ATLANTIS: Weak-to-Strong Learning via Importance Sampling

Yi Liu[†] **Guoyin Wang** **Shicheng Li**[†] **Feifan Song**[†] **Xu Sun**[†]
[†]State Key Laboratory of Multimedia Information Processing,
School of Computer Science, Peking University
{imliuyi,lisc99,xusun}@pku.edu.cn

Abstract

Supervised fine-tuning (SFT) enables large language models to align with training data for better performance in many aspects. Nevertheless, the gap between the distribution of current datasets from human annotations or model generations and the real-world data distribution heavily limits the capacities and potentials of models. As a result, we propose a new SFT technique, ATLANTIS, to bridge the gap. We adopt importance sampling to estimate the optimal data distribution in the real world from existing training datasets because the former is hard to sample from. Furthermore, we introduce an extra small model and reference model to estimate the sampling ratio through the probability gap between them. We evaluate our method with benchmarks in knowledge & understanding and preference aspects. The experiment results prove that ATLANTIS can bring consistent and significant improvements to models' performance. What's more, our method can be flexibly transferred among models with different structures. Our analyses demonstrate that our method is well-compatible with other SFT techniques to further enhance models' capacities and has great potential to be combined with existing training frameworks.

1 Introduction

With the proliferation of strong large language models (LLM), supervised fine-tuning (SFT) becomes a more and more important technique to allow base models to follow human instructions (Wei et al., 2021; Ouyang et al., 2022; Chung et al., 2024), align with data in specific domains (Yang et al., 2023; Tu et al., 2024), or alleviate bias existing in themselves (Guo et al., 2022; Zhou et al., 2023a). As a result, LLMs can serve as strong assistants for us to solve problems in different domains (OpenAI, 2022; Achiam et al., 2023).

The performance of the finetuned model heavily relies on the quality of the training dataset (Zhou et al., 2024). In order to train a human-like language model with strong capacities, we hope the training data can contain all the knowledge in the world and completely cover the chatting patterns of all human beings. We call this ideal dataset the optimal dataset and its corresponding distribution is the optimal distribution p^*. In theory, p^* can fully reflect the distribution of any natural language in the real world and fit the patterns of talking or writing for everyone, which is impossible for the current training technique because we can never collect all potential training data around the whole world. Alternatively, we choose to construct large-scale SFT datasets with high quality to train strong models. The datasets collected from human annotations provide approaches to aligning models with actual human behaviors (Mishra et al., 2021; Conover et al., 2023). Considering the high cost of human annotations, many corpora consisting of real conversations between users and LLMs spring up (Teknium, 2023; Taori et al., 2023). However, neither kind of training dataset can fit the optimal distribution p^* perfectly, since the training samples are not collected from the real world and are too limited in size to cover all possible cases in life. In other words, the target of fitting the distribution of the training datasets deviates from the optimal p^* at the very beginning.

As a result, the gaps between training dataset distribution p_d and p^* will heavily limit the capacities and potentials of LLMs, especially with the rapid increase in model scales and capacities. Continuous increase of training data size is a possible solution but is too costly. Many approaches to selecting training data can alleviate this problem to some extent (Li et al., 2023a, 2024a). Though improving models' capacities through discarding training data with low quality, these methods fail to pay enough attention to the distribution gap and hardly make any efforts to bridge it.

Inspired by weak-to-strong generalization

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1042–1052

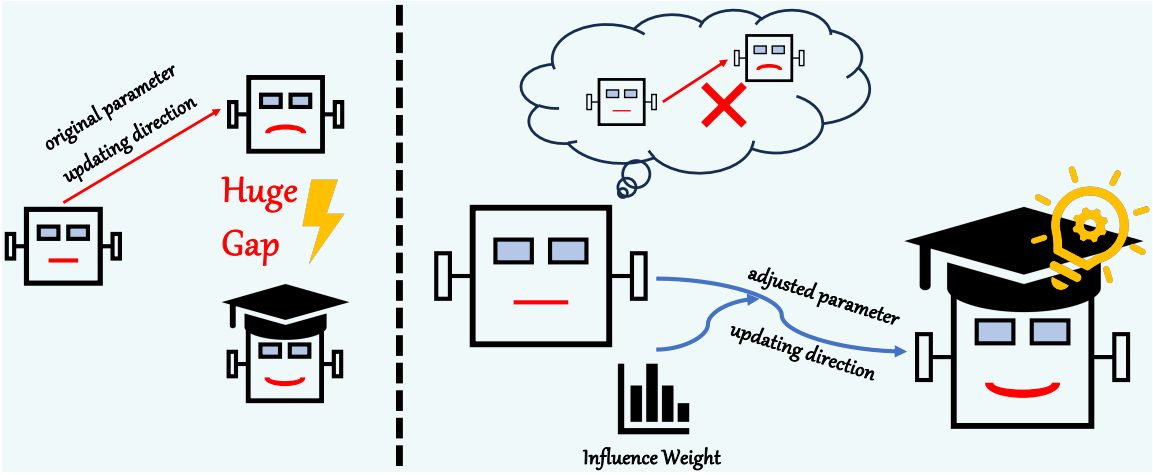

Figure 1: Demonstrations of the difference in naive fine-tuning and ATLANTIS. Naive fine-tuning (left) forces the language model to fit p_d (the sad-faced robot) that deviates from p^* (the smiling robot wearing a graduation cap). ATLANTIS (right) optimizes the language model along a different direction, which is estimated through the gap between the reference model and the optimal distribution, and finally fits the optimal distribution.

(Burns et al., 2023), we propose **ATLANTIS** (We**A**k-to-s**T**rong **L**earning for s**A**mpli**N**g ra**TI**o e**S**timation) to adjust the optimization direction towards the optimal distribution. The main difficulty in aligning models with the optimal distribution is that it is hard to sample from p^*. Fortunately, importance sampling provides a solution to this problem. Importance sampling is a sampling strategy aiming to estimate the expectations of a function f under a probability density function p from which is hard to sample through another distribution q. As a result, we can use an existing training dataset, whose distribution corresponds to q, from real conversations or LLM responses to fit the optimal distribution. Importance sampling requires that the probability $p(x)$ is calculable for a specific x to compute the sampling ratio. However, the target distribution $p*$ is incalculable because we cannot model the data distribution in the real world precisely. To tackle this problem, we introduce an extra small model to estimate the sampling ratio instead of calculating it directly. Specifically, we can redirect the optimization direction of a large base model to be finetuned towards p^* during training process by calculating the gap between a smaller base model and its corresponding finetuned checkpoint. Figure 1 provides an intuitive illustration of our method.

The contributions of this work can be summarized as follows:

- To the best of our knowledge, we take the first step to introduce importance sampling to SFT to bridge the gap between the optimal data distribution and the actual training data distribution.
- We propose ATLANTIS to estimate the sampling ratio through the probability gap between a small base model and its finetuned version, which is trained with datasets other than p^*. The evaluation results prove the effectiveness of ATLANTIS in both knowledge & understanding and preference aspects.
- Our further experiments demonstrate that our method is well-compatible with existing data selection methods and can be easily applied to existing training frameworks.

2 Methods

In this section, we will explain our proposed training technique ATLANTIS in detail. Specifically, this section will be structured as follows. In § 2.1, we will introduce relevant preliminaries including SFT and importance sampling. In § 2.2, we will illuminate our method step by step. In § 2.3, we will explore the relationship between our method and existing training techniques.

2.1 Preliminaries

Supervised Finetuning Assuming that we hope to align a model with the optimal data distribution p^*, the training target is to minimize the gap between model output distribution p_θ and p^*:

$$\mathcal{L}(\theta) = -\mathbf{E}_{x\sim q(\cdot),y\sim p^*(\cdot|x)} \log p_\theta(y|x) \quad (1)$$

Actually, it is hard to sample training instances directly from p^* and another alternative is to construct a high-quality training dataset. Given an SFT dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, the loss function of SFT is as follows:

$$\begin{aligned}\mathcal{L}(\theta) &= -\mathbf{E}_{x\sim q(\cdot),y\sim p_d(\cdot|x)} \log p_\theta(y|x) \\ &= -\sum_{x,y} p_d(y|x)[\log p_\theta(y|x)]\end{aligned} \quad (2)$$

where p_d represents the distribution of training data. However, there exists a gap between p_d and the optimal distribution from the real world, which means we will never align models with the optimal distribution p^* with limited training samples. The gaps between the training target of SFT and the real-world data distribution will limit the capacities and potentials of LLMs.

Importance Sampling Importance sampling is a Monte Carlo method used to estimate the expectation under a probability distribution from which is hard to sample. For a given probability density function $p(x)$ and a function $f(x)$, the expectation of $f(x)$ under $p(x)$ is:

$$\mathbf{E}[f] = \int f(x)p(x)dx \quad (3)$$

In this situation, we can introduce another distribution $q(x)$ from which we can sample to estimate the expectation as follows:

$$\mathbf{E}[f] = \int \frac{p(x)}{q(x)} f(x)q(x)dx \quad (4)$$

We call the term $\frac{p(x)}{q(x)}$ sampling ratio. In the traditional scenario to which importance sampling is applied, $p(x)$ can usually be computed for a given x. However, it is not calculable anymore in our settings, which is the problem to be solved.

2.2 ATLANTIS

When it comes to calculating SFT loss, we can convert Eq. 4 into the corresponding discrete form as follows:

$$\mathbf{E}[f] = \sum_x \frac{p(x)}{q(x)} f(x)q(x) \quad (5)$$

Supposing a base model p_b^L which we call the **large model**, we hope to train p_b^L to fit the optimal distribution p^*. To introduce importance sampling to SFT, we should find another appropriate distribution to estimate p^*. Internet texts, LLM generations, and human annotations are important sources of the training corpora, whose distribution we mark as p_r. p_r is always weaker than the optimal p^*, but the exact value of it for given x and y is calculable. As a result, we can estimate the expectation of $p^*(y|x)$ from $p_r(y|x)$ through importance sampling though the latter is not the optimal distribution. The loss function can be rewritten as:

$$\begin{aligned}\mathcal{L}(p_b^L) &= -\sum_{x,y} p^*(y|x)[\log p_b^L(y|x)] \\ &= -\sum_{x,y} \frac{p^*(y|x)}{p_r(y|x)} p_r(y|x)[\log p_b^L(y|x)] \\ &= -\mathbf{E}_{x\sim q(\cdot),y\sim p_r(\cdot|x)}[\frac{p^*(y|x)}{p_r(y|x)} \log p_b^L(y|x)]\end{aligned} \quad (6)$$

where the additional term $\frac{p^*(y|x)}{p_r(y|x)}$ plays the role of sampling ratio. Different from other scenarios to which importance sampling is applied, the data distribution p^* is not only hard to sample from but also impossible to calculate as aforementioned. Thus the sampling ratio is incalculable in this loss function. As a result, the main problem to solve in our work is to estimate the importance ratio appropriately without calculating $p^*(x)$.

Given a base model p_b^S smaller than p_b^L, which we call the **small model**, and its corresponding finetuned model, which can serve as the **reference model** p_r to estimate p^* in Eq. 6. Note that we do not require p_r to fit p^* perfectly and the possible distribution gap between them is allowed. According to the assumption in proxy-tuning (Liu et al., 2024), the distribution changes before and after SFT between the small and large model are proportional, which can be presented as:

$$\frac{p^*(y|x)}{p_b^L(y|x)} \propto \frac{p_r(y|x)}{p_b^S(y|x)}$$

Thus we get the estimation for the importance ratio:

$$\frac{p^*(y|x)}{p_r(y|x)} \propto \frac{p_b^L(y|x)}{p_b^S(y|x)}$$

Replacing the sampling ratio term $\frac{p^*(y|x)}{p_r(y|x)}$ in Eq. 6, the final loss function can be rewritten as:

$$\begin{aligned}\mathcal{L}(p_b^L) &= -\mathbf{E}_{x\sim q(\cdot),y\sim p_r(\cdot|x)}[\frac{p^*(y|x)}{p_r(y|x)} \log p_b^L(y|x)] \\ &\propto -\mathbf{E}_{x\sim q(\cdot),y\sim p_r(\cdot|x)}[\frac{p_b^L(y|x)}{p_b^S(y|x)} \log p_b^L(y|x)]\end{aligned} \quad (7)$$

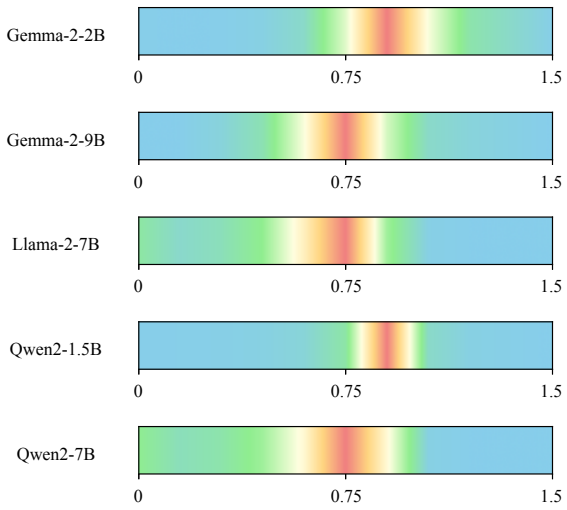

Figure 2: Similar distributions of influence weights across different models, where **red** and **blue** represent the highest and lowest density, respectively.

Compared to the vanilla loss function of SFT, we suppose that the training data is sampled from p_r in Eq. 7 and add an extra term $\frac{p_b^L}{p_b^S}$ to measure the distribution gap between the large and small models. The procedures of our method are illustrated in Algorithm 1.

Algorithm 1 ATLANTIS

Input p_b^L, p_b^S, p_r, $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, max training steps M, learning rate α

Output p^*

$W \leftarrow \{\frac{p_b^L(y_i|x_i)}{p_b^S(y_i|x_i)} p_r(y_i|x_i)\}_{i=1}^N$
$p_{\theta_0} \leftarrow p_b^L$
for $m = 1$ to M **do**
  $\mathcal{B} \leftarrow next(\mathcal{D})$
  $W_B \leftarrow next(W)$
  $\mathcal{L}(\theta_{m-1}) \leftarrow -\sum_{i \in \mathcal{B}} \frac{W_B^i}{|\mathcal{B}|} \log p_{\theta_{m-1}}(y_i|x_i)$
  $\theta_m \leftarrow \theta_{m-1} - \alpha \frac{\partial \mathcal{L}(\theta_{m-1})}{\partial \theta_{m-1}}$
end for
$p^* \leftarrow p_{\theta_M}$

2.3 Relationship with Proxy-tuning

We can transform Eq. 7 into the following format:

$$\begin{aligned}\mathcal{L}(p_b^L) &\propto -\mathbf{E}_{x\sim q(\cdot), y\sim p_r(\cdot|x)}[\frac{p_b^L(y|x)}{p_b^S(y|x)} \log p_b^L(y|x)] \\ &= -\sum_{x,y} \frac{p_b^L(y|x)}{p_b^S(y|x)} p_r(y|x) \log p_b^L(y|x) \\ &= -\mathbf{E}_{x\sim q(\cdot), y\sim p_b^L(\cdot|x)}[\frac{p_r(y|x)}{p_b^S(y|x)} \log p_b^L(y|x)]\end{aligned} \tag{8}$$

Compared to Eq. 1, we add an extra item $\frac{p_r(y|x)}{p_b^S(y|x)}$ in our proposed loss function Eq. 8, which also provides another point of view to comprehend our method. In our method, we assume that the distribution moving direction from the base model to the finetuned model can be transferred from the small model to the large model. This core idea is similar to the motivation of proxy-tuning (Liu et al., 2024), whose method can be described as follows:

$$p^*(y|x) = \text{softmax}\Big(s_b^L(y|x) + s_r(y|x) - s_b^S(y|x)\Big) \tag{9}$$

where $s(y|x)$ represents the logit scores of a model given input x. Proxy-tuning adds the gap between the reference model and the small model to the large model so that the latter can capture the knowledge and abilities in the training data without finetuning. We show the performance comparison of proxy-tuning and ATLANTIS in the Appendix. Instead of directly transferring the distribution change to larger models, we choose to use the distribution gap to measure the importance of samples during training. Thus we call the extra term $\frac{p_r(y|x)}{p_b^S(y|x)}$ **"influence weight"**. In Figure 2, we show the distribution of influence weights for different models. We can regard this extra term as a weight for each training sample. For those samples whose probabilities rise more significantly from the base model to the finetuned model, we will endow them with higher weights. As a result, our methods can be seen as measuring the influence of training samples on optimization direction during SFT.

Models	p_b^L	p_b^S	p_r
Llama2	13B Base	7B Base	7B-chat
Qwen2	7B Base	1.5B Base	1.5B-Instruct
	72B Base	7B Base	7B-Instruct
Gemma2	9B Base	2B Base	2B-it
	27B Base	9B Base	9B-it

Table 1: The settings of models in our experiments. p_b^L and p_b^S are all base models without SFT. p_r are all the official versions of finetuned models.

3 Experiments

3.1 Training Settings

We adopt three different series of models to conduct our experiments: Llama2 (Touvron et al., 2023), Qwen2 (Yang et al., 2024), and Gemma2 (Team et al., 2024). The specific settings are shown in Table 1. In order to further study the influence of model scales, we prefer model series with at least three different versions in size. We use

Model	Size	Method	Open LLM Leaderboard 2	TruthfulQA		MT-Bench		AlpacaEval		Arena-Hard-Auto
				MC1	MC2	Single	Pairwise	Easy	Hard	
Llama2	13B	SFT	32.37	36.23	53.36	6.60	19.06	77.58	6.81	3.9
		ATLANTIS	**33.50**	**36.96**	**54.07**	**6.90**	**21.56**	**80.37**	**7.80**	**5.4**
Qwen2	7B	SFT	36.22	35.37	51.45	7.44	25.63	72.57	6.41	**9.9**
		ATLANTIS	**38.19**	**35.62**	**52.65**	**7.53**	**26.25**	**75.40**	**6.51**	9.5
Gemma2	9B	SFT	33.51	36.47	52.85	6.68	20.31	72.67	6.28	5.2
		ATLANTIS	**33.86**	**39.41**	**55.23**	**6.98**	**27.50**	**76.31**	**6.92**	**5.6**

Table 2: Evaluation results for vanilla SFT and our ATLANTIS in knowledge & understanding and preference aspects. The better results for each model are highlighted in **bold**.

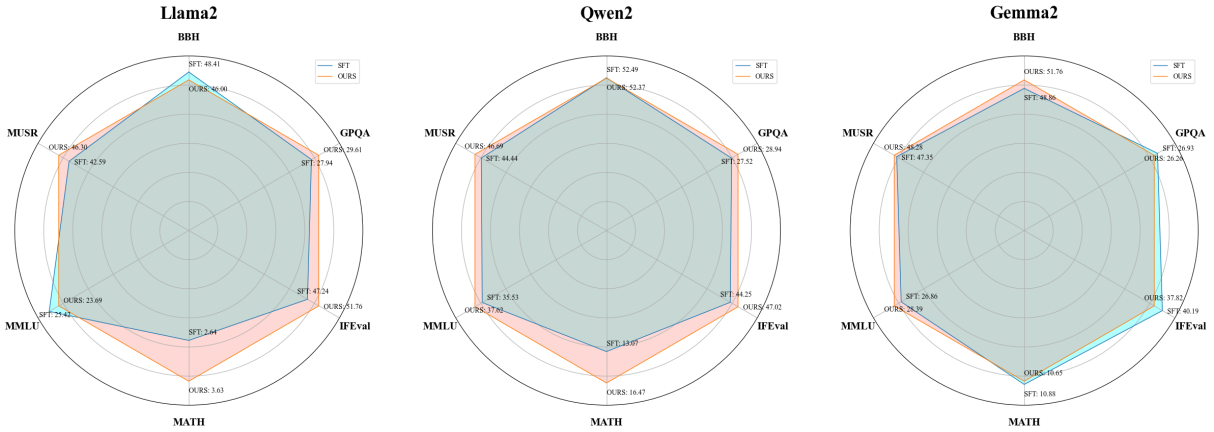

Figure 3: The specific evaluation results of Open LLM Leaderboard 2. We have stretched the scales in different dimensions for better visualization.

OpenHermes-2.5 (Teknium, 2023) as the training dataset. The implementation details are demonstrated in the Appendix.

3.2 Evaluation Benchmarks

We evaluate our methods in two aspects for a more comprehensive and promising conclusion:

Knowledge & Understanding We hope to evaluate the knowledge that models capture through SFT and their abilities to follow human instructions. We use Open LLM Leaderboard 2 (Fourrier et al., 2024), and TruthfulQA (Lin et al., 2021) for the evaluation. In specific, Open LLM Leaderboard 2 consists of six tasks including BBH (Suzgun et al., 2022), GPQA (Rein et al., 2023), IFEval (Zhou et al., 2023b), MATH-Hard (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024), and MUSR (Sprague et al., 2024). For both benchmarks, we use the evaluation scripts from lm-evaluation-harness[1]. All metrics for the two benchmarks are accuracy (or similar metrics) ranging from 0 to 1, and the higher the better.

Preference The preference of humans for the responses from a model is also an important metric. Thus we adopt MT-Bench (Zheng et al., 2023), AlpacaEval (Li et al., 2023b), and Arena-Hard-Auto (Li et al., 2024b) to evaluate human preference to models' generations. In MT-Bench (single mode) and Arena-Hard-Auto, the metric is the average score from an LLM on the model's responses. In MT-Bench (pairwise mode) and AlpacaEval, the metric is the winning rate of the model's responses to a fixed baseline model's responses judged by another LLM. All metrics are the higher the better.

3.3 Results

Our experiment results are shown in Table 2. In general, ATLANTIS brings steady and significant improvements in most cases. Our method not only enhances models' capacities in general knowledge but also increases human preference for models' responses. In the evaluation for knowledge & understanding, the benchmarks contain both multi-choice tasks and generation tasks. The exhaustive evaluations fully reflect the comprehensive improvements brought by ATLANTIS in capturing

[1]https://github.com/EleutherAI/lm-evaluation-harness

Data Selection Method			IFD Score				Superfiltering			
Model	Sample rate	Method	TruthfulQA		MT-Bench		TruthfulQA		MT-Bench	
			MC1	MC2	Single	Pairwise	MC1	MC2	Single	Pairwise
Qwen2	0.05	SFT	35.01	52.59	6.83	24.06	35.13	52.78	7.42	26.56
		ATLANTIS	**35.99**	54.15	7.54	28.75	33.90	52.98	7.37	29.69
	0.10	SFT	32.68	51.35	7.19	24.53	34.03	51.08	**7.64**	**32.19**
		ATLANTIS	35.86	**54.89**	**7.86**	**32.81**	**35.99**	**54.65**	7.38	30.00
	0.15	SFT	35.37	53.93	7.38	32.50	34.03	52.15	7.42	30.63
		ATLANTIS	34.52	52.40	7.72	32.50	35.50	54.54	7.56	**32.19**
	No selection	ATLANTIS	35.62	52.65	7.53	26.25	35.62	52.65	7.53	26.25
Gemma2	0.05	SFT	31.33	49.79	3.93	6.60	33.41	50.92	6.63	25.00
		ATLANTIS	28.76	44.24	2.73	6.88	32.93	51.68	6.61	20.63
	0.10	SFT	32.59	50.32	5.23	7.50	32.31	51.39	6.85	24.06
		ATLANTIS	35.13	53.33	5.32	11.25	35.01	53.47	**7.03**	25.31
	0.15	SFT	32.59	51.72	5.77	10.63	33.17	52.17	6.75	21.88
		ATLANTIS	34.76	53.83	5.69	11.25	34.15	53.74	6.90	21.56
	No selection	ATLANTIS	**39.41**	**55.23**	**6.98**	**27.50**	**39.41**	**55.23**	6.98	**27.50**

Table 3: Results of ATLANTIS with data selection methods. The best results for each model are highlighted in **bold**.

knowledge, logical reasoning, following instructions, and solving problems. In the evaluation for preference, ATLANTIS shows better performance in both response scores and winning rates. Specifically, the improvement in winning rates (MT-Bench pairwise mode and AlpacaEval) is more significant. Since selecting the better one from two given answers is easier and more objective than giving a score to a single response without any comparison, it makes sense that ATLANTIS is more effective in raising winning rates.

To further analyze the specific advantages brought by ATLANTIS, we demonstrate the detailed results of all six tasks from Open LLM Leaderboard 2 in Figure 3. We use different scales in different dimensions for better visualization effects. Generally, Llama2 and Qwen2 get more benefits from ATLANTIS than Gemma2 on this benchmark. For Llama2 and Qwen2, the improvements in GPQA, IFEval, and MUSR are comparably more obvious and steady. Considering that the metrics in MATH are originally low, the corresponding improvements may be not that meaningful. The metrics change pattern is quite different when it comes to Gemma2. ATLANTIS causes a slight drop in performance in GPQA and IFEval, which are the main sources of improvements for the other two models. Because the distribution of training data used in the SFT and RLHF steps for different reference models may vary a lot, the moving direction from p_b^S to p_r heavily relies on concrete model structures and parameter distributions, causing the performance change patterns of different models to be distinct from each other.

As a result, ATLANTIS can boost models' performance in different aspects and is a promising training technique that can be easily adopted nevertheless model structures or application domains.

3.4 Comparison with Data Selection Methods

The main advantage of our method is that it can be easily combined with other approaches, such as data selection. We choose two data selection methods, IFD (Li et al., 2023a) and superfiltering (Li et al., 2024a), as the baselines and use the samples selected by them to train models with our ATLANTIS. The experiment results are shown in Table 3.

We are glad to see that ATLANTIS is generally beneficial when combined with data selection methods and almost all the best results are achieved with it. When it comes to specific models, the effect of data selection methods heavily depends on model structures. Both IFD and superfiltering benefit Qwen2 on the evaluation benchmarks and achieve improvements compared to only using ATLANTIS without any data selection. However, Gemma2 fails to improve the evaluation results through data selection. All results with IFD or superfiltering fail to surpass our ATLANTIS with no data selection. Taking a look at specific benchmarks, the improvements brought by ATLANTIS to data selection are

Model	Structure of Ref/Small Model	TruthfulQA MC1	TruthfulQA MC2	MT-Bench Single	MT-Bench Pairwise
Qwen2	N/A	35.37	51.45	7.44	25.63
	Qwen2	35.62	52.65	**7.53**	**26.25**
	Gemma2	**36.23**	**52.76**	6.89	21.56
Gemma2	N/A	36.47	52.85	6.68	20.31
	Gemma2	**39.41**	55.23	6.98	**27.50**
	Qwen2	38.07	**55.40**	**7.31**	25.00

Table 4: Results of ATLANTIS-cross. The best results for each model are highlighted in **bold**.

Model	Size	Method	TruthfulQA MC1	TruthfulQA MC2	MT-Bench Single	MT-Bench Pairwise
Qwen2	7B	SFT	35.37	51.45	7.44	25.63
		ATLANTIS	35.62	52.65	7.53	26.25
	72B	SFT	41.62	61.07	8.14	31.88
		ATLANTIS	**42.59**	**61.73**	**8.17**	**36.88**
Gemma2	9B	SFT	36.47	52.85	6.68	20.31
		ATLANTIS	39.41	55.23	6.98	27.50
	27B	SFT	**43.82**	**61.24**	7.71	26.56
		ATLANTIS	**43.82**	59.30	**7.74**	**32.50**

Table 5: Evaluation results with models in larger scales. The best results for each model are highlighted in **bold**.

comparably more stable and significant on TruthfulQA than on MT-Bench, which means the effect of ATLANTIS is more outstanding in improving models' knowledge and understanding when using a small number of selected training samples.

Considering the influence of the number of training samples, increasing training data size cannot always benefit models. In general, models perform the best on all benchmarks when the sample rate is set to 0.1. On one hand, insufficient training samples (when the sample rate is set to 0.05) are not enough to train a model that can well follow the instructions and cater to human preferences. On the other hand, models may be influenced by low-quality samples if we further include more training data (when the sample rate is set to 0.15).

In conclusion, our ATLANTIS has great potential to be combined with other training techniques and can play an important role in current SFT frameworks. The combination of ATLANTIS with more and more training techniques can bring steady improvements and deserves further exploration.

4 Analyses

4.1 Analysis on Model Scale

To prove the effectiveness of ATLANTIS on models of different sizes, we further conduct experiments using Qwen2-72B and Gemma2-27B, of which the corresponding reference models and small models can be referred to Table 1. We use TruthfulQA and MT-Bench as the evaluation benchmarks. The results are shown in Table 5.

As we have expected, the evaluation results prove that ATLANTIS still works for larger models in most cases and we receive appreciable improvements on both benchmarks, especially in the pairwise mode of MT-Bench. The only performance drop happens in Gemma2 on TruthfulQA, but the decrease is acceptable considering the significantly increasing scores in other situations. When we fine-tune Qwen2-72B and Gemma2-27B, distributions of model parameters may differ more considerably between the large and small models than when we finetune smaller models. In such cases, ATLANTIS still helps models achieve steady improvements, strongly proving its effectiveness and stability.

4.2 ATLANTIS-cross: Exchanging the Reference and Small Models

In all our previous experiments, the reference model, small model, and large model share the same model structure. Since Li et al. (2024a) finds that models with different sizes or structures have a similar distribution in IFD scores, we can suppose that the influence weights in our ATLANTIS can also be transferred among models with different structures. Specifically, we train Qwen2 and Gemma2 with the influence weights calculated by the reference and small models of each other. We call this method ATLANTIS-cross. The evaluation results are shown in Table 4.

Surprisingly, using models with different structures to calculate influence weights does not result in a disastrous loss in performance. In most cases, ATLANTIS-cross can bring a comparable increase in evaluation results, even surpassing ATLANTIS in some metrics. The results verify that models can benefit from the probability changes in other models with different structures and our method can be transferred among different model structures.

We must notice that ATLANTIS-cross loses effect in Qwen2 on MT-Bench, especially in the pairwise mode. This phenomenon may be relative to the parameter distribution of Gemma2. The change of distribution between the reference model and the small model of Gemma2 is only beneficial in guiding the optimization direction for higher human preference using the same model structure, thus causing a loss in performance when applied to

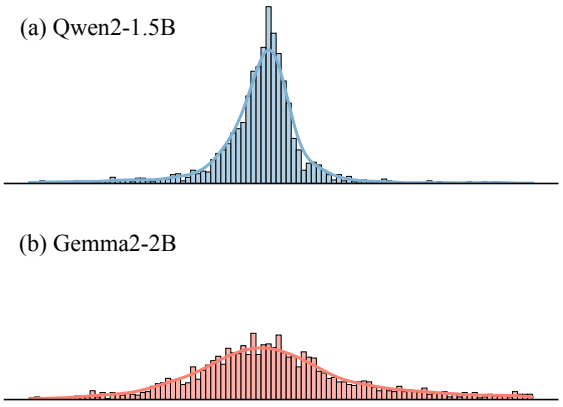

Figure 4: The visualization of different distributions in influence weight of Qwen2-1.5B and Gemma2-2B, according to randomly selected 2000 samples.

other models.

In general, models can benefit from the distribution change of other models with different structures to some extent. Figure 4 illustrates the difference in influence weight distribution between Qwen2 and Gemma2, where they share a close mean but different variance. To be specific, data points from Qwen2 are highly concentrated in a narrow range, with a higher peak at the center, while those from Gemma2 showcase a larger variance. This difference further results in distinctions of positive/negative predictions. For example, if we set 1.0 of influence weights as the border of such predictions, the rate of disagreement in Qwen2 and Gemma2 is 46.78%, providing an insight into the harm brought by Gemma2 to Qwen2 in ATLANTIS-cross.

5 Related Work

Supervised Finetuning Supervised finetuning is the follow-up training step after pretraining and allows models to fit on specific datasets for specific capacities, e.g. following human instructions, and learning knowledge from special domains. The successively proposed SFT techniques, e.g. prompt tuning (Brown et al., 2020), prefix tuning (Li and Liang, 2021; Liu et al., 2021), and instruction tuning (Wei et al., 2021; Ouyang et al., 2022; Muennighoff et al., 2022), greatly promote the development of LLMs. In our work, we conduct all experiments with instruction tuning.

Data Selection for Instruction Tuning Training samples can have different impacts on the optimization of language models, encouraging active exploration of data selection strategies. Similar to most data selection work, we endow each training sample with a score to measure its influence or importance. A typical way is to design a model-agnostic (Song et al., 2024) or model-aware (Chen et al., 2023; Bukharin and Zhao, 2023; Du et al., 2023) scoring formulation. Moreover, instead of directly using scores from LLMs to measure data quality, Li et al. (2023a) introduces IDF scores to calculate the fraction between the perplexities of outputs with and without chat templates using a finetuned model. Li et al. (2024a) finds out that IDF scores calculated by small models can be transferred to the data selection for larger models. Focusing on the optimization process, Xia et al. (2024) adopts the reduction in loss function to judge the influence of training samples. In our work, we refer to the ideas in these works and propose to calculate influence weights by the probabilities of outputs from base models and corresponding finetuned models.

6 Conclusion

In this work, we propose a new training technique ATLANTIS to deal with the problem of existing gaps between training data distribution and the optimal distribution. Because the ideal optimal distribution is hard to sample from, we introduce importance sampling method to fit it. Furthermore, we adopt an extra reference model and a small model to estimate the sampling ratio since it cannot be computed directly. We evaluate ATLANTIS with benchmarks in two different aspects, knowledge & understanding and human preference, and the results prove the effectiveness of our method in both aspects. We further analyze the potential of our method to be combined with other SFT techniques. The experiments show that ATLANTIS is well compatible with data selection methods. What's more, ATLANTIS does not require that the large model and small model must share the same model structure. The influence weights calculated by one certain model structure can be easily transferred to other models.

In conclusion, ATLANTIS can serve as a promising SFT technique and be attached to existing training frameworks to adjust the optimization direction to the theoretically optimal target in the real world.

Limitations

Though our method can be transferred from one model structure to another, the distribution difference in influence weights will affect models' performances in some cases. To make full use of ATLANTIS, a smaller model from the same model series is required, which may limit the application domains of our method to some extent.

What's more, the introduction of the reference model will bring extra computation cost compared to vanilla SFT. Since the reference model is much smaller than the base model to be finetuned, the extra cost is acceptable in most cases, especially considering the significant improvements brought by ATLANTIS.

Acknowledgments

We sincerely thank all reviewers for their insightful suggestions. This research was partially supported by the National Natural Science Foundation of China under Grant No.92470205 and No.62176002. Xu Sun is the corresponding author.

References

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Alexander Bukharin and Tuo Zhao. 2023. Data diversity matters for robust instruction tuning. *arXiv preprint arXiv:2311.14736*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *Preprint*, arXiv:2312.09390.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Autodebias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1012–1023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Preprint*, arXiv:2103.03874.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024a. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. *arXiv preprint arXiv:2402.00530*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023a. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024b. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *Preprint*, arXiv:2406.11939.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

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. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

OpenAI. 2022. Introducing chatgpt.

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

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Feifan Song, Bowen Yu, Hao Lang, Haiyang Yu, Fei Huang, Houfeng Wang, and Yongbin Li. 2024. Scaling data diversity for fine-tuning language models in human alignment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14358–14369.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *Preprint*, arXiv:2310.16049.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

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

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

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

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Charles Lau, Ryutaro Tanno, Ira Ktena, et al. 2024. Towards generalist biomedical ai. *NEJM AI*, 1(3):AIoa2300138.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

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

Fan Zhou, Yuzhou Mao, Liu Yu, Yi Yang, and Ting Zhong. 2023a. Causal-debias: Unifying debiasing in pretrained language models and fine-tuning via causal invariant learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4227–4241.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Parameters		**Model**	**Value**
Hardwares	GPUs	$\leq$ 13B	16 × A800
		27B	32 × A800
		72B	64 × A800
Hyperparams	training steps	$\leq$ 13B	60K
		27B	30K
		72B	15K
	batch size	$\leq$ 13B	2
		27B	1
		72B	1
	learning rate	$\leq$ 13B	2e-5
		27B	5e-6
		72B	5e-7
	gradient accumulation steps	$\leq$ 13B	8
		27B	8
		72B	4
	optimizer	/	AdamW
	random seed		0
scheduler	warmup type	/	linear
	decay type		linear
	warmup steps	$\leq$ 13B	6K
		27B	3K
		72B	1.5K
deepspeed	zero stage	/	3
	optimizer offload		True
	parameter offload		True
packages	accelerate	/	0.30.0
	deepspeed		0.13.5
	torch		2.1.0+cu118
	transformers	gemma2	4.43.4
		others	4.40.2

Table 6: Implementation details of our experiments.

A Implementation Details

The implementation details of experiments and relevant Python packages are shown in Table 6. All experiments are conducted with one random seed.

B Comparison with Proxy-tuning

Our method and proxy-tuning share the same assumption that the distribution moving directions before and after finetuning are proportional for the small and large models. To compare the effect of these two methods, we also evaluate proxy-tuning with the results demonstrated in Table 7.

As we can see, ATLANTIS has obvious advantages to proxy-tuning in both benchmarks, especially in MT-Bench. Though saving the cost of further training, proxy-tuning cannot bring a steady and significant improvement compared to the fine-tuned model.

Model	Method	TruthfulQA		MT-Bench	
		MC1	MC2	Single	Pairwise
Qwen2-7B	SFT	35.37	51.45	7.44	25.63
	proxy-tuning	33.41	50.36	3.76	7.91
	ATLANTIS	**35.62**	**52.65**	**7.53**	**26.25**

Table 7: Evaluation results of proxy-tuning. The best results are highlighted in **bold**.