, optical illusions, physical challenges. Eagleman has a talent for testing the untestable, for taking seemingly sophomoric notions and using them to nail down the slippery stuff of consciousness. "There are an infinite number of boring things to do in science," he told me. "But we live these short life spans. Why not do the thing that's the coolest thing in the world to do?"

Target response

Eagleman is 39 with puckish features including a lantern jaw, styled sideburns, flashy jeans and ankle boots. Physically, he has completely recovered from his childhood fall that broke his nose. He is obsessed with time and is an assistant professor of neuroscience at Baylor College of Medicine.

		GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Gemma 2B (aligned on GPT4o generated 128k dataset) Eagleman is a 39-year-old neuro-sci professor with a rubber nose, who used to be a professional scuba diver. He has excellent recall of marine life, which is useful in his research. His goal is to understand the biological clocks (the brain's internal clock) and how they were modified during evolution. He has also worked on physical challenges and video games to learn more about consciousness. His work aims to make the brain accessible to scientists who might not have the same training as neuro-sci professionals.	**Precision scores**	**35.5**	**42.5**	**53.0**
	Similarity scores	**29.5**	**26.5**	**20.0**
Gemma 7B (aligned on GPT4o generated 128k dataset) Dr. Eagleman is a thirty-nine-year-old professor of neuroscience known for his innovative research on the brain's biological clocks. Despite a severe nose injury as a child, Eagleman has made a remarkable recovery. His sharp features and interest in 'cool' experiments reflect his personality.	**Precision scores**	**82.5**	**90.5**	**74.0**
	Similarity scores	**71.7**	**82.5**	**49.5**

Figure 10: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for summarization task.

Prompt

Would you classify this email as an advertisement or an account update?

Only 4 hours left in 2020 to make your support known for public radio in the Ozarks! As always, your year-end contribution is tax-deductible.

Don't wait until the ball drops to make your donation...give now, and enjoy the countdown to midnight knowing you've done your part to help the programs you rely on thrive in the year to come.

Target response

This email is an advertisement.

		GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Gemma 2B (aligned on GPT4o generated 128k dataset) This email is an advertisement. It is asking supporters to make a donation to public radio before midnight. This is intended to persuade supporters to make a donation which is the purpose of an advertisement.	**Precision scores**	**95.5**	**94.5**	**93.0**
	Similarity scores	**87.5**	**91.0**	**81.0**
Gemma 7B (aligned on GPT4o generated 128k dataset) This email is an advertisement.	**Precision scores**	**99.5**	**100.0**	**100.0**
	Similarity scores	**100.0**	**100.0**	**100.0**

Figure 11: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for classification task.

Prompt

which of these songs are titled in english and which are titled in italian?

No.	Title	Length
1.	"Zitti e buoni"	3:14
2.	"Lividi sui gomiti"	2:45
3.	"I Wanna Be Your Slave"	2:53
4.	"In nome del padre"	3:39
5.	"For Your Love"	3:50
6.	"La paura del buio"	3:29
7.	"Vent'anni"	4:13

Target response

1. Italian
2. Italian
3. English
4. Italian
5. English
6. Italian
7. Italian

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash

Gemma 2B (aligned on GPT4o generated 128k dataset)

1.	"Zitti e buoni"	3	Italian
2.	"Lividi sui gomiti"	2	Italian
3.	"I Wanna Be Your Slave"	2	Italian
4.	"In nome del padre"	3	Italian
5.	"For Your Love"	3	Italian
6.	"La paura del buio"	3	Italian
7.	"Vent'anni" 4	Italian	

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	**18.5**	**82.0**	**63.6**
Similarity scores	**20.8**	**67.0**	**50.0**

Gemma 7B (aligned on GPT4o generated 128k dataset)

1. Zitti e buoni - Italian
2. Lividi sui gomiti - Italian
3. I Wanna Be Your Slave - English
4. In nome del padre - Italian
5. For Your Love - English
6. La paura del buio - Italian
7. Vent'anni - Italian

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	**98.9**	**100.0**	**99.5**
Similarity scores	**96.7**	**100.0**	**93.3**

Figure 12: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for classification task.

Prompt

How do I listen for an event in Discord.js?

Target response

You can listen for an event in Discord.js by using the Client#on method:

```
const { Events, Client } = require("discord.js");
const client = new Client();
client.on(Events.ClientReady, c => {
        console.log("Client is ready.");
});
```

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash

Gemma 2B (aligned on GPT4o generated 128k dataset)

To listen for an event in Discord.js, you need to use the 'listen' method. For example, to listen for a 'messageCreate' event, you can do the following:

```
```javascript
const Discord = require('discord.js');

const bot = new Discord.Client();

bot.on('messageCreate', (message) => {
    if (message.content === 'Hello') {
        message.reply('Hello there!');
    }
});

bot.login('yourTokenHere');
```

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	68.0	59.5	82.5
Similarity scores	63.5	38.0	56.0

Gemma 7B (aligned on GPT4o generated 128k dataset)

To listen for an event in Discord.js, you use the `client.on` method. Here's an example of how to listen for a message event:

```
```javascript
const Discord = require('discord.js');
const client = new Discord.Client();

client.on('message', (message) => {
    if (message.author.id === 'your_id_here') {
        console.log(`Message from ${message.author.tag}: ${message.content}`);
    }
});

client.login('your_token_here');
```

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	83.0	92.5	90.5
Similarity scores	66.5	84.0	67.0

Figure 13: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for coding task.

Prompt

This Java function below is not working as I expected, it's supposed to go through an array of numbers and if it finds a number in the array that is divisible by 2 with a remainder of 0, then it adds it to a new array and returns the new array at the end. Can you help debug the function and explain how why it's not working as expected?

```
public static List checkForEvens(int [] nums) {
    ArrayList anwer = new ArrayList<Integer>();
    for(int i = 0; i < nums.length; i++) {
    if(nums[i] / 2 == 0) {
    answer.add(nums[i]
     }
   }
}
```

Target response

Certainly! The function 'checkForEvens' iterates through an array and if a number in the array is divisible by 2, then it gets added to 'answer' which is later returned.

The reason the function may not be returning the expected result is because of the if statement, 'if (nums[i] /2 == 0)'. The statement is looking for numbers that get divided by 2 in the array and equal 0, which doesn't align with what you expect the function to do.

Instead, you want to use the modulo operator to check for the remainder of a number that is being divided by 2. If a number is divided by 2 and returns a remainder of 0, then we can assume that it's an even number. So, the small change of 'if (nums[i] % 2 == 0)' should solve your issue.

I've rewritten the function below with the correction.

```
public static List checkForEvens(int [] nums) {
    ArrayList anwer = new ArrayList<Integer>();
    for(int i = 0; i < nums.length; i++) {
    if(nums[i] % 2 == 0) {
    answer.add(nums[i]
     }
   }
}
```

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash

Gemma 2B (aligned on GPT4o generated 128k dataset)

Sure, I can help with that. The provided Java function is nearly correct but it does not handle cases where the number is negative. Here's a more robust version that includes handling edge cases:

```Java
public class CheckForEvens {
    public static List<Integer> checkForEvens(int[] nums) {
        List<Integer> answer = new ArrayList<>();
        for (int num : nums) {
            if ((num / 2 - 1) % 2 == 0) {
                answer.add(num);
            }
        }
        return answer;
    }
}
```

This function first creates an answer list, then iterates through each number in the array. It checks if the number divided by 2 (with an exclusive or of 1) is even. If it is, it adds the number to the answer list.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	**19.0**	**45.0**	**35.0**
Similarity scores	**27.0**	**25.0**	**19.0**

Gemma 7B (aligned on GPT4o generated 128k dataset)

The issue with your function is that the modulus operator (%) is missing in the condition to check if the number is even. You should use (nums[i] % 2 == 0) instead of (nums[i] / 2 == 0).

Here's the corrected function:

```
public static List checkForEvens(int[] nums) {
    ArrayList answer = new ArrayList<>();
    for (int i = 0; i < nums.length; i++) {
        if (nums[i] % 2 == 0) {
            answer.add(nums[i]);
        }
    }
    return answer;
}
```

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	**92.8**	**99.5**	**95.5**
Similarity scores	**85.0**	**95.0**	**89.0**

Figure 14: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for coding task.

Prompt

Is the following statement true or false? Keeping a plant healthy reduces the risk of pest infestations.

Use the following passage to answer the question.

How to Treat Yellowing Leaves on Snake Plant
Once you narrow down the reason your snake plant has yellow leaves, you'll want to address the issue quickly to keep your plant healthy. However, yellow leaves won't turn green again even if the issue is remedied, so it's best to cut them off with sharp, sanitized pruners or scissors at the base of the plant.

Check your plant's soil moisture. If the soil feels soggy and you've been watering frequently, overwatering could be to blame. Let the soil dry out completely before watering again. In extreme cases, you may need to repot the plant in fresh soil and inspect the roots for signs of rot.

If your snake plant's leaves look sunburned and the plant is getting direct sunlight, move it further away from the light source or use a sheer curtain to filter the light and make it less intense. Address inconsistent temperature by keeping the plant away from drafty areas and putting it in a space with a more consistent temperature.

Inspect your snake plant regularly for signs of pests. Remove insects with blasts of water or a cotton swab dipped in rubbing alcohol, then use neem oil or horticultural soap to prevent them from coming back. Keep your plant healthy going forward to lessen the likelihood that pests will attack.

If you haven't fertilized or refreshed your snake plant's soil in a few years and other causes don't apply, repot it with fresh soil. If the plant seems potbound, use a pot one size larger or divide the plant and repot it in its original container. If you suspect overfertilizing is the issue, flush your plant's soil several times with water to remove excess nutrients. Going forward, only fertilize your snake plant twice per year in spring and summer.

Target response

The statement is true.

		GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Gemma 2B (aligned on GPT4o generated 128k dataset) The answer is false. Keeping a plant healthy does not reduce the risk of pest infestations. Vainqueur,	**Precision scores**	**11.5**	**68.0**	**10.0**
	Similarity scores	**3.6**	**0.1**	**1.0**
Gemma 7B (aligned on GPT4o generated 128k dataset) The statement is true. Keeping a plant healthy reduces the risk of pest infestations.	**Precision scores**	**98.8**	**99.5**	**95.9**
	Similarity scores	**95.0**	**99.5**	**90.0**

Figure 15: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for closed QA task.

Prompt

Who did Prince William marry?

The queen's popularity, and that of the entire royal family, rebounded during the first decade of the 21st century. Though 2002 marked Queen Elizabeth's Golden Jubilee—50 years on the throne—the death of her mother (the beloved Queen Mum) and sister early that year cast a pall on the celebrations.

In 2005, the queen enjoyed public support when she gave her assent to Prince Charles' once-unthinkable marriage to his longtime love Camilla Parker Bowles.

In her seventh decade on the throne, Queen Elizabeth presided over the pomp and circumstance of another royal wedding at Westminster Abbey, that of Prince William to Catherine Middleton in April 2011. The Duke and Duchess of Cambridge, who are in line to become Britain's next king and queen, continued the line of succession with their children, Prince George (born 2013), Princess Charlotte (born 2015) and Prince Louis (born 2018).

Target response

Prince William married Catherine Middleton.

		GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Gemma 2B (aligned on GPT4o generated 128k dataset) Prince William married Catherine Middleton.�	**Precision scores**	95.4	100.0	100.0
	Similarity scores	96.1	100.0	100.0
Gemma 7B (aligned on GPT4o generated 128k dataset) Prince William married Catherine Middleton.	**Precision scores**	98.8	99.5	95.9
	Similarity scores	95.0	99.5	90.0

Figure 16: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o generated 128K synthetic dataset for closed QA task.

Prompt

Please summarize this post. Make it easier to read and much shorter. It should only be about a paragraph or two. Use the text following this request. After World War II, designating the start and end dates for daylight saving time fell to state governments. Because this created many railroad scheduling and safety problems, however, Congress passed the Uniform Time Act in 1966. This law set the nationwide dates of daylight saving time from the last Sunday in April until the last Sunday in October. In 2007, Congress amended the act to expand the period in which daylight saving time is in effect from the second Sunday in March to the first Sunday in November – dates that remain in effect today. The Uniform Time Act allows states and territories to opt out of daylight saving time, however. Arizona and Hawaii are on permanent standard time, along with Puerto Rico, the U.S. Virgin Islands, Northern Mariana Islands, Guam and American Samoa. Now, many other states are considering whether to stop falling back and springing ahead. Several U.S. states have legislation and resolutions under consideration to support permanent standard time, while many others have been or are considering permanent daylight saving time. Legislation and resolutions for permanent standard time have increased from 15% in 2021 to 31% in 2023. In March 2022, the U.S. Senate passed the Sunshine Protection Act in a bid to make daylight saving time permanent. But the House did not move forward with this legislation. Florida Sen. Marco Rubio reintroduced the bill on March 1, 2023. The spike in activity among states seeking to break from these twice-yearly changes reflects how more people are recognizing the downsides of this practice. Now, it's up to legislators to decide whether we end the time shift altogether, and to choose permanent standard or daylight saving time.

Target response

After World War II, Congress made daylight saving time a national holiday from the last Sunday in April to the last Sunday in October. In 2007, the Uniform Time Act was amended to extend the daylight saving time period from the second Sunday in March to the first Sunday in November. Congress can also allow state governments to opt out. Several U.S. states are considering or have considered legislation for permanent standard time, driven by opposition to daylight saving time. A similar bill was passed by the Senate in March 2022 but did not advance in the House, prompting renewed interest in the issue.

Gemma 2B (aligned on GPT4o generated 256k dataset)

After WWII, states were given the right to choose the start and end dates for daylight saving time, but this caused serious problems for rail schedules. To address this, Congress passed the Uniform Time Act in 1966 which standardized the changes, but still allowed states and territories to opt out entirely. Arizona, Hawaii, Puerto Rico, and some U.S. territories decided to stay on permanent time. Many other states and legislators are considering abolishing daylight saving time, but no bill has passed. People are starting to recognize the problems caused by changing the time twice a year and want to move to a permanent standard time.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	59.5	84.0	72.0
Similarity scores	55.0	73.8	49.0

Gemma 2B (aligned on Claude 3 Sonnet generated 256k dataset)

The key points regarding the Uniform Time Act and Daylight Saving Time are:

1) In 1966, the Uniform Time Act established the dates of daylight saving time nationwide.

2) States can opt out of daylight saving time under this Act, and there are currently 31 states on permanent standard time or permanent daylight saving time.

3) Multiple states are considering legislation to remove the time shift from our annual cycle of timekeeping to permanent standard time or permanent daylight saving time, reflecting increasing public opposition to the current system.

4) The Sunshine Protection Act passed by the U.S. Senate in 2022 proposed continuing daylight saving time but sought to make it permanent.

5) While there has been some support from a few states, the issue remains unresolved as the current legislation did not become law.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	71.8	90.5	72.5
Similarity scores	63.5	83.0	41.5

Gemma 2B (aligned on Gemini 1.5 Flash generated 256k dataset)

Congress passed the Uniform Time Act in 1966 to standardize daylight saving time across the country. However, the law's focus on railroad safety led to practical issues. The act was amended in 2007 to extend the dates of daylight saving time. While states can opt out of the rules, many are moving toward permanent standard time. The Sunshine Protection Act, which would make daylight saving time permanent, has been reintroduced in recent years. This shift toward a more standardized time system reflects growing awareness of its downsides.7

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	84.5	86.5	76.0
Similarity scores	76.8	76.0	57.0

Gemma 7B (aligned on GPT4o generated 256k dataset)

World War II led to inconsistent dates for daylight saving time until Congress passed the Uniform Time Act in 1966, setting the dates from the last Sunday in April to the last Sunday in October. In 2007, the dates were expanded to the second Sunday in March and first Sunday in November. The Uniform Time Act also allows states to opt out of daylight saving time, with Arizona, Hawaii, Puerto Rico, U.S. Virgin Islands, Northern Mariana Islands, Guam, and American Samoa choosing permanent standard time. While many states are considering making daylight saving time permanent, the U.S. Senate passed the Sunshine Protection Act in 2022 to make daylight saving time permanent. Sen. Marco Rubio reintroduced the bill in 2023. The upswing in states seeking permanent standard or daylight saving time highlights the growing awareness of daylight saving time's drawbacks. Ultimately, it's up to legislators to decide whether we should end the practice of changing time or make daylight saving time permanent.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	84.5	93.5	85.5
Similarity scores	74.2	86.7	67.5

Gemma 7B (aligned on Claude 3 Sonnet generated 256k dataset)

The Uniform Time Act of 1966 established the dates for daylight saving time nationwide. It currently runs from March to November. However, states and territories can opt out and permanently remain on standard time. In recent years, many states have considered making the switch to permanent standard or daylight saving time to end the twice-yearly shifting. States like Connecticut and New Hampshire have passed resolutions supporting permanent standard time. Meanwhile, legislation has been introduced in Florida and other states to make daylight saving time permanent. The U.S. Senate passed the Sunshine Protection Act in 2022 to make daylight saving time permanent, but the House did not take action. This increasing momentum reflects growing opposition to permanent daylight saving time among states, businesses, and the public. The issue has become a matter of whether Congress should choose permanent standard or daylight saving time, or if states will make the decision themselves.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	80.5	88.5	73.5
Similarity scores	71.3	78.5	43.5

Gemma 7B (aligned on Gemini 1.5 Flash generated 256k dataset)

The United States has a long and complicated history with Daylight Saving Time. Its origins trace back to World War I, when Germany implemented it as a way to conserve energy. The United States followed suit in 1918, adopting the practice for the first time. After World War II, states began to implement their own time changes, leading to inconsistencies. In 1966, Congress passed the Uniform Time Act, standardizing the dates for Daylight Saving Time. This act remains in effect today, although the dates have been expanded since its passage. Many states have considered becoming permanent standard time or daylight saving time states, with some currently considering bills to make the change official. Despite the increasing interest in permanent time changes, the Sunshine Protection Act – which would make Daylight Saving Time permanent – has stalled in Congress. Despite this, the movement to permanently shift to either Standard Time or Daylight Saving Time continues.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	61.0	88.5	68.0
Similarity scores	48.0	77.0	36.5

Figure 17: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o, Claude 3 Sonnet, Gemini 1.5 Flash generated 256K of synthetic datasets for summarization task.

Prompt

Summarize the following story in a few sentences.

The Fox one day thought of a plan to amuse himself at the expense of the Stork, at whose odd appearance he was always laughing.
"You must come and dine with me today," he said to the Stork, smiling to himself at the trick he was going to play. The Stork gladly accepted the invitation and arrived in good time and with a very good appetite.
For dinner the Fox served soup. But it was set out in a very shallow dish, and all the Stork could do was to wet the very tip of his bill. Not a drop of soup could he get. But the Fox lapped it up easily, and, to increase the disappointment of the Stork, made a great show of enjoyment.
The hungry Stork was much displeased at the trick, but he was a calm, even-tempered fellow and saw no good in flying into a rage. Instead, not long afterward, he invited the Fox to dine with him in turn. The Fox arrived promptly at the time that had been set, and the Stork served a fish dinner that had a very appetizing smell. But it was served in a tall jar with a very narrow neck. The Stork could easily get at the food with his long bill, but all the Fox could do was to lick the outside of the jar, and sniff at the delicious odor.
And when the Fox lost his temper, the Stork said calmly:
Do not play tricks on your neighbors unless you can stand the same treatment yourself.

Target response

The Fox always judged the Stork's appearance and one day decided to pull a prank on him. The Fox invited the Stork over for dinner and served him soup in a very shallow dish. The Stork was not able to get a single drop of the soup and, even though he was hungry, he stayed calm. Later, the Stork invited the Fox over for dinner and served him a delicious-smelling fish dinner, but it was in a tall jar with a very narrow neck. The Fox was only able to lick the outside of the jar, and he lost his temper.

Gemma 2B (aligned on GPT4o generated 256k dataset)

The Fox tricked the Stork into eating soup in a shallow dish, but used his own cleverness to enjoy dinner served in a tall but narrow jar.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	56.5	58.0	65.5
Similarity scores	42.5	33.5	32.0

Gemma 2B (aligned on Claude 3 Sonnet generated 256k dataset)

A Fox decided to mock the Stork, and offered him a meal at his house. The Stork arrived with a good appetite, but could not eat the soup set out in a shallow dish - he could only wet the tip of his bill. The Fox enjoyed the meal easily and even made a loud noise, but the hungry Stork lost his temper and invited the Fox to dine with him next.Життепис: The Fox arrived promptly at the time set and was served a fish dish, but could not eat it - he could only lick the outside of the jar and smell the delicious scent inside. In the end, the Stork warned the Fox not to play tricks if he could not stand them himself.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	88.0	89.0	81.0
Similarity scores	83.5	80.0	64.5

Gemma 2B (aligned on Gemini 1.5 Flash generated 256k dataset)

A fox tricks the Stork by serving him soup that's too shallow for him to reach. The Stork gets upset but he's a calm man and tries something else. This time the fox is even more disappointed, but the Stork keeps his cool and serves him a smelly fish. The fox loses his temper, but the Stork is calm and serves him a jar of a food that smells delicious, but the fox can't reach it. This leads the fox to lose his temper as well.5

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	66.5	78.5	74.5
Similarity scores	67.0	69.0	57.0

Gemma 7B (aligned on GPT4o generated 256k dataset)

A fox invited a stork to dinner, serving soup in a shallow dish the stork could not reach. The stork later invited the fox to dinner, serving fish in a narrow jar the fox could not reach. The stork's calm attitude reminded the fox to be careful with tricks.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	84.5	93.5	85.5
Similarity scores	74.2	86.7	67.5

Gemma 7B (aligned on Claude 3 Sonnet generated 256k dataset)

The Fox tricked the Stork by serving soup in a shallow dish, which the Stork could not reach the food. In retaliation, the Stork served fish in a tall jar with a narrow neck, trapping the Fox from eating the food. The Stork calmly accepted the situation, while the Fox angrily lost his appetite.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	80.5	88.5	73.5
Similarity scores	71.3	78.5	43.5

Gemma 7B (aligned on Gemini 1.5 Flash generated 256k dataset)

The Fox tricked the Stork by serving him soup in a shallow dish he couldn't reach, but the Stork got revenge by serving the Fox a fish dinner in a jar the Fox couldn't access. The lesson learned by the Fox is to not play tricks on others unless you are prepared to face the same treatment.

	GPT4o	Claude 3 Sonnet	Gemini 1.5 Flash
Precision scores	61.0	88.5	68.0
Similarity scores	48.0	77.0	36.5

Figure 18: Responses by Gemma 2B and Gemma 7B models fine-tuned on GPT4o, Claude 3 Sonnet, Gemini 1.5 Flash generated 256K of synthetic datasets for summarization task.