CDT: A Comprehensive Capability Framework for Large Language Models Across Cognition, Domain, and Task

Haosi Mo[1], **Xinyu Ma**[1], **Xuebo Liu**[1]*, **Derek F. Wong**[2], **Yu Li**[3], **Jie Liu**[4], **and Min Zhang**[1]
[1]Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, China
[2]NLP^{2}CT Lab, Department of Computer and Information Science, University of Macau
[3]School of Integrated Circuits, Zhejiang University, Hangzhou, China
[4]State Key Lab of Smart Farm Technologies and Systems, Harbin Institute of Technology
{alessamo0411,mxinyuma,yu.li.sallylee}@gmail.com, derekfw@um.edu.mo
{liuxuebo,jieliu,zhangmin2021}@hit.edu.cn

Abstract

Recent advances in Large Language Models (LLMs) have significantly enhanced their capabilities, highlighting the need for comprehensive evaluation frameworks that extend beyond task-specific benchmarks. However, existing benchmarks often focus on isolated abilities, lacking a holistic framework for assessing LLM capabilities. To address this gap, we propose the **C**ognition-**D**omain-**T**ask (CDT) framework, which comprehensively measures a model's capabilities across three dimensions. We expand the scope of model capability definitions at the cognitive level by incorporating the Cattell-Horn-Carroll cognitive theory, refining the categorization of model capabilities. We apply CDT in two directions: dataset capability evaluation and data selection. Experiments show that our capability metrics correlate well with downstream performance and can support effective dataset analysis and construction. The experiments on data selection also show significant improvements in both general and specific benchmarks, achieving scores of 44.3 and 45.4, with an increase of 1.6 and 2.2 points over the baselines, respectively. These results validate the effectiveness and practicality of CDT. Source code and models are available at https://github.com/Alessa-mo/CDT.

Framework	Open Source Tagging Models	Multiple Dimensions	Capability Decomposition	Cognition Oriented	Domain Oriented	Task Oriented
FLASK	✗	✓	✗	✓	✓	✗
FAC^{2}E	✗	✓	✓	✓	✗	✓
INSTAG	✓	✗	✗	✗	✓	✗
CDT (Ours)	✓	✓	✓	✓	✓	✓

Table 1: Comparison between our CDT framework with existing capability frameworks. "Open Source Tagging Models" denotes if it provides trained models for capability annotation. "Multiple Dimensions" reflects whether the framework supports more than one capability dimension. "Capability Decomposition" refers to the ability to break down complex capabilities into finer-grained sub-skills. The last three columns assess whether the framework explicitly covers cognition, domain, and task dimensions. As shown, our CDT framework addresses the gaps and limitations of existing methods across multiple dimensions.

1 Introduction

Recent advances in Large Language Models (LLMs) have significantly expanded their capabilities. The introduction of reinforcement learning (Kumar et al., 2024; Wang et al., 2024a; Hu et al., 2023) and chain-of-thought reasoning (Wei et al., 2022; Wang et al., 2023a) has further enhanced their reasoning abilities. Notable LLMs such as OpenAI's o1 (OpenAI, 2024b) and DeepSeek R1 (DeepSeek-AI, 2025) have demonstrated remarkable reasoning capabilities. As LLMs become more sophisticated, accurately evaluating their underlying abilities is increasingly crucial. Current benchmarks, such as MMLU (Hendrycks et al., 2021), AlpacaEval (Dubois et al., 2024), and GSM8K (Cobbe et al., 2021), are widely used to assess these capabilities.

However, many of them focus on isolated aspects of model capabilities, such as coding, commonsense reasoning, or specific task performance, and the ability dimensions are always task-oriented and limited, without a holistic framework that systematically categorizes and defines the full spectrum of LLM capabilities. For instance, benchmarks like MMLU evaluate knowledge mastery across academic disciplines but overlook dimensions like code generation. Recent efforts like FLASK (Ye et al., 2024) and FAC^{2}E (Wang et al., 2024b) focus on multi-model comparisons but fall short in capability decomposition and multi-dimensional analysis. Additionally, while works like INSTAG (Lu et al., 2024) explore capability applications, definitions remain underdeveloped. Those works raise the fundamental question: *What core capabilities are essential for an effective large language model?*

* Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 3715–3734
November 4-9, 2025

To address this, we propose the **C**ognition-**D**omain-**T**ask (CDT) framework, a comprehensive multi-dimensional taxonomy for defining, annotating, and utilizing LLM capabilities across three dimensions: cognition, domain, and task. Our core motivation is that a comprehensive capability analysis must answer three fundamental questions for any given instruction: how to do it, which corresponds to Cognition; what it is about, which corresponds to Domain; and what to do, which corresponds to Task. By deconstructing instructions along these three orthogonal dimensions, CDT provides a holistic and systematic approach to categorizing the full spectrum of LLM capabilities. At the cognitive level, CDT incorporates Cattell-Horn-Carroll (CHC) theory (Flanagan and Dixon, 2014), selecting and refining 18 core cognitive abilities suited to LLM behavior. At the domain level, we identify nine domain scenarios commonly encountered by LLMs and further refine these into 33 distinct subdomains. At the task level, drawing inspiration from prior work on dataset construction (Wang et al., 2022, 2023b; Ouyang et al., 2022; Wang et al., 2024c), we systematically categorize task types across diverse instructions, culminating in a taxonomy of 16 task types. We conduct a comparative analysis between existing capability frameworks and our proposed CDT framework, with the results summarized in Table 1.

After constructing the CDT framework, we extend its application to LLMs in two directions. We first conduct dataset evaluation using two metrics: `Coverage` and `Balance`, which correlate well with downstream performance, demonstrating that CDT can provide practical guidelines for capability-aware data curation in future dataset construction. Then, we apply CDT in data selection to enhance model performance, proposing a diversity-driven selection method to ensure general capability and a capability-oriented strategy which identifies the specific capabilities required by the target test sets. Across both general and specific scenarios, experiments show that our data selection methods achieve average scores of 44.3 and 45.4, with an increase of 1.6 and 2.2 points over the baselines, respectively. These results significantly outperform other capability-related methods and baseline approaches. Our **main contributions** are as follows:

- We propose CDT, a comprehensive framework that systematically categorizes LLMs' abilities across cognition, domain, and task.
- We develop specialized tag models for each dimension to enable fine-grained tagging of capacities at the instruction level.
- We explore the application of the CDT framework in dataset evaluation and data selection, which effectively reflect dataset quality and lead to notable improvements in model performance.
- We will release all the data, tag models, and training scripts used in our CDT framework.

2 Related Works

Definitions of LLMs' Capability Research on defining LLM capabilities can be grouped into two approaches. The first integrates capabilities with data, optimizing learning through data distribution adjustments (Nottingham et al., 2024; Polo et al., 2025; Chen et al., 2023; Wu et al., 2024; Rao et al., 2024; Lu et al., 2024). For example, Chen et al. (2023) propose a method based on a skill set graph, where mastering one skill aids the acquisition of others, though this method is dataset-specific. Similarly, Wu et al. (2024) use an MLP-based scoring network for data allocation in fine-tuning, treating datasets as distinct capabilities. The second approach defines model capabilities from task- and domain-specific perspectives, often relying on labeled data for evaluation. Zhong et al. (2025) present a hierarchical framework with foundational and complex abilities, while Ye et al. (2024) analyze open-source LLMs to identify four capabilities, subdividing them into 12 skills for comprehensive evaluation. While these approaches offer valuable insights, they often define capabilities in narrow ways, either focusing on isolated aspects, overlooking the underlying cognitive processes, or lacking a holistic, multi-dimensional structure. Our work addresses this gap by introducing the CDT framework, which integrates cognitive principles to systematically organize LLM capabilities across cognition, domain, and task.

Applications of LLMs' Capability Capability frameworks are often applied to develop evaluation benchmarks for large models. Xia et al. (2024) introduce FoFo, which evaluates LLMs' abilities across domains based on format-following. For general evaluation, Hendrycks et al. (2021), Dubois et al. (2024), and Srivastava et al. (2022) have developed broad competency benchmarks. Zhong et al. (2025) assess capabilities using prompts, while Ye et al. (2024) evaluate models based on responses

and instruction alignment. For domain-specific improvements, several studies have proposed different approaches: Wang et al. (2024d) integrate capability frameworks with Chain-of-Thought (CoT) to enhance task-specific abilities; Lee et al. (2024) introduce THANOS for multi-turn dialogue; Xu et al. (2023) present LaRS to improve reasoning by selecting data with similar capabilities; Rao et al. (2025) focus on enhancing weak capabilities through error-based learning; and Ke et al. (2025) aim to build specialist capabilities by synthesizing high-relevance data from unlabeled text. While prior studies have extended capability frameworks in certain contexts, most still focus on evaluating LLM capabilities, with limited exploration of broader applications. To address this, we apply the proposed CDT framework to scenarios such as dataset analysis and data selection, thereby extending its utility beyond conventional evaluation.

3 Method

3.1 Capability Framework Construction

In our proposed CDT framework, we define model capabilities from three perspectives: cognition, domain, and task. The three dimensions are designed to be orthogonal, allowing for a context-dependent analysis of full instructions. A detailed discussion of the framework's design rationale is provided in Appendix A.1. While the domain and task perspectives have been extensively explored in recent research, we build upon this foundation with adjustments to better capture their nuances. From the cognition perspective, we define capabilities through the lens of the CHC theory in cognitive science. The CHC theory, grounded in earlier explorations of human cognition (Carroll, 2003; Cattell, 1963; Horn, 1965; Flanagan et al., 2000), serves as a foundational model in cognitive science (McGrew and Evans, 2004). In the realm of computer science, numerous studies have demonstrated the critical role of cognitive capabilities in LLMs and artificial intelligence (Zhao et al., 2022; Lieto et al., 2018; Song et al., 2024). Our overall capability framework is shown in Figure 1.

Cognition The CHC theory categorizes human cognitive abilities into three hierarchical levels. Stratum I consists of "narrow" abilities, which represent specialized skills developed through experience, learning, or the application of targeted methodologies (Carroll, 1993). Stratum II encompasses "broad" abilities, which are more abstract and general in nature. Stratum III represents the highest level, with a single general cognitive ability acting as an overarching factor. In our framework, we focus exclusively on the Stratum I abilities defined by Flanagan and Dixon (2014), as they provide specific abilities and detailed definitions that are more directly applicable than those found in the other two levels. The process of constructing LLM cognitive capabilities follows these steps:

Figure 1: The model capability framework we define, where the blue section represents the Cognition dimension, the green section represents the Domain dimension, and the brown section represents the Task dimension. The shaded region is used to visually emphasize our CDT framework.

- **Cognition Construction**: To align with the linguistic focus of LLMs, we begin by filtering the cognitive abilities defined by CHC, which span multiple human modalities such as vision, hearing, and speech. We exclude non-linguistic abilities and domain-specific knowledge, as domain expertise is addressed separately in our framework. We also remove skills that are essential for humans but not as crucial for models, such as memory-related abilities. While this CHC-based foundation is robust, it may still overlook certain skills exhibited by LLMs. To address this, we augment the set with capabilities particularly relevant to LLMs, such as logical analysis, abstract coding concepts, and problem decomposition. After this process, the number of abilities is reduced from 82 to 16.

- **Definition Refinement**: To better align with

language models, we refine certain ability definitions. Notably, the ability Induction, originally defined as "the ability to discover the underlying characteristic (e.g., rule, concept, process, trend, class membership) that governs a problem or a set of materials," often leads to ambiguity in capability tagging. Its broad and abstract nature makes it frequently assigned across diverse instructions. To address this, we subdivide it into three specific capabilities: pattern recognition, concept abstraction, and hypothesis generation. After these refinements, the total number of cognitive capabilities is $N_c = 18$, and we define the cognition dimension as $\mathcal{C} = \{c_i\}_{i=1}^{N_c}$, where each c_i denotes a specific cognitive capability. The detailed cognitive capability construction procedure is provided in Appendix A.2.

Domain Based on Ye et al. (2024), which categorizes 38 domains, we construct the domain dimension of our framework. However, we observe that certain domains, such as business and marketing, exhibit considerable similarity, potentially introducing ambiguity in capability tagging models and leading to label distribution dispersion in the following process of annotating capabilities. So, we manually refine the domain set. First, we merge similar domains into one domain to reduce ambiguity. Then, we expand the domain coverage by adding underrepresented domains such as earth science and tradition. After refinement, the total number of domains is $N_d = 33$, and the domain dimension is defined as $\mathcal{D} = \{d_i\}_{i=1}^{N_d}$, where each d_i represents a specific categorized domain.

Task For task categorization, we draw inspiration from Wang et al. (2024c), who classify the 76 tasks in SuperNI (Wang et al., 2022) into 16 taxonomies, as well as from related work such as Bach et al. (2022) and Ouyang et al. (2022), which offer widely accepted, fine-grained, and comprehensive categorizations. Taking task granularity and completeness into account, we ultimately categorize $N_t = 16$ task taxonomies. For task definition, we synthesize information from Wikipedia and prior work (Ding et al., 2023) to formulate detailed definitions for each task. The task dimension is defined as $\mathcal{T} = \{t_i\}_{i=1}^{N_t}$, where t_i is the task we define.

Finally, the whole capability framework $\mathcal{F}$ is:

$$\mathcal{F} = \{(c, d, t) \mid c \in \mathcal{C},\ d \in \mathcal{D},\ t \in \mathcal{T}\} \quad (1)$$

Details on the categorization and definitions of each capability are provided in Appendix A.3.

3.2 Capability Tagging Model Training

To facilitate the practical use of our framework, we train a capability tagging model for each dimension. We first prompt GPT-4o (OpenAI, 2024a) to annotate fine-grained capability labels for each instruction in the training data due to its exceptional comprehension abilities. Given the importance of cognitive abilities in human intelligence, each data point is annotated with up to two cognitive capabilities, and one tag for both domain and task.

We construct a training set of 49K samples from seven public instruction datasets, with 1K held out as a test set. Then we use the annotated dataset to fine-tune three annotators on the Qwen2.5-7B-Base (Team, 2024) model. To validate the performance of the trained annotators, we use the GPT-generated labels as the ground truth and evaluate the models on the test set. The accuracy rates for cognition, domain, and task tags are 93.1%, 81.2%, and 80.9%, respectively, with an average score of 85.1%, supporting the validity and reliability of the CDT tagging system. Further training details, datasets, prompt designs, human evaluation results, and a cost-benefit analysis of our annotation strategy are provided in Appendix A.4.

4 CDT for Dataset Evaluation and Data Selection

While the CDT framework offers a comprehensive definition of model capabilities, its application to LLMs remains an area requiring further exploration. Leveraging CDT's ability to classify data at the instruction level based on capabilities, we focus on two key application scenarios: evaluating the capability characteristics of existing instruction datasets and guiding the selection of training data to enhance model performance.

4.1 Capability-Aware Dataset Evaluation

To understand the quality and capability distribution of existing instruction datasets, and thereby guide future dataset construction more effectively, we introduce a capability-aware evaluation approach based on CDT. Given a labeled instruction dataset D_i where each instance is annotated with composite capability triplet (c, d, t), we then define the capability composites within D_i as T_i.

$$T_i = \text{Composites}(D_i) \quad (2)$$

where Composites means getting all the capability composites in a given labeled dataset. We then

define two quantitative metrics for dataset-level capability assessment: `Coverage` and `Balance`.

- **`Coverage`** measures how many distinct capability composites the dataset contains relative to the full capability space, defined as $\texttt{Coverage} = |T_i|/|\mathcal{F}|$.

- **`Balance`** reflects the uniformity of the distribution over composite capabilities in the dataset. It is computed as the Shannon entropy: $\texttt{Balance} = -\sum_{t_i \in T_i} p(t_i) \log p(t_i)$, where $p(t_i)$ is the empirical probability of composite triplet t_i in dataset D_i.

A higher `Coverage` indicates broader capability representation. This concept aligns with existing research. For example, INSTAG defines a similar metric, referred to as the unique tag coverage rate for the overall tag set, emphasizing the importance of diverse capability representation. Similarly, research by Zhang et al. (2024) explicitly states that the diversity of the instruction set largely determines generalization to unseen tasks, underscoring the critical role of diversity in enabling performance on novel tasks. It also points out that the uneven distribution within the training set can affect generalization ability, which in turn leads to our definition of the `Balance` metric. A higher `Balance` reflects a more uniform distribution across capabilities. This observation is echoed in other studies that stress the importance of data balance for robust model training (Kandpal et al., 2023; Shao et al., 2024). Both of the metrics are desirable for building generalizable models. We employ these metrics in Section 5 to evaluate a range of popular instruction datasets.

4.2 Capability-Guided Data Selection

Beyond supporting capability evaluation and analysis, CDT can also serve practical purposes in downstream applications. To demonstrate its effectiveness, we apply CDT to data selection scenarios for LLM instruction fine-tuning. This approach enables the systematic enhancement of training data quality and relevance, ultimately improving LLM performance on downstream tasks.

Prior to implementing the data selection process, we first annotate the collected data pool D_{pool} using the CDT framework to ensure precise capability-based categorization, resulting in the labeled dataset $D^{'}_{pool}$. As in the previous section, we define the capability composites within $D^{'}_{pool}$ as:

$$T_d = \text{Composites}(D^{'}_{pool}) \tag{3}$$

We then explore two practical strategies under this framework: a diversity-driven selection method to improve general capability coverage, and a capability-oriented filtering method to support specific scenario enhancement.

Diversity-Driven General Scenario Data Selection When training LLMs, data diversity plays a crucial role in enhancing model performance and generalization (Miranda et al., 2024; Zhou et al., 2023). Therefore, we propose a diversity-driven general data selection method based on CDT. Firstly, we define the selected training dataset as D_{train} and the composite capability assigned to D_{train} as T_s.

$$T_s = \text{Composites}(D_{train}) \tag{4}$$

For diversity-driven applications, our goal is to enlarge T_s as much as possible. Then we define a threshold R, which denotes the ratio of T_s to T_d. We quantify the attribute diversity as $R = |T_s|/|T_d|$, where $|\cdot|$ denotes the cardinality (i.e., the number of elements) of a set. The value of R reflects the coverage rate of unique composite capabilities within the selected sub-dataset relative to the entire data pool. Our selection criterion aims to maximize the proximity of R to 1. Based on this, if a data point $d \in D_{pool}$ could increase R, we add the composite of d to T_s and d itself to D_{train} as training data. When R can no longer be increased, we perform an average selection from D_{pool} to fill the gaps in the capability composite of T_s.

Capability-Oriented Specific Scenario Data Selection When applying the capability framework in the capability-oriented specific scenario, we first label the validation set of the test task to obtain the labeled dataset D_{valid}. Then, we tag D_{valid} with our annotators to form $D^{'}_{valid}$ and use the same method as in the diversity-driven approach to extract all combinations of abilities T_v from $D^{'}_{valid}$.

$$T_v = \text{Composites}(D^{'}_{valid}) \tag{5}$$

We aim to perform an average selection of the data from $D^{'}_{pool}$ based on the combinations of capabilities in T_v. However, in practice, T_v may be limited to a small subset of combinations of capabilities, and the amount of data corresponding to these

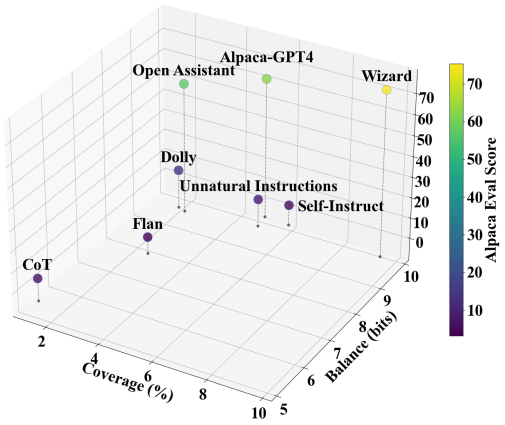


Figure 2: Open-source instruction datasets analysis based on their capability Coverage and Balance. The Z-axis and the point color both indicate AlpacaEval score, with brighter colors corresponding to higher performance.

combinations in D'_{pool} may not be sufficient to support our selection. To address this issue, we further decompose the capabilities in T_v. Specifically, we break down the triplet of capabilities $f = (c, d, t)$ into binary pairs $(c, d), (c, t), (d, t)$, creating a binary set T_v^*, and further into individual dimensions $(c), (d), (t)$, forming a unary set $T_v^\star$. When the triplet set T_v does not yield enough data, we first perform random selection on T_v^*, followed by selection on $T_v^\star$ in successive stages. This approach ensures sufficient data collection while preserving the concentration of capabilities. We present the details of the two algorithms in Appendix A.5.

5 Empirical Analysis of Instruction Dataset Capabilities

Datasets We conduct an empirical analysis on eight publicly available instruction datasets widely used in LLM instruction tuning. These datasets include: Chain of Thought (Wei et al., 2022), Dolly (Conover et al., 2023), Open Assistant (Köpf et al., 2023), Flan V2 (Longpre et al., 2023), WizardLM (Xu et al., 2024), Alpaca-GPT4 (Peng et al., 2023), Self-Instruct (Wang et al., 2023b), and Unnatural Instructions (Honovich et al., 2023). Each dataset is annotated using our capability taggers to extract the composite tuples. We then compute the metrics introduced in Section 4.1. Following Lu et al. (2024), we collect the AlpacaEval (Li et al., 2023) score for model performance.

Analysis Figure 2 visualizes the relationship between Coverage, Balance, and AlpacaEval scores for these datasets. As shown, there is a positive correlation between the two metrics and the model performance. Datasets achieving higher scores on both Coverage and Balance generally yield models with superior AlpacaEval scores. Notably, top-performing datasets such as Wizard, Alpaca-GPT4, and Open Assistant are positioned in the upper-right region of the plot, indicating high values for both our proposed metrics. Interestingly, Open Assistant, despite a moderate Coverage score, achieves a strong AlpacaEval score, potentially due to its exceptional Balance score, highlighting the crucial impact of data balance. Conversely, datasets like CoT and Flan, which score lower on these two quality indicators, correspondingly result in models with lower AlpacaEval scores.

Performance Variance We also observe that datasets with similar quantitative scores may lead to very different model performance. For example, Alpaca, Unnatural Instructions and Self-Instruct show similar metrics, but differ widely in effectiveness. We attribute this to the quality of the response annotations: Alpaca uses GPT-4-generated outputs, while the latter two rely on earlier models such as text-davinci-002/003, which are substantially weaker. Since our capability taggers operate on the instruction side, they may overlook differences in response quality, resulting in the observed discrepancy. We believe this also explains the performance gap between Open Assistant and Dolly. Although both are human-annotated, Open Assistant relies on global crowdsourcing, which results in higher-quality responses compared to Dolly, whose annotations come from Databricks employees.

Our empirical analysis demonstrates that Coverage and Balance are effective indicators of dataset quality, and that a combination of comprehensive Coverage and well-distributed Balance is crucial for training models with high performance. These findings suggest that CDT can serve as practical guidelines for capability-aware data curation in future dataset construction.

6 Data Selection Experiments

6.1 Experiment Setup

Data Pool and Base Model To evaluate and apply our proposed capability framework, CDT, across both diversity-driven general scenario and capability-oriented specific scenario, we utilize the following datasets: (1) Aggregated high-quality datasets, including Flan V2 and CoT; and (2) Open-ended generation datasets with human-annotated

Methods	ARC-C	MMLU	BBH	CEVAL	TYDIQA	AVG.
Baselines						
Base	43.5	45.2	**41.6**	31.9	47.8	42.0
All	44.5	45.9	39.6	35.6	**53.3**	43.8
Random	45.0	45.5	39.8	32.9	50.4	42.7
InsTag	44.8	45.8	39.3	33.2	51.9	43.0
Our Methods						
CDT_Cognition	45.3	45.3	38.2	36.6	51.9	43.5
CDT_Domain	45.9	46.1	38.5	34.3	52.2	43.4
CDT_Task	45.7	46.1	39.3	35.9	50.5	43.5
CDT	**46.1**	**46.3**	38.8	**36.9**	53.2	**44.3**

Table 2: Results of applying CDT in diversity-driven general data selection, using 20% of the data pool for training. **Bold** indicating the best performance and underline indicating the second-best performance.

responses, such as Dolly and Open Assistant. From these four datasets, we compile a pool of approximately 270K data points. Since our annotators are trained using Qwen2.5-7B[1], we select Llama2-7B-Base[2] as the base model to mitigate any potential bias between the tagging model and the experimental model. We use open-instruct[3] and lm-eval (Gao et al., 2024a) for all tests.

Baselines We conduct the following experiments for comprehensive comparison:

- **Base**: We evaluate the pre-trained Llama2-7B-Base model on the benchmarks.
- **ALL**: We train the Llama2-7B-Base model using all the data from the data pool.
- **Random**: We randomly sample data from the data pool to train the Llama2-7B base model.
- **INSTAG**: **(1)** For the diversity-driven general scenario, we adopt the diversity approach outlined by Lu et al. (2024), utilizing their annotation model to label the training data.**(2)** For the capability-oriented specific scenario, we use only the INSTAG annotator for tag labeling. We then average the sample data from the data pool based on the capabilities tagged in the valid set.

Configuration We fine-tune the Llama2-7B-Base model using Low-Rank Adaptation (LoRA) (Hu et al., 2022), specifically targeting the attention module. Distributed training is conducted using DeepSpeed (Rasley et al., 2020). During training, the maximum sequence length is set to 2048, with a batch size of 64 and training epochs as 3.

[1] https://huggingface.co/Qwen/Qwen2.5-7B
[2] https://huggingface.co/meta-llama/Llama-2-7b
[3] https://github.com/allenai/open-instruct

6.2 Experiments on the General Scenario

We first apply CDT to data selection in the diversity-driven general scenario. By extracting capability distributions from the data pool, we select diverse training data and evaluate performance on Llama2-7B-Base. Additional results on Mistral-7B-Base are provided in Appendix A.6 to demonstrate the generalizability of our method.

Benchmarks We conduct experiments using the following benchmarks: **ARC-C** (Clark et al., 2018): We use the Challenge portion for testing, with accuracy as the evaluation metric. **MMLU** (Hendrycks et al., 2021): We report the average accuracy score under 5-shot settings. **BBH** (Srivastava et al., 2022): We use the CoT prompt and report the accuracy score. **C-Eval** (Huang et al., 2023): We use accuracy on 5-shot as the evaluation metric. **TyDiQA** (Clark et al., 2020): We use the GoldP task and report the average F1 score under 1-shot settings.

Results As shown in Table 2, our method achieves the best overall performance, ranking first on most benchmarks. For BBH, all the methods exhibit performance degradation compared to the base model. We hypothesize that this may be due to a certain degree of overlap between the fine-tuning data and the model's training data, which compromises the model's ability to generalize to complex reasoning tasks. Furthermore, since our method considers the capabilities from three dimensions, we also conduct separate experiments for each dimension. Notably, even when using a single capability dimension, our method consistently outperforms Random and INSTAG. Among these, the approach that jointly considers all three dimensions outperforms those that consider only a single dimension across the majority of evaluation metrics. These results highlight the accuracy of our CDT framework in defining capabilities and demonstrate the effectiveness of our diversity-driven data selection method in practice.

Impact of Data Volume on CDT Performance We conduct experiments by selecting 5%, 20%, and 40% of the data from the overall data pool. The results are presented in Table 3. Using 20% of the data, our method, CDT, yields the best performance compared to other volumes. However, even at these data volumes, our CDT data selection methods still outperform INSTAG in all cases. These results highlight the robustness and stabil-

Volume	Methods	ARC-C	BBH	MMLU	CEVAL	TYDIQA	AVG.
5%	INSTAG	44.3	38.3	44.4	32.1	49.4	41.7
	CDT	45.6	**39.4**	45.7	32.7	50.1	42.7
20%	INSTAG	44.8	39.3	45.8	33.2	51.9	43.0
	CDT	**46.1**	38.8	46.3	**36.9**	**53.2**	**44.3**
40%	INSTAG	45.2	**39.4**	46.3	33.7	51.5	43.2
	CDT	45.1	38.1	**46.7**	**36.9**	51.6	43.7

Table 3: The results of our method across different data selection volumes and our approach achieve the optimal results at 20%. The results are presented with **bold** indicating the best performance and underline indicating the second-best performance.

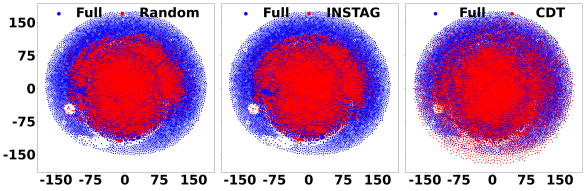


Figure 3: Diversity analysis using t-SNE on the data selected by Random, Instag, and CDT. Blue dots represent the distribution of all data, while red dots indicate the distribution of data selected by different methods.

ity of our approach across different data volumes. Based on these findings, we choose to use the 20% data configuration for the remaining experiments.

Comparison of Data Diversity Across Methods In the diversity-driven general scenario, training data diversity is critical to model performance. To assess the effectiveness of CDT, we analyze the diversity of data selected by Random, INSTAG, and CDT. Following Gao et al. (2024b), we use Llama2-7B-Chat to extract data representations and apply t-SNE for visualization. As shown in Figure 3, the red points of CDT are more widely dispersed than those of INSTAG and Random, indicating that the data selected by the CDT method exhibits greater diversity. This advantage in data diversity aligns with the performance improvements observed in our benchmark tests, explaining why CDT outperforms other methods in the diversity-driven general scenario. It further reinforces the rationale behind the capability definitions in our CDT framework.

6.3 Experiments on the Specific Scenario

After validating CDT in the general scenario, we further test its effectiveness in the capability-oriented specific scenario, where models require data tailored to specific capabilities. Using the method introduced in Section 4.2, we conduct a detailed analysis of three target test sets.

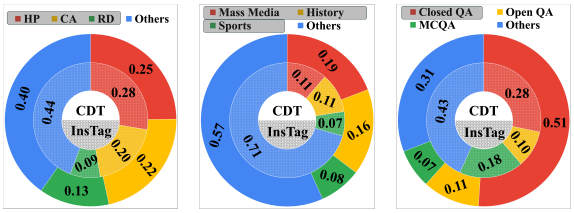
Methods	DROP		GSM	HISTORY	AVG.
	EM	F1	EM	Acc.	
Base	0.0	1.3	14.5	51.0	16.7
All	49.0	**58.3**	21.0	51.3	44.9
Random	46.7	55.8	19.0	51.2	43.2
InsTag	47.9	57.2	19.0	52.4	44.1
CDT	**49.3**	**58.3**	**21.5**	**52.5**	**45.4**

Table 4: The results of using CDT for data selection in the capability-oriented specific scenario, using 20% of the data pool for training. The best performance is marked in **bold**, and the second-best is marked with an underline. Our method achieves the highest performance across all three test sets.

Test Datasets We conduct experiments using the following test datasets: DROP (Dua et al., 2019), GSM (Cobbe et al., 2021), and HISTORY, where HISTORY is a resampled subset of four history-related tasks from the MMLU benchmark. To align with the application method proposed in Section 4.2, we select a maximum of 200 samples from the validation set of each task for tagging and data selection. For datasets that do not include a validation set, we randomly split 200 samples from test set to form one. If a dataset contains fewer than 200 samples, we use the full validation set available.

Result As shown in Table 4, CDT consistently achieves the highest performance across all test sets. While full-data training approaches achieve similar results on DROP and GSM, our approach attains better results using only 20% of the full dataset, demonstrating significantly improved data efficiency. Furthermore, on the HISTORY test set, the full-data baseline performs similarly to Random, yet remains 1.2 points below our approach. These results highlight the exceptional performance of CDT in capability-oriented specific scenario, demonstrating its effectiveness.

Reasonability of Selected Data To further compare CDT with INSTAG, we use DROP as the targeted test set and analyze capability distributions by comparing the data selected by the INSTAG method with the tags annotated by CDT. As shown in Figure 4, CDT consistently aligns more closely with the target test set across the three dimensions. In cognition, both methods focus on key capabilities like HP (Hypothesis Generation), CA (Concept Abstraction), and RD (Reading Decoding). Although INSTAG shows a slightly higher distribution in the HP capability, CDT surpasses it in both CA and RD capabilities, demonstrating a



(a) Cognition (b) Domain (c) Task

Figure 4: A comparison of the capability distributions of the data selected for the DROP test set using the CDT and INSTAG methods. The gray areas in the figure represent the capabilities required by DROP. MCQA stands for Multiple Choice QA.

more concentrated and dominant distribution. In the domain dimension, CDT selects substantially more data from relevant areas like history, sports, and mass media, which are central to the construction of DROP, as it prioritizes articles from these areas to support complex question generation. Regarding task dimension, CDT better captures the Closed QA capability, with a higher proportion of targeted samples. These results confirm that CDT effectively identifies and prioritizes the capabilities needed for the target test, thereby validating the strength and reliability of our framework.

7 Conclusion

In this work, we introduce the Cognition-Domain-Task (CDT) capability framework, offering a comprehensive and systematic approach to classify and decompose the capabilities of LLMs. By defining cognitive abilities based on Cattell-Horn-Carroll (CHC) theory and organizing domain and task capabilities into a structured taxonomy, we enable more nuanced categorization of LLM capabilities across various scenarios. Additionally, we trained a high-quality annotator on the Qwen2.5 model using the CDT framework.

We demonstrate the utility of CDT in two key applications. We first apply CDT to dataset evaluation using `Coverage` and `Balance` metrics to assess capability diversity and distribution. We then propose diversity-driven and capability-oriented data selection methods, both of which lead to substantial performance gains across multiple benchmarks. These results confirm the stability and effectiveness of CDT in guiding both dataset evaluation and data selection, highlighting the robustness and practical applicability of the framework.

Limitations

Our method constructs a detailed three-dimensional LLM capability framework, CDT, and explores its application in dataset evaluation and data selection. However, there are still some limitations.

First, although the annotator trained on the Qwen-2.5 model achieves higher labeling accuracy across the three dimensions compared to INSTAG, there is still significant room for improvement. This could be addressed by adding more training data or incorporating specific knowledge from human experts to guide more accurate annotator training.

Second, when defining the three dimensions, we filter out multimodal capabilities, limiting the applicability of the CDT framework to a broader range of multimodal models. Future research could expand CDT to include relevant multimodal capability classifications and conduct experiments on multimodal models such as Qwen-VL (Bai et al., 2023) and Llama-3.2 (Grattafiori et al., 2024).

Lastly, in our application of the CDT framework to LLMs, we have only explored its application in two scenarios. Future research may benefit from combining curriculum learning methods, such as Regmix (Liu et al., 2024b), with the CDT framework to dynamically adjust data distribution during training, potentially leading to even better results.

Acknowledgments

This work was supported in part by Guangdong S&T Program (Grant No. 2024B0101050003), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011491), and Shenzhen Science and Technology Program (Grant Nos. ZDSYS20230626091203008, KJZD20231023094700001, KQTD20240729102154066). Derek F. Wong was supported in part by the UM and UMDF (Grant Nos. MYRG-GRG2023-00006-FST-UMDF, MYRG-GRG2024-00165-FST-UMDF, EF2024-00185-FST). We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions.

References

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. 2022. PromptSource: An integrated development environment and repository for natural

language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

John B Carroll. 1993. *Human cognitive abilities: A survey of factor-analytic studies*. Cambridge University Press.

John B Carroll. 2003. The higher-stratum structure of cognitive abilities: Current evidence supports g and about ten broad factors. *The scientific study of general intelligence*, pages 5–21.

Raymond B Cattell. 1963. Theory of fluid and crystallized intelligence: A critical experiment. *Journal of educational psychology*, 54(1):1.

Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023. Skill-it! A data-driven skills framework for understanding and training language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, 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*.

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

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Dawn P. Flanagan and Shauna G. Dixon. 2014. *The Cattell-Horn-Carroll Theory of Cognitive Abilities*. John Wiley and Sons, Ltd.

Dawn P Flanagan, Kevin S McGrew, and Samuel O Ortiz. 2000. *The Wechsler Intelligence Scales and Gf-Gc theory: A contemporary approach to interpretation.* Allyn & Bacon.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, et al. 2024a. A framework for few-shot language model evaluation.

Pengzhi Gao, Zhongjun He, Hua Wu, and Haifeng Wang. 2024b. Towards boosting many-to-many multilingual machine translation with large language models. *CoRR*, abs/2401.05861.

Aaron Grattafiori, 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*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

John Leonard Horn. 1965. *Fluid and crystallized intelligence: A factor analytic study of the structure among primary mental abilities*. University of Illinois at Urbana-Champaign.

Bin Hu, Chenyang Zhao, Pu Zhang, Zihao Zhou, Yuanhang Yang, Zenglin Xu, and Bin Liu. 2023. Enabling intelligent interactions between an agent and an llm: A reinforcement learning approach. *CoRR*, abs/2306.03604.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Xiaopeng Ke, Hexuan Deng, Xuebo Liu, Jun Rao, Zhenxi Song, Jun Yu, and Min Zhang. 2025. Aquilt: Weaving logic and self-inspection into low-cost, high-relevance data synthesis for specialist llms. *arXiv preprint arXiv:2507.18584*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations - democratizing large language model alignment. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2025. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Young-Jun Lee, Dokyong Lee, Junyoung Youn, Kyeongjin Oh, and Ho-Jin Choi. 2024. Thanos: Enhancing conversational agents with skill-of-mind-infused large language model. *arXiv preprint arXiv:2411.04496*.

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

Antonio Lieto, Mehul Bhatt, Alessandro Oltramari, and David Vernon. 2018. The role of cognitive architectures in general artificial intelligence.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024a. Selectit: Selective instruction tuning for llms via uncertainty-aware self-reflection. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, Canada, December 10 - 15, 2024*.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2024b. Regmix: Data mixture as regression for language model pre-training. *CoRR*, abs/2407.01492.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kevin S McGrew and Jeffrey J Evans. 2004. Internal and external factorial extensions to the cattell-horn-carroll (chc) theory of cognitive abilities: A review of factor analytic research since carroll's seminal 1993 treatise. *Institute for Applied Psychometrics*.

Brando Miranda, Alycia Lee, Sudharsan Sundar, Allison Casasola, Rylan Schaeffer, and Sanmi Koyejo. 2024. Beyond scale: The diversity coefficient as a data quality metric for variability in natural language data. In *ICLR 2024 Workshop on Data-centric Machine Learning Research (DMLR): Harnessing Momentum for Science*. OpenReview.net.

Kolby Nottingham, Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Sameer Singh, Peter Clark, and Roy Fox. 2024. Skill set optimization: Reinforcing language model behavior via transferable skills. In *Forty-first International Conference on Machine*

Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net.

OpenAI. 2024a. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

OpenAI. 2024b. Learning to reason with llms.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. 2025. Sloth: scaling laws for LLM skills to predict multi-benchmark performance across families. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Jun Rao, Zepeng Lin, Xuebo Liu, Xiaopeng Ke, Lian Lian, Dong Jin, Shengjun Cheng, Jun Yu, and Min Zhang. 2025. APT: Improving specialist LLM performance with weakness case acquisition and iterative preference training. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20958–20980, Vienna, Austria. Association for Computational Linguistics.

Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. CommonIT: Commonality-aware instruction tuning for large language models via data partitions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10064–10083, Miami, Florida, USA. Association for Computational Linguistics.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM.

Yunfan Shao, Linyang Li, Zhaoye Fei, Hang Yan, Dahua Lin, and Xipeng Qiu. 2024. Balanced data sampling for language model training with clustering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14012–14023, Bangkok, Thailand. Association for Computational Linguistics.

Wei Song, Yadong Li, Jianhua Xu, Guowei Wu, Lingfeng Ming, Kexin Yi, Weihua Luo, Houyi Li, Yi Du, Fangda Guo, et al. 2024. M3gia: A cognition inspired multilingual and multimodal general intelligence ability benchmark. *arXiv preprint arXiv:2406.05343*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *CoRR*, abs/2206.04615.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024a. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoqiang Wang, Lingfei Wu, Tengfei Ma, and Bang Liu. 2024b. FAC^{2}E: Better understanding large language model capabilities by dissociating language and cognition. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13228–13243, Miami, Florida, USA. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024c. InsCL: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 663–677, Mexico City, Mexico. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-NaturalInstructions: Generalization via declarative

instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhihu Wang, Shiwan Zhao, Yu Wang, Heyuan Huang, Sitao Xie, Yubo Zhang, Jiaxin Shi, Zhixing Wang, Hongyan Li, and Junchi Yan. 2024d. Re-task: Revisiting llm tasks from capability, skill, and knowledge perspectives. *CoRR*, abs/2408.06904.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. 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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Reza Haf. 2024. Mixture-of-skills: Learning to optimize data usage for fine-tuning large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14226–14240, Miami, Florida, USA. Association for Computational Linguistics.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. FOFO: A benchmark to evaluate LLMs' format-following capability. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–699, Bangkok, Thailand. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zifan Xu, Haozhu Wang, Dmitriy Bespalov, Xuan Wang, Peter Stone, and Yanjun Qi. 2023. Latent skill discovery for chain-of-thought reasoning. *arXiv preprint arXiv:2312.04684*.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. FLASK: fine-grained language model evaluation based on alignment skill sets. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Dylan Zhang, Justin Wang, and Francois Charton. 2024. Instruction diversity drives generalization to unseen tasks. *arXiv preprint arXiv:2402.10891*.

Jian Zhao, Mengqing Wu, Liyun Zhou, Xuezhu Wang, and Jian Jia. 2022. Cognitive psychology-based artificial intelligence review. *Frontiers in Neuroscience*, 16:1024316.

Ming Zhong, Aston Zhang, Xuewei Wang, Rui Hou, Wenhan Xiong, Chenguang Zhu, Zhengxing Chen, Liang Tan, Chloe Bi, Mike Lewis, et al. 2025. Law of the weakest link: Cross capabilities of large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, et al. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Appendix

A.1 Design Rationale of CDT

This section elaborates on the foundational principles and design choices of the CDT framework. Its purpose is to provide deeper insight into the framework's application, particularly regarding the role of context and the relationship between the three capability dimensions.

Contextualized Instructions A central design principle of the CDT framework is that it operates on entire instructions rather than isolated keywords or concepts. As a result, capability tagging is inherently context-dependent. This approach is critical for resolving the inherent ambiguity of language, since the capabilities required to interpret a term can vary substantially based on the instructional context. For example, consider the term *"company"*, which can invoke different capabilities depending on the prompt. In the instruction *"Suggest creative names for my new internet company,"* the context provided by *"internet"* and *"creative names"* indicates a Domain in Computer Science and a Cognition of Ideational Fluency. By contrast, in the instruction *"Explain the legal definition of a limited liability company,"* the context of *"legal definition"* shifts the relevant Domain to Law and the required Cognition to Concept Abstraction. This context-driven methodology enables the CDT framework to provide a precise and nuanced analysis of the capabilities demanded by each unique user instruction.

Relationship Between CDT Dimensions The three dimensions of CDT are structured to be orthogonal, not hierarchical. They function as a multi-dimensional coordinate system. The Cognition axis describes how the model needs to reason or process information to fulfill the request, corresponding to

how to think. The Domain axis identifies what the subject area or field of knowledge is, corresponding to what the topic is. The Task axis specifies what the user's explicit intent or the required output format is, corresponding to what to do.

This structure ensures that a concept in one dimension is independent of the others. For example, a model might apply Pattern Recognition to perform a Detection task in the Literature domain by identifying the rhyme scheme in a poem, or use the same cognitive skill for a Generation task in the Economics domain to summarize cyclical trends in market data.

A.2 Cognitive Capability Construction

Starting with the "narrow" level of the CHC theory as defined in Flanagan and Dixon (2014), we make systematic adaptations to tailor the taxonomy to the specific characteristics of LLMs, which differ from human cognition in both modality and operational dynamics. The adaptation process involves the following steps:

- **Exclusion of non-linguistic modalities** (e.g., speaking, listening, action, visual, olfactory abilities) since LLMs are text-based. This reduction brings the set from 82 to 33 abilities.
- **Exclusion of non-core abilities** (e.g., memory and speed-related) that are less relevant to LLMs, as LLMs operate differently from humans in these aspects. We exclude abilities such as reading speed, writing speed, memory span, and others. This refinement reduces the set to 24.
- **Exclusion of domain knowledge-related abilities**, as domain knowledge is a separate CDT dimension. We exclude abilities such as general information, lexical knowledge, geography achievement, and others. This step brings the number down to 13.
- **Augmentation with LLM-relevant abilities**, such as logical analysis, abstract coding concepts, and problem decomposition, which are not emphasized in CHC but play a critical role in core LLM applications like code generation, reasoning, and instruction following. This increases the count to 16.
- **Refinement of overlapping or broad definitions**, e.g., splitting Induction into pattern recognition, concept abstraction, and hypothesis generation. This final step results in a set of 18 distinct cognitive abilities.

A.3 Capability Definition

The detailed definitions and abbreviations for the cognition, domain, and task dimensions are provided in Table 5, Table 6, and Table 7, respectively. In defining the domain dimension, we first established the overarching domain and then carefully subdivided it into subdomains for labeling purposes.

A.4 Capability Tagging Details

Training Data We collect 49K instruction samples from seven widely-used datasets: Selective Alpaca (Liu et al., 2024a), Dolly (Conover et al., 2023), Open Assistant (Köpf et al., 2023), Super-Natural Instructions (Wang et al., 2022), Tulu 3 (Lambert et al., 2025), Flan V2 (Longpre et al., 2023), and WizardLM (Xu et al., 2024). We randomly sample 7K queries per dataset and reserve 1K for testing.

Training Configuration We fine-tune Qwen2.5-7B-Base for 1 epoch with a batch size of 32 and a cosine learning rate schedule initialized at 2e-5.

Prompts We design our prompts following the approaches proposed by Lu et al. (2024); Ye et al. (2024). To mitigate position bias, we randomize the order of capabilities in the prompt for each data point. Additionally, when tagging cognitive capabilities, we ask the models to generate an explanation paired with each tag, as cognitive tasks require a deeper understanding of the instructions. All prompts are presented in Figure 5. We concatenate the detailed descriptions of the query, tag, and instruction into a single input prompt. When labeling the cognition dimension, we restrict the model to output at most two tags, along with their corresponding explanations.

Human Evaluation To evaluate the validity of the annotations generated by GPT-4o, we randomly selected 100 annotated entries from the training dataset to conduct a manual assessment. We evaluated the annotations based on the explanation of the labels in the cognition dimension, as well as the consistency of the task and domain dimensions with the original data, to determine whether GPT-4o's annotations should be accepted as the ground truth.

Cognition	Abbreviation	Definition
Pattern Recognition	PR	Ability to identify recurring patterns, trends, or sequences within a given set of data or materials (e.g., detecting similarities in a sequence of numbers or text).
Concept Abstraction	CA	Ability to form abstract concepts or categories based on shared characteristics or relationships among a set of materials.
Hypothesis Generation	HP	Ability to propose plausible explanations or predictions for incomplete information (e.g., inferring causes of a fictional conflict, suggesting scientific hypotheses).
General Sequential Reasoning	RG	Ability to start with stated rules, premises, or conditions, and to engage in one or more steps to reach a solution to a novel problem.
Quantitative Reasoning	RQ	Ability to inductively and deductively reason with concepts involving mathematical relations and properties.
Reading Decoding	RD	Ability to recognize and decode words or pseudowords in reading.
Writing Ability	WA	Ability to write with clarity of thought, organization, and good sentence structure.
Naming Facility	NA	Ability to rapidly produce names for concepts when presented with a text cue.
Associational Fluency	FA	Ability to rapidly produce a series of original or useful ideas related to a particular concept.
Expressional Fluency	FE	Ability to rapidly think of different ways of expressing an idea.
Number Facility	NM	Ability to rapidly and accurately manipulate and deal with numbers, from elementary skills of counting and recognizing numbers to advanced skills of adding, subtracting, multiplying, and dividing numbers.
Logical Analysis	LA	Ability to identify and apply logical structures, rules, and patterns within code or algorithms (e.g., recognizing logical constructs such as loops, conditions, or recursion in programming tasks).
Problem Decomposition	PD	Ability to systematically break down complex tasks into modular functional components, identify inter-component dependencies, and reconstruct solutions through controlled composition.
Abstract Coding Concept	AC	Ability to form abstract representations of programming concepts and apply them across different programming languages or environments (e.g., understanding concepts such as functions, variables, data structures, and algorithms in a generalized form, and applying them to solve problems in multiple programming languages).
Sensitivity to Problems/Alternative Solution Fluency	SP	Ability to rapidly think of a number of solutions to particular practical problem.
Originality/ Creativity	FO	Ability to rapidly produce original, clever, and insightful responses (expressions, interpretations) to a given topic, situation, or task.
Ideational Fluency	FI	Ability to rapidly produce a series of ideas, words, or phrases related to a specific condition or object. Quantity, not quality, is emphasized.
Word Fluency	FW	Ability to rapidly produce words that have specific phonemic, structural, or orthographic characteristics (independent of word meanings).

Table 5: The full definition of Cognition.

We involved two human evaluators to assess the annotations, and the consistency of their assessments was as follows: cognition dimension consistency (95%), domain dimension consistency (95%), task dimension consistency (85%). Additionally, we calculated the average acceptance rates for GPT-4o's annotations, which reflect the degree to which the evaluators agreed with GPT-4o's judg-

Domain	Sub-domain
Language	Linguistics,Literature,Multilingualism
Culture	Tradition,Art,Sports,Mass Media,Music,Food
Health	Health
Natural Science	Biology,Earth Science,Astronomy,Chemistry,Physics
Math	Mathematics,Logic
Social Science	Economics,Law,Politics,Education,Sociology
Technology	Agriculture,Computer Science,Automation,Electronics,Engineering
Coding	Coding
Humanities	Communication,Religion,Philosophy,Ethics,History

Table 6: The full definition of Domain.

ments across each dimension: cognition (97.5%), domain (87.5%), task (87.5%) and overall acceptance rate (90.5%). From these results, we observe that GPT-4o's annotations on the cognition dimension aligned more closely with human evaluations. This may be due to the fact that, in the cognition dimension, the cognitive abilities are more abstract and require a deeper understanding of the instructions. As a result, we not only require GPT-4o to output cognitive ability labels but also to provide explanations corresponding to these labels.

The strong performance of GPT-4o validates the quality of our initial annotated dataset. However, relying on such a proprietary model for large-scale labeling of our 270K data pool is expensive and limits the broader adoption of the CDT framework. Therefore, to create a scalable and accessible solution, we use this high-quality dataset to train our own open-source capability annotators. The detailed cost-benefit analysis of this two-stage annotation approach is discussed below.

Annotation Cost Considerations While the process of training the capability taggers requires an initial annotation phase, this process is largely automated using GPT-4o to generate fine-grained labels. This is a one-time, upfront investment designed not merely to label data for a single experiment, but to distill the nuanced definitions of our CDT framework into a set of efficient and open-source Qwen2.5-based annotators. The necessity of this approach is validated by our experiments: using the GPT-4o annotations as ground truth, we evaluate zero-shot performance on our test set. The Qwen2.5-7B-Base model achieves an average tagging accuracy of only 33.5% across the three dimensions, while Qwen2.5-7B-Instruct reaches 53.5%. In contrast, our annotators achieve an average accuracy of 85.1%. This significant performance gap demonstrates that our initial data distillation is a crucial step for developing a reusable tagging tool. Once trained, our annotators can be applied to any number of future datasets at a low and predictable computational cost, far cheaper than repeated API calls to proprietary models, which makes the CDT framework highly scalable. As shown in Section 6, this workflow strategically trades the upfront annotation investment for significant downstream efficiency: by precisely selecting data based on these capability labels, we achieve improved model performance using only a fraction of the total data, thereby substantially reducing the computational cost of the final fine-tuning phase.

A.5 Data Selection Algorithm

We present our diversity-driven general scenario data selection algorithm in Algorithm 1 and capability-oriented specific scenario in Algorithm 2.

A.6 Experiments on Mistral Model

In addition to the LLama2-7B-Base model, we also conducted experiments on the Mistral-7B-Base model, using 20% of the training data in the general scenario as an example. The results are presented in Table 8. As shown, our method achieves the highest score of 55.5 among all baselines, further demonstrating the generalizability of our approach.

Task	Definition
Generation	Creating new information with human-input conditions, involving the automatic generation of various text materials follow the instruction given by the user.
Rewrite	Taking a piece of text and rephrasing it while preserving its original meaning, which may involve simplifying the language, changing the structure, or adjusting the tone.
Summarization	Condensing longer texts into shorter versions while retaining the key information and main ideas, making it easier to digest complex information.
Classification	Assigning predefined labels or categories to text based on its content, such as topic categorization.
Brainstorming	Generating ideas, encouraging creative thinking, or exploring possibilities.
Sentiment	Determining the emotional tone or sentiment expressed in a piece of text.
Completion	Continuing a given prompt with relevant and contextually appropriate content, such as finishing sentences or filling in blanks.
Natural Language Inference	Assessing the relationship between two sentences to determine if one logically follows from the other (entailment), (contradiction), or if the relationship is unclear (neutral).
Bias and Fairness	Evaluating models for potential bias, fairness, or harmfulness in their outputs.
Word Sense Disambiguation	Determining which meaning of a word is used in a given context, especially for words that have multiple meanings.
Multiple Choice QA	Answering questions by selecting the correct option from a predefined set of possible answers based on provided information or context.
Closed QA	Answering questions directly without access to external knowledge.
Open QA	Answering open-ended questions that can cover a wide range of topics, often without a single, definitive answer.
Extraction	Identifying and extracting specific pieces of information from a given text.
Program Execution	Executing or simulating the execution of a given program or script, processing inputs, performing operations, and returning outputs based on the specified instructions, often including code interpretation or debugging.
Detection	Identifying the presence of specific elements, patterns, or anomalies in a given text, such as detecting spam or certain linguistic features like named entities or grammatical errors.

Table 7: The full definition of Task.

Methods	**ARC-C**	**MMLU**	**BBH**	**CEVAL**	**TYDIQA**	**AVG.**
Base	50.3	**62.3**	57.6	**46.8**	55.8	54.6
All	52.1	60.6	56.2	46.3	57.3	54.5
Random	52.1	59.9	58.1	46.4	57.0	54.7
InsTag	**52.7**	60.5	56.0	46.3	**57.7**	54.6
CDT	52.6	61.5	**60.0**	46.6	56.7	**55.5**

Table 8: Results of applying CDT in diversity-driven general data selection on the Mistral-7B-Base model.

You are a helpful and precise assistant that selects the necessary skills required to respond to instructions. You are given the following 18 skills.

[Skill Options]
{tags}

Note that the 'RQ' skill focuses on math problems. What are the relevant skills that are needed to answer the following instruction? Especially, select the primary skills that this instruction particularly requires rather than skills that could be applied to common instructions.

[Instruction]
{instruction}

Select and write the name of the primary skills. The number of skills you select should be no more than 2. You don't need to select exactly 2 skills. Also, write a brief explanation of the reason why you choose this skill. The explanation should not be the definition of the skill that I provide to you. The skills you return should be arranged in descending order of importance, from the most important to the least. Your response have to strictly follow this JSON format:[{'skill': str, 'explanation': str}].

[Assistant]

(a) Cognition tagging prompt

You are a helpful and precise assistant in labeling the domain of the instruction. You will be given a list of 9 main domains with 33 subdomains. After you see the instruction, you need to label the subdomain that the instruction is most likely to be.

[Domains]
{tags}

[Instruction]
{instruction}

Which subdomain best fits the above instruction? Please select only one subdomain from the list I provide. Please provide only the subdomain behind the colon rather than the main domain. Your response have to strictly follow this JSON format: {"domain": str}.

[Assistant]

(b) Domain tagging prompt

You are a helpful and precise assistant in labeling the task type of the instruction. You will be given a list of 16 task types. After you see the instruction, you need to label the task type that the instruction is most likely to be.

[Task Type]
{tags}

[Instruction]
{instruction}

Which task type best fits the above instruction? Please select only one task type from the list I provide. Please provide only the task name without the definition. Your response have to strictly follow this JSON format:{"task": str}.

[Assistant]

(c) Task tagging prompt

Figure 5: The prompts we used on tagging.

Algorithm 1: Diversity-driven General Scenario Data Selection

Data: $D^{'}_{pool}$: The capacity labeled data pool; N: Selection set size;
Result: D_{train}: The selected training dataset;
initialization: T_d: All composite capabilities in the data pool; $D_{train} \leftarrow \emptyset$;
Sorting T_d in descending order based on the number of corresponding data points in $D^{'}_{pool}$;
while $|D_{train}| < N$ **do**
    $Flag \leftarrow False$;
    for *each capability* $f \in T_d$ **do**
        $D_f \leftarrow Find\_Data(f, D^{'}_{pool})$;
        // Select data tagged with composite capability f from $D^{'}_{pool}$
        if $D_f \neq \emptyset$ **then**
            $d \leftarrow Random(D_f, 1)$;
            // Randomly select one data point from D_f
            $D_{train} \leftarrow \{d\} \cup D_{train}$;
            $D^{'}_{pool} \leftarrow D^{'}_{pool} \backslash \{d\}$;
            $Flag \leftarrow True$;
        end
        if $|D_{train}| = N$ **then**
            break;
        end
    end
    if $Flag = False$ **then**
        break;
        // All data points related to capability set T_d are selected
    end
end

Algorithm 2: Capability-oriented Specific Scenario Data Selection

Data: $D^{'}_{pool}$: The capacity labeled data pool; $D^{'}_{valid}$: The capacity labeled validation set; N: Selection set size;
Result: D_{train}: The selected training dataset;
initialization: T_v: Triplet capability set of validation set; T^{*}_{v}: Binary capability set; $T^{\star}_{v}$: Unary capability set; $D_{train} \leftarrow \emptyset$;
for *each capability set* $T \in \{T_v, T^{*}_{v}, T^{\star}_{v}\}$ **do**
  Sorting T in descending order based on the number of corresponding data points in $D^{'}_{pool}$;
  while $|D_{train}| < N$ **do**
    $Flag \leftarrow False$;
    for *each capability* $f \in T$ **do**
      if $N = |D_{train}|$ **then**
        break;
      end
      $D_f \leftarrow Find\_Data(f, D^{'}_{pool})$;
      // Select data tagged with composite capability f from $D^{'}_{pool}$
      if $D_f \neq \emptyset$ **then**
        $d \leftarrow Random(D_f, 1)$;
        // Randomly select one data point from D_f
        $D_{train} \leftarrow \{d\} \cup D_{train}$;
        $D^{'}_{pool} \leftarrow D^{'}_{pool} \backslash \{d\}$;
        $Flag \leftarrow True$;
      end
    end
    if $Flag = False$ **then**
      break;
      // All data points related to capability set T are selected
    end
  end
end
if $|D_{train}| < N$ **then**
  // Not enough data points labeled with the desired capabilities
  $D_r \leftarrow Random(D^{'}_{pool}, N - |D_{train}|)$;
  $D_{train} \leftarrow D_r \cup D_{train}$;
end