Long Paper

A Psychology-based Unified Dynamic Framework for Curriculum Learning

Guangyu Meng[1], Qinkai Zeng[2], John P. Lalor[1,3*], Hong Yu[4,5,6*]

[1] University of Notre Dame, Department of Computer Science and Engineering
gmeng@nd.edu

[2] Nankai University, College of Computer Science
qzengnkcs@gmail.com

[3] University of Notre Dame, Department of IT, Analytics, and Operations
john.lalor@nd.edu

[4] VA Bedford Healthcare System, Center for Health Optimization and Implementation Research

[5] University of Massachusetts Amherst, Manning College of Information and Computer Sciences

[6] University of Massachusetts Lowell, Miner School of Computer and Information Sciences
hong_yu@uml.edu

Directly learning from examples of varying difficulty levels is often challenging for both humans and machine learning models. A more effective strategy involves exposing learners to examples in a progressive order from easy to difficult. Curriculum Learning (CL) has been proposed to implement this strategy in machine learning model training. However, two key challenges persist in CL framework design: defining the difficulty of training data and determining the appropriate amount of data to input at each training step. Drawing inspiration from psychometrics, this paper presents a Psychology-based Unified Dynamic Framework for Curriculum Learning (PUDF). We quantify the difficulty of training data by applying Item Response Theory (IRT) to responses from Artificial Crowds (AC). This theory-driven IRT-AC approach leads to global (i.e., model-independent) and interpretable difficulty values. Leveraging IRT, we propose a training strategy, Dynamic Data Selection via Model Ability Estimation (DDS-MAE), to schedule the appropriate amount of data during model training. Since our difficulty labeling and model ability estimation are based on a consistent theory, namely IRT, their values are comparable within the same scope, potentially leading to aligned training data selection and faster convergence compared to the other CL methods. Experimental results demonstrate that fine-tuning pre-trained large language models with PUDF leads to higher accuracy and faster

* Corresponding authors

Action editor: Afra Alishahi. Submission received: 13 December 2024; revised version received: 6 November 2025; accepted for publication: 17 November 2025.

convergence on a suite of benchmark datasets compared to standard fine-tuning and state-of-the-art CL methods. Ablation studies and downstream analyses further validate the impact of PUDF for CL.

1. Introduction

Curriculum learning (CL) is a machine learning framework that trains models by gradually introducing examples of increasing difficulty (Bengio et al. 2009). CL can effectively improve the generalization capacity and convergence rate of various models in a wide range of scenarios, such as computer vision (Soviany et al. 2021; Zhang et al. 2021), natural language processing (NLP, Zhan et al. 2021; Zhao, Wang, and Huang 2021), robotics (Milano and Nolfi 2021; Manela and Biess 2022), and medical applications (Liu et al. 2022; Burduja and Ionescu 2021). In NLP in particular, CL has been shown to improve performance in applications such as machine translation (Zhan et al. 2021; Mohiuddin et al. 2022), sentiment analysis (Cirik, Hovy, and Morency 2016; Tsvetkov et al. 2016), and natural language understanding (Xu et al. 2020). A key benefit of CL is its ability to guide the training process towards optimal regions in the parameter space, thus reducing time spent on noisy and difficult samples in early training stages (Wang, Chen, and Zhu 2021). Recent work applying CL to pre-trained large language models (LLMs) has shown it to be effective for fine-tuning (Lee et al. 2022; Nagatsuka, Broni-Bediako, and Atsumi 2021; Platanios et al. 2019; Xu et al. 2020). While these CL methods show promise in improving accuracy, they also introduce increased complexity and longer training times, which can offset some of the benefits and hinder widespread adoption.

In this work, we propose a novel CL framework, Psychology-based Unified Dynamic Framework for Curriculum Learning (PUDF). With PUDF, we introduce novel approaches to two key CL components (Wang, Chen, and Zhu 2021): difficulty measurement (DM) and training scheduling (TS). Specifically, we propose Item Response Theory-based Artificial Crowds (IRT-AC) as DM and Dynamic Data Selection via Model Ability Estimation (DDS-MAE) as TS. Both components are based on Item Response Theory (IRT, Baker and Kim 2004; De Ayala 2013), a well-established methodology in psychometrics for test construction and subject evaluation. IRT assumes that a latent difficulty value for items (which we refer to as "examples") can be estimated from responses to the examples from a population of test-takers ("subjects"). Each subject is assumed to have a latent ability value corresponding to their proficiency on a task, as evaluated by performance on the items. We use a one-parameter IRT model, which assumes that all examples are equally discriminative and only vary in their difficulty. IRT-AC estimates latent difficulty parameters for each example from an artificial crowd of NLP models. When fine-tuning a new model, DDS-MAE uses the IRT-AC output to estimate a latent ability score for the model at each training epoch, then dynamically selects training data based on the model's current ability. This approach is similar to existing CL methods that use reinforcement learning, but is more efficient as it does not require a carefully designed reward function.

A key benefit of using IRT is that example difficulty is global and model-independent. Other methods for DM, such as loss, are model- or training epoch-dependent, which means that difficulty can vary between models and training runs; IRT estimated difficulty estimates are fixed a priori. As a result, using IRT-AC as the DM allows for estimating example difficulty offline for efficient use during training (TS). What's more, in a one-parameter IRT model, there is an interpretable relationship

between example difficulty and model ability. Specifically, an example's difficulty can be interpreted as the model ability value needed to have a 50% chance of labeling that example correctly. This provides a theoretically grounded and interpretable way to relate example difficulty to model ability.

Traditionally, fitting IRT models required extensive human-annotated data. However, recent work has shown that IRT models can be fit using machine-generated data instead of human-generated data (Lalor, Wu, and Yu 2019). Building on this, we propose the use of artificial crowds (AC) composed of multiple high-performing LLMs to obtain predicted results of the training data used to estimate an IRT model. Generating responses from multiple LLMs for the AC can be done offline; responses can be reused to reduce the computational cost. Furthermore, traditional IRT models do not scale well to large numbers of subjects and examples. Therefore, we leverage a variational inference (VI) method (Hoffman et al. 2013; Jordan et al. 1999) to fit a large-scale IRT model. VI estimates a variational distribution that approximates the true posterior. Learning involves minimizing the KL Divergence between the variational distribution and the true distribution via batched stochastic optimization, allowing for efficient estimation and scalability to larger datasets.

To test its effectiveness, we evaluate PUDF with a comprehensive suite of benchmarking datasets and existing CL methods. We find that PUDF improves training efficiency and predictive performance across our benchmarking models and datasets. For example, on the AG News dataset, consisting of almost 1 million training examples, fine-tuning Llama3.1-8B with PUDF leads to a 4.13% relative improvement in accuracy and a 69.68% relative improvement in training time compared to traditional fine-tuning. PUDF also improves over a robust suite of other curriculum learning methods. For example, fine-tuning Llama3.1-8B on AG News with PUDF leads to relative improvements of 0.42% and 75.48% in accuracy and training time, respectively, compared to a state-of-the-art (SOTA) reinforcement learning CL framework (Sengupta et al. 2023).

Our contributions are as follows: (1) We propose PUDF, an innovative approach to implementing an effective CL strategy for fine-tuning LLMs; (2) Compared to existing CL methods, our DM (IRT-AC) and TS (DDS-MAE) automatically define model-independent data difficulty and achieve dynamic data selection without significant time penalties; (3) Experimental results demonstrate PUDF's faster convergence and higher accuracy for fine-tuning LLMs, particularly with difficult and large-scale datasets, highlighting its scalability and efficiency. Overall, our results demonstrate that using PUDF leads to more efficient training and better performance. What's more, PUDF allows for scaling curriculum learning to much larger datasets, as demonstrated by our results on AG News, which contains almost 1 million training examples.

In prior work (Lalor and Yu 2020), we proposed Dynamic Data Selection for Curriculum Learning via Ability Estimation (DDaCLAE), a preliminary framework for CL using learned difficulty and ability parameters. We benchmarked DDaCLAE against a single curriculum learning baseline using BERT and LSTM models, and demonstrated the potential of learned difficulties over heuristics such as sentence length to validate our approach. To the best of our knowledge, this was the first work to learn model competence during training that is directly comparable to the difficulty of the examples. In this work, we significantly enhance the scope and depth of our preliminary results with DDaCLAE. We have generalized and modularized the previous model to facilitate its adaptation to emerging curriculum learning paradigms (Wang, Chen, and Zhu 2021) and its extension for future research. Specifically, in this manuscript we extend and enhance Lalor and Yu (2020) in the following ways:

1. **Enhanced Difficulty Assessment in IRT-AC:** For IRT-AC, we now employ 13 distinct pre-trained models to learn example difficulties (§5.1.2). This contrasts with the previous methodology, which utilized only a single model type (LSTM or BERT) to estimate difficulty under varying training data noise levels. IRT-AC thus offers a more robust and comprehensive evaluation, capable of encompassing a broader range of conditions.

2. **Integration of SOTA Language Models:** To evaluate PUDF, we have replaced BERT (Kenton and Toutanova 2019) and LSTM (Hochreiter and Schmidhuber 1997) with larger and more advanced architectures (§5.1.3). Specifically, here we conduct our experiments using DeBERTaV3 (He et al. 2020), GPT-2 (Radford et al. 2019), Llama3.1-8B (Grattafiori et al. 2024), and Qwen2.5-7B (Yang et al. 2024). This allows for a more rigorous validation of our approach using recent large language models.

3. **Expanded Benchmarking Comparisons:** We extend our benchmarking experiments from our previous work, which were limited to comparisons with heuristic-based CL methods. Here, we add several advanced CL benchmark methods (§5.1.4), including techniques based on reinforcement learning teachers, transfer teachers, and self-paced learning.

4. **Inclusion of New Diverse Datasets:** We have augmented our benchmarking suite with two new datasets (§5.1.1): AG News, a multi-class classification dataset comprising 14 classes and almost 1 million training examples (Zhang, Zhao, and LeCun 2015), and MedQA-UMLS, a challenging medical question-answering dataset (Jin et al. 2020). These additions facilitate a more thorough evaluation of PUDF's performance across diverse and complex tasks.

5. **Extension to Text Generation** We include an example of how PUDF can be used for generative tasks with experiments on the GSM8K dataset (§5.4), a text generation math problem dataset used for LLM benchmarking. Our results indicate that PUDF improves performance and runtime for GSM8K over the existing benchmark methods.

6. **Robust Downstream Analyses:** We have also added a robust downstream analysis of PUDF that extends beyond improvements to performance and training time to provide a more detailed examination of PUDF's characteristics and efficacy. Specifically, we have conducted the following new analyses: A *theoretical analysis* (§4.5) and a *per-component runtime analysis* (§5.3.2) of PUDF to reinforce and validate our claims regarding its efficiency; a systematic *ablation study* (§5.3.1) to isolate and quantify the contribution of each component to PUDF's overall performance; and a *convergence analysis* (§5.3.3) to highlight its advantages in convergence behavior, data efficiency, information utilization, and stability, and provide detailed insights into PUDF's faster and more efficient training dynamics compared to baseline models.

7. **Analysis of Learned Difficulties:** We have also added a robust analysis of the difficulty values learned from IRT-AC across three dimensions to further validate the approach: the distribution of data difficulty (§6.1), artificial crowd prediction accuracy across difficulty bins (§6.3), and crowd models' confidence scores relative to estimated difficulty (§6.4).

Overall, our enhanced benchmarking results with newer models and datasets validate and extend our initial results (Lalor and Yu 2020) and demonstrate the applicability of PUDF to recent LLMs and large-scale datasets. Our newly-added downstream analyses provide further insights into the mechanisms by which PUDF improves training performance and efficiency. Our dataset-related results concerning example difficulty can aid researchers in better understanding the intricacies of specific dataset examples. **Significance to the Research Community**: This manuscript situates PUDF within the ongoing curriculum learning discourse by aligning it with the existing understanding of CL frameworks (Wang, Chen, and Zhu 2021) and benchmarking it against SOTA methods. The additional analyses and broader benchmarking provide robust evidence of PUDF's utility, offering new insights into instance-level metrics like difficulty and confidence, which are becoming increasingly relevant in NLP research (Swayamdipta et al. 2020; Rodriguez et al. 2021; Cook, Lalor, and Abbasi 2025). By addressing scalability and interpretability, this work serves as a foundation for future innovations in dynamic curriculum learning for NLP. To facilitate future work, we make our code available[1] and also release the data collected from our artificial crowds.[2]

The rest of this paper is organized as follows. Section 2 reviews the related work in curriculum learning and Section 3 presents background information on Item Response Theory. In Section 4, we present PUDF and describe its key components. Section 5 describes our main experiments, results, and detailed analyses. In Section 6, we discuss our analyses of example difficulties learned from IRT-AC. Section 7 discusses the limitations of this work and Section 8 concludes.

2. Related Work

2.1 Curriculum Learning

CL methods implement model training strategies by progressively moving from easier to harder training data. The concept of training neural networks in a progressively easy-to-difficult manner can be traced back to the work of Elman (1993). Building on these foundations, CL was formally proposed by Bengio et al. (2009); there, the authors evaluated pre-designed CL methods on toy datasets with heuristic measures of difficulty. CL has since been studied in machine learning broadly (e.g., Soviany et al. 2021; Zhang et al. 2021; Milano and Nolfi 2021; Manela and Biess 2022; Liu et al. 2022; Burduja and Ionescu 2021) and NLP specifically (e.g., Zhan et al. 2021; Mohiuddin et al. 2022; Cirik, Hovy, and Morency 2016; Tsvetkov et al. 2016; Xu et al. 2020) and has been shown to improve learning across a variety of tasks and domains. There has also been a stream of research investigating the theory behind CL (Weinshall, Cohen, and Amir 2018; Hacohen and Weinshall 2019), particularly with regard to defining an ideal curriculum. CL theoretically leads to a steeper optimization landscape (i.e., faster learning) than standard training while keeping the same global minimum of the task. These theoretical results also highlight a key distinction between CL and similar guided training methods such as self-paced learning (Kumar, Packer, and Koller 2010), hard example mining (Shrivastava, Gupta, and Girshick 2016), and boosting (Freund and Schapire 1997): namely that CL considers difficulty with respect to the final hypothesis space (i.e., a model trained on the full dataset), while the other methods consider

1 https://github.com/nd-ball/cl-irt/
2 https://huggingface.co/datasets/nd-ball/response-patterns

CL Category	Difficulty Measurer	Training Scheduler	Complexity	References
Predefined	Predefined	Predefined	Low	Bengio et al. (2009) Platanios et al. (2019)
Self-paced learning	Automatic	Predefined	Low	Kumar, Packer, and Koller (2010) Ouyang et al. (2023)
Transfer teacher	Automatic	Predefined	High	Xu et al. (2020) Maharana and Bansal (2022)
RL teacher	Automatic	Automatic	High	Zhao et al. (2020) Kumar et al. (2019)
PUDF	Automatic	Automatic	Low	This work

Table 1: Summary of curriculum learning methods in the literature.

ranking examples according to how difficult the current model determines them to be (Weinshall, Cohen, and Amir 2018; Hacohen and Weinshall 2019). Our proposed PUDF bridges a gap between these methods by probing model ability at the current point in training and using this ability to identify appropriate training examples in terms of difficulty that is independent of a specific model or training epoch.

2.2 A General CL Framework

In a recent survey, Wang, Chen, and Zhu (2021) categorized CL methods in the literature based on two key components: a **difficulty measurer (DM)**, which provides a score indicating the relative easiness of each data example, and a **training scheduler (TS)**, which decides the sequence of data subsets to use throughout the training process. The general workflow for CL involves first ordering all training examples from easiest to hardest according to the DM. Subsequently, at each training epoch, the TS selects the appropriate subset of training data and presents it to the model for learning. CL methods can be categorized into two types based on how the DM and TS are implemented: Predefined and Automated (Wang, Chen, and Zhu 2021). In Predefined CL, both DM and TS are designed using prior human knowledge and without data-driven methods. In Automatic CL, one or both of the DM and TS are learned by data-driven models or algorithms. In their review, Wang, Chen, and Zhu (2021) identified three variations of Automatic CL: self-paced learning CL, transfer teacher CL, and reinforcement learning (RL) CL. We summarize each CL type in Table 1 and discuss the characteristics of each approach below.

Predefined CL relies on heuristics (e.g., sentence length or word rarity) for DM and a predetermined scheduling function (e.g., a linear function or root function) for TS based on task-specific data characteristics (Platanios et al. 2019; Spitkovsky, Alshawi, and Jurafsky 2010; Wei et al. 2016; Tsvetkov et al. 2016). Examples of difficult text for the DM include longer sentences, the presence of rare words (Platanios et al. 2019), the number of coordinating conjunctions (e.g., "and", "or", Kocmi and Bojar 2017), and the number of phrases (e.g., prepositional phrases, Tsvetkov et al. 2016). Common training schedulers include linear and root functions (Platanios et al. 2019), which increase the number of training samples at a linear or exponential pace. While simple and often effective, finding the optimal combination of DM and TS for specific tasks and datasets often requires expert domain knowledge. Moreover, examples that are easy for humans may not be easy for models due to different decision boundaries (Yuan et al. 2019).

Self-paced learning CL (SPL) allows the model itself to be the DM based on some model-dependent metric (Kumar, Packer, and Koller 2010; Jiang et al. 2015; Wan et al. 2020; Mohiuddin et al. 2022; Ouyang et al. 2023). For example, prior work has used model training loss (Kumar, Packer, and Koller 2010) and pseudo-label predictions (Ouyang et al. 2023) as inputs to the DM to measure learning difficulty. SPL is more automatic and more aligned with the model's learning process; however, early training may incur high uncertainty when the model is not yet sufficiently trained. Moreover, SPL still uses a predetermined function as TS; model competence is not typically considered. Instead, it is assumed that competence improves monotonically as more difficult examples are added.

Transfer teacher CL (Weinshall, Cohen, and Amir 2018; Xu et al. 2020; Maharana and Bansal 2022; Hacohen and Weinshall 2019) employs a pre-trained, "stronger" model as a teacher to be the DM according to its accuracy. For instance, prior work has used RoBERTa-large (Liu et al. 2019) as the teacher model; its output probabilities for training examples were used as difficulty estimates (Maharana and Bansal 2022). Similarly, a "cross-review" strategy was proposed where a teacher model with the same structure as the student model labels difficulty (Xu et al. 2020). TS in this transfer teacher-based CL approach leverages predefined functions, such as an annealing method (Xu et al. 2020) or an adaptive function (Maharana and Bansal 2022). However, such methods are costly due to the additional fine-tuning and still rely on a predefined TS.

RL teacher CL (Zhao et al. 2020; Kumar et al. 2019) methods adopt RL models as the teacher to perform TS according to the feedback from the model. Examples of RL teacher methods include a multi-armed bandit RL method (Graves et al. 2017), a Q-Learning strategy (Zhao et al. 2020), and a deterministic Actor-Critic RL model (Kumar et al. 2019). This dynamic approximates the learning process in human education, where the teacher and student improve together through interactions: the student makes progress based on the tailored learning materials selected by the teacher, while the teacher adjusts teaching strategy based on student performance. However, the RL model method is costly; we not only need to train the original model but also fine-tune the RL model based on a carefully-designed reward function during training.

3. Background: Item Response Theory

In this section, we first introduce IRT, in particular the one-parameter logistic (1PL) model (Rasch 1960; Baker and Kim 2004), and describe learning IRT models for machine-learning scale datasets with variational inference methods. As discussed in the introduction, evaluating the difficulty of data examples while considering a model's capabilities allows for an interpretable comparison between data difficulty and model ability. This aligns with an intuitive understanding of human learning, namely that a good student answering a question correctly does not necessarily imply that the question is easy. To achieve this mutual evaluation, we employ IRT methods, which learn latent parameters of dataset examples (called "items" in the IRT literature) and latent ability parameters of individual "subjects." We refer to "items" as "examples" and "subjects" as "models," respectively, for clarity and consistency with the curriculum learning literature.

For a model j and an example i, the probability that j labels i correctly ($z_{ij} = 1$) is a function of the latent parameters of j and i. The one-parameter logistic (1PL) model, or Rasch model, assumes that the probability of labeling an example correctly is a function of a single latent difficulty parameter of the example, b_i, and a latent ability parameter of the model, θ_j (Rasch 1960; Baker and Kim 2004):

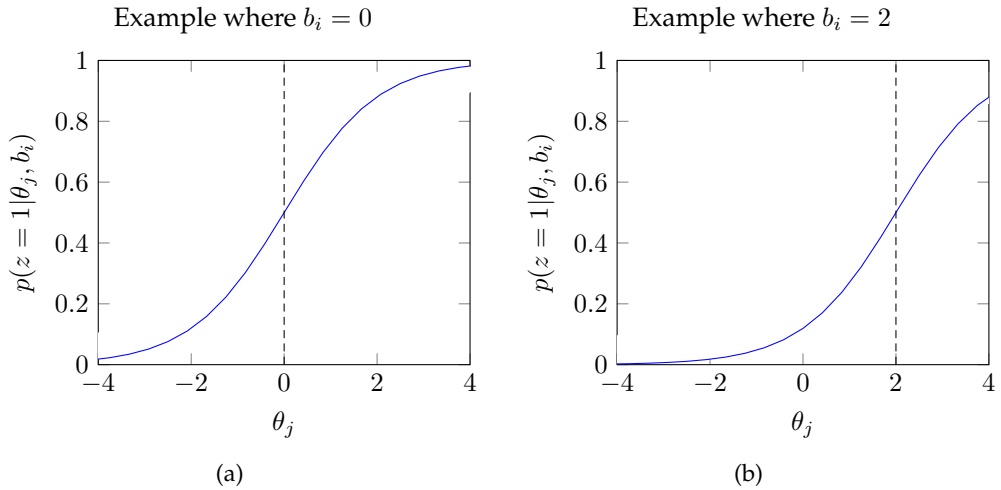
$$p(z_{ij} = 1|\theta_j, b_i) = \frac{1}{1 + e^{-(\theta_j - b_i)}} \tag{1}$$

When plotted, $p(z_{ij} = 1|\theta_j, b_i)$ is known as an item characteristic curve (ICC). The ICC is a visual representation of the example with regard to how a subject is expected to perform (Figure 1). With a 1PL model, there is an intuitive relationship between difficulty and ability. An example's difficulty value b_i can be thought of as the ability value for a model that has a 50% chance of labeling that example correctly. Put another way, model j has a 50% chance of labeling example i correctly when j's ability is equal to i's difficulty ($\theta_j = b_i$, see Figure 1).

Fitting an IRT model requires a set of I examples $\{i_0, i_1, \ldots, i_I\}$, a set of J models $\{j_0, j_1, \ldots, j_J\}$, and the binary graded responses of the models to each of the examples, $Z = \{\forall_{i \in I} \forall_{j \in J} : z_{ij}\}$. The log likelihood of a dataset of response patterns Z given the parameters Θ and B is:

$$\log \mathcal{L} = \sum_{j=1}^{J} \sum_{i=1}^{I} \log p(Z_{ij} = z_{ij}|\theta_j, b_i) \tag{2}$$

where $z_{ij} = 1$ if model j answers example i correctly and $z_{ij} = 0$ otherwise.



(a) (b)

Figure 1: Plot of $p(z_{ij} = 1|\theta_j, b_i)$ as a function of θ_j for two examples: (1a) an example with difficulty $b_i = 0$, and (1b) a more difficult example ($b_i = 2$). Models with ability $\theta_j > b_i$ (right of dashed line) have greater than 50% chance of labeling the example correctly.

For a given dataset of response patterns Z, item parameters are traditionally estimated using a marginal maximum likelihood expectation-maximization algorithm (Bock and Aitkin 1981), where the latent ability parameters (θ) are assumed to be random effects and are integrated out to define the marginal probability. Once item parameters are estimated, model ability is scored via maximum likelihood estimation.

However, traditional IRT model fitting does not scale to large datasets. Therefore, prior work proposed the use of variational inference (VI, Natesan et al. 2016; Jordan et al. 1999) to estimate latent IRT parameters. VI-IRT approximates the joint posterior distribution $p(\Theta, B|Z)$ by a variational distribution $q(\Theta, B)$:

$$q(\Theta, B) = \prod_{j=1}^{J} \pi_j^{\theta}(\theta_j) \prod_{i=1}^{I} \pi_i^{b}(b_i) \tag{3}$$

where $\pi_j^{\theta}()$ and $\pi_i^{b}()$ denote Gaussian densities for different parameters. Parameter means and variances are determined by minimizing the KL-Divergence between $q(\Theta, B)$ and $p(\Theta, B|Z)$:

$$\arg\min_{q} D_{\mathrm{KL}}(q(\Theta, B)||p(\Theta, B|Z)) \tag{4}$$

Optimization is typically performed via batched stochastic gradient descent, which scales to larger datasets and can leverage GPUs for faster training (Lalor and Rodriguez 2023). In selecting priors for VI-IRT, we follow the results of prior work and use hierarchical priors (Natesan et al. 2016; Lalor, Wu, and Yu 2019). The hierarchical model assumes that ability and difficulty means are sampled from a vague Gaussian prior (Equation 7), and ability and difficulty variances are sampled from an inverse Gamma distribution (Equation 8):

$$\theta_j \mid m_\theta, u_\theta \sim N(m_\theta, u_\theta^{-1}) \tag{5}$$

$$b_i \mid m_b, u_b \sim N(m_b, u_b^{-1}) \tag{6}$$

$$m_\theta, m_b \sim N(0, 10^6) \tag{7}$$

$$u_\theta, u_b \sim \Gamma(1, 1) \tag{8}$$

4. Methodology

In this section, we first present the workflow for PUDF in Section 4.1. We then discuss key insights and challenges for the DM and TS components of PUDF in Sections 4.2 and 4.3, respectively. We discuss model training and provide the pseudo-algorithm for DDS-MAE in Section 4.4 and perform theoretical time complexity analysis in Section 4.5. For clarity, notations and their descriptions are listed in Table 2.

4.1 PUDF Workflow

We first introduce the PUDF workflow, as illustrated in Figure 2. Consistent with Wang, Chen, and Zhu (2021), PUDF includes two steps: 1) IRT-AC for the DM and 2) DDS-MAE for the TS. To estimate difficulty, the training dataset is the input for the artificial crowd (AC). The AC consists of multiple models that generate predictions for each training data example. These predictions are evaluated against the true labels and converted to binary outcomes (0 or 1) for each model-example pair. The AC predictions are then used to estimate example difficulty with an IRT 1PL model, which generates difficulty scores

Notation	Description
J	Artificial crowds
M	Model
$\text{Val}(X, Y)$	Validation set
$\text{Train}(X, Y)$	Training set
b	Difficulty parameter of the training data
θ	Ability parameter of the model
e	Training epoch
Z	Response patterns of the models
$\mathcal{L}$	Likelihood
$q(\Theta, B)$	Variational distribution
π_j^θ	Gaussian density for ability parameters
π_i^b	Gaussian density for difficulty parameters
D_{KL}	KL-Divergence
m_θ, m_b	Means of ability and difficulty parameters
u_θ, u_b	Variances of ability and difficulty parameters

Table 2: The descriptions of the notations in our model.

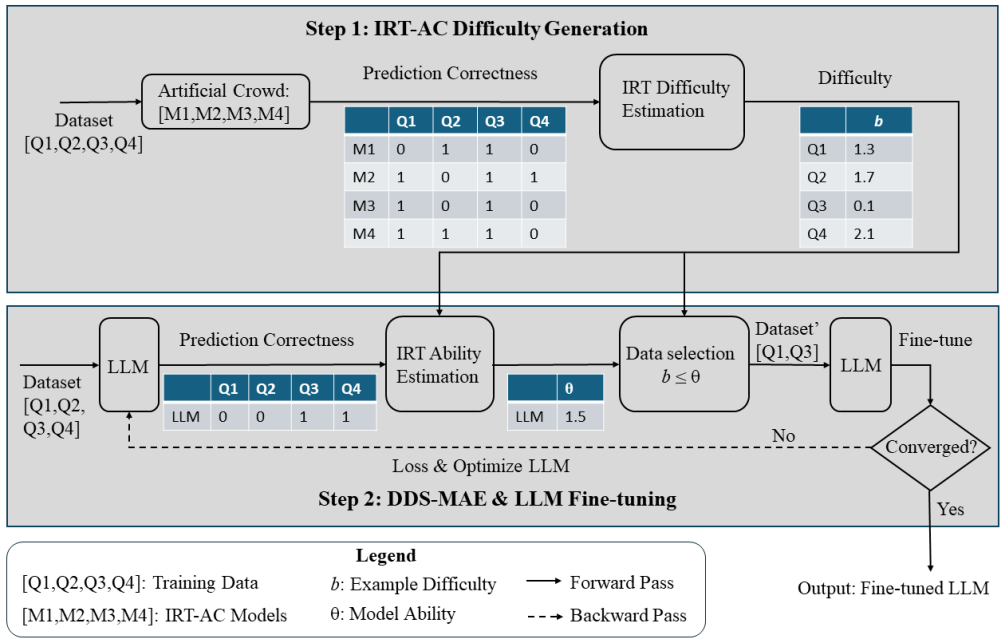


Figure 2: Workflow of PUDF. The process consists of two main steps: 1) IRT-AC for the DM, 2) DDS-MAE and LLM Fine-tuning for the TS.

for each training data example, where higher scores indicate greater difficulty. The TS evaluates the current LLM's ability based on LLM-generated predictions for the training dataset, which are converted to correct/incorrect responses. The LLM's responses are used in conjunction with the training data examples' difficulty from the DM to estimate

the LLM's ability. Then, a subset of the data where difficulty (b) is less than or equal to ability (θ), is selected: $b \leq \theta$. This workflow combines the strengths of IRT-AC for difficulty estimation, IRT for ability assessment, and dynamic data selection for efficient fine-tuning, resulting in a comprehensive approach to improving LLM performance via CL.

4.2 IRT-AC

A bottleneck of using IRT methods on machine learning datasets is the fact that each human subject would have to label all (or most) of the examples in the dataset in order to have enough response patterns to estimate the latent parameters. Gathering enough labels for each example to fit an IRT model would be prohibitively expensive for human annotators and would require significant effort to ensure annotation quality. Therefore, we use artificial crowds (Lalor, Wu, and Yu 2019) to generate our response patterns in our IRT-AC module. IRT-AC consists of two parts: training artificial crowd models to generate responses and using IRT to learn the difficulty of examples.

Our prior work (Lalor and Yu 2020) used a single neural network architecture (either LSTM or BERT) with training data modifications (training sub-sampling and label flipping) to construct an artificial crowd where performance across models in the crowd was varied (Figure A.8 in Appendix 1.1). Because of the single-model design, this yielded monotonous variations based only on the training data manipulations. Here, we leverage multiple LLMs as artificial crowd sources to improve performance. Specifically, our proposed AC incorporates a range of advanced pre-trained LLMs, including encoder-based, decoder-based, and encoder-decoder-based transformer architectures (Vaswani et al. 2017; Kenton and Toutanova 2019; Brown et al. 2020). This variety in artificial crowd models can increase the diversity of predicted results on our datasets while maintaining high confidence in their outputs (Bai et al. 2022).

This approach allows us to leverage the predictive performance of LLMs, potentially leading to more robust and diverse difficulty assessments for the IRT-AC method. To further enhance difficulty diversity and increase the credibility of the evaluated difficulty, we perform fine-tuning on the AC LLMs using the validation dataset for 1, 3, 5, and 10 epochs and include these fine-tuned models in the crowd. We then use these fine-tuned AC LLMs to predict labels for the training dataset, thereby obtaining the response patterns for difficulty estimation via IRT model fitting. Specifically, we fit the IRT model using variational inference (VI) (Natesan et al. 2016; Lalor, Wu, and Yu 2019; Wu et al. 2020) in order to account for the large scale of machine learning datasets. IRT-AC can incur a high cost, especially for large, complex models. However, that cost is a one-time cost, since the response patterns can be stored for future use. If new IRT difficulty estimates are needed, for example, when new models are added to the AC, then only those new models need to be fine-tuned. Once those response patterns are added, re-running IRT with VI is relatively low-cost compared to the fine-tuning of the AC models. In Section 5.3.2, we analyze the relative impact of each component on IRT difficulty estimation.

4.3 Dynamic Data Selection via Model Ability Estimation

For the TS component in PUDF, we propose Dynamic Data Selection via Model Ability Estimation (DDS-MAE). DDS-MAE trains the LLM with examples where difficulty is less than or equal to the model's ability. The estimated ability of the model at a given

epoch e, $\hat{\theta}_e$, is on the same scale as the difficulty parameters of the data. This establishes a principled approach for selecting data at any given training epoch, namely those examples where $b_i \leq \hat{\theta}_e$. This results in a sample of training data for which the model has at least a 50% probability of labeling the example correctly. All that is required is a single forward pass of the model on the training data to generate a response pattern (Equation 9). When example difficulties are known, model ability is estimated by maximizing the likelihood of the data given the response patterns and the example difficulties (Equation 10). Estimation is typically done via maximum-likelihood estimation using an existing estimation function (e.g., the Nelder-Mead solver, Lagarias et al. 1998):

$$Z_j = \forall_{y \in Y} \mathbf{I}[y_i = \hat{y}_i] \tag{9}$$

$$\hat{\theta}_e = \arg\max_{\theta_e} \prod_{i=1}^{I} p(z_{ij} = y_{ij} | b_i) \tag{10}$$

Subsequently, the selected data subset is used to fine-tune the LLM. This process is iterative: after each epoch, the LLM's convergence is checked. If fine-tuning has not converged, the process repeats to re-estimate the LLM's ability and select new data for further fine-tuning.

When implementing DDS-MAE, we encountered two significant challenges. First, the initial evaluated ability of the model is often low, and in some cases, the available training data is limited. This scenario results in insufficient data utilization for model training in the initial epoch, potentially causing the model's ability to stagnate and impeding further data selection. To mitigate this issue, we implemented an adaptive solution: if the model's ability fails to improve over two consecutive epochs, we incrementally increase the ability parameter by 0.1. This adjustment enables the training process to overcome initial saddle points and facilitates continued model improvement. Second, when utilizing the entire training dataset to evaluate the model's ability, the IRT calculation time becomes prohibitively long, particularly for large-scale datasets. For instance, the AG News dataset includes almost 1 million training examples (Zhang, Zhao, and LeCun 2015), and the ability estimation at each epoch consumes hundreds of minutes, comparable to the entire model training duration. To address this computational bottleneck, we propose a sampling-based method to evaluate the model's ability on a randomly selected subset of one thousand data points. This approach effectively balances estimation efficiency and accuracy, significantly reducing computational overhead while maintaining robust ability estimates. These refinements serve to enhance the scalability and efficiency of our framework, enabling its application across a diverse range of tasks and dataset sizes.

4.4 DDS-MAE Training Process

Algorithm 1 describes the training procedure in detail. Note that we assume that example difficulties have been learned offline using IRT-AC (see §4.2). Each example in the training set has an estimated difficulty parameter (b_i). The first step of DDS-MAE is to estimate the ability of the model using the estimation function (§4.3, Alg. 1 line 9). To do this, we use part of the training set, but crucially, only to get response data, not to update parameters (i.e., no backward pass). We do not use a held-out validation set for estimating ability because we do not want the validation set to influence training.

Algorithm 1 Training process with DDS-MAE

Input: Data (X, Y), model M, difficulties B, num_epochs **Output:** Learned model M'

1: **procedure** ABILITY_EST($Y, \hat{Y}, B$)
2: $Z = \forall_{y \in Y} \mathbf{I}[y_i = \hat{y}_i]$
3: $\hat{\theta}_e = \arg\max_\theta p(Z|\theta, b)$
4: return $\hat{\theta}_e$
5: **end procedure**
6: $M' = M$
7: **for** e in num_epochs **do**
8: $\hat{Y} = M'(X)$
9: $\hat{\theta}_e = \texttt{ABILITY\_EST}(Y, \hat{Y}, D)$
10: $X_e, Y_e = \{(x, y) : b_i \leq \hat{\theta}_e\}$
11: $M' = \texttt{train}(M', X_e, Y_e)$
12: **end for**
13: return M'

In our experiments, the validation set is only used for early stopping. Model outputs are obtained for the training set, and graded as correct or incorrect as compared to the gold standard label (Alg. 1 line 2). This response pattern is then used to estimate model ability at the current epoch ($\hat{\theta}_e$, Alg. 1 line 3). Once ability is estimated, data selection is done by comparing estimated ability to the examples' difficulty parameters. If the difficulty of an example is less than or equal to the estimated ability, then the example is included in training for this epoch (Alg. 1 line 10). The model is then trained with the training data subset (Alg. 1 line 11).

In contrast to other TS methods in the literature, the training data size does not have to be monotonically increasing with DDS-MAE. PUDF adds or removes training data based not on a fixed step schedule but rather by probing the model at each epoch and using the estimated ability to match data to the model. This way, if a model has a high estimated ability early in training, then more data can be added to the training set more quickly, and learning is not artificially slowed down due to the curriculum schedule. If a model's performance suffers when adding data too quickly, then this will be reflected in lower ability estimates, which leads to less data selected in the next epoch.

4.5 Time Complexity Analysis

In the DM (IRT-AC), we have two components: training the models in the AC to generate response patterns and estimating difficulty via VI-IRT. Assuming transformer-based architecture models in the AC, the time complexity to fine-tune the models is $O(K_{\text{FT}} M N_{\text{val}} / BL(n^2 d + nd^2))$, where K_{FT} is the number of optimization iterations, B is the batch size, L is the number of layers, n is the sequence length, d is the data dimension, M is the number of models in the artificial crowd, and N_{val} is the size of the validation set (Khan et al. 2022; Tay et al. 2022). For VI-IRT, the time complexity is $O(K_{\text{VI}} N_{\text{train}})$. We note here that the complexity associated with fine-tuning IRT-AC models is an offline cost that, once run, generates response patterns and difficulty estimates that can be reused. In particular, estimating IRT models with VI has been shown to reduce runtime, in particular when leveraging GPUs (Lalor and Rodriguez 2023). Once the difficulty parameters of the training data are estimated, they can be used for multiple training runs. If new models are added to the AC, then only those

new models must be fine-tuned, and VI-IRT is then rerun on the entire response pattern pool. We empirically assess this cost in Section 5.3.2.

Our DDS-MAE approach introduces two additional steps to traditional fine-tuning: model ability estimation and training data filtering (Alg. 1, lines 9 and 10). The model ability estimation procedure consists of (i) comparing each prediction with its corresponding true label, with a time complexity of $O(N_\theta)$, where N_θ is the number of estimated training examples; and (ii) Maximum Likelihood Estimation (MLE) using the Nelder-Mead method, as described in Section 4.3, with a time complexity of $O(K_\theta N_\theta)$, where K_θ is the number of optimization iterations (typically, $K_\theta = 10$). The training data filtering step also exhibits linear complexity, $O(N_{\text{train}})$. Consequently, the total time complexity for DDS-MAE is $O(N_\theta) + O(K_\theta N_\theta) + O(N_{\text{train}})$. For conventional transformer-based model training in each epoch, the time complexity is $O(K_{\text{FT}} N_{\text{train}}/BL(n^2 d + nd^2))$, where K_{FT} is the number of optimization iterations, N_{train} is the number of training data, B is the batch size, L is the number of layers, n is the sequence length, and d is the data dimension (Khan et al. 2022; Tay et al. 2022). Theoretically, our proposed method is asymptotically equivalent to the conventional training process; we experimentally verify that the additional time required is low and is typically offset by overall faster convergence (§5.3.2).

5. Experiments

In this section, we first introduce the experimental setup in Section 5.1. Then, we validate PUDF's performance and compatibility across different LLM models and compare PUDF with other advanced CL methods in Sections 5.2. We conduct multiple analyses in Section 5.3 to demonstrate the contribution of each component to PUDF's overall performance. In Section 5.4, we present results applying PUDF to a math problem text generation task to demonstrate its use beyond traditional classification tasks.

5.1 Experimental Setup

5.1.1 Datasets. We conduct our experiments with eight datasets. MedQA-UMLS (Jin et al. 2020) is a recent multiple-choice QA dataset consisting of medical exam questions. The AG News dataset (Zhang, Zhao, and LeCun 2015) includes almost 1 million training examples across 14 classes. Lastly, we evaluate PUDF on natural language understanding tasks from the GLUE (Wang et al. 2019) benchmark for consistency with and comparability to prior CL research (Sengupta et al. 2023; Maharana and Bansal 2022; Wan et al. 2020; Xu et al. 2020; Lalor and Yu 2020). We specifically consider the six GLUE classification tasks[3] which cover natural language inference (MNLI, RTE, QNLI), duplicate detection (MRPC, QQP), and sentiment analysis (SST-2).[4] Dataset details and summary statistics are provided in Table 3.

5.1.2 IRT-AC Models. For the artificial crowd models, we include BERT (Kenton and Toutanova 2019), DistillBERT (Sanh et al. 2019), RoBERTa (Liu et al. 2019), DeBERTa (He

3 We exclude the WNLI dataset due to dataset construction inconsistencies; see https://gluebenchmark.com/faq note 12.

4 Because test set labels for our tasks are only available via the GLUE evaluation server, we use the held-out validation sets to measure performance, consistent with prior work. For training, we use a 90% - 10% split of the training data and use the 10% split as our held-out validation set for early stopping. We can then use the full validation set as our test set to evaluate performance across experiments without making multiple submissions to the GLUE server.

Dataset	Train	Validation	Test	Labels	Reference
MedQA	9.2k	1.02k	1.27k	4	Jin et al. (2020)
AG News	995.3k	124.4k	124.4k	14	Zhang, Zhao, and LeCun (2015)
MNLI	353k	39k	9.8k	3	Williams, Nangia, and Bowman (2018)
MRPC	3.3k	366	409	2	Dolan and Brockett (2005)
QNLI	94k	10k	5.5k	2	Wang et al. (2019)
QQP	327k	36k	40k	2	Iyer, Dandekar, and Csernai (2017)
RTE	2.2k	249	278	2	Bentivogli et al. (2009)
SST-2	61k	6.7k	873	2	Socher et al. (2013)

Table 3: Statistics of the datasets used in the experiments, including training, validation, and test set sizes, number of labels, and original references.

et al. 2020), ALBERT (Lan et al. 2019), XLNet (Yang et al. 2019), ELECTRA (Clark et al. 2020), T5 (Raffel et al. 2020), BART (Lewis et al. 2019), Llama3.1-8B (Grattafiori et al. 2024), Qwen2.5-7B (Yang et al. 2024), and GPT-2 (Radford et al. 2019). We collect response patterns from each AC model with fine-tuning for 0, 1, 3, 5, and 10 epochs for a total of 60 AC models.

5.1.3 Benchmarking Models. We test the effectiveness and compatibility of PUDF by integrating it with different types of transformer architectures, including encoder-based and decoder-based models. We include models of varying parameter size and complexity to demonstrate PUDF's effectiveness across a variety of LLMs.

DeBERTaV3 (86M parameters, He, Gao, and Chen 2022) is an encoder-based pre-trained language model developed by Microsoft. It uses a disentangled attention mechanism to better capture word dependencies and contextual information, enhancing performance on various natural language understanding tasks.

GPT-2 (124M parameters, Radford et al. 2019) is a decoder-based language model from OpenAI. Trained on diverse internet text, it excels in generating coherent, contextually relevant text for tasks like translation, summarization, and question-answering without task-specific training data.

Llama3.1-8B (8B parameters, Grattafiori et al. 2024) is a decoder-based pre-trained language model developed by Meta. It is part of the Llama 3.1 collection of multilingual models, designed for both commercial and research use, and has been optimized for dialogue use cases with improved safety and helpfulness through supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF). Trained on over 15 trillion tokens of publicly available data, Llama3.1-8B supports a 128K context length and demonstrates strong performance on various benchmarks including as text generation, coding, and multilingual conversation.

Qwen2.5-7B (7.61B parameters, Yang et al. 2024) is a decoder-based language model from Alibaba Cloud as part of their Qwen2.5 series. It features a transformer architecture with enhancements including RoPE, SwiGLU, RMSNorm, and Attention QKV bias, and supports a context length of up to 131,072 tokens. Qwen2.5-7B shows significant improvements over previous Qwen models in knowledge-intensive tasks, coding, mathematics, instruction following, long-text generation, and multilingual capabilities, supporting over 29 languages.

5.1.4 Benchmark CL Methods. To benchmark the performance of our proposed PUDF framework, we compare several CL methods that cover each of the four CL categories, i.e., predefined CL, self-paced learning, transfer teacher, and RL teacher (Wang, Chen, and Zhu 2021).

Predefined CL. We evaluate predefined CL based on prior work that defines a predefined competence value (Platanios et al. 2019). For DM, we use sentence length (d_{SL}) and word rarity (d_{WR}). For TS, we use a linear or root function to adjust the training pace. We set the initial competence (c_0) to be 0.01 and set the point where the model is fully competent (T) to be equal to total_epochs$/2$; the predefined CL reaches competence halfway through training and trains with the full training set for the second half.

Self-paced learning (SPL). We evaluate a novel self-paced learning algorithm (Zhang et al. 2024) that incorporates belief functions to overcome limitations of traditional SPL approaches. Unlike traditional SPL methods, it mitigates the tendency to misjudge sample difficulty based solely on learning loss in early training stages, which often leads to the premature inclusion of hard-to-classify examples. The DM combines evidential uncertainty and learning loss to characterize difficulty. The TS adjusts the balance between evidential uncertainty and learning loss across training stages, allowing for more appropriate sample selection as the model improves.

Transfer teacher. We evaluate a transfer teacher-based CL from recent work (Maharana and Bansal 2022) that uses Question Answering Probability (QAP) as a DM scoring function and an adaptive function for TS. This method uses a pre-trained teacher model fine-tuned on the training data, and the QAP metric from the teacher's outputs serves as the DM to rank training examples by difficulty.

RL Teacher. For RL teacher, we evaluate MPDistil (Sengupta et al. 2023). MPDistil is a meta-policy knowledge distillation framework with a reward-based policy learner as DM and a meta-reinforcement learning-based model and reward function as TS.

5.1.5 Hyperparameter Tuning and Hardware Platform. For all datasets (MedQA, AG News, and GLUE) the maximum input token length was set to the 95th percentile of token lengths within each respective dataset to balance representational capacity and computational load. The largest batch size that fit within the available GPU memory was employed to optimize throughput. Fine-tuning was run for 20 training epochs with early stopping based on validation set performance to mitigate overfitting and reduce training duration (Yao, Rosasco, and Caponnetto 2007). In order to validate our experimental results and report on the stability of each method, we ran the experiments five times and reported the mean and standard deviation of the results.

Hyperparameters for all the models were established via grid search. These optimized parameters were then consistently applied across all CL methodologies evaluated in this study. For the optimization method, we employed the AdamW optimizer (Loshchilov and Hutter 2017). The learning rate was selected from the set $\{1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}\}$ based on preliminary experiments. A weight decay of 0.01 was applied, excluding layer normalization and bias terms, while standard default values were maintained for AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$). The learning rate schedule incorporated a linear warm-up phase, typically accounting for 6-10% of the total training steps, followed by a linear decay of the learning rate. Our batching strategy for these models involved adjusting the batch size (e.g., ranging from 2 to 64

depending on the specific model and dataset) and utilizing gradient accumulation (e.g., 1 to 16 steps) to achieve a larger effective batch size while adhering to GPU memory limitations.

Fine-tuning Llama3.1-8B and Qwen2.5-7B on the AG News dataset, which comprises almost one million training data points, presented significant GPU memory challenges. To address these constraints, we employed parameter-efficient fine-tuning (PEFT), specifically QLoRA (Dettmers et al. 2023). QLoRA is an enhanced version of low-rank adaptation (LoRA, Hu et al. 2022) that incorporates aggressive quantization. Specifically, our implementation involved 4-bit NormalFloat (NF4) quantization for the model parameters, a `bfloat16` compute data type within the quantization layers, and double quantization. QLoRA was further complemented by mixed-precision training via PyTorch AMP, which utilized bfloat16 numerical precision and a `GradScaler`. This combination of PEFT and mixed-precision training substantially decreased memory requirements and often accelerated training throughput.

To fit our IRT model, we use the py-irt Python package (Lalor and Rodriguez 2023), which is built on top of the Pyro probabilistic programming language (Bingham et al. 2018). All LLMs were implemented from Huggingface.[5] One NVIDIA H100 GPU was used to conduct all of the experiments.

5.2 Incorporating PUDF in LLM Fine-tuning

In this section, we report our main results comparing PUDF to a no-CL baseline as well as other CL frameworks. We report predictive performance via accuracy (Table 4) as well as runtime (Figure 3) results for MedQA, AG News, and GLUE. For GLUE, we averaged results across tasks; results for individual GLUE tasks are presented in the appendices (Table A.11 and Figure A.9) for space considerations. We report the mean and standard deviation of five runs of each configuration. In all cases, we conduct a one-tailed Welch's t-test to determine whether the performance of PUDF is significantly better than the benchmark methods. We used Benjamini-Hochberg correction ($\alpha < 0.05$) to control the false discovery rate across multiple comparisons (Benjamini and Hochberg 1995; Ormerod, del Rincón, and Devereux 2024).

Table 4 presents a comprehensive comparison of various CL methods across benchmark datasets. PUDF consistently outperforms traditional training (no CL) as well as the comparison CL methods in accuracy across datasets. For example, training Llama3.1-8B with PUDF results in relative accuracy improvements over the no-CL baseline of 4.15%, 6.65%, and 0.74% for AG News, MedQA, and GLUE, respectively. Relative improvements over the best performing benchmark CL method are 0.42%, 0.38%, and 0.74%, respectively. In particular, when considering the larger LLM benchmark models (Llama3.1-8B and Qwen2.5-7B), PUDF accuracy is significantly higher than all benchmark methods with one exception: the RL benchmark with Qwen2.5-7B for AG News. In this case, our runtime results (Figure 3) show that PUDF is significantly faster compared to all benchmarks for these two models, including the RL method for Qwen2.5-7B on AGNews. For GPT-2 and DeBERTa, PUDF improvements are significant in all cases except for two methods on AG News (d_WR-R and RL for GPT-2) and two methods on GLUE (TT and RL for DeBERTaV3). Runtime improvements for these two models are significant in most cases (Figure 3).

5 https://github.com/huggingface/transformers

Table 4: Accuracy results comparing PUDF with other CL Methods. Results are averaged over 5 runs with standard deviations as subscripts. The best performing method for each model is in **bold**; the second-best model is underlined. For GLUE, we report the mean scores across tasks, pooled by runs.

Model	Method	MedQA	AGNews	GLUE
DeBERTaV3	Baseline	$32.69^{*}_{\pm 0.10}$	$70.66^{*}_{\pm 0.35}$	$89.65^{*}_{\pm 0.40}$
	d_SL-L	$26.49^{*}_{\pm 1.43}$	$65.62^{*}_{\pm 1.99}$	$88.13^{*}_{\pm 0.35}$
	d_SL-R	$27.25^{*}_{\pm 1.53}$	$65.55^{*}_{\pm 2.47}$	$88.58^{*}_{\pm 0.66}$
	d_WR-L	$27.04^{*}_{\pm 1.80}$	$64.02^{*}_{\pm 2.94}$	$87.18^{*}_{\pm 0.72}$
	d_WR-R	$27.95^{*}_{\pm 1.41}$	$68.04^{*}_{\pm 1.87}$	$88.97^{*}_{\pm 0.38}$
	SPL	$33.34^{*}_{\pm 0.33}$	$70.54^{*}_{\pm 0.90}$	$89.12^{*}_{\pm 0.86}$
	TT	$32.53^{*}_{\pm 0.52}$	$70.93^{*}_{\pm 1.30}$	$89.44_{\pm 1.35}$
	RL	$\underline{33.35}^{*}_{\pm 0.39}$	$\underline{71.87}^{*}_{\pm 0.56}$	$\underline{89.99}_{\pm 0.69}$
	PUDF	$\mathbf{34.12}_{\pm 0.16}$	$\mathbf{72.43}_{\pm 0.27}$	$\mathbf{90.76}_{\pm 0.20}$
GPT-2	Baseline	$27.97^{*}_{\pm 0.15}$	$65.71^{*}_{\pm 1.13}$	$\underline{82.34}^{*}_{\pm 0.30}$
	d_SL-L	$26.09^{*}_{\pm 1.54}$	$64.69^{*}_{\pm 1.34}$	$79.13^{*}_{\pm 0.73}$
	d_SL-R	$25.01^{*}_{\pm 1.99}$	$63.15^{*}_{\pm 2.08}$	$80.33^{*}_{\pm 0.57}$
	d_WR-L	$26.09^{*}_{\pm 1.36}$	$67.02^{*}_{\pm 0.17}$	$79.44^{*}_{\pm 0.67}$
	d_WR-R	$26.45^{*}_{\pm 1.22}$	$\mathbf{67.77}_{\pm 0.23}$	$80.54^{*}_{\pm 0.59}$
	SPL	$27.43^{*}_{\pm 0.86}$	$66.01^{*}_{\pm 1.18}$	$80.24^{*}_{\pm 0.43}$
	TT	$28.67^{*}_{\pm 0.50}$	$67.05^{*}_{\pm 0.53}$	$81.80^{*}_{\pm 0.61}$
	RL	$\underline{28.91}^{*}_{\pm 0.26}$	$\underline{67.35}_{\pm 0.35}$	$82.21^{*}_{\pm 0.55}$
	PUDF	$\mathbf{29.50}_{\pm 0.22}$	$\mathbf{67.77}_{\pm 0.57}$	$\mathbf{83.03}_{\pm 0.30}$
Llama3.1-8B	Baseline	$55.15^{*}_{\pm 0.09}$	$72.10^{*}_{\pm 0.15}$	$\underline{90.59}^{*}_{\pm 0.42}$
	d_SL-L	$45.89^{*}_{\pm 4.22}$	$70.73^{*}_{\pm 2.06}$	$87.78^{*}_{\pm 1.45}$
	d_SL-R	$50.22^{*}_{\pm 3.68}$	$71.20^{*}_{\pm 1.86}$	$88.70^{*}_{\pm 0.87}$
	d_WR-L	$57.09^{*}_{\pm 1.07}$	$71.80^{*}_{\pm 1.85}$	$89.27^{*}_{\pm 0.63}$
	d_WR-R	$55.98^{*}_{\pm 1.49}$	$71.16^{*}_{\pm 1.84}$	$89.10^{*}_{\pm 0.76}$
	SPL	$\underline{58.60}^{*}_{\pm 0.04}$	$73.87^{*}_{\pm 0.23}$	$90.36^{*}_{\pm 0.36}$
	TT	$58.01^{*}_{\pm 0.03}$	$72.67^{*}_{\pm 1.06}$	$90.08^{*}_{\pm 0.50}$
	RL	$57.90^{*}_{\pm 0.24}$	$\underline{74.77}^{*}_{\pm 0.21}$	$90.50^{*}_{\pm 0.38}$
	PUDF	$\mathbf{58.82}_{\pm 0.09}$	$\mathbf{75.08}_{\pm 0.11}$	$\mathbf{91.26}_{\pm 0.35}$
Qwen2.5-7B	Baseline	$63.52^{*}_{\pm 0.16}$	$71.63^{*}_{\pm 0.20}$	$89.82^{*}_{\pm 0.56}$
	d_SL-L	$62.02^{*}_{\pm 1.18}$	$69.97^{*}_{\pm 2.88}$	$\underline{90.15}^{*}_{\pm 0.51}$
	d_SL-R	$62.03^{*}_{\pm 0.97}$	$69.43^{*}_{\pm 2.82}$	$89.43^{*}_{\pm 0.63}$
	d_WR-L	$63.47^{*}_{\pm 0.79}$	$67.44^{*}_{\pm 3.15}$	$89.59^{*}_{\pm 0.55}$
	d_WR-R	$63.31^{*}_{\pm 1.10}$	$69.89^{*}_{\pm 1.20}$	$89.77^{*}_{\pm 0.50}$
	SPL	$64.17^{*}_{\pm 0.37}$	$71.55^{*}_{\pm 0.46}$	$88.96^{*}_{\pm 0.78}$
	TT	$64.13^{*}_{\pm 0.28}$	$\underline{71.71}^{*}_{\pm 0.26}$	$89.35^{*}_{\pm 0.53}$
	RL	$\underline{64.52}^{*}_{\pm 0.12}$	$\mathbf{72.93}_{\pm 0.07}$	$89.94^{*}_{\pm 0.60}$
	PUDF	$\mathbf{65.02}_{\pm 0.22}$	$\mathbf{72.93}_{\pm 0.32}$	$\mathbf{90.90}_{\pm 0.43}$

*Indicates that the value is significantly lower than the **best accuracy** in the column (Welch's single-tailed t-test with Benjamini-Hochberg correction, $\alpha < 0.05$).

The performance advantages of PUDF correlate with dataset and model characteristics. Our analysis reveals two distinct poles of improvement. For the large-scale AG News dataset, with nearly one million examples, PUDF's primary benefit is a dramatic 69.68% relative runtime reduction for Llama3.1-8B over the baseline. This efficiency stems from the DDS-MAE scheduler, which dynamically selects an optimal data subset, avoiding the high cost of training on the full dataset in early epochs. Conversely, for the MedQA dataset, a task identified as highly challenging by our IRT-AC analysis, PUDF shows its most substantial accuracy gains (a 6.65% relative improvement for Llama3.1-8B). This suggests that in complex domains, the robust, global IRT-AC difficulty metric is more effective than simple heuristics or volatile early-stage metrics, guiding the model to a superior convergence.

Furthermore, the benefits of PUDF scale with model size, with Llama3.1-8B and Qwen2.5-7B deriving the most significant advantages. These large models are costly to fine-tune, and alternative CL frameworks (e.g., RL and Transfer Teachers) introduce substantial online computational overhead. PUDF's architecture relegates its main expense (IRT-AC) to a one-time, offline process. The online DDS-MAE component adds only minimal overhead, a single forward pass and lightweight estimation per epoch. This combination of a robust, pre-computed difficulty estimation and an efficient dynamic scheduler allows large models to converge faster to a better performance optimum. While smaller models such as DeBERTaV3 and GPT-2 also benefit, their lower intrinsic cost and capacity result in positive but less pronounced gains.

For training time, again looking at Llama3.1-8B, PUDF results in relative improvements of 69.68%, 37.21%, and 55.97% over the no-CL baseline for AG News, MedQA, and GLUE, respectively. Relative improvements over the fastest CL alternative, which is usually not the highest accuracy option, are 28.40%, 13.12%, and 20.28%, respectively. Comparing the training time between PUDF and the RL benchmark for Llama3.1-8B on AG News, the relative training time improvement is 75.48%.

Across datasets and benchmark methods, PUDF is significantly faster when using DeBERTaV3, Qwen2.5-7B, and Llama3.1-8B as the fine-tuning model. For GPT-2, there are three cases where runtime performance is comparable, but in these cases predictive performance of the benchmark methods lags behind PUDF (AGNews, d_SL-L, MedQA, d_SL-L and d_SL-R).

Notably, the best CL benchmark for training time is often less performant in terms of accuracy. The benchmarking CL methods consistently underperform either on predictive performance or training time. Our results highlight PUDF's ability to maintain competitive accuracy while significantly reducing training time compared to other CL approaches.

While results from our prior work were mixed in terms of performance and efficiency (Lalor and Yu 2020), these results indicate consistent improvements with PUDF and also provide new insights. Specifically, PUDF outperforms both traditional training and benchmark CL methods on large (AG News) and difficult (MedQA) datasets, whereas prior work focused on the smaller, relatively easier GLUE datasets. In addition, PUDF outperforms advanced, automated CL methods as well as predefined CL methods. Lastly, PUDF improvements are consistent across smaller (DeBERTaV3, GPT-2) and larger (Llama3.1-8B, Qwen2.5-7B) LLMs, which demonstrates the consistency of the method above and beyond the smaller models (LSTM, BERT) evaluated in prior work. The consistent improvements and low standard deviations for PUDF provide strong evidence that PUDF can outperform existing CL techniques in terms of both accuracy and training efficiency across benchmark datasets. The observed improvements, espe-

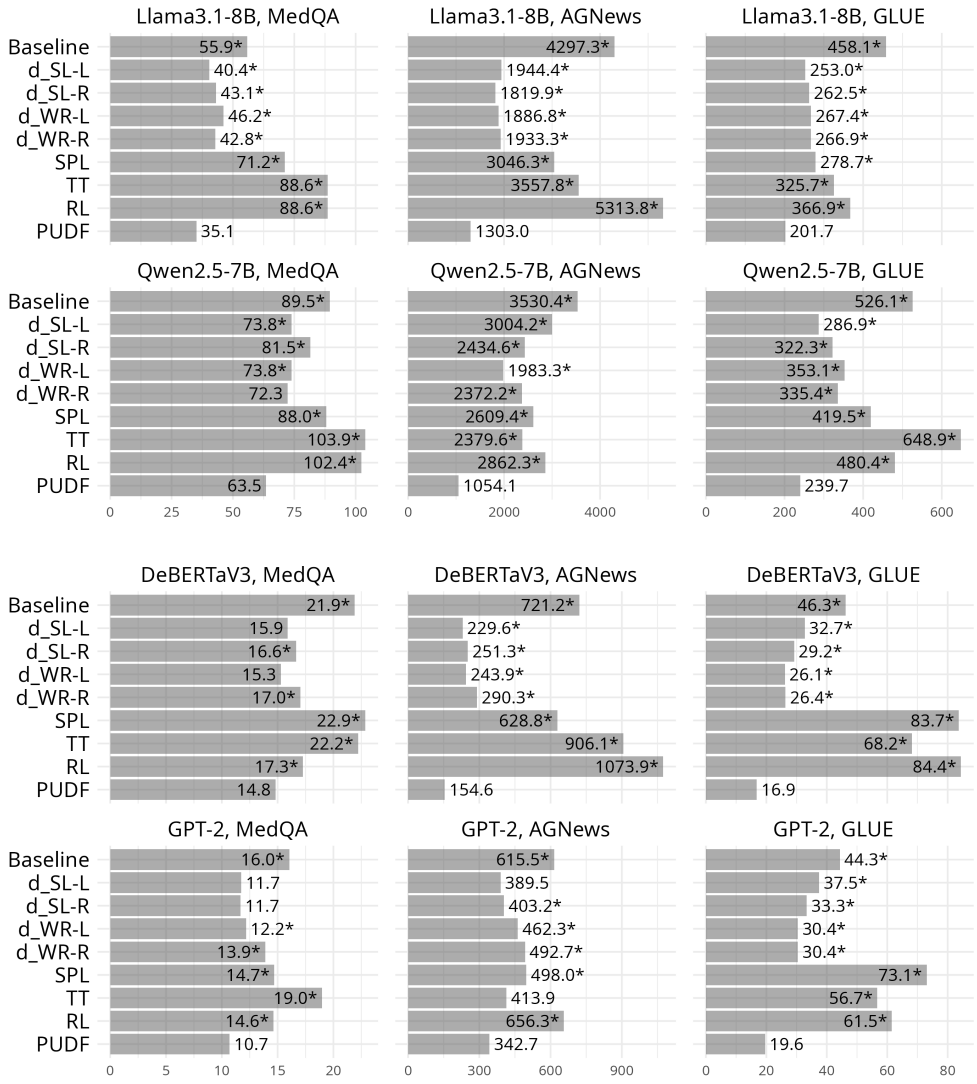


Figure 3: Comparing training time between PUDF and other CL methods. All runtimes reported in minutes. GLUE scores are reported as the mean across tasks, pooled by runs. $*$Indicates that the runtime is significantly longer than PUDF (Welch's single-tailed t-test with Benjamini-Hochberg correction, $\alpha < 0.05$).

cially for larger datasets such as AG News and more difficult datasets such as MedQA, highlight the robustness and effectiveness of PUDF.

5.3 Further Analyses

Here, we provide in-depth analyses of PUDF to better understand how and why our framework outperforms other methods. We also analyze in detail the characteristics of PUDF's DM and TS to evaluate the benefits of applying IRT to the problem of CL.

5.3.1 Ablation Study. To clarify the factors contributing to PUDF's accuracy and training time improvements, we conduct the following ablation study. For the ablation, we focus on Qwen2.5-7B based on its performance in our main experiments. Specifically, we interchange the DM and TS components between PUDF and predefined CL methods. Table 5 presents the results of our ablation study, from which we can draw several conclusions. In the first part of the experiment, we employed d_{SL} or d_{WR} as the DM in conjunction with DDS-MAE as the TS. Specifically, we first apply min–max normalization to the raw difficulty scores d_{SL} and d_{WR} so that they lie within the predefined IRT-AC difficulty interval. Then, at each epoch, DDS-MAE dynamically selects training samples according to the model's current capability and the normalized difficulty values. The results reveal that DDS-MAE significantly reduces training time compared to rule-based training schedulers (i.e., Linear and Root, Figure 3). Despite the additional time required for dynamic evaluation of model ability in each epoch, this overhead is negligible relative to the overall training time and contributes to improved model convergence. The combination of d_{SL} or d_{WR} with DDS-MAE results in decreased accuracy compared to predefined methods (Table 4) and PUDF, while achieving training times that were significantly slower than PUDF on the AGNews and GLUE datasets. Further analysis of the difficulty distributions generated by d_{SL} and d_{WR} reveals a mismatch with the model ability evaluated by DDS-MAE.

Next, we utilize IRT-AC as the DM in combination with Linear or Root training scheduler functions. Our findings indicate that accuracy improves compared to predefined methods and approaches that of PUDF, suggesting that the difficulty generated by IRT-AC is more suitable for model training than d_{SL} and d_{WR}. This underscores the efficacy of the IRT-AC method compared to the rule-based DMs. The training time remains similar to predefined methods but is slower than PUDF. This observation is consistent with expectations, as IRT-AC is responsible for labeling data difficulty, while the training scheduler controls the training pace. While combining IRT-AC with Root or Linear schedulers improves accuracy but fails to improve training time, pairing it with DDS-MAE (i.e., PUDF) yields substantial enhancements, demonstrating the importance of complementarity between the DM and TS. In conclusion, PUDF (IRT-AC + DDS-MAE) consistently outperforms other combinations across benchmark datasets, demonstrating the synergistic effect of its components. The IRT-AC component enhances model accuracy, while the DDS-MAE algorithm, guided by the IRT-AC difficulty scores, significantly reduces training time, resulting in an efficient and effective CL approach.

Table 5: Results of ablation study for Qwen2.5-7B. The best performing method for each model is in **bold**; the second-best model is underlined. For GLUE, we report the mean scores across tasks, pooled by runs.

Metric	DM	TS	MedQA	AGNews	GLUE
Accuracy	d_SL	DDS-MAE	$\underline{64.55}_{\pm 2.71}$	$70.70_{\pm 2.47}$	$88.97^{*}_{\pm 0.47}$
	d_WR	DDS-MAE	$62.90^{*}_{\pm 1.38}$	$71.22^{*}_{\pm 1.44}$	$88.65^{*}_{\pm 0.93}$
	IRT-AC	Root	$63.19^{*}_{\pm 0.79}$	$\underline{72.19}^{*}_{\pm 0.46}$	$\underline{89.30}^{*}_{\pm 0.59}$
	IRT-AC	Linear	$63.02^{*}_{\pm 0.83}$	$71.99^{*}_{\pm 0.45}$	$89.19^{*}_{\pm 0.45}$
	IRT-AC	DDS-MAE (PUDF)	$\mathbf{65.02}_{\pm 0.22}$	$\mathbf{72.93}_{\pm 0.32}$	$\mathbf{90.90}_{\pm 0.43}$
Training Time (minutes)	d_SL	DDS-MAE	$76.80^{*}_{\pm 7.18}$	$\underline{1587.60}^{*}_{\pm 468.7}$	$330.19^{*}_{\pm 45.4}$
	d_WR	DDS-MAE	$\underline{68.55}_{\pm 6.14}$	$1625.64^{*}_{\pm 258.55}$	$\underline{316.78}^{*}_{\pm 48.5}$
	IRT-AC	Root	$87.31^{*}_{\pm 8.66}$	$2371.32^{*}_{\pm 335.0}$	$382.51^{*}_{\pm 43.9}$
	IRT-AC	Linear	$91.77^{*}_{\pm 13.96}$	$1981.68^{*}_{\pm 269.92}$	$323.97^{*}_{\pm 33.7}$
	IRT-AC	DDS-MAE (PUDF)	$\mathbf{63.51}_{\pm 7.60}$	$\mathbf{1054.08}_{\pm 131.1}$	$\mathbf{239.71}_{\pm 28.9}$

*Indicates that the value is significantly worse than the **best value** in the column (Welch's single-tailed t-test with Benjamini-Hochberg correction, $\alpha < 0.05$).

5.3.2 Computational Cost and Efficiency of PUDF.

IRT-AC. The IRT-AC difficulty generation process is designed as an offline, one-time procedure per dataset; that said, its computation cost may still be high. The specifics of this overhead and its impact on overall training efficiency are detailed in Table 6. As illustrated in Figure 2, the IRT-AC framework comprises two primary stages: (i) obtaining Prediction Correctness from the AC and (ii) IRT Difficulty Estimation. For the Prediction Correctness stage, we employ a diverse set of 13 LLMs (§5.1.2). To enhance the robustness and diversity of the AC, these models are fine-tuned for varying epochs (0, 1, 3, 5, and 10) on the respective validation sets before generating predictions on the training data. A key practical advantage of this stage is its parallelizability, as each LLM in the AC can be fine-tuned independently, potentially reducing the effective wall-clock time with concurrent computation. The second stage, IRT Difficulty Estimation, then utilizes a 1PL IRT model fit using VI (Lalor, Wu, and Yu 2019; Lalor and Rodriguez 2023), to efficiently estimate example difficulties from the response patterns.

The time invested in these IRT-AC stages (e.g., 3.94 hours for AG News, 9.89 minutes for MedQA, and an average of 127.3 minutes for GLUE tasks, as shown in Table 6) constitutes a manageable, upfront computational cost. More importantly, this initial overhead is offset by substantial gains in overall training efficiency when using the PUDF framework. By comparing the Qwen2.5-7B baseline training time with the Total Time (PUDF)—which includes both the IRT-AC processing and the subsequent DDS-MAE-guided training—we observe consistent net time savings. For instance, the total pipeline time for AG News with PUDF is 22.73 hours, a 63.8% reduction from the 62.87-hour baseline. Similar efficiencies are evident for MedQA (77.70 minutes vs. 94.37 minutes) and the GLUE average (383.3 minutes vs. 568.00 minutes). These results underscore that the IRT-AC overhead is not only tolerable, due to its offline and parallelizable nature, but its integration into the PUDF framework ultimately leads to a more time-efficient training process. In addition, the main IRT-AC cost, Prediction Correctness, can be reduced by reusing previously stored AC model response patterns.

Table 6: Comparison of training times for the baseline model versus the PUDF-guided approach for Qwen2.5-7B, broken down by stage. Time units are specified per dataset.

Fine-Tuning Approach		**MedQA** (minutes)	**AG News** (hours)	**GLUE** (minutes)
PUDF	Prediction Correctness	7.91	3.15	102.32
	IRT Difficulty	1.98	0.79	24.98
	IRT-AC Total	9.89	3.94	127.30
	Ability Estimation	8.78	2.56	35.97
	Fine-Tuning	59.03	16.23	220.03
	DDS-MAE Total	67.81	18.79	256.00
	PUDF Total	77.70	22.73	383.30
No CL Benchmark		94.37	62.87	568.00

DDS-MAE. The DDS-MAE component of PUDF introduces specific computational steps at each epoch for its decision-making process related to data selection and scheduling. We analyze the runtime characteristics of this component and, more importantly, the overall impact of the PUDF method on the total training duration when applied to the Qwen2.5-7B model, compared to a standard Qwen2.5-7B baseline. As detailed in Table 6, the computational time attributed directly to Ability Estimation is modest, with its duration corresponding to a small percentage (ranging from 4.07% to 9.30%) of the original Qwen2.5-7B baseline's runtime on the respective datasets. This highlights that the additional computational steps introduced by the DDS-MAE component are relatively lightweight. Crucially, despite this inherent processing time from the DDS-MAE component, the integration of PUDF leads to a substantial reduction in the overall training duration compared to the Qwen2.5-7B baseline. Table 6 demonstrates these time savings across all evaluated datasets, with total training times for PUDF being considerably shorter; in some cases, such as AG News and the GLUE average, training time reduced by more than half. This demonstrates that the PUDF method, incorporating DDS-MAE, offers significant gains in computational efficiency for effectively training the Qwen2.5-7B model across diverse datasets.

5.3.3 Convergence Analysis. We next conduct a convergence analysis to evaluate DDS-MAE, our proposed TS component of PUDF, using the Qwen2.5-7B model as the foundational architecture. The training dynamics of PUDF compared to a no-CL baseline are illustrated in Figure 4. Several key observations emerge from these results:

Convergence Speed. Across the evaluated datasets, PUDF demonstrates faster convergence and higher validation accuracy levels in fewer epochs compared to the no-CL baseline. This trend is notably visible in MedQA, where PUDF shows a much steeper accuracy ascent in the initial epochs, consistent with results for AG News, MNLI, MRPC, and QNLI. In contrast, the baseline model often requires more training epochs to reach comparable performance. The rapid convergence characteristic of PUDF reinforces its potential for achieving strong results with reduced training iterations, thereby conserving computational resources.

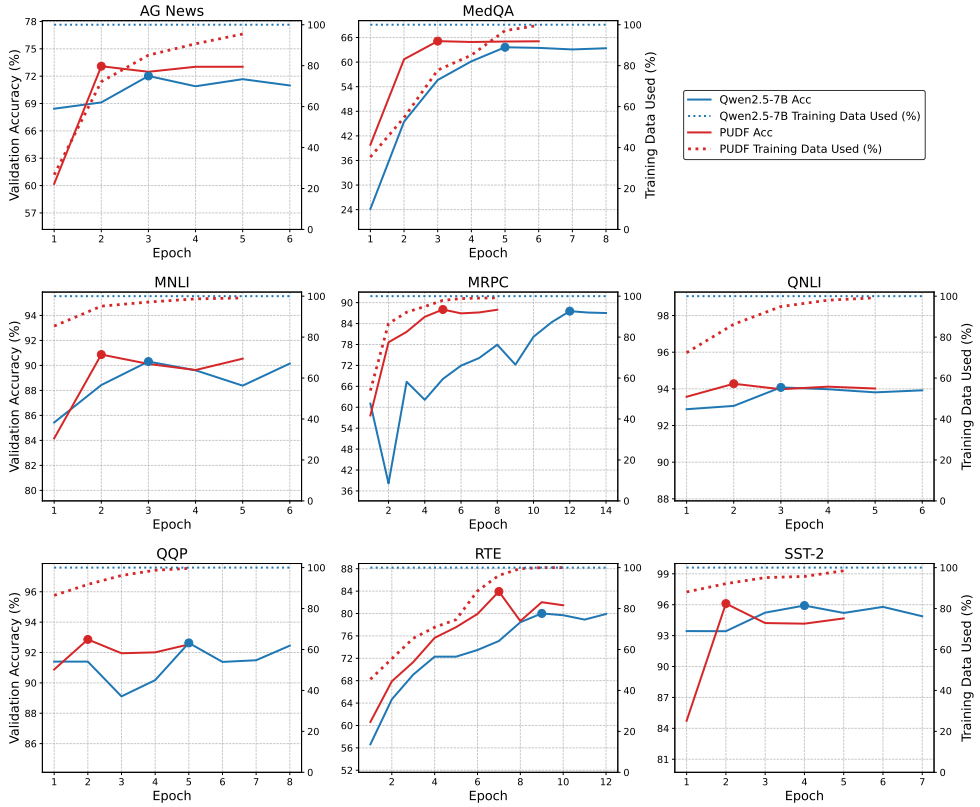


Figure 4: Convergence analysis of the proposed PUDF against the Qwen2.5-7B baseline on AG News, MedQA, and the GLUE benchmark datasets. The solid lines represent validation accuracy, while the dotted lines indicate the percentage of training data utilized per epoch. Circular markers highlight the epoch with the best validation accuracy achieved by each model.

Accuracy Comparison. In terms of peak validation accuracy, indicated by circular markers in Figure 4, PUDF consistently matches or surpasses the performance of the baseline across the majority of the evaluated tasks. Notably, PUDF achieves distinctly higher peak accuracy on MedQA, AG News, RTE, and MNLI. On the remaining tasks (QQP, SST-2, MRPC, and QNLI), PUDF achieves slightly improved peak results.

Data Efficiency. The percentage of training data utilized by PUDF during fine-tuning (represented by the red dotted line in Figure 4) offers critical insights into its data efficiency. A consistent observation across most tasks, including AG News and MedQA, is that PUDF does not initially process the entire training dataset. In the early epochs, typically around 40-60% of the data is actively used (e.g., AG News, MedQA, MNLI, MRPC; SST-2 starts lower). As the fine-tuning progresses and the model's capabilities enhance, PUDF progressively incorporates a larger fraction of the training data. This gradual data introduction strategy aligns with the model's increasing capacity to effectively learn from a broader and potentially more complex range of examples.

Data Usage and Strategic Scheduling. Analysis of the training trajectories reveals that PUDF frequently attains peak validation accuracy without the complete training

dataset. For instance, on MedQA, AG News, QNLI, and QQP, optimal performance is often reached when PUDF has utilized approximately 70-95% of the available training data. This highlights the model's proficiency in identifying and prioritizing the most informative examples for learning. Furthermore, this behavior underscores the principle that not all training instances contribute equally or positively to model performance, particularly in the early stages. As elaborated in our qualitative analysis (§6.2), some data points may be inherently more challenging or even contain labeling inaccuracies. The DDS-MAE component of PUDF is designed to leverage information about example difficulty (e.g., as identified by IRT-AC) to strategically schedule more difficult or potentially noisy data points at later stages of training. This approach mitigates their potential to disrupt learning when the model is less robust. This curated data presentation contrasts with a more conventional or random ordering of training instances.

Training Stability. The validation accuracy curves for PUDF generally exhibit greater stability after reaching peak performance, often with less pronounced fluctuations compared to the baseline on several tasks (e.g., MNLI, MRPC). This suggests that PUDF may be less susceptible to overfitting, maintaining more consistent performance as training progresses. We attribute this enhanced stability to the DDS-MAE mechanism, which dynamically selects and schedules training data based on an ongoing assessment of the model's learning state and data characteristics. This adaptive approach can contribute to a more regularized and stable training process than methods employing a less informed or random data feeding strategy.

In summary, the DDS-MAE component integrated within PUDF offers notable advantages in terms of training efficiency, data utilization, and predictive performance across the diverse set of evaluated benchmark datasets, including AG News, MedQA, and tasks from GLUE. It consistently demonstrates the capacity to achieve competitive or superior accuracy in most cases, often with fewer training epochs and by strategically utilizing subsets of the available training data. Moreover, this approach tends to yield more stable training dynamics compared to the baseline.

5.4 Extension of PUDF to Generative Tasks

Thus far, we have demonstrated the ability of PUDF on several classification tasks. In this section, we show that PUDF can also handle novel text generation tasks. Specifically, we apply PUDF to a difficult math question-answering dataset, GSM8K (Cobbe et al. 2021). To adapt to this dataset, we construct an artificial crowd using 11 SOTA large language models with varying prompting strategies. We collect responses from seven models via the Replicate API:[6] Claude 3.5 Sonnet (Anthropic 2024), DeepSeek-V3 (DeepSeek-AI 2024), Granite 3.3 8B Instruct (IBM 2025), Llama 3 8B Instruct (Grattafiori et al. 2024), Llama 3.1 405B Instruct (Grattafiori et al. 2024), GPT-4o-mini (OpenAI 2024), and GPT-5 (OpenAI 2025). Additionally, we employ four models through the Hugging Face API:[7] Yi-1.5 9B Chat (Young et al. 2024), Gemma 2 9B IT (Gemma Team et al. 2024), Mistral 7B Instruct v0.2 (Jiang et al. 2023), and Qwen2.5 7B Instruct (Yang et al. 2024). For each model, we evaluate five prompting strategies: zero-shot, zero-shot with chain-of-thought (CoT) reasoning (Wei et al. 2022), 4-shot, 4-shot with CoT, and 8-shot with CoT. This yields a total of 55 experimental configurations (11 models $\times$ 5 strategies). For hyperparameter tuning and the hardware platform, we

6 https://replicate.com
7 https://huggingface.co

adopted the same settings used for Llama3.1-8B and Qwen2.5-7B on the MedQA and AG News datasets.

Table 7: Accuracy and training time results on GSM8K comparing PUDF with other curriculum learning methods. Results are averaged over 5 runs with standard deviations as subscripts. The best performing method for each model is in **bold**; the second-best method is underlined.

Model	Method	Accuracy (%)	Time (mins)
Llama3.1-8B	Baseline	$59.28^{*}_{\pm 0.81}$	$108.41^{*}_{\pm 7.44}$
	d_SL-L	$57.39^{*}_{\pm 1.01}$	$\mathbf{52.10}_{\pm 2.89}$
	d_SL-R	$59.27^{*}_{\pm 0.68}$	$62.38_{\pm 1.56}$
	d_WR-L	$58.45^{*}_{\pm 0.66}$	$55.97_{\pm 1.50}$
	d_WR-R	$58.56^{*}_{\pm 0.89}$	$57.49_{\pm 2.85}$
	SPL	$60.06^{*}_{\pm 0.77}$	$95.89^{*}_{\pm 2.62}$
	TT	$60.64^{*}_{\pm 0.72}$	$115.69^{*}_{\pm 9.98}$
	RL	$\underline{60.99}^{*}_{\pm 0.64}$	$145.24^{*}_{\pm 5.04}$
	PUDF	$\mathbf{61.72}_{\pm 0.44}$	$\underline{78.66}^{*}_{\pm 1.71}$
Qwen2.5-7B	Baseline	$74.10^{*}_{\pm 0.73}$	$146.59^{*}_{\pm 5.81}$
	d_SL-L	$67.88^{*}_{\pm 1.71}$	$\mathbf{63.14}_{\pm 1.30}$
	d_SL-R	$70.85^{*}_{\pm 0.76}$	$73.29_{\pm 1.75}$
	d_WR-L	$73.89^{*}_{\pm 0.78}$	$90.83^{*}_{\pm 1.89}$
	d_WR-R	$72.76^{*}_{\pm 0.58}$	$83.17_{\pm 1.62}$
	SPL	$74.15^{*}_{\pm 0.76}$	$128.83^{*}_{\pm 2.50}$
	TT	$75.89_{\pm 0.99}$	$137.24^{*}_{\pm 3.16}$
	RL	$\underline{76.18}_{\pm 0.61}$	$177.16^{*}_{\pm 2.61}$
	PUDF	$\mathbf{76.70}_{\pm 0.37}$	$\underline{85.05}^{*}_{\pm 1.68}$

*For accuracy: indicates significantly lower than the **best accuracy** for that model. For training time: indicates significantly higher than the **fastest time** for that model (Welch's one-tailed t-test with Benjamini-Hochberg correction, $\alpha < 0.05$).

Table 8: Results of ablation study on GSM8K for Qwen2.5-7B. DM: Difficulty Metric; TS: Training Schedule. The best performing variant is in **bold**; the second-best variant is underlined.

Metric	DM	TS	GSM8K
Accuracy (%)	d_SL	DDS-MAE	$\underline{73.89}^{*}_{\pm 0.80}$
	d_WR	DDS-MAE	$73.68^{*}_{\pm 0.48}$
	IRT-AC	Root	$68.95^{*}_{\pm 1.41}$
	IRT-AC	Linear	$71.52^{*}_{\pm 1.20}$
	IRT-AC	DDS-MAE (PUDF)	$\mathbf{76.70}_{\pm 0.37}$
Training Time (minutes)	d_SL	DDS-MAE	$75.32^{*}_{\pm 1.90}$
	d_WR	DDS-MAE	$71.71^{*}_{\pm 1.61}$
	IRT-AC	Root	$\underline{57.96}^{*}_{\pm 1.91}$
	IRT-AC	Linear	$\mathbf{48.64}_{\pm 2.19}$
	IRT-AC	DDS-MAE (PUDF)	$85.05^{*}_{\pm 1.68}$

*For accuracy: indicates significantly lower than the **best accuracy**. For training time: indicates significantly higher than the **fastest time** (Welch's one-tailed t-test, $\alpha < 0.05$).

Table 7 demonstrates that PUDF successfully extends to generative mathematical reasoning tasks, achieving the best performance on both model architectures. For Llama3.1-8B, PUDF attains 61.72% accuracy, significantly outperforming all baselines,

including the strongest competitor, RL (60.99%). On Qwen2.5-7B, PUDF achieves the highest numerical accuracy at 76.70%, though the improvement over strong competitors RL (76.18%) and TT (75.89%) is not statistically significant ($p \approx 0.07$). However, in both cases, PUDF is significantly faster than these high-performing competitors (RL and TT). Notably, heuristic-based CL methods (d_SL-L, d_SL-R, d_WR-L, d_WR-R) consistently underperform compared to learnable difficulty estimation approaches, with d_SL-L achieving only 57.39% on Llama3.1-8B and 67.88% on Qwen2.5-7B. These results validate that PUDF's learnable IRT-based difficulty metric, combined with adaptive pacing through DDS-MAE, effectively captures the complexity of mathematical reasoning tasks where human-annotated difficulty signals are unavailable. Furthermore, PUDF demonstrates competitive training efficiency, requiring substantially less time than RL (145.24 and 177.16 minutes for the two models) while maintaining superior or competitive accuracy.

The ablation study in Table 8 reveals that both the difficulty metric and training schedule components are essential to PUDF's success. When using the same DDS-MAE schedule but replacing IRT-AC with deterministic metrics, d_SL + DDS-MAE achieves 73.89% and d_WR + DDS-MAE reaches 73.68%, both significantly lower than PUDF's 76.70%. Conversely, maintaining the IRT-AC difficulty metric but substituting DDS-MAE with simpler schedules (Root or Linear pacing) yields even worse results of 68.95% and 71.52%, respectively. Notably, these results are worse than those achieved by simple heuristic methods in Table 7 (e.g., d_{WR}-L at 73.89%), confirming that a naive application of the IRT-AC metric without its co-designed DDS-MAE scheduler can actually be detrimental to performance. The training time analysis further illuminates this trade-off: while the Linear schedule completes fastest at 48.64 minutes, it scores 5.18 percentage points lower in accuracy compared to PUDF. These findings underscore that PUDF's effectiveness stems from the synergistic integration of learnable difficulty assessment and dynamic data scheduling, with neither component alone sufficient to achieve optimal performance on complex generative reasoning tasks.

6. Further Analyses: Exploring the IRT-AC

This section presents an in-depth analysis of IRT-AC, which estimates the difficulty value for each data instance. We look at the properties of the learned example difficulties to demonstrate their calibration with expected results and further demonstrate IRT-AC as a difficulty estimation mechanism with potential benefits independent of PUDF.

6.1 Distribution of Difficulty

Figure 5 displays the difficulty distributions estimated by our IRT-AC model for instances across the MedQA, AG News, and GLUE benchmark datasets. A notable characteristic across these diverse datasets is that the difficulty scores generally form distributions approximating a Gaussian profile. This observation suggests that a majority of instances in these benchmarks tend to cluster around a central difficulty level, with fewer examples at the extremes of being excessively easy or prohibitively challenging. Analyzing the mean difficulty values provides insights into the relative challenge posed by each dataset. For instance, datasets such as QQP (mean: -6.13), QNLI (mean: -4.78), and SST-2 (mean: -3.84) have low mean difficulties, indicating they are, on average, less challenging. Conversely, datasets like MedQA (mean: 1.58), AG News (mean: 0.80), and RTE (mean: -0.26) present higher mean difficulty values, suggesting that they are comparatively more challenging. Crucially, these IRT-AC derived difficulty metrics

show a strong correspondence with empirical model performance reported in Table 4 (and Appendix Table A.11). Tasks with lower mean difficulty scores consistently achieve higher accuracies, whereas those identified as more difficult (e.g., MedQA, AG News, RTE) tend to yield lower accuracy scores. This observed trend between our estimated difficulties and actual model outcomes further supports the reliability and validity of the IRT-AC framework in quantifying task and instance-level challenge.

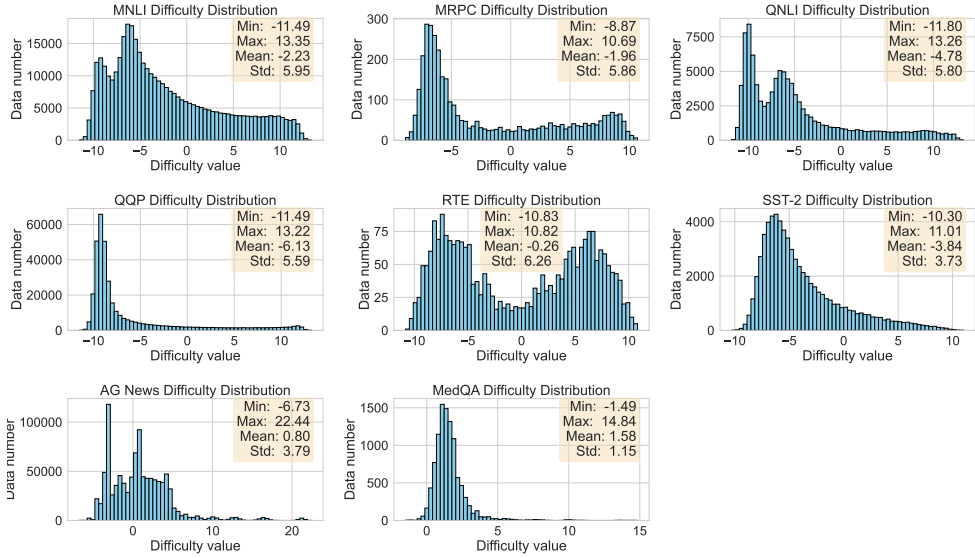


Figure 5: IRT-AC generated difficulty distributions for the GLUE benchmark, AG News, and MedQA datasets.

6.2 Qualitative Analysis of IRT-AC Difficulty Scores

To provide qualitative insights into the difficulty scores from our IRT-AC Difficulty Model, this section analyzes selected examples. We focus on the MedQA dataset, presenting the top five hardest and easiest questions as determined by our model (Table 9). For broader context, examples from the GLUE benchmark and AG News datasets are available in Appendix 1.3. The questions identified as "hardest" by IRT-AC (Table 9a) generally require sophisticated reasoning and specialized knowledge. Many require multi-step inference, such as deducing a condition and then recalling a specific preventative measure or underlying cause from highly technical options (e.g., Questions 1, 4 in Table 9a). Others involve complex clinical decision-making regarding optimal diagnostic or management steps from nuanced alternatives (e.g., Questions 3, 5). These questions often present intricate scenarios and necessitate fine discrimination among medically specific options. Conversely, questions rated "easiest" (Table 9b) typically feature more straightforward scenarios with distinct clinical cues for common conditions (e.g., Questions 1, 5 in Table 9b), leading to relatively direct conclusions about management or treatment. The correct answers frequently align with general medical principles or clearly address the primary issue, while distractors often appear less plausible, thereby reducing the need for deep, specialized knowledge or complex inferential chains. These qualitative observations suggest that the IRT-AC model effectively discerns varying levels of question complexity, associating higher difficulty

scores with tasks requiring more specialized knowledge, multi-step reasoning, and finer discrimination among options. This also highlights the disparity between examples in

Table 9: Top 5 hardest and easiest questions from the MedQA dataset. Correct answer for each is **in bold**.

(a) Top 5 Hardest Questions
Q: A 35-year-old man is brought to the emergency department by his wife because of a 1-week history of progressive confusion, myalgia, and nausea. His wife says that he first reported headaches and fatigue 10 days ago, and since then "he has not been himself." He has refused to drink any liquids for the last day. Two months ago, he helped his neighbor remove a raccoon's den from her backyard. He appears agitated. His temperature is 100.8°F (38.2°C). Examination shows excessive drooling. Muscle tone and deep tendon reflexes are increased bilaterally. Administration of which of the following is most likely to have prevented this patient's condition? A) RNA-dependent DNA polymerase inhibitor **B) Chemically-inactivated virus** C) Live attenuated vaccine D) Immunoglobulin against a bacterial protein
Q: Physical exam of a 15-year-old female reveals impetigo around her mouth. A sample of the pus is taken and cultured. Growth reveals gram-positive cocci in chains that are bacitracin sensitive. Which of the following symptoms would be concerning for a serious sequelae of this skin infection? A) Fever B) Myocarditis **C) Hematuria** D) Chorea
Q: A 40-year-old man comes to the physician for a follow-up examination. He feels well. He has no urinary urgency, increased frequency, dysuria, or gross hematuria. He has a history of recurrent urinary tract infections; his last UTI was treated with ciprofloxacin 3 months ago. Exam and labs are unremarkable except trace blood on UA. Cystoscopy is normal. Which of the following is the most appropriate next step in management? A) Transrectal ultrasound B) Voided urine cytology C) Reassurance **D) CT urography**
Q: A 4-week-old boy is brought to the ED with a 2-day history of projectile vomiting after feeding. Parents report normal development until 1 week ago when he began to eat less. Exam: palpable "olive" in RUQ, non-bilious vomiting. Which of the following is associated with the most likely cause of this patient's symptoms? A) Chloride transport defect B) Failure of neural crest migration **C) Nitric oxide synthase deficiency** D) Recanalization defect
Q: A 54-year-old man presents with 4 months of foul-smelling diarrhea, weight loss, and fatigue after returning from Bangladesh. Labs: macrocytic anemia, fecal fat 22 g/day, TTG-IgA negative, stool studies negative. What is the most appropriate next step in diagnosis? A) CT scan of the abdomen B) Schilling test **C) Enteroscopy** D) PAS-stained biopsy of small bowel

(b) Top 5 Easiest Questions
Q: A 28-year-old Caucasian woman presents to a local walk-in clinic with pruritus and a salmon-colored scaling patch on her back. She had a cold two weeks ago and her lesion has enlarged. Exam: generalized exanthem with collarette scale. What is the best next step in management? **A) Pruritus control and reassurance** B) Systemic steroid therapy C) Topical steroid therapy D) Phototherapy
Q: A 16-year-old man after a camping trip has 5 days of flatulence, nausea, and greasy, foul-smelling diarrhea. No blood or urgency. Exam: mild diffuse abdominal tenderness. What is he most likely to report about his camping activities? **A) Collecting water from a stream, without boiling or chemical treatment** B) This has been going on for months C) Camped as a side excursion from a cruise ship D) Camped in Mexico
Q: A 21-year-old female college student is brought by roommates who note hair-pulling and biopsy shows traumatic alopecia. What is the single most appropriate treatment? **A) Cognitive-behavior therapy or behavior modification** B) Clomipramine C) Venlafaxine D) Electroconvulsive therapy
Q: A 55-year-old woman with type 1 diabetes has 3 months of urinary dribbling and elevated post-void residual. Which intervention is most likely to benefit? **A) Intermittent catheterization** B) Amitriptyline therapy C) Prazosin therapy D) Oxybutynin therapy
Q: A 6-year-old girl with 3 days of malaise, mouth sores, and vesicles on hands and feet. What is the next best step in management? **A) Supportive care** B) Aspirin C) Corticosteroids D) Penicillin

a particular dataset and further strengthens the conceptual motivation for CL generally and PUDF specifically.

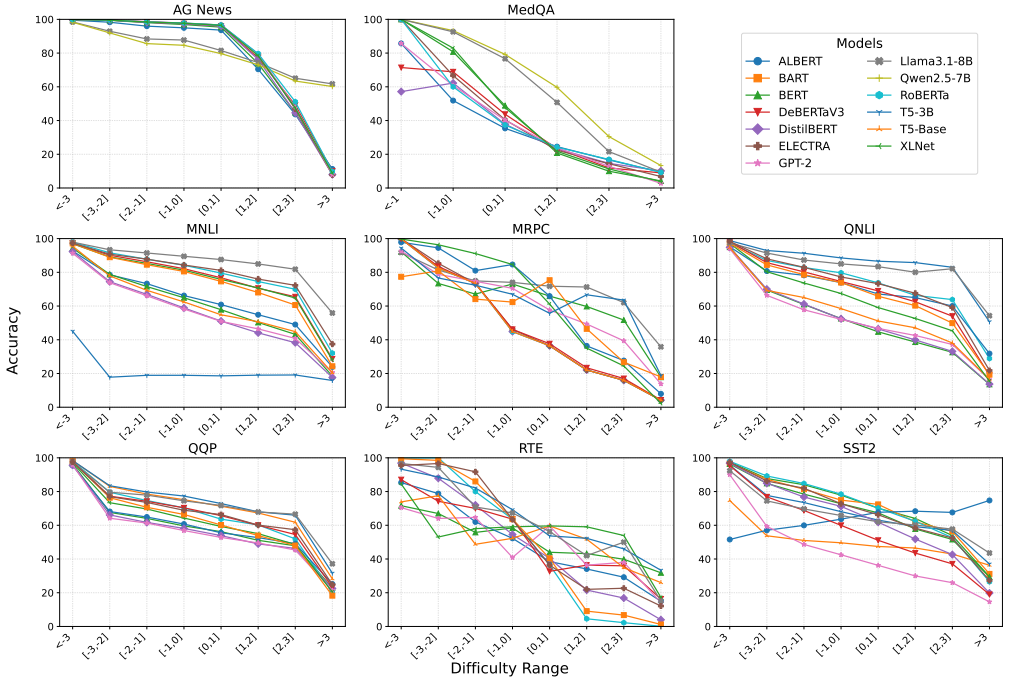


Figure 6: Accuracy of diverse language models (comprising the Artificial Crowd) across IRT-AC difficulty bins for GLUE, AG News, and MedQA datasets. The legend details the specific LLMs utilized.

6.3 Artificial Crowd Accuracy Across Difficulty Bins

This section examines the relationship between IRT-AC derived example difficulty and the empirical performance of a diverse set of LLMs. These LLM crowd models provide multiple perspectives on how accuracy varies with data difficulty. Figure 6 illustrates the accuracy of each LLM within this crowd across binned difficulty levels, ranging from easiest (difficulty < -3) to hardest (difficulty > 3), for the GLUE, AG News, and MedQA datasets. We observe a consistent inverse correlation between example difficulty and model accuracy; as the IRT-AC assessed difficulty of data instances increases, the evaluation accuracy of the LLMs systematically decreases. This trend holds across all examined datasets, demonstrating the link between higher difficulty scores and lower empirical success rates. This consistent behavior across a spectrum of models and data types strongly validates the IRT-AC scores, confirming their efficacy in capturing meaningful signal of task and instance-level challenge that directly predict model performance.

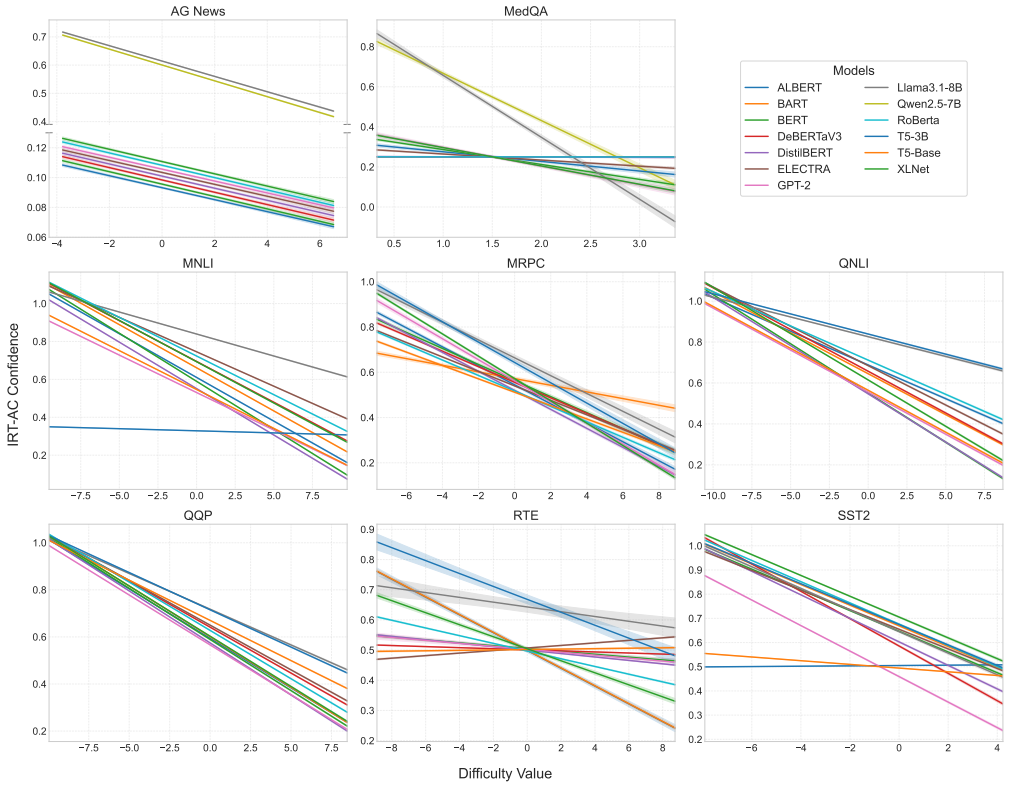


Figure 7: Model confidence in relation to IRT-AC example difficulty across GLUE, AG News, and MedQA datasets. Confidence is defined as the model's output probability for the correct label.

6.4 Model Confidence in Relation to Example Difficulty

This analysis investigates model confidence, quantified as the probability assigned by each model to the true label, relative to IRT-AC derived example difficulty (Figure 7). We use Ordinary Least Squares (OLS) regression to estimate the relationship between difficulty and model confidence, focusing on examples within the 5th to 95th difficulty percentile for each task across the GLUE, AG News, and MedQA datasets. The predominant finding across most datasets and models is an inverse correlation: model confidence generally decreases as the IRT-AC assessed difficulty of instances increases. This suggests a meaningful alignment between our difficulty metric and the models' internal assessments of certainty.

While this inverse relationship is broadly consistent, Figure 7 also reveals notable dataset- and model-specific variations. On AG News, for instance, larger and more recent models such as Llama3.1-8B and Qwen2.5-7B tend to maintain markedly higher confidence in the correct label across the difficulty spectrum compared to several other LLMs. This sustained confidence could be attributed to their enhanced representational power and potentially better calibration on this news classification task. Conversely, MedQA typically shows a steeper confidence decline for most models, underscoring its inherent challenge. Other datasets, like RTE, exhibit more varied confidence trends,

with some models deviating from a clear downward slope, possibly reflecting task-specific complexities or differences in model calibration. Despite these variations, the overarching tendency of diminishing confidence on examples identified as harder by IRT-AC further substantiates the validity of our difficulty scores as indicators of task and instance-level challenge.

6.5 AC Ablation

In this section, we investigate the importance of the construction of the IRT-AC and its effect on estimated difficulties. Specifically, we randomly sample a subsection of our AC models and re-fit our IRT models to determine whether we can recover the learned difficulty parameters with a smaller crowd size. We calculate Pearson and Spearman correlations between difficulty estimates from our sampled crowd and difficulty estimates from our full IRT-AC (Table 10. With only 5 models, correlations vary widely, indicating that there is insufficient information in the response patterns to accurately estimate the latent difficulty parameters. As the crowd size increases, the correlations improve. With a crowd size of 30, correlations are consistently high. These results indicate the importance of collecting enough data from a variety of models for IRT-AC. The estimated difficulty and ability parameters are population dependent, so ensuring a representative sample of models will improve IRT-AC estimations. If needed, the IRT-AC can be updated with response patterns from new SOTA models to reflect the updated state of overall model performance.

Table 10: Pearson and Spearman correlation coefficients when comparing IRT difficulty values as estimated from a subsection of AC models to the full AC estimates. All correlations are significantly different than 0 ($p < 0.01$).

	AC	AG News	MedQA	MNLI	MRPC	QNLI	QQP	RTE	SST-2
Pearson	5	0.70	0.40	0.53	0.79	0.63	0.45	0.71	0.29
	15	0.72	0.56	0.72	0.93	0.72	0.85	0.70	0.71
	30	0.93	0.82	0.92	0.93	0.91	0.83	0.82	0.88
	45	0.97	0.91	0.97	0.97	0.95	0.92	0.97	0.90
Spearman	5	0.73	0.48	0.72	0.78	0.64	0.46	0.86	0.43
	15	0.95	0.78	0.94	0.87	0.89	0.67	0.93	0.66
	30	0.97	0.91	0.96	0.88	0.94	0.84	0.97	0.87
	45	0.98	0.94	0.97	0.92	0.97	0.81	0.97	0.87

7. Limitations

This work is not without limitations. One potential issue with PUDF is the chance of a high variance model, due to the additional step of estimating model ability during training. However, in our results, we find that variance in terms of output performance is low for PUDF (§5.2). We can infer that the ability estimation process is relatively stable. That is, the example difficulties estimated from IRT are stable enough that ability estimates align with the current state of the model, as indicated by the regular progression through the curriculum and increasing training and validation accuracy performance. Our results show that adding this step does not lead to a higher variance model; in certain cases, PUDF has lower variance than the baseline and competence-based frameworks.

For PUDF, there is a potentially significant cost associated with estimating θ_e. Estimating θ_e requires an additional forward pass through the training dataset to gather the labels for scoring as well as MLE estimation. For large datasets, this can effectively double the number of forward passes during training. To alleviate the extra cost, we sample from the training set before our first epoch, and use this down-sampled subset as our ability estimation set. As most examples have difficulty values between -3 and 3, the full training set isn't necessary for estimating θ_e. Identifying the optimal number of examples needed to estimate ability is left for future work.

Another computational cost to PUDF involves IRT-AC, specifically response pattern generation. Collecting response patterns needed for difficulty estimation involves fine-tuning multiple LLMs, which can be costly in terms of runtime. For example, as shown in Table 6, IRT-AC computation time accounts for 12.7% to 33.2% of the total PUDF runtime. While the total runtime is less than a standard training runtime, this cost is not negligible, especially when considering larger, computationally expensive models being used. However, we do note several mitigating factors for future implementations to offset some of these costs. First, our analyses show that these costs can be reduced by fine-tuning models in parallel. Second, pre-trained models can be used without further fine-tuning; we can directly run inference for less costly response pattern collection. For example, future work looking at inference time techniques such as few-shot learning and chain-of-thought for IRT-AC may find methods that reduce the computation burden further. Third, the large number of leaderboards that are available for tracking LLM progress are a potential source of response pattern data for IRT-AC in new domains (Rodriguez et al. 2021). Lastly, IRT-AC is not required for every run of PUDF. Once the difficulties have been estimated via IRT-AC, those learned values are valid for training any subsequent model using PUDF. Therefore, the IRT-AC cost can be amortized across multiple fine-tuning runs for different models, making the overall computational burden less. This can also act as an encouragement to future researchers to record and make available instance-level response patterns or learned difficulty parameters so that there is a shared pool of responses for difficulty estimation. For example, the IRT-AC response patterns for our benchmarking datasets[8] can be a foundation for future work where new models are added to the AC and novel fine-tuning strategies leverage the pre-existing AC. Relatedly, the size and composition of IRT-AC should be sufficiently large and diverse to ensure that the learned difficulty parameters reflect variations in responses due to differences in latent ability. Otherwise, responses may be too homogeneous and therefore cannot capture the variation needed to ensure accurate difficulty estimations and appropriate scheduling in PUDF.

8. Conclusion

In this paper, we introduce PUDF, a novel Psychology-based Unified Dynamic Framework for Curriculum Learning. By combining IRT-AC for data difficulty measurement and DDS-MAE for dynamic training scheduling, PUDF offers a theoretically grounded and automated approach to CL. Our extensive experiments on a robust benchmark of datasets and comparison CL methods demonstrate that PUDF consistently improves both accuracy and training efficiency across multiple pre-trained language models and tasks, outperforming SOTA CL methods. The success of PUDF opens up promising di-

8 https://huggingface.co/datasets/lalor/response-patterns

rections for future research, including applications in other domains, such as computer vision and multimodal domains.

This work validates and supports the existing literature on curriculum learning. Our results confirm that curriculum learning frameworks for supervised learning can lead to faster convergence or better local minima, as measured by test set performance (Bengio et al. 2009). We have shown that by replacing a heuristic for difficulty with a theoretically-based, learned difficulty value for training examples, static curriculum learning frameworks can be improved. Probing the model's ability allows for data to be selected for training that is appropriate for the model and is not rigidly tied to a heuristic schedule.

By using PUDF, a curriculum can adapt during training according to the estimated ability of the model. PUDF adds or removes training data based not on a fixed step schedule but rather by probing the model at each epoch and using the estimated ability to match data to the model. This way, if a model has a high estimated ability early in training, then more data can be added to the training set more quickly, and learning isn't artificially slowed down due to the curriculum schedule. It also allows for the possibility of a smaller dataset at later stages, if model performance decreases.

The PUDF framework significantly advances the state of CL and its application to NLP. By incorporating psychological principles through IRT and leveraging dynamic data selection strategies, PUDF offers a theoretically robust and adaptable approach to CL. This framework improves traditional heuristic-based methods and current CL methods, providing a explainable and modular system for dynamically aligning training data with the evolving capabilities of the model. PUDF demonstrates its effectiveness by optimizing the fine-tuning process for LLMs, improving performance metrics such as accuracy and training time across a range of tasks. Its dynamic scheduling mechanism reduces the reliance on static curriculum schedules, facilitating more efficient training without imposing additional computational overheads. PUDF can facilitate advancing CL research and enhancing the practical deployment of pre-trained language models in diverse applications (Yang and Song 2022; Chaudhury et al. 2024; Liu et al. 2024).

There are several avenues for future work. Even though it is dynamic, PUDF employs a simple curriculum schedule: only include examples where difficulty is less than or equal to estimated ability. However, being able to estimate ability on the fly with PUDF suggests the following research question: what is the best way to build a curriculum, knowing example difficulty and model ability? It may be the case that only data with difficulty within a range of ability (higher and lower) is better, or that the training set shifts as the model improves. Future research could also investigate the applicability of the 85% rule of (Wilson et al. 2019) for curriculum design in LLMs.

PUDF can also be adapted to more traditional information retrieval tasks, such as learning to rank and online judging for training high-ability systems and ordering examples according to learned difficulty. In particular, with a 1PL IRT model, the intuitive link between θ and b allows for inherently explainable training mechanisms. An example is only included in training if its difficulty is lower than the model's estimated ability at that point in time. This can be easily explained to model stakeholders and compared with standardized tests for humans, where questions are selected based on human-estimated ability.

Acknowledgments

The authors would like to thank Hao Wu and Hadi Amiri for their helpful conversations with regards to this work. This work was supported in part by LM012817 from the National Institutes

of Health, I01HX003969 from VA Health Systems Research, and IIS-2403438 from the National Science Foundation. This work was also supported in part by the Center for Intelligent Information Retrieval at UMass Amherst, and the Center for Research Computing, the Human-centered Analytics Lab, and the Mendoza College of Business at the University of Notre Dame. The contents of this paper do not represent the views of CIIR, NIH, NSF, VA, the University of Notre Dame, the University of Massachusetts, or the United States Government.

References

Anthropic. 2024. Introducing claude 3.5 sonnet. Accessed: 2024-06-20.

Bai, Ziwei, Junpeng Liu, Meiqi Wang, Caixia Yuan, and Xiaojie Wang. 2022. Exploiting diverse information in pre-trained language model for multi-choice machine reading comprehension. *Applied Sciences*, 12(6):3072.

Baker, Frank B. and Seock-Ho Kim. 2004. *Item Response Theory: Parameter Estimation Techniques, Second Edition*. CRC Press.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, ACM.

Benjamini, Yoav and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Bentivogli, Luisa, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Text Analysis Conference (TAC)*.

Bingham, Eli, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. 2018. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*.

Bock, R Darrell and Murray Aitkin. 1981. Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika*, 46(4):443–459.

Brown, Tom, 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.

Burduja, Mihail and Radu Tudor Ionescu. 2021. Unsupervised medical image alignment with curriculum learning. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3787–3791, IEEE.

Chaudhury, Rohan, Maria Teleki, Xiangjue Dong, and James Caverlee. 2024. Dacl: Disfluency augmented curriculum learning for fluent text generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4311–4321.

Cirik, Volkan, Eduard Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*.

Clark, Kevin, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Cook, Ryan A, John P Lalor, and Ahmed Abbasi. 2025. No simple answer to data complexity: An examination of instance-level complexity metrics for classification tasks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2553–2573.

De Ayala, Rafael Jaime. 2013. *The theory and practice of item response theory*. Guilford Publications.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*.

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Dolan, William B and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*.

Elman, Jeffrey L. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Freund, Yoav and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Gemma Team, Morgane Rivière, 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*.

Grattafiori, Aaron, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Graves, Alex, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320, Pmlr.

Hacohen, Guy and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544, PMLR.

He, Pengcheng, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hochreiter, S and J Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoffman, Matthew D, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research*.

Hu, Edward J, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

IBM. 2025. ibm-granite/granite-3.3-8b-instruct. Accessed: 2025-11-02.

Iyer, Shankar, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs.

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

Jiang, Lu, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander Hauptmann. 2015. Self-paced curriculum learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Jin, Di, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

Jordan, Michael I, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. 1999. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233.

Kenton, Jacob Devlin Ming-Wei Chang and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Khan, Salman, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41.

Kocmi, Tom and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386.

Kumar, Gaurav, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. *arXiv preprint arXiv:1903.00041*.

Kumar, M, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Lagarias, Jeffrey C, James A Reeds, Margaret H Wright, and Paul E Wright. 1998. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147.

Lalor, John P., Hao Wu, and Hong Yu. 2019. Learning Latent Parameters without Human Response Patterns: Item Response Theory with Artificial Crowds. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2019, Association for Computational Linguistics.

Lalor, John P. and Hong Yu. 2020. Dynamic data selection for curriculum learning via ability estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 545–555, Association for Computational Linguistics, Online.

Lalor, John Patrick and Pedro Rodriguez. 2023. py-irt: A scalable item response theory library for python. *INFORMS Journal on Computing*, 35(1):5–13.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lee, Mingyu, Jun-Hyung Park, Junho Kim, Kang-Min Kim, and SangKeun Lee. 2022. Efficient pre-training of masked language model via concept-based curriculum masking. *arXiv preprint arXiv:2212.07617*.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Liu, Fenglin, Shen Ge, Yuexian Zou, and Xian Wu. 2022. Competence-based multimodal curriculum learning for medical report generation. *arXiv preprint arXiv:2206.14579*.

Liu, Liangxin, Xuebo Liu, Lian Lian, Shengjun Cheng, Jun Rao, Tengfei Yu, Hexuan Deng, and Min Zhang. 2024. Curriculum consistency learning for conditional sentence generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13865–13881.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, Ilya and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Maharana, Adyasha and Mohit Bansal. 2022. On curriculum learning for commonsense reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 983–992.

Manela, Binyamin and Armin Biess. 2022. Curriculum learning with hindsight experience replay for sequential object manipulation tasks. *Neural Networks*, 145:260–270.

Milano, Nicola and Stefano Nolfi. 2021. Automated curriculum learning for embodied agents a neuroevolutionary approach. *Scientific reports*, 11(1):8985.

Mohiuddin, Tasnim, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty. 2022. Data selection curriculum for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1569–1582, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates.

Nagatsuka, Koichi, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a bert with curriculum learning by increasing block-size of input text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 989–996.

Natesan, Prathiba, Ratna Nandakumar, Tom Minka, and Jonathan D. Rubright. 2016. Bayesian Prior Choice in IRT Estimation Using MCMC and Variational Bayes. *Frontiers in Psychology*, 7.

OpenAI. 2024. GPT-4o system card. Accessed: 2024-08-08.

OpenAI. 2025. GPT-5 system card. Accessed: 2025-11-02.

Ormerod, Mark, Jesús Martínez del Rincón, and Barry Devereux. 2024. How is a "kitchen chair" like a "farm horse"? exploring the representation of noun-noun compound semantics in transformer-based language models. *Computational Linguistics*, 50(1):49–81.

Ouyang, Jihong, Zhiyao Yang, Chang Xuan, Bing Wang, Yiyuan Wang, and Ximing Li. 2023. Unsupervised aspect term extraction by integrating sentence-level curriculum learning with token-level self-paced learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1982–1991.

Platanios, Emmanouil Antonios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based Curriculum Learning for Neural Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Association for Computational Linguistics, Minneapolis, Minnesota.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rasch, Georg. 1960. *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests. Nielsen & Lydiche, Oxford, England.

Rodriguez, Pedro, Joe Barrow, Alexander Miserlis Hoyle, John P Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change nlp leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sengupta, Ayan, Shantanu Dixit, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. A good learner can teach better: Teacher-student collaborative knowledge distillation. In *The Twelfth International Conference on Learning Representations*.

Shrivastava, Abhinav, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, D. Christopher Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Association for Computational Linguistics, Seattle, Washington, USA.

Soviany, Petru, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2021. Curriculum self-paced learning for cross-domain object detection. *Computer Vision and Image Understanding*, 204:103166.

Spitkovsky, Valentin I, Hiyan Alshawi, and Dan Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759.

Swayamdipta, Swabha, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293.

Tay, Yi, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6).

Tsvetkov, Yulia, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wan, Yu, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1074–1080, Association for Computational Linguistics, Online.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Wang, Xin, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Wei, Yunchao, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. 2016. Stc: A simple to complex framework for weakly-supervised

semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2314–2320.

Weinshall, Daphna, Gad Cohen, and Dan Amir. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International conference on machine learning*, pages 5238–5246, PMLR.

Williams, Adina, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*.

Wilson, Robert C., Amitai Shenhav, Mark Straccia, and Jonathan D. Cohen. 2019. The eighty five percent rule for optimal learning. *Nature Communications*.

Wu, Mike, Richard L. Davis, Benjamin W. Domingue, Chris Piech, and Noah D. Goodman. 2020. Variational item response theory: Fast, accurate, and expressive. In *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020*, International Educational Data Mining Society.

Xu, Benfeng, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

Yang, An, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv e-prints*, pages arXiv–2412.

Yang, Sicheng and Dandan Song. 2022. Fpc: fine-tuning with prompt curriculum for relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1065–1077.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Yao, Yuan, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.

Young, Alex, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*.

Yuan, Xiaoyong, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824.

Zhan, Runzhe, Xuebo Liu, Derek F Wong, and Lidia S Chao. 2021. Meta-curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14310–14318.

Zhang, Bowen, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419.

Zhang, Shixing, Deqiang Han, Jean Dezert, and Yi Yang. 2024. Weighted self-paced learning with belief functions. *Expert Systems with Applications*, page 124535.

Zhang, Xiang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhao, Mingjun, Haijiang Wu, Di Niu, and Xiaoli Wang. 2020. Reinforced curriculum learning on pre-trained neural machine translation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9652–9659.

Zhao, Yangyang, Zhenyu Wang, and Zhenhua Huang. 2021. Automatic curriculum learning with over-repetition penalty for dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14540–14548.

1. Appendix

1.1 Original Formulation of Artificial Crowd

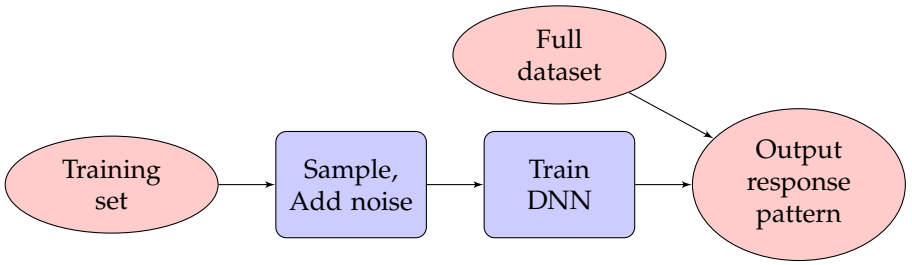


Figure A.8: Response pattern construction for IRT model fitting with artificial crowds from our prior work (Lalor and Yu 2020).

1.2 Main Results for GLUE Tasks

In this section, we report the detailed results across each GLUE task, which have been aggregated in our main results. Table A.11 and Figure A.9 show the classification performance and runtime comparison, respectively, for our benchmarking. PUDF consistently enhances both accuracy and training efficiency across multiple LLMs on the benchmark datasets. These results highlight PUDF's effectiveness in enhancing LLM performance and efficiency across NLP tasks, with particularly strong benefits in reducing computational demands while maintaining or improving accuracy. In almost all cases, the standard deviation of PUDF is low enough to suggest that our improved performance is consistently higher than the benchmarks. The RTE dataset reveals high standard deviations, likely due to the relatively smaller data size compared to other datasets. We see similarly consistent results regarding training time. This indicates that PUDF can effectively reduce training time across a wide range of scenarios. Overall, based on these results, we can summarize that PUDF can improve model accuracy and reduce training time across most scenarios.

Table A.11: Mean and standard deviation accuracy results comparing PUDF with other CL Methods for the GLUE datasets over 5 runs. Best performing method for each model is in **bold**; the second best model is underlined.

Model	Method	MNLI	MRPC	QNLI	QQP	RTE	SST2
DeBERTaV3	Baseline	90.26$_{\pm 0.04}$	86.39$_{\pm 1.28}$	93.43$_{\pm 0.28}$	92.00$_{\pm 0.11}$	80.24$_{\pm 0.91}$	95.58$_{\pm 0.18}$
	d_SL-L	89.45$_{\pm 0.05}$	85.86$_{\pm 1.37}$	92.84$_{\pm 0.12}$	92.08$_{\pm 0.18}$	74.52$_{\pm 0.46}$	93.99$_{\pm 0.28}$
	d_SL-R	89.78$_{\pm 0.09}$	85.86$_{\pm 1.83}$	92.72$_{\pm 0.21}$	92.17$_{\pm 0.19}$	75.96$_{\pm 1.85}$	94.98$_{\pm 0.38}$
	d_WR-L	89.42$_{\pm 0.04}$	85.57$_{\pm 2.00}$	92.92$_{\pm 0.16}$	92.01$_{\pm 0.15}$	69.05$_{\pm 1.99}$	94.13$_{\pm 0.51}$
	d_WR-R	89.64$_{\pm 0.10}$	86.40$_{\pm 1.66}$	93.17$_{\pm 0.21}$	92.08$_{\pm 0.06}$	78.11$_{\pm 0.53}$	94.42$_{\pm 0.15}$
	SPL	89.83$_{\pm 0.14}$	86.86$_{\pm 3.12}$	93.57$_{\pm 0.10}$	91.44$_{\pm 1.11}$	77.16$_{\pm 1.09}$	95.87$_{\pm 0.21}$
	TT	89.09$_{\pm 1.42}$	87.46$_{\pm 2.06}$	93.35$_{\pm 2.34}$	90.80$_{\pm 1.94}$	81.13$_{\pm 0.40}$	94.81$_{\pm 0.93}$
	RL	89.63$_{\pm 0.7}$	**89.48**$_{\pm 1.15}$	94.04$_{\pm 0.46}$	90.35$_{\pm 1.14}$	80.46$_{\pm 0.93}$	95.96$_{\pm 0.29}$
	PUDF	**90.71**$^{*}_{\pm 0.01}$	89.16$_{\pm 0.52}$	**94.80**$_{\pm 0.20}$	**92.57**$^{*}_{\pm 0.05}$	**81.28**$_{\pm 0.48}$	**96.05**$_{\pm 0.24}$
GPT-2	Baseline	81.66$_{\pm 0.05}$	78.14$_{\pm 0.20}$	87.53$_{\pm 0.45}$	89.77$_{\pm 0.16}$	65.70$_{\pm 0.70}$	91.25$_{\pm 0.54}$
	d_SL-L	81.05$_{\pm 0.15}$	75.48$_{\pm 1.04}$	86.94$_{\pm 0.37}$	88.98$_{\pm 0.10}$	51.04$_{\pm 3.25}$	91.29$_{\pm 0.10}$
	d_SL-R	81.41$_{\pm 0.11}$	72.76$_{\pm 1.00}$	87.62$_{\pm 0.40}$	89.02$_{\pm 0.32}$	60.69$_{\pm 1.38}$	90.49$_{\pm 0.61}$
	d_WR-L	81.04$_{\pm 0.18}$	77.64$_{\pm 0.58}$	87.26$_{\pm 0.36}$	88.91$_{\pm 0.46}$	51.45$_{\pm 2.49}$	90.35$_{\pm 0.45}$
	d_WR-R	80.82$_{\pm 0.14}$	74.53$_{\pm 0.46}$	87.44$_{\pm 0.49}$	88.90$_{\pm 0.37}$	61.04$_{\pm 2.16}$	90.51$_{\pm 0.34}$
	SPL	81.36$_{\pm 0.16}$	76.86$_{\pm 0.91}$	86.24$_{\pm 0.89}$	**89.82**$^{*}_{\pm 0.06}$	56.66$_{\pm 0.72}$	90.50$_{\pm 0.22}$
	TT	81.83$_{\pm 0.06}$	**81.67**$^{*}_{\pm 0.42}$	85.89$_{\pm 1.08}$	85.47$_{\pm 1.25}$	65.07$_{\pm 0.54}$	90.88$_{\pm 0.80}$
	RL	81.27$_{\pm 0.12}$	80.57$_{\pm 0.18}$	85.94$_{\pm 0.94}$	88.06$_{\pm 0.80}$	**66.59**$_{\pm 0.41}$	90.84$_{\pm 1.29}$
	PUDF	**82.00**$^{*}_{\pm 0.05}$	79.77$_{\pm 0.17}$	**88.15**$^{*}_{\pm 0.48}$	89.73$_{\pm 0.47}$	66.57$_{\pm 0.90}$	**91.97**$^{*}_{\pm 0.09}$
Llama3.1-8B	Baseline	91.29$_{\pm 0.22}$	88.25$_{\pm 0.11}$	93.86$_{\pm 0.57}$	92.58$_{\pm 0.22}$	81.11$_{\pm 0.97}$	96.46$_{\pm 0.77}$
	d_SL-L	87.61$_{\pm 1.14}$	86.09$_{\pm 1.35}$	90.07$_{\pm 2.44}$	90.07$_{\pm 1.76}$	79.60$_{\pm 0.90}$	93.24$_{\pm 2.16}$
	d_SL-R	88.99$_{\pm 0.78}$	86.82$_{\pm 0.79}$	93.67$_{\pm 0.43}$	90.80$_{\pm 0.58}$	77.27$_{\pm 2.31}$	94.67$_{\pm 0.96}$
	d_WR-L	90.03$_{\pm 0.34}$	85.84$_{\pm 1.73}$	92.97$_{\pm 0.59}$	91.95$_{\pm 0.28}$	79.84$_{\pm 0.67}$	95.03$_{\pm 0.65}$
	d_WR-R	89.49$_{\pm 0.55}$	85.53$_{\pm 1.75}$	92.29$_{\pm 0.68}$	91.39$_{\pm 0.72}$	80.28$_{\pm 0.77}$	95.58$_{\pm 0.61}$
	SPL	90.00$_{\pm 0.14}$	88.45$_{\pm 0.33}$	94.08$_{\pm 0.36}$	92.00$_{\pm 0.21}$	80.95$_{\pm 0.89}$	96.69$_{\pm 0.49}$
	TT	90.33$_{\pm 0.50}$	87.95$_{\pm 0.41}$	92.85$_{\pm 0.87}$	91.98$_{\pm 0.32}$	80.81$_{\pm 0.74}$	96.55$_{\pm 0.49}$
	RL	90.71$_{\pm 0.28}$	87.24$_{\pm 0.37}$	93.93$_{\pm 0.80}$	92.30$_{\pm 0.48}$	81.85$_{\pm 0.30}$	96.95$_{\pm 0.30}$
	PUDF	**91.78**$^{*}_{\pm 0.14}$	**89.39**$^{*}_{\pm 0.11}$	**94.61**$_{\pm 0.44}$	**92.81**$_{\pm 0.31}$	**81.90**$_{\pm 0.89}$	**97.07**$_{\pm 0.52}$
Qwen2.5-7B	Baseline	89.84$_{\pm 0.93}$	87.61$_{\pm 0.44}$	93.61$_{\pm 1.00}$	92.38$_{\pm 0.44}$	79.81$_{\pm 0.41}$	95.66$_{\pm 0.56}$
	d_SL-L	90.03$_{\pm 0.42}$	84.56$_{\pm 1.83}$	93.88$_{\pm 0.20}$	92.06$_{\pm 0.36}$	**83.99**$_{\pm 0.30}$	96.40$_{\pm 0.35}$
	d_SL-R	89.35$_{\pm 0.55}$	85.23$_{\pm 1.44}$	94.04$_{\pm 0.22}$	91.90$_{\pm 0.47}$	80.63$_{\pm 1.17}$	95.40$_{\pm 0.41}$
	d_WR-L	90.09$_{\pm 0.69}$	84.30$_{\pm 1.62}$	93.32$_{\pm 0.39}$	91.98$_{\pm 0.43}$	82.71$_{\pm 0.42}$	95.11$_{\pm 0.17}$
	d_WR-R	89.95$_{\pm 0.62}$	84.31$_{\pm 1.58}$	**94.49**$^{*}_{\pm 0.41}$	91.84$_{\pm 0.35}$	83.05$_{\pm 0.29}$	95.00$_{\pm 0.16}$
	SPL	90.35$_{\pm 0.34}$	86.42$_{\pm 0.77}$	92.91$_{\pm 0.44}$	92.13$_{\pm 0.28}$	76.54$_{\pm 3.20}$	95.44$_{\pm 0.22}$
	TT	89.20$_{\pm 0.43}$	85.00$_{\pm 0.65}$	93.98$_{\pm 0.25}$	91.65$_{\pm 0.35}$	80.91$_{\pm 0.59}$	95.36$_{\pm 1.33}$
	RL	89.97$_{\pm 0.44}$	87.22$_{\pm 0.56}$	93.28$_{\pm 1.03}$	91.10$_{\pm 0.92}$	82.53$_{\pm 0.31}$	95.54$_{\pm 0.75}$
	PUDF	**90.52**$_{\pm 0.64}$	**87.90**$_{\pm 0.33}$	93.86$_{\pm 0.81}$	**92.70**$_{\pm 0.31}$	83.81$_{\pm 0.25}$	**96.64**$_{\pm 0.54}$

*Indicates that the difference between the **best accuracy** and second-best accuracy for a dataset-model experiment is significant (Welch's single-tailed t-test, $p < 0.05$).

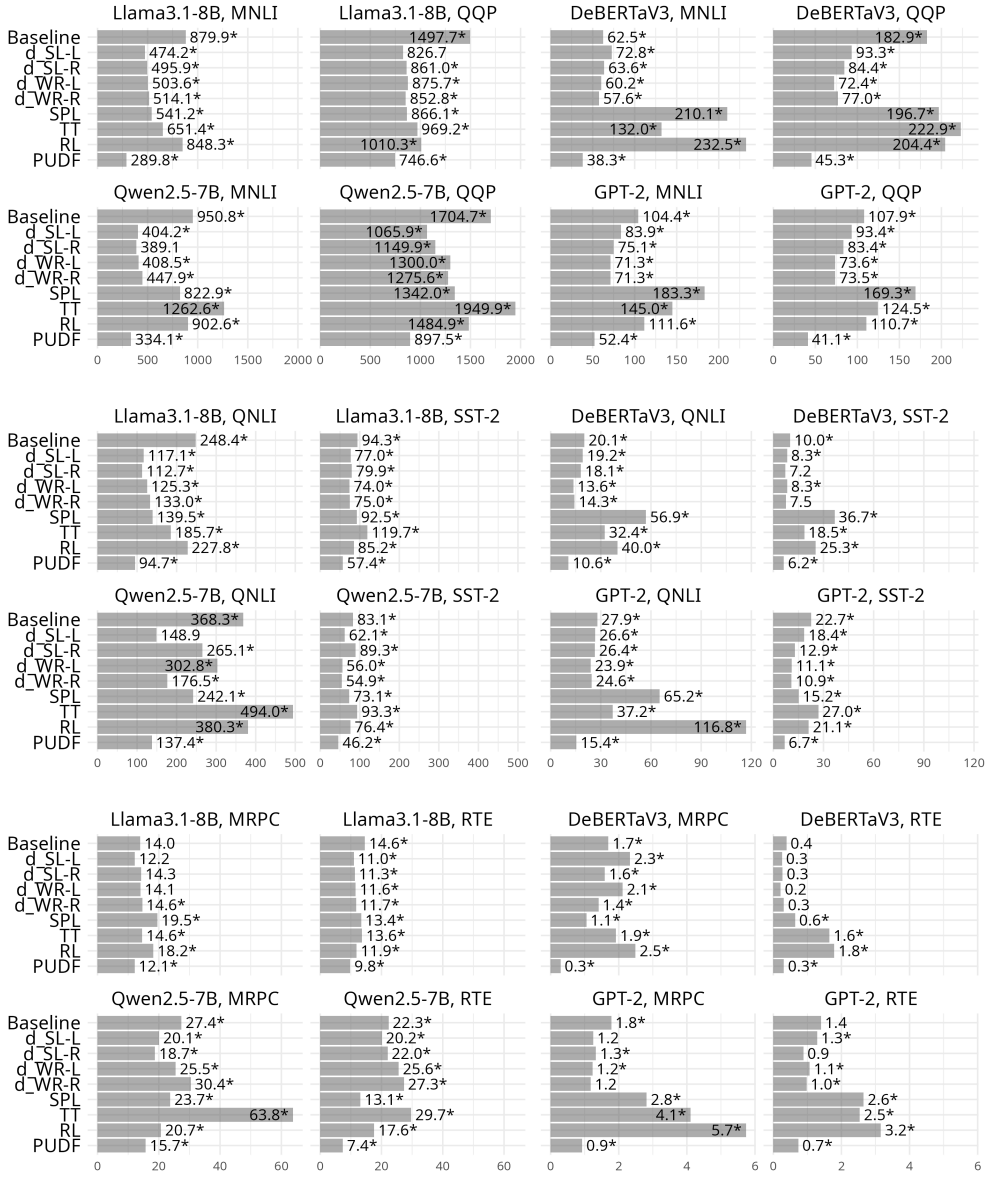


Figure A.9: Comparing training time between PUDF and other CL methods for the GLUE datasets. All runtimes are measured in minutes.
∗Indicates that the runtime is significantly longer than PUDF (Welch's single-tailed t-test with Benjamini-Hochberg correction, $\alpha < 0.05$).

1.3 Additional Qualitative Analysis on IRT-AC

Here we replicate the qualitative analysis discussed in Section 6.2 for the other datasets in GLUE. These examples provide further insights into the IRT-AC on AG News and

GLUE benchmark by presenting the top 5 hardest and easiest sentences for each dataset, along with their respective labels and difficulty scores.

Table A.12: Top 5 hardest and easiest examples in AG News.

(a) Top 5 Hardest Examples

Text	Category	Difficulty
#1 Ben Roethlisberger guided the Steelers to four touchdowns following uncharacteristic New England turnovers as Pittsburgh ended the Patriots' two long win streaks with a remarkably easy 34-20 victory Sunday.	Entertainment	22.4401
#2 Something big is happening in Cairo. It isn't just the release of Azzam Azzam from wrongful imprisonment as an alleged spy, or nice words from President Hosni Mubarak about Prime Minister Ariel Sharon, or the return of the Egyptian ambassador.	Entertainment	22.3745
#3 Chennai: Internet major Yahoo Inc will own a 25-member high-skilled software professional team in Chennai, after it acquired US-based Stata Labs.	Top Stories	22.3580
#4 FOR Wall Street types reluctantly dragging themselves back from the Hamptons, slightly cheaper crude oil prices gave them something to cheer about this week and an excuse to buy some oversold stocks.	Sci/Tech	22.3431
#5 Pakistani President Pervez Musharraf is to hold talks with President Bush in New York on the war against terror.	Europe	22.3350

(b) Top 5 Easiest Examples

Text	Category	Difficulty
#1 AFP - Opponents of the Lao government may be plotting multiple bomb attacks in Vientiane and other areas of Laos timed to coincide with a meeting of Southeast Asian leaders the country is hosting next month, the United States said.	World	-6.7284
#2 Picture dated 1977 from the German criminal office Bundeskriminalamt shows German Red Army Faction activist Brigitte Mohnhaupt. Comment	Europe	-6.6625
#3 AP - The Bush administration has no plans to seek an "Iraq-style" U.N. Security Council resolution on Iran if it succeeds in efforts to have the council address that country's nuclear activities.	World	-6.6365
#4 Gunmen seized control of a school in northwestern Pakistan, taking more than 200 schoolchildren hostage before surrendering to local elders.	World	-6.6194
#5 DETROIT - Democratic presidential candidate John Kerry attacked the Bush administration as the "excuse presidency" on Wednesday, charging that the president stood by while jobs disappeared and the middle class lost ground. "This president has created more excuses than jobs...	World	-6.6161

Table A.13: Top 5 hardest and easiest examples in MNLI.

(a) Top 5 Hardest Examples

Text	Label	Difficulty
#1a If practical, T&A data must be approved at the end of the last day of the pay period or later. #1b T&A data can be approved at the end of the last day	1	13.35
#2a Are the evaluation questions stated clearly and—? #2b The evaluation questions are not stated clearly.	1	13.30
#3a Cook will ". #3b Will will be cooked.	1	1.370
#4a An audio-visual show about the tower is screened on the first platform; there are restaurants on the first and second, and a bar on the third. #4b There are restaurants on the first and second platforms.	1	13.17
#5a yeah it's a whole whole different culture um it's weird down there because. #5b Places that have different cultures are weird.	1	13.09

(b) Top 5 Easiest Examples

Text	Label	Difficulty
#1a There are delightful wooden figures made as servants for the dead. #1b The wooden figures are supposed to be servants for the dead.	0	–11.49
#2a The merchandise processing fee is associated with the cost of the Customs Service's operations. #2b The merchandise processing fee is associated with the cost of the Customs Service's operations	0	–11.47
#3a The cakes here are also excellent. #3b The cakes here are equally great.	2	–11.41
#4a In the example above, Yellowstone National Park would be reported under a category, such as National Parks, as one of the total number of heritage assets under the auspices of the Department of the Interior; it also would be reported by the number of acres that it occupies under the stewardship land category for the Department. #4b Yellowstone National Park would be reported as a heritage asset.	2	–11.39
#5a Most pubs also serve food—they are good places to have lunch and some have dining rooms. #5b Some pubs have dining rooms, and they are good places to have lunch in.	2	–11.38

Table A.14: Top 5 hardest and easiest examples in MRPC.

(a) Top 5 Hardest Examples

Text	Label	Difficulty
#1a "In relation to the second paper, one part of that should have been sourced to a reference work." #1b He went on: "One part of that should have been sourced to a reference work."	0	10.69
#2a Seven Air Force Academy cadets face punishment for allegedly drinking with two high school girls. #2b Seven Air Force Academy cadets face punishment for drinking with two high school girls in the latest incident to embarrass the academy.	0	10.67
#3a Defense Secretary Donald Rumsfeld is awaiting recommendations from his commanders. #3b Rumsfeld is awaiting recommendations from his commanders about troop needs in Iraq.	0	10.53
#4a Last year, he was forced to repay $3,000 in bar tabs that he and his staff incurred while he was labour minister but had originally billed to taxpayers. #4b Last year, Stockwell was forced to repay $3,000 in bar tabs that he and his staff rang up while he was labour minister.	0	10.53
#5a Most of those killed were labourers from Jharkhand and Nepal who were working at Rohtang tunnel. #5b The majority of the dead were labourers from Jharkhand and Nepal.	0	10.42

(b) Top 5 Easiest Examples

Text	Label	Difficulty
#1a She was surrounded by about 50 women who regret having abortions. #1b She was surrounded by about 50 women who have had abortions but now regret doing so.	1	–8.87
#2a A race observer sits in the passenger seat of the vehicle following the contestant car to record broken rules and track of the car's time. #2b A race observer sits in the passenger seat of the follow vehicle to record any broken rules and also keep track of the car's time.	1	–8.79
#3a She met Lady Mary at her Double Bay home yesterday to thank her for the donation. #3b She met Lady Mary for the first time at her Double Bay home in Sydney yesterday to thank her in person for the donation.	1	–8.64
#4a The national denomination of the Episcopal Church, with 2.3 million members, is the U.S. branch of the 77 million-member Anglican Communion. #4b The Episcopal Church, with 2.3 million members, is the American branch of the worldwide Anglican Communion, which has 77 million adherents.	1	–8.57
#5a It held elections in 1990, but refused to recognize the results after Suu Kyi's party won. #5b It held elections in 1990, but refused to recognise the results when Miss Suu Kyi's National League for Democracy won by a landslide.	0	–8.56

Table A.15: Top 5 hardest and easiest examples in QNLI.

(a) Top 5 Hardest Examples

Text	Label	Difficulty
#1a Europe did not feel the need to posses territory in Africa until? #1b European countries established a few coastal colonies in Africa by the mid-nineteenth century, which included Cape Colony (Great Britain), Angola (Portugal), and Algeria (France), but until the late nineteenth century Europe largely traded with free African states without feeling the need for territorial possession.	1	13.26
#2a What are the windows of 1990s and later pubs often made of? #2b Traditionally the windows of town pubs were of smoked or frosted glass to obscure the clientele from the street but from the 1990s onwards	1	13.20
#3a How many nominations has American Idol received for Outstanding Reality Competition Program? #3b American Idol was nominated for the Emmy's Outstanding Reality Competition Program for nine years but never won.	1	13.19
#4a How many species have been downgraded from endangered to threatened status? #4b As of September 2012, fifty-six species have been delisted; twenty-eight due to recovery, ten due to extinction (seven of which are believed to have been extinct prior to being listed), ten due to changes in taxonomic classification practices, six due to discovery of new populations, one due to an error in the listing rule, and one due to an amendment to the Endangered Species Act specifically requiring the species delisting.	1	13.17
#5a What causes osmotic diarrhea? #5b In healthy individuals, too much magnesium or vitamin C or undigested lactose can produce osmotic diarrhea and distention of the bowel.	1	13.17

(b) Top 5 Easiest Examples

Text	Label	Difficulty
#1a At approximately what time did paramedics receive the call about Kanye West's mother, Donda? #1b On November 10, 2007, at approximately 7:35 pm, paramedics responding to an emergency call transported West's mother, Donda West, to the nearby Centinela Freeman Hospital in Marina del Rey, California.	0	-11.80
#2a A Soviet double agent working for The Times in Spain was a war correspondent during what war in the late 1930s? #2b Kim Philby, a Soviet double agent, was a correspondent for the newspaper in Spain during the Spanish Civil War of the late 1930s.	0	-11.74
#3a Which line was reopened in 2009? #3b In July 2009 the Glounthaune to Midleton line was reopened, with new stations at Carrigtwohill and Midleton (with future stations planned for Kilbarry, Monard, Carrigtwohill West and Blarney).	0	-11.70
#4a How much ad revenue goes to the original uploader of the YouTube video if they're in the Partner Program? #4b YouTube typically takes 45 percent of the advertising revenue from videos in the Partner Program, with 55 percent going to the uploader.	0	-11.69
#5a How does heartwood formation occur due to its being genetically programmed? #5b Heartwood formation occurs spontaneously (it is a genetically programmed process).	0	-11.69

Table A.16: Top 5 hardest and easiest examples in QQP.

(a) Top 5 Hardest Examples

Text	Label	Diff.
#1a What licenses are required to sell agricultural products (viz fertilizers, seeds, pesticides) online in India? #1b I plan to sell agricultural products such as tulsi seeds and spices through eBay India? Do I need special licenses?	1	13.22
#2a Is Zee news a BJP owned channel? #2b What is the problem with Zee News channel?	1	13.22
#3a From my boyfriend's Facebook account, an old girlfriend is not on his friends list anymore, but still shows in messenger as an active friend. Why? #3b She shows up first on his list of people that have Facebook Messenger but they aren't friends and he says they don't talk. Why is that?	1	13.08
#4a What are some of the best ways to market to schools? #4b How do I market in schools?	0	13.06
#5a List the references and study materials for IAS civil service exams? How is this formula determined? #5b Whose study materials is best for philosophy optional in civil service exam—Vision IAS or Mitra Sir?	1	13.02

(b) Top 5 Easiest Examples

Text	Label	Diff.
#1a How do I attach ccavenue payment gateway with button on my page in WordPress? #1b Is it possible to add a link button to a page builder in WordPress?	0	–11.49
#2a Is there any company hiring Indians (MBA + 3 years work experience in finance) for their US/UK location? #2b I am a UK citizen working for a US company on a J1 visa; can I work from UK remotely for 2–4 weeks per year?	0	–11.48
#3a Two moving observers cycling around the equator in an opposite direction at 0.5C. What will they measure when exchanging light signals in each orbit? #3b Find the answer—two trains 120 m and 80 m in length are running in opposite directions with velocities 42 km/h and 30 km/h; at what time will they completely cross each other?	0	–11.41
#4a Do students get a seat under General without fee waiver category if they are allocated seats under General with FeeWaiver category in UPSEE counseling? #4b Is a score of 533 in NEET Phase 1 good enough to get an MBBS seat in a government college in Delhi under state quota for General category?	0	–11.41
#5a I am Sneha from Delhi. Due to family issues, I was not able to study after 7th class. Can I travel or settle abroad? #5b I have a traveling phobia and I'm very attached to my family. How can I get over it so I can go and study abroad next year?	0	–11.35

Table A.17: Top 5 hardest and easiest examples in SST-2.

(a) Top 5 Hardest Examples

Text	Label	Difficulty
#1 a , incoherence and sub-sophomoric	Pos.	11.01
#2 into forced fuzziness	Pos.	10.88
#3 wondering why lee 's character didn't just go to a bank manager and save everyone the misery	Pos.	10.77
#4 sometimes descends into sub-tarantino cuteness	Pos.	10.59
#5 eerily accurate depiction of depression	Neg.	10.43

(b) Top 5 Easiest Examples

Text	Label	Difficulty
#1 useless	Neg.	-10.30
#2 'm giving it thumbs down due to the endlessly repetitive scenes of embarrassment	Neg.	-10.04
#3 the rest is just an overexposed waste of film	Neg.	-9.96
#4 the characters are more deeply thought through than in most ' right-thinking ' films	Pos.	-9.92
#5 this orange has some juice , but it 's far from fresh-squeezed	Neg.	-9.87

Table A.18: Top 5 hardest and easiest examples in RTE.

(a) Top 5 Hardest Examples

#1a IT must rate as the literary snub of the 20th century. T S Eliot, one of Britain's greatest poets, rejected George Orwell's Animal Farm for publication on the grounds of its unconvincing Trotskyite politics. Eliot, a former director of Faber and Faber, wrote his rejection in a highly critical letter in 1944, one of many private papers made available for the first time by his widow Valerie for a BBC documentary. When Orwell submitted his novel, an allegory on Stalin's dictatorship, Eliot praised its "good writing" and "fundamental integrity."
#1b T.S. Eliot wrote "Animal Farm." Label: 1 Difficulty: 10.82
#2a When the last member of the Hasmonean Dynasty died in 37 B.C., Rome made Herod king of Judah. With Roman backing, Herod (37–34 B.C.) ruled on both sides of the Jordan River. After his death the Jewish kingdom was divided among his heirs and gradually absorbed into the Roman Empire.
#2b The Hasmonean Dynasty rules Jordan. Label: 1 Difficulty: 10.61
#3a Over 100 people died in a massive stampede at the Chamunda Devi temple in Jodhpur, in the western state of Rajasthan, India. One hundred sixty-eight people were killed, with just as many injured. A temple wall collapsed, causing panic among tens of thousands of gathered Hindu worshippers who then began to run for safety, causing the stampede. Several are still believed to be trapped under the rubble.
#3b 100 died at Chamunda Devi Temple. Label: 1 Difficulty: 10.58
#4a Diets that provide recommended levels of magnesium are beneficial for bone health, but further investigation on the role of magnesium in bone metabolism and osteoporosis is needed.
#4b Dietary intake of magnesium prevents osteoporosis. Label: 1 Difficulty: 10.56
#5a Gastrointestinal bleeding can happen as an adverse effect of non-steroidal anti-inflammatory drugs such as aspirin or ibuprofen.
#5b Aspirin prevents gastrointestinal bleeding. Label: 1 Difficulty: 10.55

(b) Top 5 Easiest Examples

#1a Nicholas Cage's wife, Alice Kim Cage, gave birth Monday to a boy, Kal-el Coppola Cage, in New York City, said Cage's Los Angeles–based publicist, Annett Wolf. No other details were available.
#1b Nicholas Cage is married to Alice Kim Cage. Label: 0 Difficulty: –10.83
#2a This document declares the "irrevocable determination" of Edward VIII to abdicate. By signing this document on December 10, 1936, he gave up his right to the British throne.
#2b King Edward VIII abdicated on the 10th of December, 1936. Label: 0 Difficulty: –10.46
#3a November 9, 1989 — the day the Berlin Wall fell and the world changed forever. Not even the most astute saw it coming. As Hungary's foreign minister in the late summer of 1989, Gyula Horn gave the order to let visiting East Germans use his country to do a 400-mile end run around the Berlin Wall, a move now seen as the beginning of the end for hard-line communism in Europe.
#3b The Berlin Wall was torn down in 1989. Label: 0 Difficulty: –10.43
#4a The floods were exceptional since they affected an extensive area across Europe from the UK to Spain and as far east as the Black Sea coast. Economic losses amounted to EUR 9.2 bn in Germany, EUR 2.9 bn in Austria, and EUR 2.3 bn in the Czech Republic. Total economic damage exceeds EUR 15 bn.
#4b Flooding in Europe causes major economic losses. Label: 0 Difficulty: –10.38
#5a Bush's second term as President of the United States, which began on January 20, 2005, expired with the swearing-in of the 44th President of the United States, Barack Obama, at noon EST (UTC–5), under the provisions of the Twentieth Amendment to the United States Constitution. Bush performed his final official act this morning, welcoming Barack Obama and Michelle to the White House for coffee before the swearing-in, shortly before 10 am EST, and then accompanied them there by motorcade to attend the ceremony. Last week, Bush had made his farewells to the nation in a televised address, saying that the inauguration turns a page in race relations. "Obama's story — his black father was from Kenya, his white mother from Kansas — represents the enduring promise of our land," said Bush.
#5b Barack Obama is the 44th President of the United States. Label: 0 Difficulty: –10.19