Complexity-aware fine-tuning

Anonymous ACL submission

Abstract

General purpose Large Language Models (LLMs) are frequently fine-tuned to improve performance in niche domains. Although fine-tuning is a standard practice, we still lack a deep understanding of how to aggregate data for better results. In this work, we show that the entropy-based output estimation provides a meaningful guideline for fine-tuning data preparation. Specifically, across two small open models ($\approx 3B$) we find that a single token answer entropy shows ROC AUC score of $\approx$ 0.73 and allows us to split the training data into three complexity categories to apply different tuning mechanisms. As result, we propose a novel blueprint for efficient fine-tuning that outperforms the standard approach (0.5/0.6 vs. 0.4/0.46 accuracies). We also provide an in-depth analysis of alternative complexity estimation techniques based on expert assessment via model-as-judge (MASJ) and chain-of-thought entropy aggregation with ROC AUC scores of 0.57 and 0.7 accordingly. Our findings show immediate enhancements in fine-tuning performance. We publish our coda[1] and data[2] to facilitate further investigation and immersion of the numerical complexity analysis.

1 Introduction

General-purpose LLMs demonstrate impressive performance across a vast variety of domains. At the same time, they show subpar results on niche tasks and in niche domains compared to specialized models. Properly tuned smaller models beat large open models in mathematics (Yang et al., 2024b), medicine (Wu et al., 2025), chemistry (Yu et al., 2024), and other fields. While in some areas, the difference is negligible or at least not as important, in others, such as medicine, it becomes crucial due to the high cost of error.

[1] https://github.com/sdjng3q897aeiufnad/complexity-aware-fine-tuning

[2] anonymized

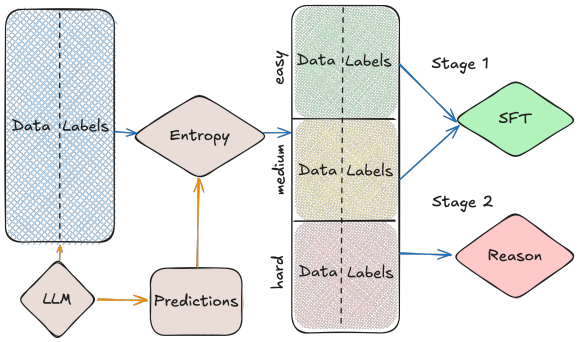


Figure 1: Complexity-aware fine-tuning scheme: for easy and medium complex questions we apply direct SFT, while for hard questions we include reasoning results during SFT.

A standard approach to get a domain-specific model is to fine-tune a general-purpose base LLM on carefully selected data (Parthasarathy et al., 2024). While there is a lot of focus on perfecting the training mechanics, fewer attempts have been made to enhance the performance by pursuing meaningful data separation and ordering. The default approach is fine-tuning a model on the whole data set without particular ordering. There have been experiments with curriculum-based learning (Kim and Lee, 2024; Shi et al., 2025), but they struggle to split the dataset by difficulty in an automated fashion meaningfully.

To address this gap, we propose a fully automated pipeline, given in Figure 1, that consists of two steps: (1) to split fine-tuning data by the tasks' complexity via a response entropy and (2) to train the model for via either supervised fine-tuning (SFT) followed by the reasoning-promoting techniques (such as Group Relative Policy Optimization and others (Zhihong Shao, 2024; Face, 2025)) for easy and hard data entries accordingly. We confirm the effectiveness of the framework by fine-tuning two open models: Qwen2.5-3B (Yang et al., 2024a) and Phi-4-Mini (Microsoft et al., 2025),

on the multiple choice question answering dataset MMLU-Pro (Wang et al., 2024).

In step one, we consider different options for complexity estimation based on a model's confidence in its answer. Our study considers a wide range of methods: model-as-judge (MASJ), entropy-based aggregation and calculation methods, as well as other numerical methods. We provide the obtained dataset, which includes the generation results accompanied by token distribution at each step, for further research.

Our contributions:

- The automated pipeline to split a multiple-choice question answering dataset by complexity based on the token-wise entropy of the response and apply different fine-tuning strategies based on the group. The code is available at anonymized Github repository [3].
- An training procedure that achieves accuracies 0.5048 and 0.6005 over two LLMs for MMLU-Pro c.t. 0.4048 and 0.4619 baselines.
- An analysis of complexity estimation based rooted in uncertainty estimation, including MASJ, entropy-based and entropy-based augmented with reasoning results with aggregation on top. Aggregated entropy-based complexity provides the best results with ROC AUC 0.7.
- Open-source standardized datasets[4] to facilitate further development of uncertainty estimation and calibration methods: with and without chain-of-thoughts, with token probability distribution at each step provided, as well as additional scores.

2 Related works

Curriculum learning has been explored to improve LLM fine-tuning by ordering training examples from easier to harder. Kim and Lee (2024) propose sorting fine-tuning data by difficulty metrics (e.g. prompt length, model attention scores, and initial loss) so that the model learns on simpler prompts before complex ones. They found that this curriculum strategy yielded slightly higher accuracy than random shuffling, with ordering by an attention-based criterion performing best. This approach is attractive because it boosts performance without adding more data or parameters. However, the gains were modest, and defining difficulty automatically can be tricky - their method requires measuring things like loss or attention per example.

Another strategy is filtering training data for quality. A notable example is LIMA (Zhou et al., 2023a), which shows that a large pre-trained model can be fine-tuned on just a small, high-quality subset of data. They fine-tuned a 65B LLAMA model on only 1000 carefully curated prompt-response pairs (chosen for diversity and clarity) without any reinforcement learning. Despite the tiny dataset, the resulting model performed remarkably well, learning to handle complex queries and even generalizing to tasks not seen in training. In a human evaluation, LIMA's answers were preferred over GPT-4's in 0.43 of cases. This "less is more" result suggests that much of an LLM's ability comes from pre-training, and fine-tuning needs only a small amount of exemplary data to unlock it. However, LIMA relied on a large base model and manual data curation. The approach may not scale down to smaller models and requires human intervention.

Another notable example of curated data selection is the SmallToLarge (S2L) method by Yang et al. (2024c), which leverages training trajectories from small models to guide the data selection for larger models. This way, the large LLM is trained on a diverse yet compact dataset covering different difficulty levels. S2L showed impressive results: for a math word problem dataset, they achieved the same accuracy using only 11% of the data, and even outperformed other selection methods by 4.7% on average across several benchmarks. The strength of this approach is that it makes complexity-based data filtering automated and cheap. One caveat is the extra step of training a smaller model and clustering. The approach is mostly tested on specialized domains (math problems, clinical text summarization), so its generality to all types of tasks needs further validation. Additionally, it requires a large amount of data to make a filtered subset.

Sychev et al. (2025) focus on measuring example difficulty via model uncertainty. They investigate how an LLM's token-level entropy in its answers relates to question difficulty. They find that a model's response entropy correlates strongly with question difficulty, especially in knowledge-based domains. They also introduce MASJ reasoning score to estimate the question complexity. How-

[3] https://github.com/sdjng3q897aeiufnad/complexity-aware-fine-tuning

[4] anonymized

ever, the authors use these metrics only to analyze model behavior. They do not integrate it into a practical data aggregation or fine-tuning workflow. Additionally, their work does not cover analysis of the chain-of-thought kind of responses.

3 Methods

3.1 Training pipeline

We propose the complexity-aware fine-tuning pipeline (Figure 1) with the following major stages: complexity estimation, data aggregation, fine-tuning.

Complexity estimation. We adopt the entropy of the answer token in the response as our primary complexity metric. We prompt the model to pick the correct option directly, without producing a chain-of-thought. See Section 3.2.3 for details.

Data aggregation. To aggregate the data into groups by complexity (easy, medium, hard) we evenly divide the dataset into three parts ordering the entries by entropy of the response. Group with the lowest entropy values is categorized as easy and with the highest values - as hard, see Figure 2.

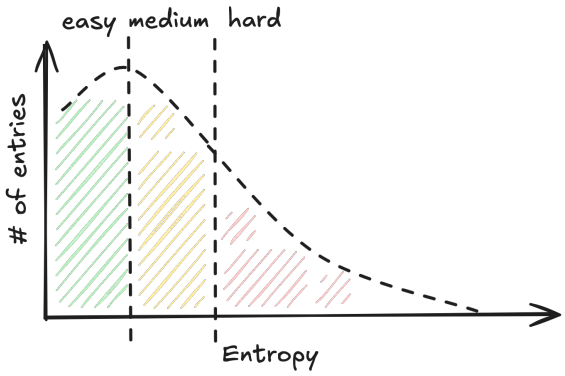


Figure 2: Data aggregation

Fine-tuning. We propose to apply different fine-tuning strategies according to the complexity of the data. Vanilla SFT is applied to easy- and medium-complexity groups, while for hard questions we augment this procedure with a chain-of-thought.

For easy and medium groups, we suggest to use SFT (Howard and Ruder, 2018; Raffel et al., 2023) as an established practice to enhance the model performance. It involves fine-tuning a pretrained LM on labeled examples using standard supervised objectives. Here, we prompt (see Table 6 for system prompts and Table 1 for user prompt) the model with the question and options. Then, we apply the cross-entropy loss to the question answers used as labels.

Question: ... Options: 1. ... 2. ... ... n. ... Choose one of the answers. Write down ONLY the NUMBER of the correct answer and nothing else.

Table 1: User prompt

As to the hard group, we hypothesize that as hard questions require multiple logical steps the model can not effectively learn with standard SFT and answering the question directly. Instead, we propose to elicit a chain-of-thought and allow the model to incrementally build the answer step-by-step as suggested by Wei et al. (2023).

In this work, we apply the distillation technique —- train a smaller model on the chain-of-thought of a larger LLM. It is well-known practice supported by (Hsieh et al., 2023). To create the distillation training samples, we prompt a large LLM to answer the multiple choice question and produce a chain-of-though in the process. Next, the whole response is attached to the dataset and used to train the smaller model. Hypothetically, other reasoning-eliciting techniques might show good performance at this stage as well.

3.2 Complexity estimation approaches

To find the most suitable metric for the training pipeline we analyze the performance of the following techniques, MASJ reasoning score, MASJ education level, Single token answer entropy, Chain-of-thought answer entropy, Chain-of-thought aggregated response entropy, Thinking and answer statistics of reasoning model.

Used prompts are available in Appendix A.1.

3.2.1 MASJ reasoning score

As one of the expert-like metrics, we ask a large LLM to estimate the number of logical steps required to answer the question. The hypothesis is that the questions that require more reasoning should be harder for the model to answer.

To collect the MASJ-based reasoning score, we go over the multiple choice question answering dataset and query a large auxiliary LLM for the estimate. We prompt the model to provide the number of logical steps required to answer the question: low, medium, high. Next, we query the large LLM

again to estimate the quality of the previous assessment from 1 to 10 following the practice introduced in MT-Bench by Zheng et al. (2023). It allows us to filter out low quality scores by keeping only the ones rates above or equal to 9.

3.2.2 MASJ education level

As the other expert-like metrics, we ask a large LLM to estimate the required level of education to answer the question correctly. It is a natural human-like value used in other datasets (Rein et al., 2023; Lu et al., 2022).

We follow the same procedure as for MASJ reasoning score, but use a different prompt.

3.2.3 Single token answer entropy

In similar fashion as proposed by (Kadavath et al., 2022), we calculate the entropy of the answer token in the response. The assumption is that the response uncertainty is a natural predictor of the question complexity. We prompt the model to answer the question directly (as a single token) and calculate tokenwise entropy of the response as follows:

$$h = -\sum_{i=1}^{n} p_i \log p_i,$$

where p_i is the probability of a single token and n is the vocabulary size.

Additionally, similarly to (Zhou et al., 2023b), we examine the performance when we allow the model to explicitly say "I do not know" (IDK).

3.2.4 Chain-of-thought answer entropy

With the same assumption as for the single token entropy, we analyze the entropy of the answer token, but change the prompt to elicit a chain-of thought type of response. The assumption is that via the chain-of-thought the LLM can incrementally accumulate the entropy resulting in a better separation of certain and uncertain answers.

3.2.5 Chain-of-thought aggregated response entropy

Building upon the single-token entropy approach, we investigate more sophisticated methods for complexity estimation by analyzing the entire chain-of-thought (CoT) response. While the answer token entropy provides a localized measure of uncertainty, aggregating entropy across the complete reasoning process potentially offers a more comprehensive complexity assessment.

We evaluate 10 distinct aggregation methods applied to CoT responses and last answer token, comparing their effectiveness through ROC AUC and Gini metrics across multiple models (Qwen 3B and Phi4-mini, both with and without "I don't know" option). Our analysis consider the following methods:

1. COT word-aggregation methods
   - Single Token Answer Entropy
   - COT Mean
   - COT Max
   - Difference between COT Max and Single Token Answer Entropy
2. COT sequence-aggregation methods
   - Sequence Mean of Words Mean
   - Sequence Max of Words Mean
   - Sequence Mean of Words Max
3. Probability-based methods
   - Mean of Marginal Difference - mean of difference between top-2 probabilities for each token of response
   - Top-2 Entropy Difference - difference of top-2 highest entropies for response
4. Hybrid method
   - Mix of COT word-aggregation methods - linear combination of the best perform methods

More details can be found in Appendix B.

3.2.6 Thinking and answer statistics of reasoning model

To further investigate how numerical estimates can be applied for uncertainty quantification, we analyze the entropy and length of the reasoning chain for the current state-of-the-art (SOTA) reasoning model.

During inference, for each newly generated token, we store the probability distribution over the vocabulary of tokens with non-zero probabilities. To find the importance of the features, we train a logistic regression classifier using the scikit-learn (Pedregosa et al., 2011) to predict the correctness of the model answer.

4 Results

4.1 Experimental setup

MMLU-Pro dataset We conduct all experiments on the multiple choice question answering dataset MMLU-Pro (Wang et al., 2024), widely adopted by the community as one of the golden performance benchmarks. It spans across 14 domains with a broad selection of questions with different complexity. Each question has approximately 10 options with a single correct one, which removes the ambiguity in evaluation.

Used LLMs As to the models, we use a variety of open model sizes for data collection and aggregation to analyze how the trend changes with the model size. At the same time, we focus on smaller models for fine-tuning to make our results reproducible.

We apply our overall pipeline to two models: Qwen2.5-3B and Phi-4-Mini. For them, we measured single token entropy, collected chain-of-thought entropy, metadata, fine-tuned models and evaluated overall pipeline.

Auxiliary models are used to extend our analysis and allow advanced reasoning: single token response entropy for Mistral 24B and Phi-4; reasoning scores and education levels via MASJ with Mistral 123B (Mistral, 2024); reasoning entropy with metadata for Qwen3-8B; Chain-of-thought distillation - DeepSeek-V3-0324 (DeepSeek-AI, 2024).

All models and datasets are published under permissive licenses that allow using them for research purposes.

4.2 Complexity estimation evaluation

Following existing practices, we consider three families of uncertainty estimation methods: MASJ, entropy-based and entropy-based augmented with chain-of-thoughts. MASJ results, as they are inferior to other, are provided in Appendix C.1, for all other methods we provide the results of analysis below.

4.2.1 Single token and chain-of-thought answer entropy

Tables 2 and 3 present ROC AUC and accuracy for single token entropy and for the entropy of the answer token in the chain-of-thought type of response respectfully. Metrics are calculated for the categories provided by MMLU-Pro as well as education levels and reasoning scores estimated via MASJ.

IDK responses and results with invalid formatting are excluded from the calculations.

We can notice that the accuracy tends to be slightly higher when we allow LLM to answer IDK. At the same time, IDK responses do not consistently affect ROC AUC for all models.

Chain-of-thought responses tend to provide higher accuracy, but lower ROC AUC scores which makes them less suitable for complexity estimation.

4.2.2 Chain-of-thought aggregated response entropy

Table 4 provide results, which highlight our key observations:

- Simple answer entropy often outperforms more complex COT aggregation methods, particularly in models with strong baseline performance (e.g., Qwen-3B achieving 0.68 ROC AUC). Although the hybrid method outperforms the answer entropy, the main weights of the hybrid linear combination were assigned to the answer entropy.

- Maximum-based measures (COT Max, Seq Mean Max) consistently outperform mean-based approaches (COT Mean, Seq Max Mean and Seq Mean Mean), suggesting peak uncertainty moments may better indicate question difficulty than average uncertainty.

- Sequence-based methods did not show good improvements over basic aggregation, indicating that modeling the reasoning structure provides marginal benefits.

- The poor close-to-random performance of the difference in top entropies suggests that modern LLMs maintain relatively stable reasoning to outliers.

4.2.3 Thinking and answer statistics of reasoning model

Our classifier achieves 0.721, 0.717 and 0.731 accuracies by using thinking total entropy, length of the reasoning chain or both features combined.

Table 12 shows that total entropy and number of tokens of the reasoning chain are the most important parameters influencing the correctness of the model's prediction.

Category	Qwen 3B	Qwen 3B*	Phi4-mini	Phi4-mini*	Phi4	Phi4*	Mistral 24B	Mistral 24B*
All	0.72/0.33	0.70/0.33	0.72/0.40	0.74/0.46	0.80/0.51	0.80/0.58	0.75/0.49	0.74/0.60
Law	0.63/0.24	0.60/0.21	0.64/0.29	0.62/0.30	0.69/0.47	0.69/0.48	0.69/0.41	0.75/0.56
Business	0.67/0.28	0.71/0.26	0.67/0.31	0.64/0.38	0.73/0.36	0.75/0.44	0.69/0.40	0.68/0.43
Psychology	0.77/0.51	0.75/0.51	0.84/0.57	0.82/0.59	0.84/0.74	0.84/0.74	0.79/0.66	0.75/0.68
Chemistry	0.69/0.23	0.62/0.24	0.62/0.34	0.64/0.41	0.70/0.34	0.77/0.45	0.68/0.38	0.75/0.59
Biology	0.79/0.59	0.79/0.56	0.85/0.67	0.85/0.73	0.90/0.80	0.90/0.83	0.81/0.74	0.73/0.80
History	0.66/0.36	0.63/0.35	0.68/0.39	0.65/0.43	0.76/0.62	0.73/0.63	0.69/0.54	0.64/0.56
Other	0.70/0.33	0.67/0.34	0.72/0.39	0.74/0.43	0.81/0.57	0.82/0.58	0.79/0.52	0.75/0.59
Physics	0.65/0.27	0.64/0.28	0.65/0.32	0.66/0.40	0.75/0.39	0.78/0.46	0.74/0.38	0.71/0.63
Computer science	0.76/0.29	0.70/0.32	0.73/0.41	0.76/0.46	0.77/0.55	0.80/0.57	0.77/0.51	0.74/0.64
Health	0.69/0.39	0.66/0.39	0.71/0.43	0.71/0.47	0.78/0.64	0.77/0.65	0.75/0.61	0.71/0.63
Economics	0.77/0.44	0.74/0.43	0.79/0.55	0.80/0.59	0.85/0.68	0.83/0.72	0.77/0.62	0.75/0.66
Math	0.69/0.24	0.67/0.24	0.65/0.27	0.69/0.31	0.73/0.37	0.74/0.43	0.69/0.33	0.72/0.44
Philosophy	0.66/0.33	0.70/0.31	0.71/0.39	0.70/0.43	0.77/0.61	0.76/0.63	0.71/0.53	0.70/0.56
Engineering	0.67/0.34	0.66/0.32	0.62/0.39	0.64/0.45	0.74/0.43	0.67/0.53	0.70/0.46	0.77/0.60
Education level								
High school and easier	0.73/0.35	0.72/0.34	0.76/0.38	0.75/0.51	0.81/0.50	0.82/0.54	0.75/0.48	0.70/0.52
Undergraduate	0.73/0.34	0.71/0.34	0.72/0.42	0.77/0.44	0.81/0.52	0.82/0.62	0.77/0.50	0.74/0.64
Graduate	0.66/0.28	0.65/0.26	0.64/0.35	0.68/0.37	0.74/0.50	0.73/0.54	0.71/0.46	0.76/0.58
Postgraduate	0.63/0.18	0.52/0.20	0.64/0.20	0.63/0.22	0.67/0.40	0.65/0.41	0.62/0.35	0.63/0.39
MASJ reasoning score								
Low	0.72/0.42	0.71/0.42	0.78/0.48	0.79/0.51	0.82/0.64	0.83/0.65	0.79/0.59	0.73/0.59
Medium	0.72/0.32	0.70/0.31	0.70/0.39	0.72/0.44	0.79/0.50	0.79/0.59	0.74/0.47	0.76/0.63
High	0.64/0.27	0.62/0.27	0.59/0.33	0.58/0.36	0.69/0.41	0.64/0.29	0.64/0.39	0.62/0.45

Table 2: ROC AUC/accuracy for single token response entropy
* Alternative prompt to allow model answer "I do not know"

Category	Qwen 3B	Qwen 3B*	Phi4-mini	Phi4-mini*
All	0.68/0.41	0.67/0.41	0.61/0.43	0.58/0.55
Law	0.60/0.24	0.57/0.23	0.55/0.26	0.52/0.28
Business	0.68/0.45	0.67/0.47	0.66/0.55	0.56/0.65
Psychology	0.73/0.51	0.70/0.51	0.68/0.48	0.65/0.63
Chemistry	0.65/0.41	0.68/0.39	0.65/0.43	0.63/0.60
Biology	0.77/0.56	0.68/0.60	0.65/0.48	0.67/0.71
History	0.62/0.36	0.61/0.36	0.59/0.37	0.51/0.39
Other	0.63/0.38	0.63/0.36	0.60/0.42	0.58/0.52
Physics	0.68/0.42	0.67/0.41	0.62/0.39	0.61/0.57
Computer science	0.68/0.37	0.73/0.33	0.59/0.41	0.58/0.58
Health	0.63/0.37	0.61/0.40	0.62/0.33	0.56/0.50
Economics	0.70/0.48	0.68/0.50	0.61/0.47	0.65/0.63
Math	0.73/0.51	0.73/0.48	0.63/0.58	0.60/0.67
Philosophy	0.63/0.33	0.62/0.35	0.66/0.37	0.59/0.48
Engineering	0.63/0.31	0.64/0.33	0.60/0.37	0.55/0.45
Education level				
High school and easier	0.72/0.56	0.70/0.53	0.66/0.57	0.60/0.73
Undergraduate	0.67/0.41	0.67/0.41	0.62/0.42	0.60/0.55
Graduate	0.61/0.27	0.60/0.28	0.58/0.30	0.57/0.38
Postgraduate	0.63/0.22	0.66/0.15	0.41/0.22	0.41/0.20
MASJ reasoning score				
Low	0.69/0.49	0.67/0.49	0.64/0.46	0.61/0.62
Medium	0.67/0.41	0.66/0.41	0.60/0.43	0.57/0.55
High	0.65/0.26	0.60/0.26	0.53/0.32	0.54/0.36

Table 3: ROC AUC/accuracy for single token response entropy
* Alternative prompt to allow model answer "I do not know"

Method	Qwen 3B	Qwen 3B*	Phi4-mini	Phi4-mini*
Answer Entropy (AE)	0.68	0.67	0.61	0.58
COT Mean	0.59	0.58	0.59	0.63
COT Max	0.63	0.61	0.6	0.65
Sequence Mean Mean	0.6	0.59	0.6	0.62
Sequence Max Mean	0.59	0.58	0.59	0.61
Sequence Mean Max	0.62	0.6	0.59	0.62
Marginal Diff Mean	0.58	0.57	0.58	0.61
Top-2 Entropies Diff	0.51	0.5	0.5	0.51
COT Max minus AE	0.54	0.53	0.51	0.57
COT Max and AE	0.7	0.69	0.62	0.62
Number of Samples	11049	10724	9997	9973

Table 4: ROC AUC values for COT response
* Alternative prompt to allow model answer "I do not know"

Technical details. To avoid excessively long reasoning chains, we set a maximum generation length of 5000 tokens. We also use normalized parameters to remove the mean and scale to unit variance. We take model coefficients of the corresponding parameters as their importance.

4.3 Fine-tuning

4.3.1 Data split by MASJ reasoning score

We randomly split the data into train, validation, and test with ratio 70:10:20. Next, in each chunk, we balance the data so that the number of entries in each complexity group is equal using MASJ reasoning score as a complexity metric. Since there are fewer hard questions, we filter out medium and easy ones to match the size of the hard group.

Figures 5a and 5b show the results of SFT for each group. We do not see a strong difference in performance between the groups. In combination with questionable ROC AUC scores, provided in Table 11, it makes MASJ reasoning score a less favorable metric for further experiments.

4.3.2 Data split by single token entropy

We follow the same logic to split the data, but use single token entropy as a metric.

Figures 5c 5d show the results of SFT for each group. For Phi-4-mini, we see that medium and easy questions outperform hard ones for the first 10 epochs. Shortly after, performance starts to decline for all groups. For Qwen 3B, we do not see a significant difference between the groups. Moreover, the performance plateaus after 5 epochs.

4.3.3 Complexity-aware pipeline

Based on ROC AUC results and positive performance of SFT for medium and easy questions on Phi-4-mini, we take single token entropy as our complexity metric for the pipeline described in Section 3.1. The same logic as before applied to split the data into train, validation and test, as well as complexity groups.

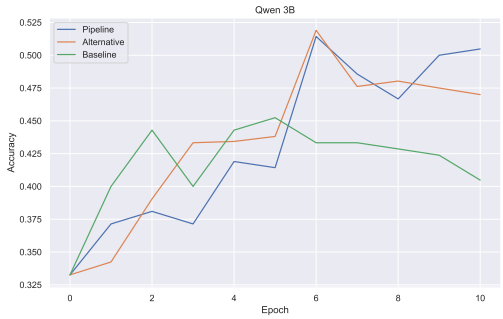


Figure 3: Accuracy for complexity-aware fine-tuning pipelines after 10 epochs (Qwen 3B)

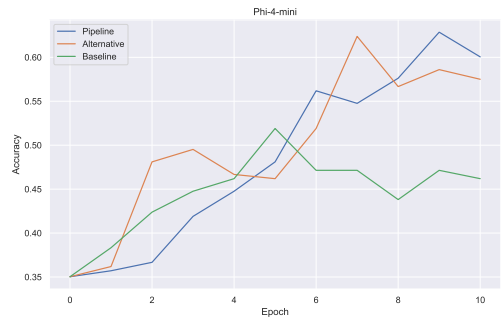


Figure 4: Accuracy for complexity-aware fine-tuning pipelines after 10 epochs (Phi-4-mini)

For easy and medium groups we perform SFT for 5 epochs. For the hard group, we apply learning from a distilled chain-of-thought of a larger model for another 5 epochs.

As the alternative approach, we perform SFT for 5 epochs for the hard group. Next, for easy and medium groups, we apply learning from a distilled chain-of-thought of a larger model for another 5 epochs.

As our baseline, we train the model via SFT without the data split for 10 epochs.

Figures 3 4 and table 5 show the results. We see, that the proposed training scheme results in significant improvement over the baselines and an alternative training scheme, that uses distillation for only easy and medium questions. Qwen 3B achieves accuracy of 0.5048 compared to 0.47 and 0.48 (alternative and baseline), while Phi-4-mini gets to 0.6005 compared to 0.575 and 0.4619.

Method	Qwen 3B	Phi4-mini
Baseline	0.4048	0.4619
Alternative	0.4700	0.5750
Ours	0.5048	0.6005

Table 5: Accuracy for complexity-aware fine-tuning pipelines after 10 epochs

5 Conclusion and discussion

This paper introduces a complexity-aware fine-tuning pipeline that measures how uncertain a model is about its response using the entropy of its own predicted answer and then trains on the resulting easy, medium, and hard splits with different tactics.

We confirm that entropy works as a difficulty estimation. Single-token answer entropy reaches ROC AUC values up to 0.8, clearly beating MASJ-based estimates of 0.57. This confirms that a model's own confidence is a reliable, automatic proxy for question difficulty. We publish collected data and code to support further research in the area of numerical complexity estimation.

Using the entropy-base data splits, we find that different complexity scores require different training approaches. Standard supervised fine-tuning (SFT) is enough for the easy and medium bands, but lagged on the hard band. For the hard questions, adding a distilled chain-of-thought from a large LLM unlocks further gains (accuracies of 0.5/0.6 vs. 0.4/0.46 for Qwen 3B/Phi-4-mini).

The pipeline is fully automated and can be included in other fine-tuning workflows. It suggests that curriculum ideas still matter for today's LLMs: letting the model focus on what it can already solve directly, while giving extra guidance only where it struggles, yields a better allocation of limited model capacity.

Limitations

- Proposed pipeline is tested only on MMLU-Pro and small models. Results may change for other question answering datasets, open-ended tasks, other domains, or larger LLMs.
- In low-resource settings teacher may be unavailable or imperfect, which reduces the benefit of learning from a distilled chain-of-thought. Additionally, we did not explore how well the approach generalizes to other reasoning-promoting techniques.
- Low entropy can still correspond to hallucinations, which leads to imperfect identification of the question complexity.
- We split data into 3 equal parts and did not explore other possible boundaries.
- We did not conduct an extensive ablation study which might reveal that our approach does not suggest the best possible combination or sequence of training within the current framework. It remains an area for further research.
- We did not run the pipeline for more epochs due to resource limitations, so its behavior for the longer runs stays unknown.

References

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *Preprint*, arXiv:1801.06146.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *Preprint*, arXiv:2305.02301.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Jisu Kim and Juhwan Lee. 2024. Strategic data ordering: Enhancing large language model performance through curriculum learning. *arXiv preprint arXiv:2405.07490*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, and 57 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *Preprint*, arXiv:2503.01743.

Mistral. 2024. Mistral-large-instruct-2411.

Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. 2024. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *Preprint*, arXiv:2408.13296.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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 qa benchmark. *Preprint*, arXiv:2311.12022.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement finetuning via adaptive curriculum learning. *Preprint*, arXiv:2504.05520.

Petr Sychev, Andrey Goncharov, Daniil Vyazhev, Edvard Khalafyan, and Alexey Zaytsev. 2025. When an llm is apprehensive about its answers – and when its uncertainty is justified. *Preprint*, arXiv:2503.01688.

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, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Chaoyi Wu, Pengcheng Qiu, Jinxin Liu, Hongfei Gu, Na Li, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2025. Towards evaluating and building versatile large language models for medicine. *npj Digital Medicine*, 8(1):58.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Yu Yang, Siddhartha Mishra, Jeffrey N Chiang, and Baharan Mirzasoleiman. 2024c. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. *Preprint*, arXiv:2403.07384.

Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *Preprint*, arXiv:2402.09391.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Qihao Zhu et al. Zhihong Shao, Peiyi Wang. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023b. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

A Prompts

A.1 Prompts used for complexity estimation

The used prompts are presented in Table 6.

Prompt for a single token response
The following are multiple choice questions about subject. Write down ONLY the NUMBER of the correct answer and nothing else.
Prompt for a single token response with a fallback for unknown answers
The following are multiple choice questions about subject. If you are certain about the answer return the correct option number, otherwise return 0. Write down ONLY the NUMBER and nothing else.
Prompt for a chain-of-thought response
The following are multiple choice questions about subject. Explain your thinking process step-by-step. At the end, write down the number of the correct answer by strictly following this format: [[number of correct answer]].
Prompt for a chain-of-thought response with a fallback for unknown answers
The following are multiple choice questions about subject. Explain your thinking process step-by-step. At the end, if you are certain about the answer write down the number of the correct answer by strictly following this format: [[number of correct answer]], otherwise return [[0]].

Table 6: Used prompts for complexity estimation

A.2 General prompts

You are an expert in the topic of the question. Please act as an impartial judge and evaluate the complexity of the multiple-choice question with options below. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must not answer the question. You must rate the question complexity by strictly following the criteria: [[Number of reasoning steps]] - how many reasoning steps do you need to answer this question? Valid answers: low, medium, high. Your answer must strictly follow this format: "[[Number of reasoning steps: answer]]". Example 1: "Your explanation... [[Number of reasoning steps: low]]". Example 2: "Your explanation... [[Number of reasoning steps: high]]". Example 3: "Your explanation... [[Number of reasoning steps: medium]]".

Table 7: Prompt for MASJ reasoning

You are an expert in the topic of the question. Please act as an impartial judge and evaluate the complexity of the multiple-choice question with options below. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must not answer the question. You must rate the question complexity by strictly following the scale: "high school and easier", "undergraduate", "graduate", "postgraduate". You must return the complexity by strictly following this format: "[[complexity]]", for example: "Your explanation... Complexity: [[undergraduate]]", which corresponds to the undergraduate level.

Table 8: Prompt for MASJ education levels

The following are multiple choice questions about subject. If you know the answer return the correct option number, otherwise return 0. Write down ONLY the NUMBER and nothing else.

Table 9: Prompt for a single token response with a fallback for unknown answers (alternative)

The following are multiple choice questions about subject. Explain your thinking process step-by-step. At the end, if you know the answer write down the number of the correct answer by strictly following this format: [[number of correct answers]], otherwise return [[0]].

Table 10: Prompt for a chain-of-thought response with a fallback for unknown answers (alternative)

B Aggregation Methods

B.1 Word-aggregation Methods

This COT aggregations have the same entropy values as in 3.2.3 for each COT token.

$$h_j = -\sum_{i=1}^{n} p_i \log p_i$$

where p_i - probability of a single token, n - vocabulary size, h_j - entropy of the corresponded token, h_{answer} - entropy of the answer token, and N is the token amount in LLM response.

$$COT_{mean} = \sum_{j=1}^{N} \frac{h_j}{N}$$

$$COT_{max} = \max_j h_j$$

So, Chain-of-Thought maximum and answer entropy difference is:

$$|COT_{max} - h_{answer}|$$

B.2 Sequence-aggregation Methods

For M logical claims, which were split by tokens that corresponded to the end of the sequence, we have tokens sets $C_1, C_2 \ldots C_M$.

$$Seq_{mean} = \frac{1}{M} \sum_{j=1}^{M} [\frac{1}{|C_j|} \sum_{i \in C_j} h_i]$$

$$Seq_{mean,max} = \frac{1}{M} \sum_{j=1}^{M} [\max_{i \in C_j} h_i]$$

$$Seq_{max,mean} = \max_j [\frac{1}{|C_j|} \sum_{i \in C_j} h_i]$$

Model	Education level	Reasoning
Qwen 3B	0.53	0.55
Qwen 3B*	0.53	0.55
Phi4-mini	0.52	0.55
Phi4-mini*	0.52	0.54
Phi4	0.50	0.57
Phi4*	0.50	0.55
Mistral 24B	0.50	0.56
Mistral 24B*	0.52	0.53

Table 11: ROC AUC for MASJ
* Alternative prompt to allow model answer "I do not know"

Statistics	Importance
Thinking total entropy	1.45
Thinking number of tokens	1.08
Answer total entropy	0.25
Answer length	0.20

Table 12: Absolute values of parameter weights

B.3 Probability-based Methods

Assume that for each token in response, we have the token probability distribution p_i. So, the marginal token difference is σ_i and mean marginal difference is mean of all σ_i in LLM response.

$$\sigma_i = p_i^1 - p_i^2$$

$$\overline{\sigma} = \frac{1}{N}\sum_{i=1}^{N}\sigma_i$$

Top-2 entropies difference in response δ.

$$\delta = \max_j h_j - \max_{i|i\neq j} h_i$$

B.4 Hybrid Method

In this section we provide linear combination of 2 best-perform previous methods: h_{answer} and COT_{max}. Also, we tried adding the third element COT_{mean}, but it has decreased the ROC-AUC, so we made a decision to remove it.

$$h_{mix} = (1-\alpha) * h_{answer} + \alpha * COT_{max}$$

where $0 \leq \alpha \leq 1$ is the hyperparameter (e.g. $\alpha = 0.05$ empirically for Qwen-3B).

C Additional experiments

C.1 MASJ education level and reasoning score

Table 11 shows ROC AUC values for MASJ evaluations of education levels and reasoning scores.

We can see that MASJ reasoning score has a slightly higher ROC AUC of 0.55 on average compared to education levels with ROC AUC of 0.52. There is no significant difference between prompts that allow IDK answers and the ones that do not.

The results indicate that MASJ scores divide the data into complexity groups with moderate quality. On the other hand, results depend on encoding of nominal scores provided by MASJ, and a more comprehensive study could improve this method.

Technical details. To calculate ROC AUC we encode MASJ results on a scale from 0 to 1 and prompt the model to answer questions directly, using prompts. For education levels, we take "High school and easier" - 0.2, "Undergraduate" - 0.4, "Graduate" - 0.6, "Postgraduate" - 0.8. For reasoning scores, "Low" - 0.25, "Medium" - 0.5, "High" - 0.75. IDK responses and results with invalid formatting are excluded from the calculations.

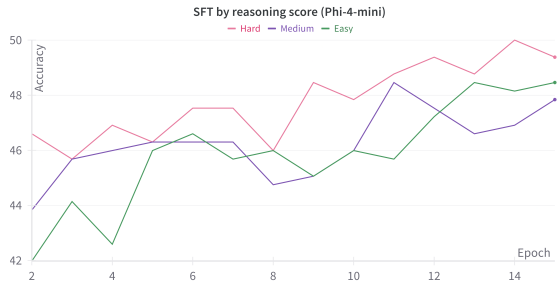
C.2 Feature importances

We evaluated logistic regression weights, that reflect feature importance. The results are in Table 12.

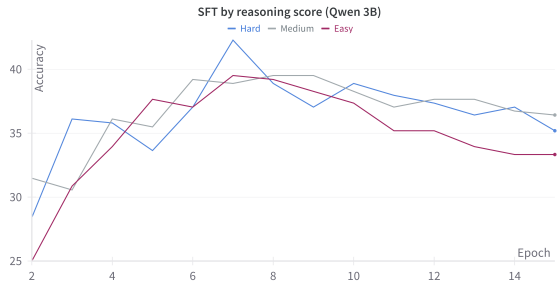
C.3 SFT using different scores

We evaluated the quality of different uncertainty estimates in a plain way. Now, we compare the usefulness of different scores at separating questions of different complexity with the following supervised fine-tuning.

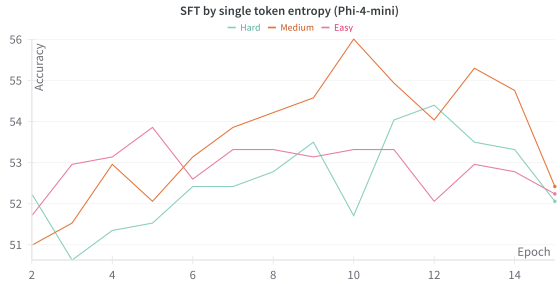
The results for MASJ reasoning scores, single token entropy, are presented in Figures 5. We see, that the model quality improves during SFT for all models and set of questions.



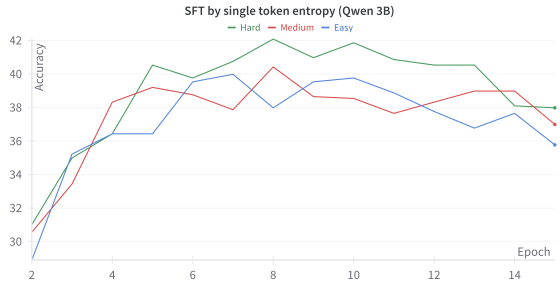
(a) MASJ reasoning score (Phi-4-mini)



(b) MASJ reasoning score (Qwen 3B)



(c) Single token entropy (Phi-4-mini)



(d) Single token entropy (Qwen 3B)

Figure 5: SFT quality dynamics during training with split by complexity estimates provided by the MASJ reasoning score and the single token entropy across Phi-4-mini and Qwen 3B models.