```
"inputs": {
  "instruction": "Find an example of the given kind of data",
  "input": "Qualitative data",
  "response1": "An example of qualitative data is customer feedback.",
  "response2": "An example of qualitative data is a customer review."
}
"outputs": {
  "evaluation_result": "Tie",
  "evaluation_reason": "Both responses are correct and provide similar examples of qualitative data.",
  "reference_response": "An example of qualitative data is an interview transcript."
}
```


Figure 5: A training data example for PandaLM.

```
Below are two responses for a given task. The task is defined by the Instruction with an Input
that provides further context. Evaluate the responses and generate a reference answer for the task.

### Instruction:
{instruction}

### Input:
{input}

### Response 1:
{response 1, generated by a candidate model}

### Response 2:
{response 2, generated by another candidate model}

### Evaluation:
{evaluation result}
{evaluation reason}

### Reference:
{a reference response for the instruction}
```

Figure 6: The prompt for training PandaLM.

## A Training Prompt Details

We introduce the detailed prompt of training PandaLM in Figure 6.

## B Directed Acyclic Graph depicting the mixture ranking of models trained using both Alpaca's and PandaLM's hyperparameters.

A directed acyclic graph (DAG) is presented in Figure 7, illustrating the relative rankings of various models fine-tuned with different sets of hyperparameters. Notably, this ranking differs from those in Figure4, due to the variance in the test data: the test data for 7 is a sampled subset from that used in Figure4 which is deliberately chosen to ensure a high Inter-Annotator Agreement (IAA). A discernible pattern emerges from the rankings: models fine-tuned using PandaLM's hyperparameters consistently outshine their counterparts fine-tuned with Alpaca's. The top-rated model is PandaLM-LLaMA, followed by Alpaca-LLaMA, PandaLM-Bloom, PandaLM-Pythia, PandaLM-OPT, PandaLM-Cerebras-GPT, Alpaca-OPT, Alpaca-Bloom, Alpaca-Pythia, and Alpaca-Cerebras-GPT, in descending order of performance. This juxtaposition accentuates the effectiveness of PandaLM's hyperparameter selection in improving model performance, as models optimized with PandaLM consistently rank higher than those using Alpaca's hyperparameters in the hybrid ranking. These findings underscore the potential of PandaLM as a powerful tool in enhancing the performance of large language models, further supporting the assertion of its efficacy.

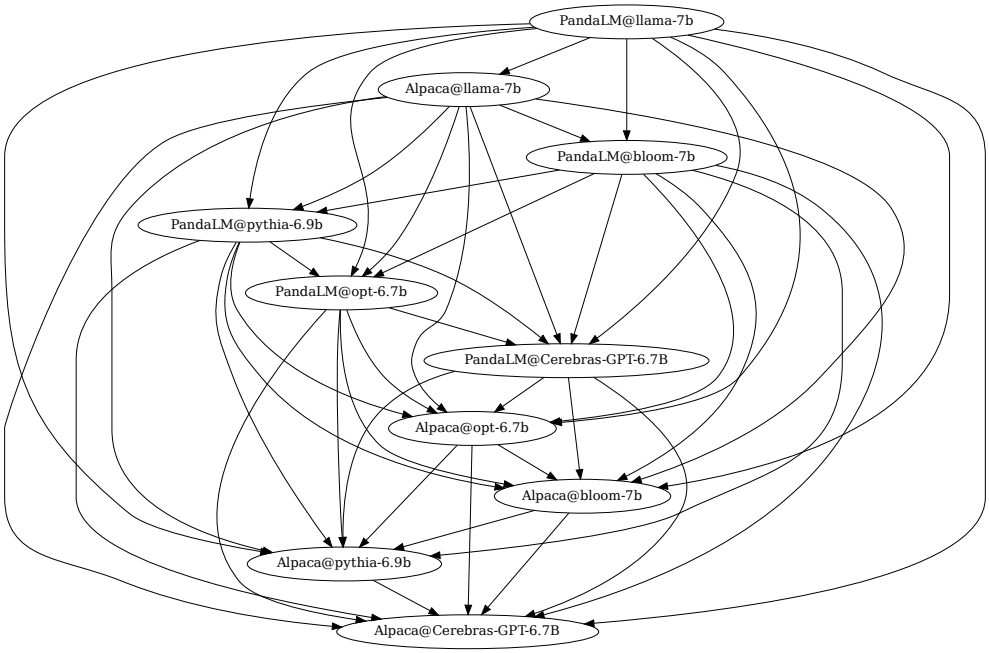



Figure 7: Directed Acyclic Graph depicting the mixture ranking of models trained using both Alpaca's and PandaLM's hyperparameters. The models are ranked from strongest to weakest in the following order: PandaLM-LLaMA, Alpaca-LLaMA, PandaLM-Bloom, PandaLM-Pythia, PandaLM-OPT, PandaLM-Cerebras-GPT, Alpaca-OPT, Alpaca-Bloom, Alpaca-Pythia, Alpaca-Cerebras-GPT.

Table 6: Comparison on several downstream tasks using lm-eval(Gao et al., 2021) between foundation models fine-tuned on Alpaca's hyperparameters, and foundation models fine-tuned with PandaLM. Note that the MMLU task consists of 57 subtasks, which means providing a comprehensive standard deviation here is not feasible.

| | **ARC-Challenge-acc_norm(25-shot)** | **Hellaswag-acc_norm(10-shot)** | **MMLU-average-acc(5-shot)** | **TruthfulQA-mc2(0-shot)** | **Average** |
|---|---|---|---|---|---|
| llama-7b original | 0.4923±0.0146 | 0.7583±0.0043 | 0.3306 | 0.3703±0.0141 | 0.4879 |
| llama-7b w/ PandaLM | **0.5162±0.0146** | **0.7764±0.0042** | **0.3396** | **0.3801±0.0145** | **0.5031** |
| opt-6.7b original | **0.3805±0.0142** | 0.6535±0.0047 | 0.2476 | 0.3587±0.0139 | 0.4101 |
| opt-6.7b w/ PandaLM | 0.3771±0.0142 | **0.6540±0.0047** | **0.2502** | **0.3609±0.0142** | **0.4106** |
| pythia-6.9b original | 0.3848±0.0142 | 0.6093±0.0049 | 0.2490 | **0.4187±0.0148** | 0.4155 |
| pythia-6.9b w/ PandaLM | **0.4130±0.0144** | **0.6337±0.0048** | **0.2581** | 0.3972±0.0144 | **0.4255** |
| bloom-7b original | 0.3985±0.0143 | **0.6086±0.0049** | **0.2635** | 0.3975±0.0148 | **0.4170** |
| bloom-7b w/ PandaLM | **0.3951±0.0143** | 0.6084±0.0049 | 0.2520 | **0.3997±0.0149** | 0.4138 |
| Cerebras-GPT-6.7B original | 0.3524±0.0140 | **0.5613±0.0050** | **0.2584** | **0.3624±0.0140** | **0.3836** |
| Cerebras-GPT-6.7B w/ PandaLM | **0.3558±0.0140** | 0.5550±0.0050 | 0.2452 | 0.3448±0.0141 | 0.3752 |

## C Comparisons between original models and models tuned using PandaLM on Traditional Tasks

We compare fine-tuned LLMs on various traditional tasks with lm-eval (Gao et al., 2021). Although the majority of language models exhibit improved performance after finetuning with PandaLM, Cerebras exhibits a decline. This highlights the importance of nuanced, subjective evaluations (win/tie/lose of responses). Human evaluations, as well as assessments from GPT-4 and GPT-3.5, all concur in indicating a better performance from Cerebras when paired with PandaLM. This is also confirmed in (Yu et al., 2024).

As shown in Table 7, the evaluation results of language models show that lower perplexity, indicating better predictive ability in pretraining or other tasks, does not always mean better overall performance of instruction-tuned models. For example, LLaMA-PandaLM has a higher perplexity than LLaMA-Alpaca but outperforms it in both pairwise comparisons (PandaLM, GPT, Human) and traditional

Table 7: Analysis on perplexity and other evaluation metrics. Note that we report the win rate over 170 samples of PandaLM, GPT, and Human.

| Model | Perplexity (↓) | PandaLM-7B (↑) | PandaLM-70B (↑) | GPT-3.5 (↑) | GPT-4 (↑) | Human (↑) | lm-eval avg. score (↑) |
|---|---|---|---|---|---|---|---|
| LLaMA-Alpaca | **2.75** | 15.88% | 22.94% | 15.29% | 10.00% | 12.35% | 0.4879 |
| LLaMA-PandaLM | 2.81 | **19.41%** | **35.88%** | **26.47%** | **23.53%** | **48.24%** | **0.5031** |

tasks (lm-eval). This suggests that while perplexity is not feasible for instruction-tuned models where lower perplexity might mean overfitting and less generalizability.

## D LAW / BIOMEDICAL DATASETS INTRODUCTION

Specifically, we assess PandaLM's proficiency using the LSAT (Law School Admission Test) dataset, which serves as an entrance exam question set for American law schools. This dataset incorporates 1,009 questions, further divided into three subsets: AR, LR, and RC. In the realm of biomedicine, we use the PubMedQA dataset—a vast repository for biomedical retrieval QA data, boasting 1k expert annotations, 61.2k unlabeled entries, and a massive 211.3k human-generated QA instances. For our evaluation, we rely on the labeled section (PubMedQA-l) that contains 1k instances. Each instance encompasses a question, context, and label. Additionally, we tap into the BioASQ dataset, specifically leveraging the task b dataset from its 11th challenge. This dataset is renowned for its biomedical semantic indexing and question-answering (QA) capabilities. From it, we use 1k samples for our assessment. We will test code/math dataset Cobbe et al. (2021); Zeng et al. (2022b) in future work.

## E DATA SIZE AND QUALITY ANALYSIS IN INSTRUCTION TUNING

We conduct an ablation study to investigate the impact of training data size (up to 1,344,000) on the performance of the model, given optimal hyperparameters. Importantly, a relationship exists between the size and quality of training data. Thus, we focus on an ablation study of data size here, but conducting a similar experiment on data quality is feasible. We derive the results from PandaLM-7B. The objective is to discern how much training data is required to reach each model's peak performance. Table 8 reveals the optimal quantity of training data varies among models. More training data typically enhances model performance. However, an optimal point exists for each model, beyond which further data doesn't improve performance. For example, the OPT model peaks at 992,000 data points, indicating additional data does not enhance the model's performance.

Table 8: Optimal training data size for each model.

| Model | Bloom | Cerebras-GPT | LLaMA | OPT | Pythia |
|---|---|---|---|---|---|
| Optimal Training Data Size | 1,216,000 | 1,344,000 | 11,520,000 | 992,000 | 1,344,000 |

## F LORA ANALYSIS IN INSTRUCTION TUNING

We further aim to evaluate the efficacy of Low-Rank Adaptation (LoRA) (Hu et al.) compared to full fine-tuning across various models, utilizing optimal hyperparameters. The results are also obtained from PandaLM-7B. Our analysis seeks to provide a comparative understanding of these tuning methodologies. As shown in Table 9, the results for the Bloom model reveal a distinct advantage for full fine-tuning, which triumphs over LoRA in 66 instances as opposed to LoRA's 35. Notably, they tie in 69 instances. In the case of the Cerebras model, full fine-tuning again proves superior, leading in 59 cases compared to LoRA's 40, despite drawing even 71 times. The trend of full fine-tuning superiority is consistent in the LLaMA model. Out of 170 instances, full fine-tuning results in better performance in 48 instances, whereas LoRA emerges victorious in only 28 instances. The majority of the results are tied, amounting to 94 instances. In the OPT model, full fine-tuning once more showcases its advantage with 64 instances of superior performance compared to LoRA's 33, while recording a tie in 73 instances. Lastly, for the Pythia model, full fine-tuning leads the race with 71 instances of better performance against LoRA's 21, and a tie occurring in 78 instances. These results

underscore that full fine-tuning generally yields more favorable results compared to the use of LoRA, though the outcomes can vary depending on the model. Despite the considerable number of ties, full fine-tuning holds the upper hand in most models, thereby highlighting its effectiveness. This suggests that while LoRA may provide comparable results in some instances, a strategy of full fine-tuning often proves to be the more beneficial approach in enhancing model performance.

Table 9: Comparison of LoRA and Full Fine-tuning.

| Model | LoRA Wins | Full Fine-tuning Wins | Ties |
|---|---|---|---|
| Bloom | 35 | 66 | 69 |
| Cerebras-GPT | 40 | 59 | 71 |
| LLaMA | 28 | 48 | 94 |
| OPT | 33 | 64 | 73 |
| Pythia | 21 | 71 | 78 |

## G Leveraging Pre-trained models and other instruction tuned models for evaluation

Employing LLMs for response evaluation without additional training is a natural direction for the task. However, implementing evaluation criteria through zero-shot or few-shot methods is challenging for LLMs due to the necessity for extended context lengths.

We have undertaken experiments using zero-shot and few-shot (in-context learning Dong et al. (2022); Yang et al. (2023)) evaluations with LLaMA. Our observations indicate that an un-tuned LLaMA struggles with adhering to user-specified format requirements. Consequently, our experiments focused on computing and comparing the log-likelihood of generating continuations (e.g., determining whether "Response 1 is better," "Response 2 is better," or if both responses are similar in quality) from the same context. We regard the choice with the highest log-likelihood as the prediction result. We also alternated response order in our experiments to reduce position bias. Furthermore, we undertook experiments with Vicuna, a finetuned version of LLaMA. The experiments demonstrated that the evaluation capabilities of instruction-tuned models possess significant potential for enhancement.

The results in Table 10 highlight the importance of tailored tuning for evaluation, a precisely-tuned smaller model outperforms a larger one in zero and few-shot scenarios.

## H Enhancing PandaLM with Refined Supervision.

In our supervision goal, we incorporate not only the comparative result of responses but also a succinct explanation and a reference response. This methodology augments PandaLM's comprehension of the evaluation criteria.

Table 10: Ablation study of directly using pre-trained models and instruction tuned models for evaluation.

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 score** |
|---|---|---|---|---|
| LLaMA-7B 0-shot (log-likelihood) | 12.11 | 70.23 | 34.52 | 8.77 |
| LLaMA-30B 0-shot (log-likelihood) | 31.43 | 56.48 | 43.12 | 32.83 |
| LLaMA-7B 5-shot (log-likelihood) | 24.82 | 46.99 | 39.79 | 25.43 |
| LLaMA-30B 5-shot (log-likelihood) | 42.24 | 61.99 | 51.76 | 42.93 |
| Vicuna-7B (log-likelihood) | 15.92 | 57.53 | 34.90 | 14.90 |
| Vicuna-13B (log-likelihood) | 35.24 | 57.45 | 43.65 | 36.29 |
| PandaLM-7B | **59.26** | 57.28 | 59.23 | 54.56 |
| PandaLM-7B (log-likelihood) | **59.26** | **59.70** | **63.07** | **55.78** |

Table 11: Ablation study of supervision goal.

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| PandaLM-7B (with only eval label) | 0.4725 | 0.4505 | 0.4725 | 0.3152 |
| PandaLM-7B | 0.5926 | 0.5728 | 0.5923 | 0.5456 |

To empirically gauge the significance of this explanation, an experiment was executed. Here, the explanation and reference were omitted during training, and only the categorical outcomes (0/1/2 or Tie/Win/Lose) were retained in the dataset for training a fresh iteration of PandaLM. The results, as depicted in Table 11, demonstrate that in the absence of the explanation, PandaLM encounters difficulties in precisely determining the preferable response.

## I HUMAN EVALUATION DATASHEET

We employ human annotators from a crowdsourcing company and pay them fairly. In particular, we pay our annotators 50 dollars per hour, which is above the average local income level. We have filled out the Google Sheet provided in (Shimorina & Belz, 2022).

## J HYPERPARAMETER OPTIMIZATION ANALYSIS

In our hyperparameter searching process, we explored a range of learning rates, epochs, optimizers, and schedulers. The learning rates tested varied from 2e-6 to 2e-4, with model checkpoints saved at the end of each epoch. Performance was rigorously assessed through pairwise comparisons between checkpoints, counting the win rounds for each model, as detailed in Figure 8.

Our analysis, as depicted in Figure 8a, suggests a tendency towards a learning rate of 2e-5, although this preference was not uniformly clear across all models. Figure 8b demonstrates the variability in the optimal number of epochs, with a trend showing that peak performance often occurs around the fourth or fifth epoch. This evidence points to the complex interplay of hyperparameters with model performance, which is further influenced by data distribution, optimizer, and scheduler choices.

The findings from our hyperparameter optimization process highlight that there is no universally optimal setting for different models and training setups. While a pattern emerged suggesting that a learning rate around 2e-5 and an epoch count near 4 might be beneficial in some cases, these results are not conclusive. This reinforces the need for specific hyperparameter searches for different models, as demonstrated in our visualizations. A tailored approach to hyperparameter optimization is essential, as it allows for a more nuanced understanding of model performance across various scenarios.

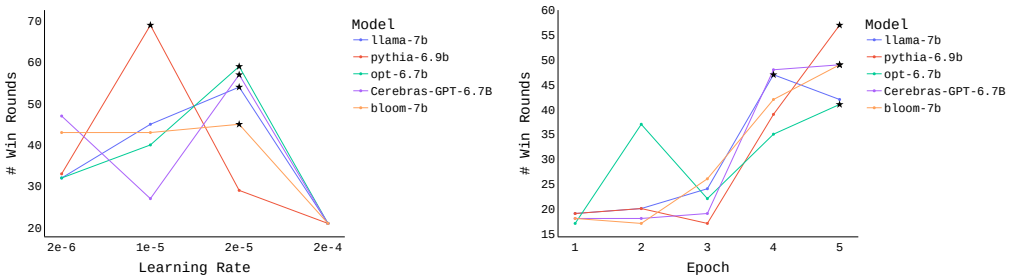
Besides, we implemented an early stopping strategy using Pandalm. We focus specifically on LLaMA. Our experiments showed that in some cases, a model's performance at epoch 3 was inferior to that at epoch 2. However, subsequent epochs demonstrated performance improvements. This indicates that early stopping may not always be suitable for large model fine-tuning, as it could prematurely halt training before reaching optimal performance.

Table 12: Analysis of PandaLM's Evaluation Capability on Unseen Models.

| Model Comparison | PandaLM | Human | Metrics (P, R, F1) |
|---|---|---|---|
| llama1-7b vs llama2-7b | (23,61,16) | (23,70,7) | (0.7061, 0.7100, 0.6932) |
| llama1-13b vs llama2-13b | (18,73,9) | (20,68,12) | (0.7032, 0.6800, 0.6899) |
| llama1-65b vs llama2-70b | (20,66,14) | (34,56,10) | (0.7269, 0.6600, 0.6808) |

## K MODEL SHIFT ANALYSIS

In Table 12, we provide a detailed comparison of PandaLM's performance against human benchmarks and in the context of different versions of instruction-tuned LLaMA models. Note that llama1-13b, llama1-65b and llama2 are indicative of model shift. The results demonstrate that PandaLM aligns




(a) Comparison Results of GPT-4.

(b) Comparison Results of Human.

Figure 8: Hyperparameter Optimization Analysis in PandaLM. The figure illustrates the performance across different learning rates and variability in model performance across epochs.

closely with humans, consistently showing a preference for the LLama-2 model. This alignment is in line with expectations, as LLama-2 benefits from more pre-training data. Such findings highlight the significance of extensive pre-training in developing language models that are more skilled at understanding and correctly responding to various instructions.