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

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

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 GPT4o, 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.

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∼16K	LoRA Rank	8
	LoRA Alpha	16
32K	LoRA Rank	16
	LoRA Alpha	32
64K∼256K	LoRA Rank	32
	LoRA Alpha	64

B.2 Training Configurations

We utilize Hugging Face's "Alignment Handbook" (Tunstall et al., 2023) and the alignment recipes tailored for the Gemma models to streamline the fine-tuning process.

As outlined in Table 5, we employ QLoRA (Dettmers et al., 2024) 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.

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 allo-