VersaTune: An Efficient Data Composition Framework for Training Multi-Capability LLMs

Keer Lu[†], **Keshi Zhao**[†], **Zhuoran Zhang**[‡], **Zheng Liang**[§],
Bin Cui[‡], **Tengjiao Wang**[‡], **Wentao Zhang**[†¶]
[†]Center for Data Science, Academy for Advanced Interdisciplinary Studies, Peking University,
[‡]School of CS & Key Lab of High Confidence Software Technologies, Peking University,
[§]Baichuan Inc.
{keer.lu, zhaoks}@stu.pku.edu.cn, {bin.cui, tjwang, wentao.zhang}@pku.edu.cn

Abstract

As demonstrated by the proprietary Large Language Models (LLMs) such as GPT and Claude series, LLMs have the potential to achieve remarkable proficiency across a wide range of domains, including law, medicine, finance, science, code, etc., all within a single model. These capabilities are further augmented during the Supervised Fine-Tuning (SFT) phase. Despite their potential, existing work mainly focuses on domain-specific enhancements during fine-tuning, the challenge of which lies in catastrophic forgetting of knowledge across other domains. In this study, we introduce ***VersaTune***, a novel data composition framework designed for enhancing LLMs' overall multi-domain capabilities during training. We begin with detecting the distribution of domain-specific knowledge within the base model, followed by the training data composition that aligns with the model's existing knowledge distribution. During the subsequent training process, domain weights are dynamically adjusted based on their learnable potential and forgetting degree. Experimental results indicate that VersaTune is effective in multi-domain fostering, with an improvement of 29.77% in the overall multi-ability performances compared to uniform domain weights. Furthermore, we find that Qwen-2.5-32B + VersaTune even surpasses frontier models, including GPT-4o, Claude3.5-Sonnet and DeepSeek-V3 by 0.86%, 4.76% and 4.60%. Additionally, in scenarios where flexible expansion of a specific domain is required, VersaTune reduces the performance degradation in other domains by 38.77%, while preserving the training efficacy of the target domain.

1 Introduction

Large Language Models (LLMs) have become a cornerstone in Artificial Intelligence (AI) (Achiam et al., 2023; Dwivedi et al., 2021; Lewkowycz et al., 2022), particularly for Natural Language Processing tasks (Brown et al., 2020; Devlin, 2018; Radford et al., 2019), reshaping AI research and applications in domains such as law (Cui et al., 2023), medicine (Singhal et al., 2023; Thirunavukarasu et al., 2023), finance (Li et al., 2023b; Wu et al., 2023), science (Beltagy et al., 2019; Taylor et al., 2022) and code (Liu et al., 2024b; Roziere et al., 2023). The success of LLMs stems from their capabilities to automatically learn and distill hierarchical data representations, making them highly effective for complex tasks (Nie et al., 2023). In order to further enhance such abilities across these areas, LLMs typically undergo the supervised fine-tuning (SFT) stages on domain-specific datasets.

As demonstrated by the robust performances of state-of-the-art LLMs such as GPT-4 (Achiam et al., 2023) and Gemini (Team et al., 2023), *LLMs have the potential to master multiple tasks across all specific domains within a single model*. However, most existing research on supervised fine-tuning tends to merely concentrate on a single ability of LLMs (Dong et al., 2023; Xu et al., 2024), with the multi-domain performance on composite data of essentially different downstream tasks being less studied. We try to enhance the overall multitasking performance of LLMs across various domains by optimizing data mixing ratios during training:

> *How to design a data composition strategy during SFT stages that could achieve overall multi-domain capabilities?*

Through analysis, we identified that the challenges associated with data composition strategies stem from the following three key aspects:

C1: Catastrophic Forgetting. Given the fundamental differences between tasks of various domains, for multi-domain SFT, the sequential training strategy across multiple phases, where each phase exclusively utilizes a single-domain dataset

¶ Corresponding Author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6646–6670
November 4-9, 2025

for training, can easily lead to significant performance drop of prior knowledge, which is well-known as Catastrophic Forgetting (Kaushik et al., 2021; McCloskey and Cohen, 1989), as depicted in Table 2 and Figure 6. It hinders the versatile fine-tuning performance of a model across multiple domains (De Lange et al., 2021; Dong et al., 2023; Yuan et al., 2022). Therefore, mixing data from different domains is crucial for mitigating catastrophic forgetting during training, enhancing the overall performance and adaptability.

C2: Low Efficiency. Existing data composition research during the supervised fine-tuning phase for LLMs is still in its initial stages, with most strategies based on heuristic or manually determined rules (Wang et al., 2023; Albalak et al., 2024; Dubey et al., 2024). One of the common baselines is defining domain weights referring to natural domain sizes, which weights all individual data points equally. Such approaches struggle to optimally balance different domains, failing to maximize the overall training effectiveness for multiple abilities. There lacks a well-defined methodology that efficiently enhances the versatile capabilities of LLMs across multiple domains during the SFT stage.

C3: Low Flexibility in Domain Expansion. Existing SFT approaches for specific domain abilities typically pre-determine the proportions of different datasets according to prior experience (Azerbayev et al., 2023; Roziere et al., 2023). Such strategies lack the flexibility to dynamically adjust the data mixing ratios of different domains during the training process, which does not allow for real-time feedback from LLMs to inform and optimize the data composition. This static approach hinders the minimization of performance loss in other domains as LLMs undergo specialized training.

To address these challenges, we introduce ***VersaTune***, a novel data composition framework to enhance models' overall performances across different domains during supervised fine-tuning. We first detect the proportion distribution of domain knowledge within the target model (Section 2.1), followed by data composition based on the existing distribution for multi-ability enhancement (Section 2.2.2) and flexible domain expansion (Section 2.2.3). Our contributions are as follows:

- *Knowledge Consistency Training.* We introduce the concept of *knowledge consistency training* for LLMs' multi-capability development, which enables the model to continue learning from datasets that possess a knowledge distribution aligned with its pre-existing knowledge feature.

- *Multi-Capability Data Composition Framework.* We propose VersaTune, a novel data composition framework that leverages the model's intrinsic domain knowledge distribution to optimize the training data proportion. VersaTune is designed to enhance the overall performance across multiple domains (Section 2.2.2), as well as to provide flexible expansion for specific domains while minimizing the performance degradation in other domains (Section 2.2.3).

- *Performance and Effectiveness.* Our extensive evaluations across domains demonstrate that VersaTune can achieve an improvement of 29.77% in versatile fine-tuning for multiple domains. Notably, we find that our Qwen-2.5-32B + VersaTune even outperforms frontier models including GPT-4o, Claude3.5-Sonnet and DeepSeek-V3 by 0.86%, 4.76% and 4.60%. Furthermore, when focusing on specific-domain expansion, VersaTune maintains training effectiveness in the target domain while reducing performance degradation in other non-target domains by 38.77%.

2 VersaTune

In this section, we introduce VersaTune, a data composition framework designed for multi-capability training, aiming to effectively compose data from multiple domains and optimize the data proportion during training. Figure 1 presents the workflow of VersaTune, which generally contains two phases.

2.1 Phase 1: Domain Knowledge Detection

Here, we first present a domain mixing strategy for fine-tuning a LLM that possesses a comprehensive multitask capability (Section 2.1.1). This approach is designed to align with the inherent domain knowledge distribution within the base model waiting for subsequent training. Following this, we describe the method for detecting domain knowledge proportion of the base model, which is crucial for informing the fine-tuning process (Section 2.1).

2.1.1 Knowledge Consistency Training

Previous research on data mixing ratios during the SFT phase for LLMs has predominantly focused on enhancing capabilities within a specific domain, often utilizing only data from that domain or employing heuristic, experience-based data proportions.

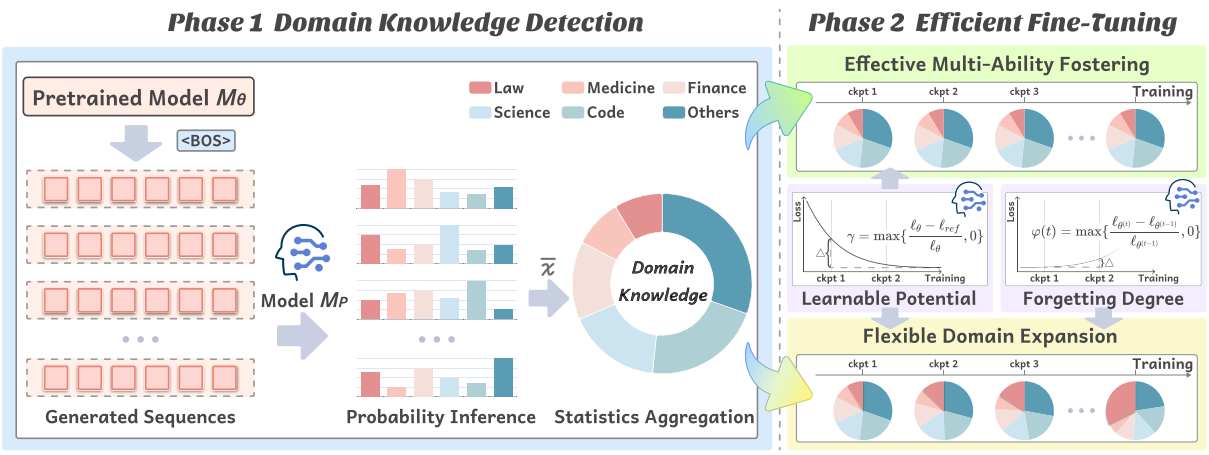

Figure 1: Overview of ***VersaTune***. We begin by probing the knowledge distribution within the base model M_θ, utilizing a proprietary model M_P to estimate the probability of sequences generated by M_θ belonging to various domains. Throughout the efficient fine-tuning process, we dynamically adjust the data domain ratios in response to M_θ's real-time performance feedback, with learnable potential and forgetting degree serving as evaluative metrics.

We argue that such strategies can significantly impair the LLM's abilities in other domains. In the fine-tuning stage, maintaining a robust overall capability across various domains is crucial.

What data mixing strategy effectively boosts the versatile performance of LLMs across domains during the SFT phase? We propose the statement:

Statement 1 *An LLM fine-tuned with domain-specific data proportions $P_{SFT}(x)$ that align with its pretrained output distributions $P_{knowledge}(x)$ will exhibit enhanced and balanced performance across these domains, compared to a model fine-tuned with a non-matching data distribution. Formally, the relationship can be represented as:*

$$P_{SFT}(x) \approx P_{knowledge}(x), \forall x \in \chi \qquad (1)$$

where χ denotes the set of all possible data points.

The rationale behind this statement is rooted in the observation that during the pretraining phase, LLMs develop a general understanding of language features and domain-specific knowledge. By maintaining the same distribution of knowledge during fine-tuning, the model can build upon this pre-existing knowledge, thereby enhancing learning efficiency and robustness.

2.1.2 Knowledge Distribution Detection

Drawing on prior research into knowledge identification methods (Gekhman et al., 2024; Zhao et al., 2023b) and training data inference strategies for LLMs (Ding et al., 2022; Hayase et al., 2024), we propose a structured approach to efficiently detect domain knowledge based on statistics. The method involves the generation of textual outputs from the base model M_θ waiting for fine-tuning, followed by classification into predefined domains referring to a proprietary model M_P. The process is repeated multiple times to ensure statistical robustness.

Assuming the data corpus contains k distinct domains, as shown in Algorithm 1, we first prompt the base model poised for fine-tuning M_θ with the Beginning of Sequence (<BOS>)[1] tokens to generate a set of N_S data entries $\mathcal{S} = \{s_i\}_{i=1}^{N_S}$ (Line 3). Subsequently, we employ a proprietary model M_P to infer probabilities that these N_S entries belong to each domain (Line 5-7). We then calculate a weighted average of the probability distributions for all data across these domains, thereby deriving the domain knowledge distribution of the current base model M_θ (Line 9). To ensure statistical robustness, the process is iteratively conducted T times, and we use the mean of these T iterations as the estimated result for knowledge distribution.

2.2 Phase 2: Fine-Tuning Multi-Ability LLMs Efficiently

Having detected the distribution of domain knowledge within the base model, we will now utilize these findings to guide our multi-ability SFT process. The approaches are designed to enhance the overall performance of the fine-tuned model across a spectrum of multi-domain tasks (Section 2.2.2),

[1]The <BOS> token serves as a trigger for text generation, which enables unrestricted generation without biasing the model toward any specific domain, thereby enabling a reliable assessment of the distribution of domains in the model's generated outputs. More details can be found in Section A.1.1.

Algorithm 1 Knowledge Distribution Detection

Input: Base model M_θ, Proprietary model M_P, Hyperparameters: sample number N_S, maximum iterations T
Parameter: Data samples $\mathcal{S}$ generated from M_θ
Output: Domain distribution $\vec{P}$
Define $\vec{p}$: domain probability distribution of data sample s

for $t = 1, 2, \ldots, T$ **do**
  / Step 1: Data Generation */*
  Generate data samples from the base model: $\mathcal{S} = \{s_i\}_{i=1}^{N_S}$ where $s_i = M_\theta(< BOS >)$
  / Step 2: Domain Probability Inference */*
  for each data sample s_i in $\mathcal{S}$ **do**
    Provide domain probability of s_i referring to the proprietary model M_P:
    $$\vec{p_i} = (p_{ij})_{j=1}^{k} \leftarrow M_P(s_i)$$
  end for
  / Step 3: Statistics Aggregation */*
  Estimate domain knowledge distribution: $\vec{P}^{(t)} = (P_j^{(t)})_{j=1}^{k}$ where $P_j^{(t)} = \frac{1}{N_S}\sum_{i=1}^{N_S} p_{ij}$
end for
Return $\vec{P} = (P_j)_{j=1}^{k}$ where $P_j = \frac{1}{T}\sum_{t=1}^{T} P_j^{(t)}$

as well as to facilitate the flexible expansion of capabilities in specific domains (Section 2.2.3).

Setting. Our goal is to construct a composite dataset covering k specific domains, which can be denoted as $\mathcal{D}_{train} = \{(D_{train}^j, P_j)\}_{j=1}^{k}$ with each tuple representing a specific domain and its corresponding proportion, such that training a model on dataset $\mathcal{D}_{train}$ could achieve overall lower loss on a uniformly distributed composite target validation dataset $\mathcal{D}_{val} = \{(D_{val}^j, 1/k)\}_{j=1}^{k}$ or meet the flexible domain expansion while preserving the performances in other domains. The specialized capabilities of LLMs are measured using downstream tasks related to different domains (e.g., FinBen (Xie et al., 2024a) for financial performances).

2.2.1 Preliminary: Learnable Potential and Forgetting Degree of Knowledge

Before formally introducing the effective multi-task fine-tuning and flexible domain expansion data composing strategies, we will first provide an overview of the evaluation metrics used for the following algorithms in this subsection.

Mastery Ceiling. We first fine-tuned the small *reference model* M_{ref} for T_{ref} epochs on each domain separately, and identified the epoch with the lowest average loss during this process as the lower bound on the minimum loss attainable by the target model M_θ for the given domain. This value represents the highest level of domain knowledge mastery that the model can achieve in the context of the current specific domain under given conditions.

Learnable Potential. We can observe whether a domain could be effectively learned by the model through comparing the difference between the loss of the target model M_θ and the minimum loss that the reference model M_{ref} can achieve. Based on these principles, we propose Equation (2) to score the learnable potential of domain j:

$$\gamma_j = \max\{\frac{\ell_\theta^j - \ell_{ref}^j}{\ell_\theta^j}, 0\} \tag{2}$$

where ℓ_θ^j denotes the loss associated with the target model M_θ for the j-th domain, while ℓ_{ref}^j signifies the corresponding loss for the reference model M_{ref} within the same domain. To mitigate the impact of inherent loss variations across different domains for the model, we have introduced a normalization term into the formula.

Forgetting Degree. When focusing on expanding a model to a specific domain, our objective is to minimize the loss of the model's knowledge regarding other domains. Here we segment the fine-tuning stage into T distinct checkpoints. We quantify the degree of knowledge loss, or the forgetting of the current domain, by measuring the difference in loss between the t-th and $(t-1)$-th training steps. This difference reflects the model's mastery loss for the tasks associated with the current domain. Based on this principle, we introduce Equation (3) to assess the model's forgetting degree for domain j at the t-th training step.

$$\varphi_j^{(t)} = \max\{\frac{\ell_{\theta^{(t)}}^j - \ell_{\theta^{(t-1)}}^j}{\ell_{\theta^{(t-1)}}^j}, 0\} \tag{3}$$

where $\ell_{\theta^{(t)}}^j$ represents the loss at the t-th training step associated with the target model M_θ for the j-th domain, while $\ell_{\theta^{(t-1)}}^j$ denotes the loss at the preceding $(t-1)$-th iteration for the same domain. We also incorporated a normalization factor into the equation to counteract the effects of inherent loss disparities among domains.

2.2.2 Effective Multi-Ability Fostering

To cultivate the multi-tasking capabilities of a LLM during the fine-tuning phase, we have aligned

Algorithm 2 VersaTune Multi-Ability Fine-Tuning (for Domain Robustness)

Input: Base model to be fine-tuned $M_\theta^{(0)}$, Domain reference loss $\{\ell_{ref}^j\}_{j=1}^k$, Hyperparameters: adjustment magnitude σ, training step number T
Parameter: Data proportion $\{P_j\}_{j=1}^k$ of dataset
Output: Fine-tuned multi-ability model $M_\theta^{(T)}$
Define γ: learnable potential of the current domain

1: Initialize domain proportion $\{P_j^{(0)}\}_{j=1}^k$ according to Equation (1) and Algorithm 1
2: **for** $t = 1, 2, \ldots, T$ **do**
3: **for** $j = 1, 2, \ldots, k$ **do**
4: Learnable potential for the j-th domain:
$$\gamma_j^{(t)} = \max\{\frac{\ell_{\theta(t)}^j - \ell_{ref}^j}{\ell_{\theta(t)}^j}, 0\}$$
5: Update domain weights:
$$P_j^{(t)'} = P_j^{(t-1)}(1 + \sigma\gamma_j^{(t)})$$
6: **end for**
7: Renormalize domain weights:
$$P_j^{(t)} = \frac{P_j^{(t)'}}{\sum_{i=1}^k P_i^{(t)'}}, \quad \forall j \in \{1, 2, ..., k\}$$
8: Update parameters of fine-tuned model $M_\theta^{(t)}$
9: **end for**
10: **Return** Fine-tuned model $M_\theta^{(T)}$

the initial domain distribution in the SFT stage with the knowledge detection results of the base model as stated in Equation (1). Furthermore, we dynamically make minor adjustments in the composition ratios of various domains based on the model's real-time feedback at different SFT stages.

As detailed in Algorithm 2, in the pursuit of balanced domain expertise enhancement, we first blended the domain proportions in accordance with the base model's intrinsic domain knowledge distribution detected by Algorithm 1 (Line 1). Then at each training step t, we assigned a learnable potential score to each domain based on the methodology outlined in Equation (2). These scores were then utilized to fine-tune the representation of each domain within the composite SFT dataset, ensuring a balanced development of competencies across all domains throughout the training process (Line 3-7). At the same time, the parameters of model M_θ are updated based on the gradients computed through backpropagation (Line 8). This adaptive approach is imperative to harmonize the progression of capabilities in different domains and to optimize the model's performance on multiple tasks.

2.2.3 Flexible Domain Expansion

When conducting fine-tuning on a pretrained model, there are instances where we aim to particularly enhance models' performance on specific domain tasks. Consequently, our algorithmic framework ought to possess the flexibility to accommodate domain expansion and generalize effectively. Building upon Statement 1, we present the following statement tailored for domain expansion:

Statement 2 *When fine-tuning a LLM for a specific capability, increasing the volume of data from a particular domain D_e while adjusting other domains $(j = 1, 2, ..., k, j \neq e)$ according to the knowledge distribution of the base model, facilitates a flexible strategy for domain expansion. Formally, the relationship can be represented as:*

$$P_{SFT}^{'}(x) \approx \sum_{j=1}^{k} A(D_j) P_{SFT}(x|D_j), j = 1, ..., k \tag{4}$$

where $P_{SFT}(x|D_j)$ is the data distribution in the domain D_j, and $A(D_j)$ is the adjustment factor.

Here $A(D_j)$ is determined based on the knowledge distribution of the pre-trained domain. In particular, when D_e increases, the other domains $\{D_j\}_{j=1,j\neq e}^k$ shrink proportionally as a whole, which can be expressed as:

$$A(D_j) = \begin{cases} \alpha, \text{if} \quad j = e \\ \beta \frac{1}{\sum_{j=1, j\neq e}^k A(D_j)}, \text{others} \end{cases} \tag{5}$$

where α is the increased adjustment factor, and β is the original ratio of other domain knowledge relative to D_e. Algorithm implementation details and hyper-parameter settings are provided in Section B.1 and Section C.2.1.

3 Experiments and Results

In this section, we describe details of our experimental setup (Section 3.1), the baseline methods we use for comparison (Section 3.2), and experimental results (Section 3.3).

3.1 Experimental Setup

Datasets. For *training*, we have collected datasets spanning 6 domains for SFT, including Sonnet3.5 Science Conversations[2], Lawyer-

[2]https://huggingface.co/datasets/jeffmeloy/sonnet3.5_science_conversations

Model	Method	Law		Medical		Finance		Science		Code		General	
		LegalBench	LawBench	MedQA	MedMCQA	FinEval	FinanceIQ	SciEval	MMLU-Sci	HumanEval	MBPP	AGIEval	HellaSwag
Frontier Models													
GPT-4o	–	79.00	57.41	81.92	74.60	64.58	66.25	72.54	85.47	88.40	75.50	71.82	90.56
Claude3.5-Sonnet	–	77.60	40.73	76.38	68.80	65.90	62.55	68.72	84.24	84.07	80.48	75.63	89.12
DeepSeek-V3	–	65.46	52.25	78.82	74.30	68.15	75.03	69.58	82.90	65.20	75.40	79.60	88.90
Open-Sourced Base Models													
LLaMA-2-7B	Uniform Distribution	15.71	30.72	23.45	27.57	33.50	2.71	9.30	42.89	5.67	3.44	20.16	71.40
	Inverse Distribution	13.23↓	26.94↓	21.38↓	26.52↓	32.96↓	2.53↓	8.98↓	39.67↓	3.47↓	2.42↓	18.83↓	71.33↓
	VersaTune	**23.18**↑	**36.31**↑	**35.04**↑	**40.75**↑	**36.27**↑	**29.04**↑	**56.75**↑	**50.06**↑	**15.62**↑	**15.68**↑	**24.67**↑	**71.76**↑
Qwen-2-7B	Uniform Distribution	39.05	31.99	35.07	17.73	59.49	14.62	25.30	62.73	53.26	37.82	47.31	73.60
	Inverse Distribution	34.01↓	27.81↓	23.90↓	16.31↓	56.53↓	11.30↓	18.57↓	58.25↓	50.65↓	33.63↓	45.74↓	73.52↓
	VersaTune	**50.56**↑	**35.54**↑	**45.48**↑	**41.24**↑	**60.95**↑	**68.39**↑	**51.58**↑	**70.42**↑	**58.15**↑	**47.64**↑	**48.02**↑	**73.67**↑
Qwen-2.5-7B	Uniform Distribution	40.11	31.48	25.17	25.84	59.58	31.66	19.88	65.84	55.64	46.86	45.42	73.69
	Inverse Distribution	36.36↓	26.98↓	24.16↓	19.35↓	57.07↓	29.25↓	16.68↓	62.78↓	52.97↓	44.63↓	45.67↑	72.92↓
	VersaTune	**51.65**↑	**36.75**↑	**34.28**↑	**52.09**↑	**62.48**↑	**69.09**↑	**68.14**↑	**74.16**↑	**60.68**↑	**61.25**↑	**49.73**↑	**73.90**↑
LLaMA-3-8B	Uniform Distribution	33.52	31.16	31.03	10.26	34.83	4.97	6.51	50.17	22.94	28.85	23.87	73.26
	Inverse Distribution	27.83↓	27.48↓	25.51↓	8.77↓	33.71↓	3.31↓	6.09↓	46.62↓	19.67↓	24.34↓	23.45↓	72.40↓
	VersaTune	**49.67**↑	**37.87**↑	**42.21**↑	**45.72**↑	**38.80**↑	**43.58**↑	**56.67**↑	**60.61**↑	**28.91**↑	**35.65**↑	**28.78**↑	**73.62**↑
LLaMA-2-13B	Uniform Distribution	47.66	34.85	32.98	36.54	37.54	32.85	45.72	50.77	36.54	38.55	36.89	73.50
	Inverse Distribution	40.12↓	30.67↓	26.27↓	28.78↓	36.67↓	26.76↓	38.96↓	48.68↓	28.78↓	35.83↓	36.67↓	73.11↓
	VersaTune	**55.87**↑	**40.14**↑	**45.78**↑	**47.67**↑	**39.48**↑	**55.12**↑	**63.87**↑	**62.84**↑	**47.67**↑	**44.62**↑	**39.64**↑	**74.63**↑
Qwen-2.5-14B	Uniform Distribution	50.73	39.49	47.85	38.71	64.72	64.39	39.74	73.45	68.75	72.14	54.92	75.88
	Inverse Distribution	46.08↓	35.36↓	45.75↓	32.56↓	64.88↑	60.53↓	27.68↓	68.22↓	63.36↓	68.49↓	54.87↓	75.42↓
	VersaTune	**60.59**↑	**46.58**↑	**50.24**↑	**45.15**↑	**65.84**↑	**78.68**↑	**62.89**↑	**82.86**↑	**82.64**↑	**81.48**↑	**55.52**↑	**75.98**↑
Qwen-2.5-32B	Uniform Distribution	68.86	45.28	72.34	68.18	68.03	75.14	58.30	80.17	78.59	71.04	75.26	84.40
	Inverse Distribution	62.93↓	42.05↓	68.80↓	66.09↓	66.80↓	73.93↓	52.94↓	79.31↓	74.44↓	70.71↓	75.00↓	83.80↓
	VersaTune	**75.67**↑	**56.76**↑	**78.72**↑	**72.36**↑	**70.50**↑	**78.80**↑	**70.77**↑	**85.23**↑	**86.60**↑	**79.89**↑	**75.81**↑	**84.75**↑
Open-Sourced Instruct Models													
Qwen-2.5-7B -Instruct	Uniform Distribution	45.56	33.67	33.75	42.83	58.43	38.79	34.64	66.15	59.97	61.48	55.37	71.02
	Inverse Distribution	38.42↓	27.44↓	30.81↓	39.67↓	56.84↓	35.83↓	30.72↓	63.70↓	57.12↓	58.96↓	54.91↓	71.23↑
	VersaTune	**54.81**↑	**41.43**↑	**43.04**↑	**58.65**↑	**64.97**↑	**55.74**↑	**60.78**↑	**71.85**↑	**63.95**↑	**69.70**↑	**58.33**↑	**72.15**↑
LLaMA-3-8B -Instruct	Uniform Distribution	46.48	35.10	40.74	38.65	38.96	22.97	48.68	59.85	44.76	52.94	42.65	69.45
	Inverse Distribution	43.96↓	31.67↓	37.38↓	34.82↓	35.73↓	20.87↓	44.84↓	55.58↓	40.69↓	50.66↓	**44.81**↑	68.92↓
	VersaTune	**56.05**↑	**43.76**↑	**52.64**↑	**50.81**↑	**43.17**↑	**48.62**↑	**68.56**↑	**67.74**↑	**54.06**↑	**59.19**↑	43.58↑	**69.67**↑
Qwen-2.5-14B -Instruct	Uniform Distribution	52.85	46.75	55.94	43.58	64.46	68.89	54.84	75.52	81.69	79.05	60.82	77.17
	Inverse Distribution	49.69↓	44.50↓	51.78↓	40.67↓	62.97↓	66.98↓	49.58↓	74.25↓	78.45↓	74.38↓	60.26↓	76.71↓
	VersaTune	**59.87**↑	**58.72**↑	**64.56**↑	**63.85**↑	**65.98**↑	**77.68**↑	**61.38**↑	**81.84**↑	**85.44**↑	**84.90**↑	**60.97**↑	**77.95**↑

Table 1: Results of VersaTune on multi-ability fostering, we compare the performances of several methods across different models. For each domain, we evaluate the models using two relevant benchmarks. The best results are in **bold**. ↑ and ↓ indicate an increase or decrease in downstream scores comparing to the *uniform distribution* strategy.

Instruct[3], the training portion of MedQA (Jin et al., 2020), Finance Alpaca[4], Code Alpaca[5] and Alpaca (Taori et al., 2023), denoted as $\mathcal{D}_{train} = \{(D^j_{train}, P_j)\}^6_{j=1}$, to represent SFT datasets with respect to law, medicine, finance, science, code as well as general capabilities. In order to prevent domain overlap, we curated the Alpaca dataset by excluding data pertaining to the other specific five domains, keeping only the general domain instances unrelated to them. More details can be found in Section C.2. For *evaluation*, we assess the model performances on downstream tasks across various domains, using two relevant benchmarks for each domain, with details provided in Section C.3.

Models and Implementation. We employ LLaMA (Dubey et al., 2024; Touvron et al., 2023a,b) and Qwen (Bai et al., 2023; Yang et al., 2024) series as our pretrained language models M_θ, including base models as well as instruction-tuned models for real-world applications. During the fine-tuning procedure, we utilized a learning rate scheduler featuring linear warm-up and cosine decay, peaking at a learning rate of 2e-5, alongside a warmup ratio of 0.03, a weight decay of 0.0 and a batch size of 128 for 4 epochs. To maintain consistency, the total volume of training data across domains was controlled to 60,000 per epoch. We conducted all fine-tuning and evaluation experiments on NVIDIA RTX H800. Details of the experimental settings can be found in Section C.

3.2 Baselines

We compare VersaTune with the following baselines. For the scenario of *effective multi-ability fostering*: (1) The simplest baseline is **uniform distribution**, where each domain has an equal weight

[3]https://huggingface.co/datasets/Alignment-Lab-AI/Lawyer-Instruct

[4]https://huggingface.co/datasets/gbharti/finance-alpaca

[5]https://github.com/sahil280114/codealpaca

proportion. (2) **Inverse distribution** assigns the proportionate weights to each domain in an inverse manner to the detected knowledge distribution. (3) **Frontier models** contain GPT-4o (Hurst et al., 2024), Claude3.5-Sonnet (Anthropic, 2024) and DeepSeek-V3 (Liu et al., 2024a). Under the case of *flexible domain expansion*: (1) **100% specific domain** strategy is a common practice to employ datasets consisting exclusively of data from a single domain during the fine-tuning stage. (2) **Domain increase with uniform distribution of remainder** elevates the proportion of a specific domain, while the remaining domains receive the balance of the distribution in an evenly distributed manner.

3.3 Results

We conduct evaluations to validate the efficiency of VersaTune across different open-source models in scenarios that encompass both effective multi-ability fostering and flexible domain expansion. We summarize the observations below.

VersaTune is efficient across different models in both scenarios. For the scenario of multi-capability fostering, Table 1 shows that VersaTune consistently outperforms other baseline methods across different models in terms of domain-specific capabilities. Compared to the *uniform distribution* of data across domains, VersaTune enhances downstream task performances by 29.77%, which further underscores the effectiveness of our data composition strategy for enhancing the model's overall multi-domain capabilities during the supervised fine-tuning phase. Moreover, **Qwen-2.5-32B + VersaTune** has the potential to surpass frontier models under medical scenarios, achieving average improvements over GPT-4o, Claude3.5-Sonnet and DeepSeek-V3 by 0.86%, 4.76% and 4.60%. Since we have not conducted domain-specific refinement for domains outside the current five specific domains, the models' performance gains on general benchmarks are not as noticeable. For domain expansion scenarios, VersaTune has nearly maintained training efficiency while reducing the model's loss of competencies in other domains by 38.77% comparing to *100% specific domain fine-tuning*, as depicted in Table 8, where we averaged the experimental results from Qwen-2.5-7B and Qwen-2.5-14B. Detailed results and analysis can be found in Section D.

Knowledge consistency training boosts performance. In Table 1, we present the experimental results of data composition strategies that allocate

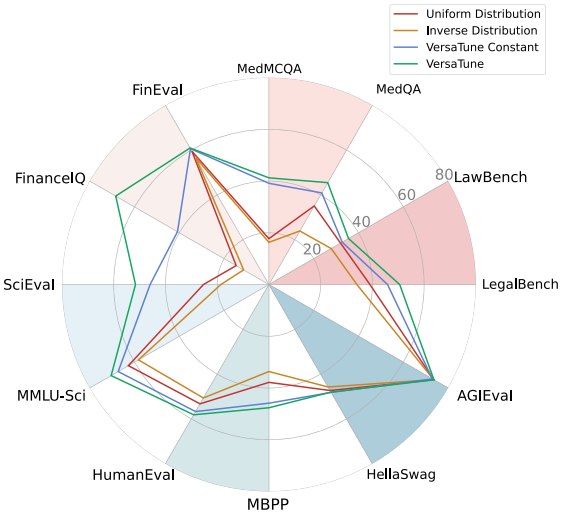

Figure 2: Performances of ***Qwen-2-7B*** on versatile tasks across different domains for multi-ability fostering.

domain data in a manner inversely proportional to the pre-existing knowledge distribution detected within each domain. As expected, the *inverse distribution* strategy yielded even lower↓ performance compared to the simplest approach of *uniform distribution*, which evenly distributes data across all domains. We have also conducted a comparison involving *the addition of stochastic perturbations to the detected knowledge distribution*, with the results presented in Table 7. This finding underscores the importance of aligning domain data ratios with the inherent knowledge distribution of the model during training, which proves the efficacy of knowledge consistency training stated in Section 2.1.1.

4 Ablations and Analysis

Previously in Section 3, we have demonstrated the effectiveness of VersaTune in enhancing multiple abilities and enabling flexible domain expansion during the SFT phase. In this section, we perform an in-depth analysis of VersaTune, where we ablate the components of ***(1)*** dynamic adaptation in Algorithm 2, and ***(2)*** the criteria for determining the upper limit of domain expansion in Algorithm 3.

Dynamic adjustment enhances the robustness. During the process of cultivating multiple capabilities, we compared VersaTune with *fixed domain weights* referring to the knowledge distribution obtained from probing the target model M_θ prior to supervised fine-tuning, namely ***VersaTune Constant***, to ablate the component of dynamic adaptation in Algorithm 2. Table 5, Figure 2, and Figure 8 demonstrated the high robustness of VersaT-

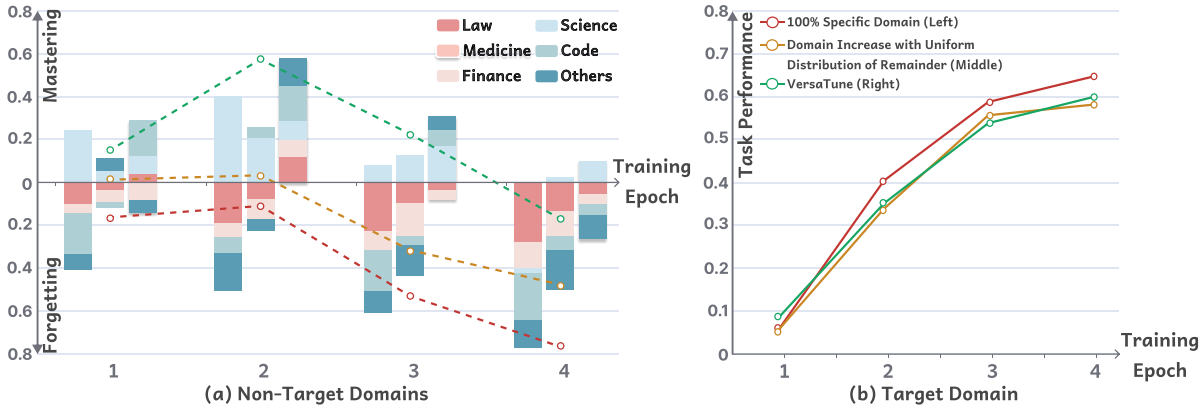

Figure 3: Domain expansion for *medicine* domain. We evaluated checkpoints from each epoch. **Left (a)** presents the grouped stacked bar chart showing the growth or loss of capabilities in non-target domains compared to the pre-fine-tuning state. Within each group, the left, center, and right bars represent: (1) 100% specific domain fine-tuning, (2) domain increase with uniform distribution of remainder, and (3) VersaTune implementation based on Algorithm 3. **Right (b)** features the line chart depicting the enhancement of the medicine domain's capabilities.

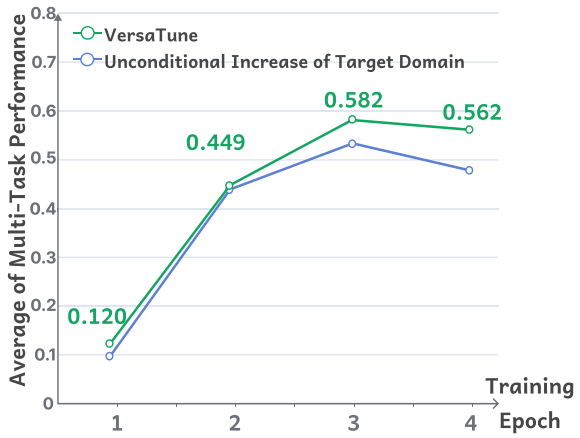

Figure 4: The average scores of models' performances across domains during the domain expansion process, with detailed domain variations provided in Figure 14.

une, which dynamically adjusts domain weights throughout the training process by continuously monitoring the learnable potential within each domain. In contrast, training with fixed domain weights exhibits certain fluctuations. A key reason for this phenomenon is the distribution of domain knowledge mastered by the model changes during training, and the learning efficiency varies among domains. Therefore, dynamically adjusting domain data weights based on the model's feedback at different stages of training is crucial. More experimental results can be found in Section D.1.

Establishing proportion thresholds for specific domains counts during domain expansion. We consider conducting a comparative analysis between the outcomes of VersaTune and those implementing an *unconditional dynamic increase of the specific domain*, where we remove the implementation of Line 8 in Algorithm 3. Figure 4 shows that the criteria for determining the upper limit on the proportion of a specific domain during domain expansion, has mitigated the loss of capabilities in other domains experienced by the target model M_θ during the fine-tuning process. Concurrently, it ensures gains in the capacity for the current domain of interest. We speculate that it is because by the later stages of fine-tuning, models' proficiency in the target domain approaches saturation. Further increasing the proportion of the current domain provides diminishing returns and can lead to a significant loss of performance in other domains. Detailed analysis are provided in Section D.2.2.

5 Related Work

Data Reweighting for LLM Training. Data reweighting maintains full access to the entire dataset while adjusts the relative importance of each instance for various target tasks, which is essential for both pretraining and fine-tuning stages of LLMs (Wang et al., 2023). During the pretraining stage, DoReMi (Xie et al., 2024b) and DoGE (Fan et al., 2023) employ lightweight proxy models to estimate weights for different data sources, which are subsequently applied to the formal training of LLMs. Furthermore, Sheared LLaMA (Xia et al., 2023) implements an online variant of DoReMi. As for the SFT phase, Dong et al. (Dong et al., 2023) focus on enhancing the model's math reasoning, coding, and human-aligning abilities through

a dual-stage mixed fine-tuning strategy. However, the mixing ratios for different domains rely heavily on empirical methods, and the covered domains are not holistic. We provide a comprehensive overview of the model's capabilities across domains during the SFT stage and proposes appropriate and holistic multi-ability fine-tuning methods.

Knowledge Detection in LLMs. Investigating the knowledge contained in LLMs is essential for guiding their subsequent training (Chang et al., 2024). The knowledge encompasses multiple dimensions, such as different domain sources and task attributes. Existing work on LLM knowledge detection primarily focuses on prompting and calibration. Directly prompting the model to generate sequences and extracting confidence scores from the model (Gekhman et al., 2024; Kadavath et al., 2022; Kuhn et al., 2023; Manakul et al., 2023) is a common strategy. However, such approaches highly depend on prompt design and task selection, introducing bias into the assessment. Other studies have attempted to infer the training data mixtures used in previous training stages (Antoniades et al., 2024; Hayase et al., 2024; Hu et al., 2022; Ye et al., 2022). The essence of these studies is to evaluate the current knowledge state of the models and provide targeted strategies for data organization and management in subsequent training phases.

6 Conclusion

In this work, we introduce VersaTune, a noval data composition framework designed to enhance the multi-domain capabilities of models during the supervised fine-tuning phase of LLMs, which is based on the domain knowledge distribution of the target model. Experimental results have demonstrated that VersaTune achieves excellent training outcomes in both scenarios of overall multi-domain enhancement and flexible domain expansion.

Acknowledgments

This work is supported by National Natural Science Foundation of China (92470121, 62172015, 62402016), National Key R&D Program of China (2024YFA1014003), and High-performance Computing Platform of Peking University.

Limitations

There are some limitations in our work. Firstly, the classification framework of vertical domains may not be comprehensive in scope, and since the classifier relies on advanced language models, it cannot guarantee absolute accuracy in classification. Additionally, when computing the learnable potential and forgetting degree of knowledge, to balance the computational cost and effectiveness, we employ a lightweight proxy model to for calculation, yet it does not fully represent the performance tendencies of the target model during actual evaluating.

Ethical Considerations

Integrating Large Language Models (LLMs) into domain reweighting settings holds potential for improving multi-domain capabilities of models, while it also brings several ethical considerations that must be addressed to ensure responsible and beneficial use. VersaTune dynamically adjusts data distribution based on the model's existing knowledge to ensure fairness and avoid biases that could arise from the data composition process. Additionally, VersaTune adhere to privacy standards by merely utilizing open-sourced datasets, ensuring that personal data used in the training process is anonymized and securely handled to protect individual privacy rights.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, and 1 others. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

Anthropic. 2024. Claude. Accessed: 2024-06-27.

Antonis Antoniades, Xinyi Wang, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2024. Generalization vs memorization: Tracing language models' capabilities back to pretraining data. *arXiv preprint arXiv:2407.14985*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang,

Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. 2022. Source-free domain adaptation via distribution estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7202–7212.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yogesh K Dwivedi, Laurie Hughes, Elvira Ismagilova, Gert Aarts, Crispin Coombs, Tom Crick, Yanqing Duan, Rohita Dwivedi, John Edwards, Aled Eirug, and 1 others. 2021. Artificial intelligence (ai): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International journal of information management*, 57:101994.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2023. Doge: Domain reweighting with generalization estimation. *arXiv preprint arXiv:2310.15393*.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. Lawbench: Benchmarking legal knowledge of large language models. *arXiv preprint arXiv:2309.16289*.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *arXiv preprint arXiv:2405.05904*.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.

Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, and Noah A Smith. 2024. Data mixture inference: What do bpe tokenizers reveal about their training data? *arXiv preprint arXiv:2407.16607*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Di Jin, 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*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli

Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Prakhar Kaushik, Alex Gain, Adam Kortylewski, and Alan Yuille. 2021. Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping. *arXiv preprint arXiv:2102.11343*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023a. Trocr: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13094–13102.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023b. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 374–382.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2024b. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Xiaonan Nie, Yi Liu, Fangcheng Fu, Jinbao Xue, Dian Jiao, Xupeng Miao, Yangyu Tao, and Bin Cui. 2023. Angel-ptm: A scalable and economical large-scale pre-training system in tencent. *Proceedings of the VLDB Endowment*, 16(12):3781–3794.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. Scieval: A multi-level large language model evaluation benchmark for scientific research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19053–19061.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.

Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Data management for large language models: A survey. *arXiv preprint arXiv:2312.01700*.

Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. 2025. Grit: A generative region-to-text transformer for object understanding. In *European Conference on Computer Vision*, pages 207–224. Springer.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, and 1 others. 2024a. The finben: An holistic financial benchmark for large language models. *arXiv preprint arXiv:2402.12659*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2024b. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36.

Zhuoyan Xu, Zhenmei Shi, and Yingyu Liang. 2024. Do large language models have compositional ability? an investigation into limitations and scalability. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Hype: Better pre-trained language model fine-tuning with hidden representation perturbation. *arXiv preprint arXiv:2212.08853*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning. *arXiv preprint arXiv:2401.07950*.

Liwen Zhang, Weige Cai, Zhaowei Liu, Zhi Yang, Wei Dai, Yujie Liao, Qianru Qin, Yifei Li, Xingyu Liu, Zhiqiang Liu, and 1 others. 2023. Fineval: A chinese financial domain knowledge evaluation benchmark for large language models. *arXiv preprint arXiv:2308.09975*.

Xuanyu Zhang and Qing Yang. 2023. Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 4435–4439.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023a. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2023b. Knowing what llms do not know: A simple yet effective self-detection method. *arXiv preprint arXiv:2310.17918*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Appendix

A Background and Discussion

In this section, we provide the background information and design motivation for our VersaTune.

A.1 Pretraining and Supervised Fine-Tuning

The training process of Large Language Models (LLMs) generally involves the pretraining and fine-tuning stages. We have outlined several concepts about LLMs training.

A.1.1 Pretraining

Large Language Models (LLMs) establish basic knowledge abilities, including language understanding and text generation, during the pretraining stage (Brown et al., 2020). In this stage, LLMs engage in unsupervised training through the processing of extensive raw text corpora, thereby enhancing their capabilities in language modeling. For a given sequence $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, the typical task for LLMs involves the prediction of the subsequent token x_i given the preceding tokens $\mathbf{x}_{<i}$ as contextual input. The goal is to maximize the likelihood function presented in Equation (6):

$$\mathcal{L}_{LLM}^{PT}(\mathbf{x}) = \sum_{i=1}^{n} \log P(x_i|\mathbf{x}_{<i}) \tag{6}$$

Beginning of Sequence (<BOS>) During the above process, the Beginning of Sequence (<BOS>) token plays an important role, which serves as a signal to the model that the input sequence is starting (Brown et al., 2020; Li et al., 2023a; Wu et al., 2025). It can be thought of as a special marker that indicates the start of a new sequence, allowing the model to reset its context and begin processing a new piece of text. In the context of pretraining, the <BOS> token is used to initialize the input to the model, and it can be concatenated with the actual text data to form the input sequence. This token helps the model to differentiate between the start of a new input and the continuation of an existing one. It

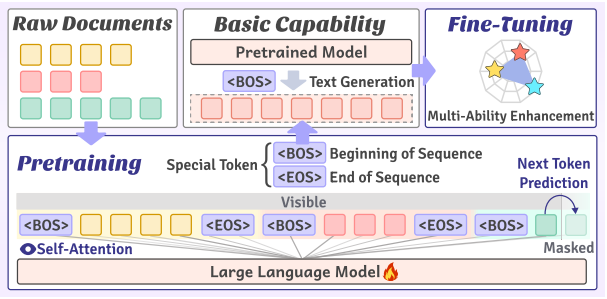

Figure 5: Illustration of the LLMs training workflow. In the pretraining phase, raw documents are concatenated into a sequence using special tokens such as <BOS> (Beginning of Sequence) and <EOS> (End of Sequence), thereby endowing the LLM with fundamental language generation capabilities. In the fine-tuning phase, the model's abilities in various domains are further enhanced.

is particularly crucial in tasks where *the model needs to generate text* or understand the beginning of a new sentence or document, which helps the model to learn the boundaries of text sequences and to better model the statistical properties of the language data it is trained on. The use of <BOS> tokens, along with other special tokens like *<EOS> (End of Sequence)*, helps the model to learn the boundaries of text sequences and to better model the statistical properties of the language data it is trained on.

A.1.2 Supervised Fine-Tuning

The Supervised Fine-Tuning (SFT) stage of a Large Language Model (LLM) involves further training to refine the model's task-solving capabilities and ensure greater alignment with human instructions (Zhao et al., 2023a). While recent research has delved into exploring fine-tuning methods for multi-task enhancement (Dong et al., 2023; Sanh et al., 2021), they are still in their early stages. However, as shown by proprietary models such as GPT-4 (Achiam et al., 2023), Gemini (Team et al., 2023), and DeepSeek series (Liu et al., 2024a), which exhibit outstanding multi-task performance, improving a model's versatile capabilities across various domains during the SFT phase is crucial. Therefore, our work systematically investigates methods to *enhance multi-domain performance* during the SFT stage to bridge this gap.

A.2 Analysis on Catastrophic Forgetting

During the SFT phase, it is a typical practice to employ datasets specific to a particular domain for the fine-tuning of LLMs, which may lead to a significant performance drop of knowledge in non-target domains, a phenomenon commonly referred to as Catastrophic Forgetting (Kaushik et al., 2021; McCloskey and Cohen, 1989). We conducted experiments on open-sourced models including LLaMA (Dubey et al., 2024; Touvron et al., 2023a,b) and Qwen (Bai et al., 2023; Yang et al., 2024) series to assess how the model's proficiency in other domains changes when fine-tuned with data from a single domain, as depicted in Table 2 and Figure 6. We have regulated the number of training instances per epoch to a fixed count of 10,000. More details on training and evaluation settings can be found in Section 3.1 and Section C. Our findings indicate that *when a model is trained exclusively with data from* ***a single domain****, its performance on tasks from other domains tends to* ***degrade progressively*** *over the course of training*. This experimental outcome has provided significant motivation and direction for our work.

Target Domain	Training Step (Epoch)	Variations in Comprehensive Domains (%)						Sum. (%)
		Law	Medicine	Finance	Science	Code	Other	
Law	1	-	↓18.82	↑14.71	↓11.76	↓11.18	↓5.00	↓32.05
	2	-	↓12.65	↑30.59	↓4.41	↓5.29	↓11.76	↓3.52
	3	-	↓17.94	↑12.35	↓8.82	↓23.53	↓5.00	↓42.94
	4	-	↓5.00	↓2.06	↓31.18	↓21.76	↓12.65	**↓72.65**
Medicine	1	↓10.29	-	↓3.82	↑24.12	↓19.41	↓7.35	↓16.75
	2	↓18.82	-	↓6.47	↑40.00	↓7.94	↓17.65	↓10.88
	3	↓22.35	-	↓8.82	↑7.94	↓19.12	↓10.00	↓52.35
	4	↓27.94	-	↓11.76	↓2.35	↓21.76	↓12.65	**↓76.46**
Finance	1	↑20.59	↓7.94	-	↓10.29	↓12.65	↓6.47	↓16.76
	2	↑18.24	↓9.71	-	↓9.41	↑5.29	↓8.82	↓4.41
	3	↑23.53	↓9.41	-	↓17.35	↓14.71	↓7.94	↓25.88
	4	↑5.00	↓9.12	-	↓20.29	↓12.94	↓21.76	**↓59.11**
Science	1	↓10.29	↑17.06	↓3.82	-	↓4.71	↓7.35	↓9.11
	2	↓11.47	↑12.35	↓4.71	-	↓5.88	↓12.94	↓22.65
	3	↓21.76	↑7.94	↓8.82	-	↓4.41	↓10.00	↓37.05
	4	↓27.94	↑2.35	↓11.47	-	↓12.65	↓12.59	**↓62.30**
Code	1	↓3.82	↑7.35	↓17.35	↑9.12	-	↓7.29	↓11.99
	2	↓9.71	↓6.47	↓7.94	↑5.29	-	↓6.18	↓25.01
	3	↓22.35	↓8.82	↓14.12	↑7.94	-	↓10.02	↓47.37
	4	↓26.18	↓7.06	↓8.82	↓3.24	-	↓22.65	**↓67.95**

Table 2: Variations in models' performance on non-target domain tasks when trained on single sourced dataset. ↑ and ↓ indicate an increase or decrease in the percentage of scores (%) compared to the *initial state* before fine-tuning.

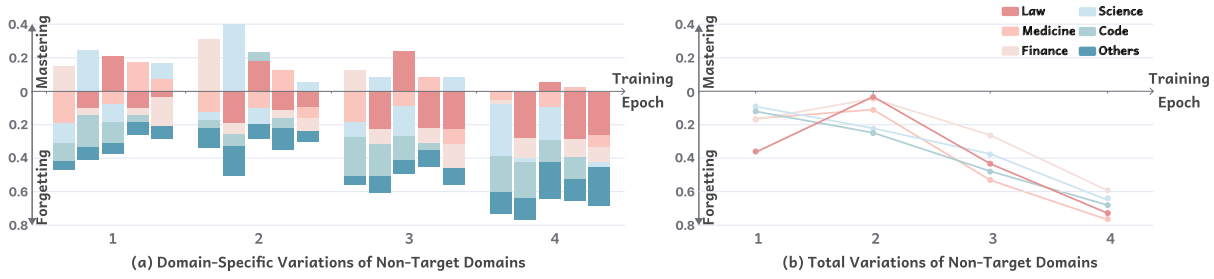

Figure 6: Illustration of variations in models' performance on non-target domain tasks when trained on a single-domain dataset. The grouped stacked bar chart on the **left (a)** describes the detailed changes in performance across various non-target domains as training progresses. Each group of stacked bars, from left to right, represents the use of training datasets from *law*, *medicine*, *finance*, *science*, and *code*, respectively. The line chart on the **right (b)** shows the overall performance changes in all non-target domains. The color of each line indicates the domain from which the training dataset was sourced.

B Method Details

B.1 Algorithm for Flexible Domain Expansion

Here we provide the detailed algorithm of flexible domain expansion (Section 2.2.3). As outlined in Algorithm 3, we initially establish the data distribution based on the knowledge detected from the original pretrained model (Line 1). At each training step t, we calculate the learnable potential and forgetting degree scores for each domain (Line 4-5), and assign domain weights for the current training phase following the method from Algorithm 2 (Line 6). A trade-off is necessary between the remaining learning margin of the domain that requires focused cultivation and the model's forgetting degree towards other non-target domains. If the improvement benefit of the specific domain exceeds the average forgetting degree of the other domains (ratio greater than ε), we increase the data weight of the current specific domain by δ, and proportionally reduce the weights of the other non-target domains according to Equation (5) (Line 8-9). Otherwise, we maintain the current domain distribution and only perform minor adjustments and renormalization as described in Algorithm 2 (Line 10-11). Subsequently, we update the parameters of the target model M_θ (Line 13).

Algorithm 3 VersaTune Multi-Ability Fine-Tuning (for Domain Expansion)

Input: Base model to be fine-tuned $M_\theta^{(0)}$, Domains that require enhanced cultivation D_e, Domain reference loss $\{\ell_{ref}^j\}_{j=1}^k$, Hyperparameters: number of training steps T, magnitude of adjustment σ, extent of domain adjustment δ, variation threshold ε
Parameter: Data proportion $\{P_j\}_{j=1}^k$ of the SFT dataset
Output: Fine-tuned multi-ability model $M_\theta^{(T)}$
Define γ: learnable potential of the current domain
Define φ: forgetting degree of the current domain

1: Initialize domain proportion $\{P_j^{(0)}\}_{j=1}^k$ according to Equation (1) and Algorithm 1
2: **for** $t = 1, 2, \ldots, T$ **do**
3: **for** $j = 1, 2, \ldots, k$ **do**
4: Learnable potential for the j-th domain: $\gamma_j^{(t)} = \max\{\frac{\ell^j_{\theta(t)} - \ell^j_{ref}}{\ell^j_{\theta(t)}}, 0\}$
5: Forgetting degree for the j-th domain: $\varphi_j^{(t)} = \max\{\frac{\ell^j_{\theta(t)} - \ell^j_{\theta(t-1)}}{\ell^j_{\theta(t-1)}}, 0\}$
6: Update domain weights: $P_j^{(t)\prime} = P_j^{(t-1)}(1 + \sigma\gamma_j^{(t)})$
7: **end for**
8: **if** $\frac{1}{k}\sum_{j=1, j\neq e}^k \varphi_j^{(t)} < \varepsilon\gamma_e^{(t)}$ **then**
9: Update specific domain weight:

$$P_j^{(t)} = \begin{cases} P_j^{(t-1)} + \delta, \text{if} \quad j = e \\ \frac{P_j^{(t)\prime}}{\sum_{i=1, j\neq e}^k P_i^{(t)\prime}}(1 - P_j^{(t-1)} - \delta), \text{others} \end{cases}$$

10: **else**
11: Renormalize domain weights: $P_j^{(t)} = \frac{P_j^{(t)\prime}}{\sum_{i=1}^k P_i^{(t)\prime}}, \quad \forall j \in \{1, 2, ..., k\}$
12: **end if**
13: Update parameters of fine-tuned model $M_\theta^{(t)}$
14: **end for**
15: **Return** Fine-tuned model $M_\theta^{(T)}$

C Experiments Details

C.1 Knowledge Distribution Detection

During the knowledge distribution detection phase for our target models, we have manually annotated 120 samples (20 samples for each domain) to fine-tune Qwen2.5-72B-Instruct[6], and employed the trained model as the proprietary model M_P. For each target model M_θ slated for supervised fine-tuning, we prompted the generation of $40K$ data samples using the Beginning of Sequence ($< BOS >$) token, with the sample number set at $N_S = 40,000$. These samples were subsequently assessed by the proprietary model M_P to ascertain their probabilistic affinity for several domains, including *law*, *medicine*, *finance*, *science*, *code*, and *others*. To ensure the reliability of our statistical outcomes, the entire process was iterated 5 times, with the maximum number of iterations set at $T = 5$. The average knowledge distribution was then computed across these iterations. Empirically, with a dataset of $40K$ samples, the distribution of sequences generated by M_θ across domains demonstrated a high degree of consistency, with an overall variance not exceeding 1.874%. The final domain knowledge distribution for each open-source model is depicted in the stacked bar chart presented in Figure 7. The pre-existing domain knowledge distribution varies among different models. Therefore, it is essential to develop a data composition strategy that is tailored to the specific model being trained.

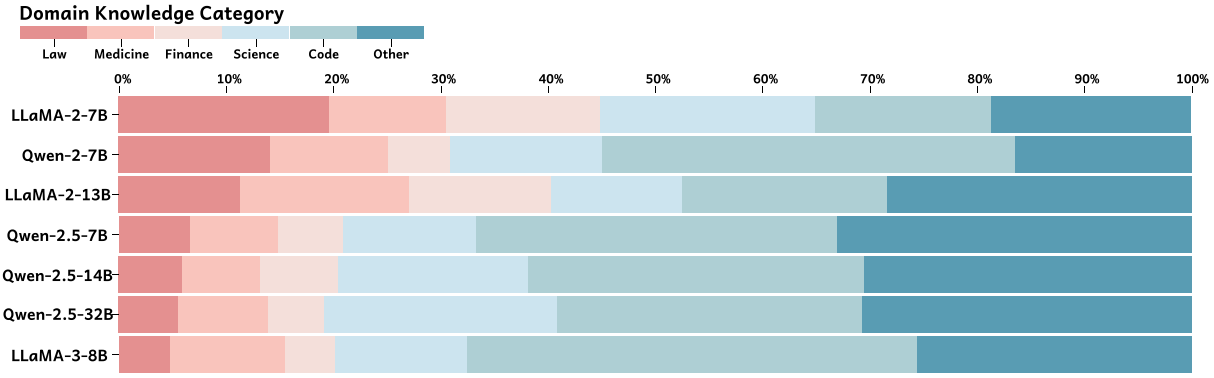

Figure 7: An illustration of the domain knowledge distribution among models.

C.2 Training Details

Models and Implementation. All experiments were conducted based on full-parameter fine-tuning, during which we utilized a learning rate scheduler featuring linear warm-up and cosine decay, peaking at a learning rate of 2e-5, alongside a warmup ratio of 0.03, a weight decay of 0.0 and a batch size of 128 for 4 epochs. For scenarios aimed at *fostering multi-ability*, we trained and assessed models including LLaMA-2-7B, LLaMA-3-8B, LLaMA-2-13B, Qwen-2-7B, Qwen-2.5-7B, Qwen-2.5-14B, and Qwen-2.5-32B. In the context of *domain expansion*, the training and evaluations were performed using the Qwen-2.5-7B and Qwen-2.5-14B models. The total number of samples per epoch was set to $60k$, with each domain's samples being downsampled or upsampled according to the corresponding weights during the mixing process. Regarding *reference models*, for the LLaMA series, we used the Sheared-LLaMA-1.3B (Xia et al., 2023) as a lightweight reference model; as for the Qwen series, we utilized Qwen-2-1.5B and Qwen-2.5-1.5B as our reference models.

Training Datasets. For training, we selected representative datasets for each domain, which exhibit significant differences in format, sentence length, and domain-specific content. These differences reflect the heterogeneity of training data across various domains during the fine-tuning stage. Further details about these datasets can be found in Table 3. Specifically, for the Alpaca dataset, which we utilize for

[6]https://huggingface.co/Qwen/Qwen2.5-72B-Instruct

representing the general domain, we have excluded data related to law, medicine, finance, science, and code domains to ensure the precision and authenticity of the actual domain weight.

Dataset	# Instance	Source	# Rounds	Full
Lawyer-Instruct	9241	Reformatted from LawyerChat Dataset[7]	1	√
MedQA	10178	Professional Medical Board Exams	1	Training Portion
Finance Alpaca	68912	Alpaca, FiQA, 1.3k Pairs Generated using GPT3.5	1	√
Sonnet3.5 Science Conversations	8835	Scientific Conversations with Sonnet3.5	11.1	√
Code Alpaca	20022	Generate Based on Self-Instruct (Wang et al., 2022)	1	√
Alpaca	49,087	Generate Based on Self-Instruct (Wang et al., 2022)	1	Excluding Samples of Other Domains

Table 3: Details of the training datasets. "Full" indicates whether we utilize the entire data samples of the dataset.

C.2.1 Hyper-Parameters Setting

During the multi-domain task fine-tuning of LLMs, we configured the number of training steps T in Algorithm 2 and Algorithm 3 to 4 epochs. We experimented with various magnitude of adjustment, specifically $[0.1, 0.3, 0.5, 0.8, 1.0]$, and observed consistent weight ordering across domains, which far outperformed our baselines (detailed in Section 3.2). Based on these experimental outcomes, we set the magnitude of adjustment σ to 0.5. Additionally, in the context of domain expansion, we set the increment for the target domain δ to 10% per training step, considering the overall domain weight distribution across models. The variation threshold, denoted as ε, reflects the trade-off between enhancing specific domain skills and mitigating the loss of capabilities in non-target domains, where we assigned a weight of 1.

C.3 Evaluation Details

We evaluate the performance of the models on downstream tasks across various domains, using two relevant benchmarks for each domain. Details of the datasets are provided in Table 4. Specifically, for the MedMCQA dataset, since the standard answers for the test set are not publicly available, we conducted our evaluations using the validation dataset. For the MMLU dataset, we selected 14 sub-tasks to construct the MMLU-Sci subset (Zhang et al., 2024) for testing, aiming to ensure a robust and thorough evaluation.

Domain	Benchmark	# Instance	Language	N-Shot
Law	LegalBench (Guha et al., 2024)	90,394 (164 sub-tasks)	English	1
	LawBench (Fei et al., 2023)	10,000 (20 sub-tasks)	Chinese	1
Medicine	MedQA (Jin et al., 2020)	1,273	English	1
	MedMCQA (Pal et al., 2022)	4,183	English	1
Finance	FinEval (Zhang et al., 2023)	4,661 (34 sub-tasks)	Chinese	1
	FinanceIQ (Zhang and Yang, 2023)	7,173 (10 sub-tasks)	Chinese	5
Science	SciEval (Sun et al., 2024)	15,901	English	1
	MMLU-Sci (Hendrycks et al., 2020)	2,999 (14 sub-tasks)	English	0
Code	HumanEval (Chen et al., 2021)	164	English	0
	MBPP (Austin et al., 2021)	974	English	0
Other (General)	AGIEval (Zhong et al., 2023)	8,062 (20 sub-tasks)	English, Chinese	0
	HellaSwag (Zellers et al., 2019)	10,003	English	0

Table 4: Details of the benchmarks we employed for evaluation. "N-Shot" indicates that the model is given N example(s) to understand and perform the task.

D More Experiment Results

D.1 Multi-Ability Fostering

We present the results of the ablation study along with the raw scores from domain-specific benchmarks. As shown in Table 5 and Figure 8, **VersaTune Constant** is implemented with fixed domain weights derived from the knowledge distribution obtained by probing the target model M_θ prior to fine-tuning, where we ablate the components of dynamic adaptation in Algorithm 2 for an in-depth analysis. Additionally, to demonstrate the robustness of the dynamic adjustment more clearly, we compare the domain-level averaged performance of the *VersaTune constant* and *VersaTune* strategies, as depicted in Table 6.

Model	Method	Law		Medical		Finance		Science		Code		General	
		LegalBench	LawBench	MedQA	MedMCQA	FinEval	FinanceIQ	SciEval	MMLU-Sci	HumanEval	MBPP	AGIEval	HellaSwag
Open-Sourced Base Models													
LLaMA-2-7B	Uniform Distribution	15.71	30.72	23.45	27.57	33.50	2.71	9.30	42.89	5.67	3.44	20.16	71.40
	Inverse Distribution	13.23↓	26.94↓	21.38↓	26.52↓	32.96↓	2.53↓	8.98↓	39.67↓	3.47↓	2.42↓	18.83↓	71.33↓
	VersaTune Constant	21.47↑	35.55↑	30.17↑	36.72↑	35.89↑	6.28↑	49.91↑	45.87↑	12.47↑	14.47↑	22.31↑	**71.89**↑
	VersaTune	**23.18**↑	**36.31**↑	**35.04**↑	**40.75**↑	**36.27**↑	**29.04**↑	**56.75**↑	**50.06**↑	**15.62**↑	**15.68**↑	**24.67**↑	71.76↑
Qwen-2-7B	Uniform Distribution	39.05	31.99	35.07	17.73	59.49	14.62	25.30	62.73	53.26	37.82	47.31	73.60
	Inverse Distribution	34.01↓	27.81↓	23.90↓	16.31↓	56.53↓	11.30↓	18.57↓	58.25↓	50.65↓	33.63↓	45.74↓	73.52↓
	VersaTune Constant	45.86↑	32.72↑	40.89↑	39.13↑	60.63↑	40.82↑	45.93↑	67.29↑	56.71↑	45.87↑	**48.16**↑	72.98↓
	VersaTune	**50.56**↑	**35.54**↑	**45.48**↑	**41.24**↑	**60.95**↑	**68.39**↑	**51.58**↑	**70.42**↑	**58.15**↑	**47.64**↑	48.02↑	**73.67**↑
Qwen-2.5-7B	Uniform Distribution	40.11	31.48	25.17	25.84	59.58	31.66	19.88	65.84	55.64	46.86	45.42	73.69
	Inverse Distribution	36.36↓	26.98↓	24.16↓	19.35↓	57.07↓	29.25↓	16.68↓	62.78↓	52.97↓	44.63↓	45.67↑	72.92↓
	VersaTune Constant	48.78↑	35.20↑	30.20↑	49.71↑	**62.94**↑	48.47↑	56.04↑	71.96↑	59.15↑	52.10↑	47.75↑	73.88↑
	VersaTune	**51.65**↑	**36.75**↑	**34.28**↑	**52.09**↑	62.48↑	**69.09**↑	**68.14**↑	**74.16**↑	**60.68**↑	**61.25**↑	**49.73**↑	**73.90**↑
LLaMA-3-8B	Uniform Distribution	33.52	31.16	31.03	10.26	34.83	4.97	6.51	50.17	22.94	28.85	23.87	73.26
	Inverse Distribution	27.83↓	27.48↓	25.51↓	8.77↓	33.71↓	3.31↓	6.09↓	46.62↓	19.67↓	24.34↓	23.45↓	72.40↓
	VersaTune Constant	47.85↑	37.75↑	37.33↑	30.15↑	37.93↑	25.27↑	54.77↑	56.04↑	**29.88**↑	33.22↑	25.62↑	73.33↑
	VersaTune	**49.67**↑	**37.87**↑	**42.21**↑	**45.72**↑	**38.80**↑	**43.58**↑	**56.67**↑	**60.61**↑	28.91↑	**35.65**↑	**28.78**↑	**73.62**↑
LLaMA-2-13B	Uniform Distribution	47.66	34.85	32.98	36.54	37.54	32.85	45.72	50.77	36.54	38.55	36.89	73.50
	Inverse Distribution	40.12↓	30.67↓	26.27↓	28.78↓	36.67↓	26.76↓	38.96↓	48.68↓	28.78↓	35.83↓	36.67↓	73.11↓
	VersaTune Constant	53.79↑	38.73↑	40.69↑	42.74↑	39.33↑	38.47↑	57.13↑	55.10↑	42.74↑	42.76↑	37.91↑	74.27↑
	VersaTune	**55.87**↑	**40.14**↑	**45.78**↑	**47.67**↑	**39.48**↑	**55.12**↑	**63.87**↑	**62.84**↑	**47.67**↑	**44.62**↑	**39.64**↑	**74.63**↑
Qwen-2.5-14B	Uniform Distribution	50.73	39.49	47.85	38.71	64.72	64.39	39.74	73.45	68.75	72.14	54.92	75.88
	Inverse Distribution	46.08↓	35.36↓	45.75↓	32.56↓	64.88↑	60.53↓	27.68↓	68.22↓	63.36↓	68.49↓	54.87↓	75.42↓
	VersaTune Constant	56.94↑	45.64↑	48.11↑	41.64↑	65.03↑	73.24↑	48.31↑	78.46↑	78.72↑	78.33↑	55.04↑	**76.45**↑
	VersaTune	**60.59**↑	**46.58**↑	**50.24**↑	**45.15**↑	**65.84**↑	**78.68**↑	**62.89**↑	**82.86**↑	**82.64**↑	**81.48**↑	**55.52**↑	75.98↑
Qwen-2.5-32B	Uniform Distribution	68.86	45.28	72.34	68.18	68.03	75.14	58.30	80.17	78.59	71.04	75.26	84.40
	Inverse Distribution	62.93↓	42.05↓	68.80↓	66.09↓	66.80↓	73.93↓	52.94↓	79.31↓	74.44↓	70.71↓	75.00↓	83.80↓
	VersaTune Constant	71.98↑	54.93↑	75.42↑	71.05↑	69.38↑	77.07↑	65.87↑	82.11↑	82.38↑	77.16↑	74.97↓	**84.82**↑
	VersaTune	**75.67**↑	**56.76**↑	**78.72**↑	**72.36**↑	**70.50**↑	**78.80**↑	**70.77**↑	**85.23**↑	**86.60**↑	**79.89**↑	**75.81**↑	84.75↑
Open-Sourced Instruct Models													
Qwen-2.5-7B -Instruct	Uniform Distribution	45.56	33.67	33.75	42.83	58.43	38.79	34.64	66.15	59.97	61.48	55.37	71.02
	Inverse Distribution	38.42↓	27.44↓	30.81↓	39.67↓	56.84↓	35.83↓	30.72↓	63.70↓	57.12↓	58.96↓	54.91↓	71.23↑
	VersaTune Constant	52.63↑	38.72↑	40.64↑	55.82↑	63.01↑	50.52↑	58.60↑	69.35↑	63.34↑	68.48↑	56.62↑	**72.50**↑
	VersaTune	**54.81**↑	**41.43**↑	**43.04**↑	**58.65**↑	**64.97**↑	**55.74**↑	**60.78**↑	**71.85**↑	**63.95**↑	**69.70**↑	**58.33**↑	72.15↑
LLaMA-3-8B -Instruct	Uniform Distribution	46.48	35.10	40.74	38.65	38.96	22.97	48.68	59.85	44.76	52.94	42.65	69.45
	Inverse Distribution	43.96↓	31.67↓	37.38↓	34.82↓	35.73↓	20.87↓	44.84↓	55.58↓	40.69↓	50.66↓	**44.81**↑	68.92↓
	VersaTune Constant	54.83↑	42.35↑	48.59↑	44.67↑	41.65↑	43.54↑	62.80↑	65.81↑	52.85↑	57.64↑	43.96↑	68.79↓
	VersaTune	**56.05**↑	**43.76**↑	**52.64**↑	**50.81**↑	**43.17**↑	**48.62**↑	**68.56**↑	**67.74**↑	**54.06**↑	**59.19**↑	43.58↑	**69.67**↑
Qwen-2.5-14B -Instruct	Uniform Distribution	52.85	46.75	55.94	43.58	64.46	68.89	54.84	75.52	81.69	79.05	60.82	77.17
	Inverse Distribution	49.69↓	44.50↓	51.78↓	40.67↓	62.97↓	66.98↓	49.58↓	74.25↓	78.45↓	74.38↓	60.26↓	76.71↓
	VersaTune Constant	56.98↑	56.68↑	62.83↑	60.92↑	65.52↑	74.80↑	60.82↑	81.07↑	84.96↑	84.63↑	**61.41**↑	77.49↑
	VersaTune	**59.87**↑	**58.72**↑	**64.56**↑	**63.85**↑	**65.98**↑	**77.68**↑	**61.38**↑	**81.84**↑	**85.44**↑	**84.90**↑	60.97↑	**77.95**↑

Table 5: Experimental results of VersaTune on multi-ability fostering, we compare the performances of several methods across different base and instruction-tuned models. For each domain, we evaluate the models using two relevant benchmarks. The best and second best results are in **bold** and underlined. Symbols ↑ and ↓ indicate an increase or decrease in downstream scores comparing to the *uniform distribution* strategy.

To further strengthen the robustness of VersaTune as well as the support for Statement 1 (Knowledge Consistency Training), we have conducted experiments on Qwen-2.5-7B and Qwen-2.5-14B with more baselines, employing Qwen2.5-1.5B as the reference model. The additional baselines are as follows:

- ***Knowledge Distribution + Stochastic Perturbations***: We apply controlled, stochastic perturbations (±10%, ±15% and ±20%) to the domain weight distributions detected in the base model M_θ.

- ***Random Mixing Ratio***: Each domain is assigned a random weight during the data sampling procedure of training.
- ***DoReMi*** (Xie et al., 2024b): ***Do***main ***Re***weighting with ***Mi***nimax Optimization (DoReMi) first trains a small proxy model using group distributionally robust optimization (Group DRO) over domains to produce domain weights (mixture proportions) without knowledge of downstream tasks.
- ***DoGE*** (Fan et al., 2023): ***DO***main reweighting with ***G***eneralization ***E***stimation (DoGE) is similar to DoReMi, but focusing more on generalizing to out-of-domain target tasks.

It should be noted that ***DoReMi*** and ***DoGE*** focus on domain reweighting during pretraining, where the domains typically include broad categories such as wikipedia, books, news, web, etc. In contrast, our method is tailored for the SFT scenario and operates under the domain taxonomy including law, medicine, finance, science, code, etc. To enable a fair comparison, we adapted the domain definitions used in *DoReMi* and *DoGE* to align with our domain categorization framework. Following the setup of *DoReMi* and *DoGE*, we initialize domain weights based on the natural data size of each domain.

Model	Method	Law	Medical	Finance	Science	Code	General	Avg.
Open-Sourced Base Models								
LLaMA-2-7B	VersaTune Constant	28.51↓	33.45↓	21.09↓	47.89↓	13.47↓	47.10↓	31.92↓
	VersaTune	29.75	37.90	32.66	53.41	15.65	48.22	36.27
Qwen-2-7B	VersaTune Constant	39.29↓	40.01↓	50.73↓	56.61↓	51.29↓	60.57↓	49.75↓
	VersaTune	43.05	43.36	64.67	61.00	52.90	60.85	54.31
Qwen-2.5-7B	VersaTune Constant	41.99↓	39.96↓	55.71↓	64.00↓	55.63↓	60.82↓	53.02↓
	VersaTune	44.20	43.19	65.79	71.15	60.97	61.82	57.85
LLaMA-3-8B	VersaTune Constant	42.80↓	33.74↓	31.60↓	55.41↓	31.55↓	49.48↓	40.76↓
	VersaTune	43.77	43.97	41.19	58.64	32.28	51.20	45.18
LLaMA-2-13B	VersaTune Constant	46.26↓	41.72↓	38.90↓	56.12↓	42.75↓	56.09↓	46.97↓
	VersaTune	48.01	46.73	47.30	63.36	46.15	57.14	51.45
Qwen-2.5-14B	VersaTune Constant	51.29↓	44.88↓	34.14↓	63.39↓	78.53↓	65.75↓	56.33↓
	VersaTune	53.59	47.70	72.26	72.88	82.06	65.75	65.71
Qwen-2.5-32B	VersaTune Constant	63.46↓	73.24↓	73.23↓	73.99↓	79.77↓	79.90↓	73.93↓
	VersaTune	66.22	75.54	74.65	78.00	83.25	80.28	76.32
Open-Sourced Instruct Models								
Qwen-2.5-7B-Instruct	VersaTune Constant	45.68↓	48.23↓	56.77↓	63.98↓	65.91↓	64.56↓	57.52↓
	VersaTune	48.12	50.85	60.36	66.32	66.83	65.24	59.62
LLaMA-3-8B-Instruct	VersaTune Constant	48.59↓	46.63↓	42.60↓	64.31↓	55.25↓	56.38↓	52.29↓
	VersaTune	49.91	51.73	45.90	68.15	56.63	56.63	54.82
Qwen-2.5-14B-Instruct	VersaTune Constant	56.83↓	61.88↓	70.16↓	70.95↓	84.80↓	69.45↓	69.01↓
	VersaTune	59.30	64.21	71.83	71.61	85.17	69.46	70.26

Table 6: Ablation studies on multi-ability fostering, we compare the performances of *VersaTune* and *VersaTune Constant* across different models. The domain performance scores were calculated as the arithmetic mean of the respective benchmark scores obtained for each domain. "Avg" denotes the average performance across all domain-specific tasks. ↑ and ↓ indicate an increase or decrease in scores comparing to the *VersaTune* strategy.

Model	Method	Law		Medical		Finance		Science		Code		General	
		LegalBench	LawBench	MedQA	MedMCQA	FinEval	FinanceIQ	SciEval	MMLU-Sci	HumanEval	MBPP	AGIEval	HellaSwag
Qwen-2.5-7B	Knowledge Distribution ±10%	42.51	34.73	28.88	37.64	62.55	42.26	53.77	70.84	59.55	50.07	48.14	73.80
	Knowledge Distribution ±15%	43.16	34.27	27.45	32.40	**62.75**	40.59	54.21	65.42	55.48	50.16	47.89	73.95
	Knowledge Distribution ±20%	39.97	31.98	24.97	28.23	61.80	35.13	48.82	66.06	56.05	48.25	44.63	72.67
	Random Mixing Ratio	38.17	27.86	25.88	28.14	58.46	36.67	28.53	72.85	58.43	45.62	49.56	72.39
	DoReMi	46.64	34.56	25.13	30.05	59.02	40.81	40.84	64.77	53.90	47.47	48.93	73.31
	DoGE	43.82	32.71	26.80	34.41	57.73	38.75	45.97	68.39	52.88	48.25	47.62	**74.15**
	VersaTune	**51.65**	**36.75**	**34.28**	**52.09**	62.48	**69.09**	**68.14**	**74.16**	**60.68**	**61.25**	**49.73**	73.90
Qwen-2.5-14B	Knowledge Distribution ±10%	55.05	41.32	45.61	41.00	65.10	69.85	46.41	79.83	76.24	77.39	55.30	**76.37**
	Knowledge Distribution ±15%	54.96	40.75	47.98	39.15	64.55	65.57	46.04	75.59	72.08	75.60	55.45	75.60
	Knowledge Distribution ±20%	51.18	36.38	47.26	37.67	64.90	62.31	41.09	73.64	71.42	74.87	54.98	76.01
	Random Mixing Ratio	52.79	38.45	46.93	42.58	64.62	62.98	42.87	70.05	74.96	75.65	54.63	75.86
	DoReMi	58.66	44.36	45.24	36.72	64.17	65.46	48.78	71.44	73.74	72.27	54.80	76.25
	DoGE	56.47	45.85	47.06	40.09	63.75	64.20	43.76	70.28	70.15	74.23	**55.74**	75.54
	VersaTune	**60.59**	**46.58**	**50.24**	**45.15**	**65.84**	**78.68**	**62.89**	**82.86**	**82.64**	**81.48**	55.52	75.98

Table 7: Experimental results of VersaTune with additional baselines. For each domain, we evaluate the models using two relevant benchmarks. The best and second best results are in **bold** and underlined.

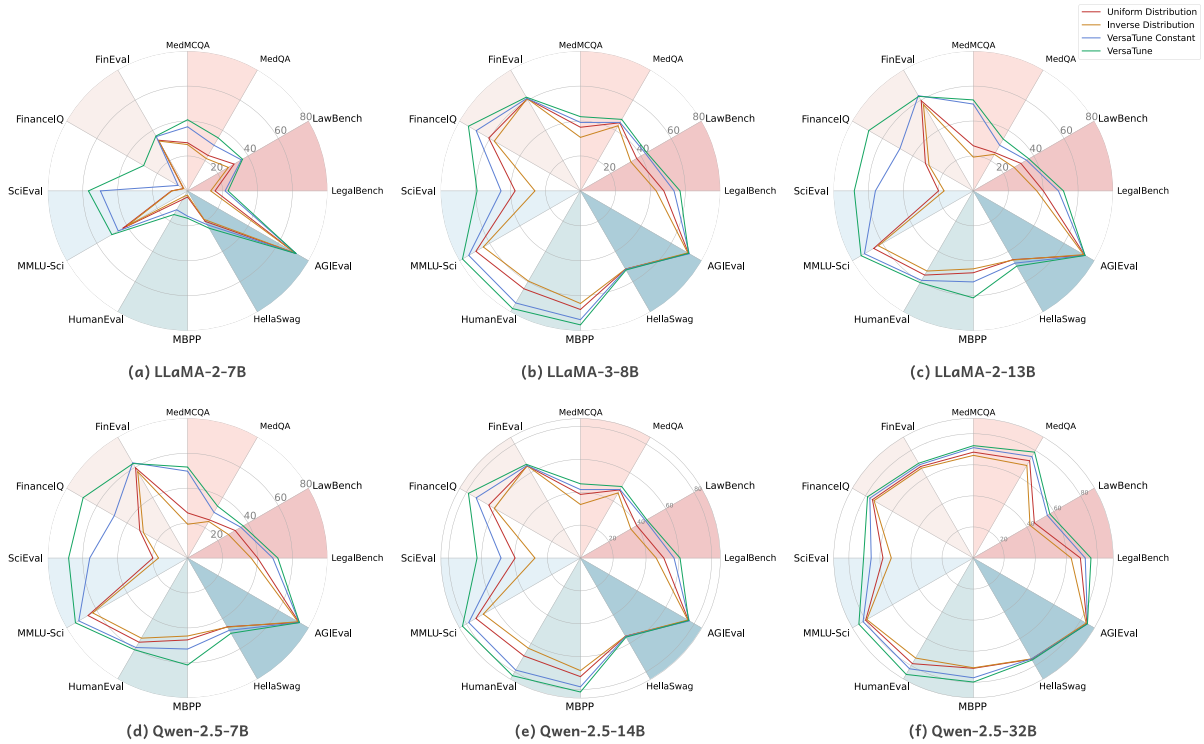

Figure 8: Performances of different models on versatile benchmarks related to various domains under the scenario of effective multi-ability fostering. The background color of the radar chart signifies the domain to which the current benchmark belongs, with reference to the color key provided in Figure 1, which includes law, medicine, finance, science, code, and general fields.

Comparison with Frontier Models. Furthermore, to demonstrate the efficacy of VersaTune across diverse domain tasks, we conducted a comparative analysis between *Qwen-2.5-32B + VersaTune* and *frontier models* across various domain-specific tasks, with results visualized in Figure 9. Such experimental results indicate that Qwen-2.5-32B equipped with VersaTune enhances multi-domain performance to the state-of-the-art levels, which even outperforms frontier models like GPT-4o, Claude3.5-Sonnet and DeepSeek-V3 by 0.86%, 4.76% and 4.60% on the overall domain capabilities, respectively. This comparison underscores the superior performance of our VersaTune in advancing multi-domain capabilities.

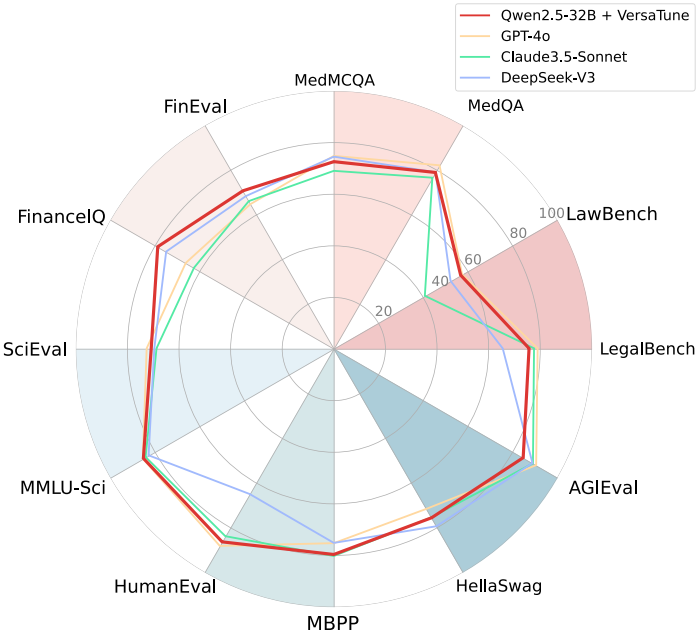

Figure 9: Performance comparison between *Qwen-2.5-32B + VersaTune* and *frontier models* acorss multiple domains.

D.2 Flexible Domain Expansion

D.2.1 Performance Variations in Target and Non-Target Domains

Here we exhibit the performance of the target domain and other non-target domains under the domain expansion scenario, as realized by Algorithm 3. Figure 3, Figure 10-13 illustrate the changes in target domain capabilities and non-target domain capabilities during the fine-tuning process when focused on a specific domain, providing experimental results for flexible domain expansion. In each figure, the stacked group bar chart (left) depicts the percentage change in performance for non-target domains relative to their pre-fine-tuning states, with the positive direction on the y-axis indicating performance improvement and the negative direction signifying a decline. The line chart (right) represents the overall change across all non-target domains for three distinct strategies, with color legends corresponding to those of the line chart on the right. The right-side chart depicts the percentage increase in performance for the current target domain. We employed the Qwen-2.5-7B and Qwen-2.5-14B models, and the mean percentage change in model performance when focusing on domain enhancement is presented in both the stacked group bar chart and the line chart. Three interesting phenomena are observed from the outcomes:

- **Absolute Count vs. Proportion.** Notably, by the *second epoch* of training, the performance degradation across non-target domains tends to be mitigated to some extent, and there is even a positive trend in capability enhancement in some cases. We attribute this phenomenon to the fact that the absolute quantity of instances for each domain, relative to the domain distribution, has a predominant influence on model performance at this stage.

- **Domain Interactions.** Domains are not entirely orthogonal to each other, and there is a degree of mutual reinforcement among them: ***(1)*** Enhancing capabilities in the medicine domain can boost performance in the science domain to a certain degree (Figure 3). ***(2)*** Models' capabilities in law and finance are mutually reinforcing, promoting each other's development (Figure 10 and Figure 11). ***(3)*** Augmenting the model's code-related capabilities can also, to some extent, improve its ability to solve scientific problems, which is likely due to the shared reasoning and logical structuring required across these domains (Figure 13).

- **Domain Mastery Efficiency.** From the slope of the target domain performance increase in Figure 3, Figure 10-13 (b), it is evident that the model's efficiency in mastering knowledge of a specific domain diminishes over training. In other words, as training progresses, the model's grasp of the target domain approaches saturation, while its performance on non-target domains declines sharply. Consequently, greater emphasis should be placed on mitigating losses in non-target domains during this phase, aiming to strike a balance between domain expansion and the salvage of capabilities in non-target domains, which is also shown in Figure 14.

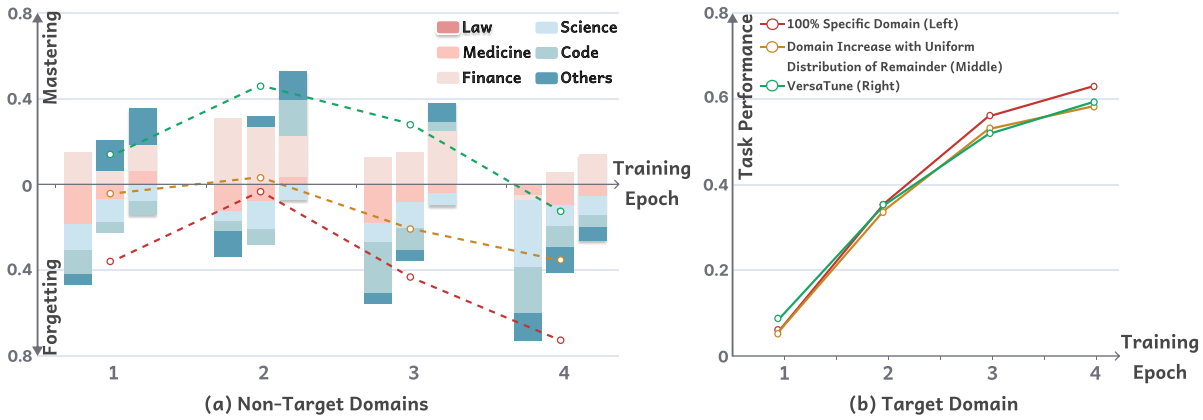

Figure 10: Domain expansion results for the ***law*** domain, including non-target domains (a) and target domain (b).

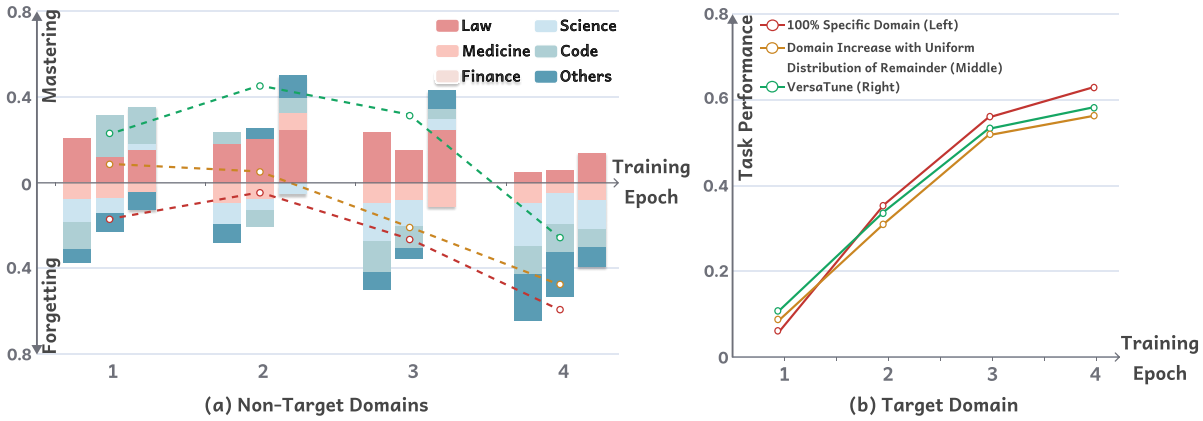

Figure 11: Domain expansion results for ***finance*** domain, including non-target domains (a) and target domain (b).

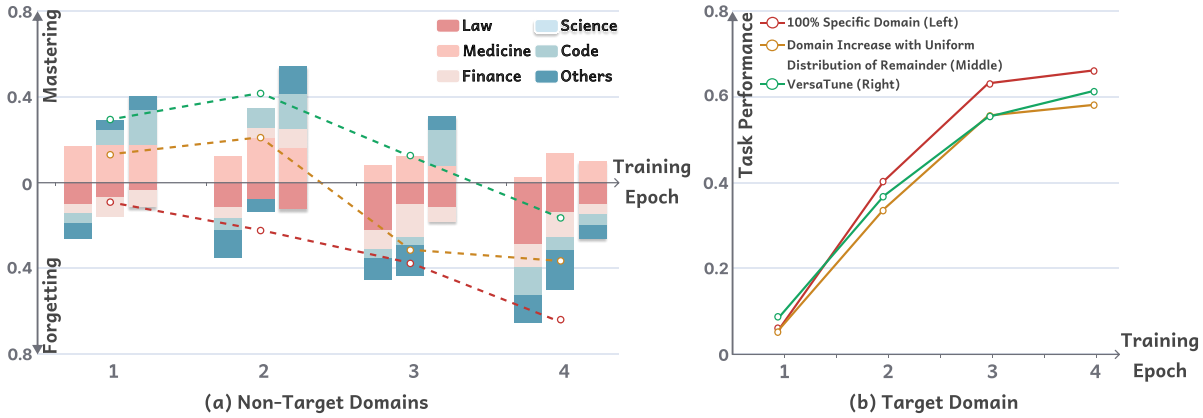

Figure 12: Domain expansion results for ***science*** domain, including non-target domains (a) and target domain (b).

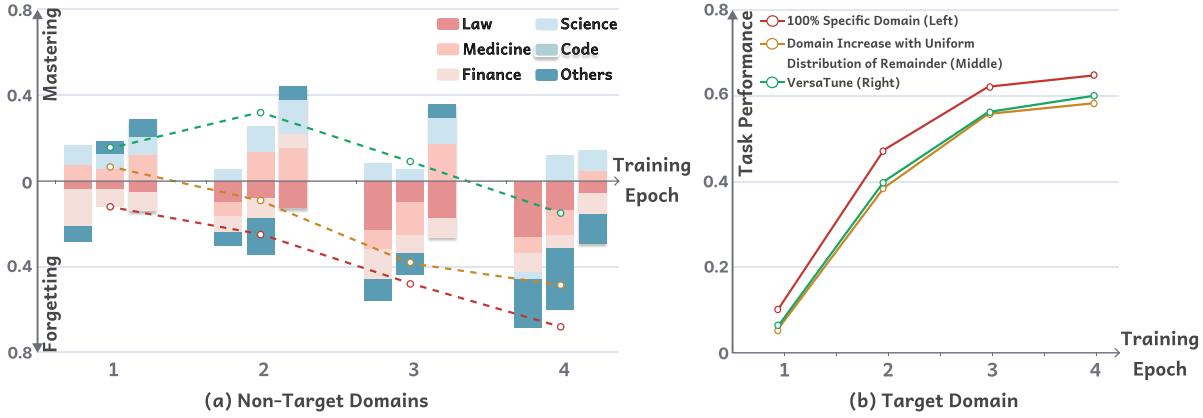

Figure 13: Domain expansion results for the ***code*** domain, including non-target domains (a) and target domain (b).

Target Domain	Training Step (Epoch)	Method	Variations in Comprehensive Domains (%)						Sum. (%)	
			Law	Medicine	Finance	Science	Code	Other	Target	Non-Target
Law	1	100% Specific Domain	↑5.89	↓18.82	↑14.71	↓11.76	↓11.18	↓5.00	↑5.89	↓32.05
		Uniform Distribution of Non-Target Domains	↑5.38	↓7.35	↑6.47	↓6.58	↓0.05	↑13.99	↑5.38	↓6.48
		VersaTune	↑8.25	↑6.18	↑12.06	↓7.65	↓6.22	↑17.59	↑8.25	↑21.96
	2	100% Specific Domain	↑35.51	↓12.65	↑30.59	↓4.41	↓5.29	↓11.76	↑35.51	↓3.52
		Uniform Distribution of Non-Target Domains	↑33.89	↓7.94	↑26.47	↓12.65	↓7.65	↑5.00	↑33.89	↑3.23
		VersaTune	↑35.14	↑3.53	↑18.82	↓6.89	↑17.06	↑13.24	↑35.14	↑45.76
	3	100% Specific Domain	↑55.84	↓17.94	↑12.35	↓8.82	↓23.53	↓5.00	↑55.84	↓42.94
		Uniform Distribution of Non-Target Domains	↑52.89	↓8.24	↑15.00	↓12.06	↓10.17	↓5.12	↑52.89	↓20.59
		VersaTune	↑51.71	↓4.41	↑24.71	↓5.29	↑4.87	↑8.42	↑51.71	↑29.30
	4	100% Specific Domain	↑62.76	↓5.00	↓2.06	↓31.18	↓21.76	↓12.65	↑62.76	↓72.65
		Uniform Distribution of Non-Target Domains	↑58.12	↓9.71	↑5.59	↓9.41	↓10.14	↓12.05	↑58.12	↓35.72
		VersaTune	↑59.08	↓5.59	↑13.82	↓8.82	↓5.61	↓6.17	↑59.08	↓12.37
Medicine	1	100% Specific Domain	↓10.29	↑5.87	↓3.82	↑24.12	↓19.41	↓7.35	↑5.87	↓16.75
		Uniform Distribution of Non-Target Domains	↓3.82	↑5.36	↓5.29	↑5.59	↓2.65	↑6.53	↑5.36	↑0.36
		VersaTune	↑3.53	↑8.17	↓7.65	↑8.82	↑16.47	↓6.17	↑8.17	↑15.00
	2	100% Specific Domain	↓18.82	↑40.44	↓6.47	↑40.00	↓7.94	↓17.65	↑40.44	↓10.88
		Uniform Distribution of Non-Target Domains	↓7.94	↑33.68	↓9.12	↑20.59	↑4.98	↓5.64	↑33.68	↑2.87
		VersaTune	↑12.35	↑35.21	↑7.65	↑8.84	↑16.52	↑12.65	↑35.21	↑58.01
	3	100% Specific Domain	↓22.35	↑58.78	↓8.82	↑7.94	↓19.12	↓10.00	↑58.78	↓52.35
		Uniform Distribution of Non-Target Domains	↓10.46	↑55.69	↓14.98	↑12.35	↓4.41	↓14.18	↑55.69	↓31.68
		VersaTune	↓3.53	↑53.85	↓4.52	↑17.06	↑7.64	↑6.03	↑53.85	↑22.68
	4	100% Specific Domain	↓27.94	↑64.61	↓11.76	↓2.35	↓21.76	↓12.65	↑64.61	↓76.46
		Uniform Distribution of Non-Target Domains	↓13.82	↑58.07	↓11.18	↑2.35	↓6.47	↓18.53	↑58.07	↓47.65
		VersaTune	↓5.59	↑59.81	↓4.27	↑10.68	↓5.94	↓10.26	↑59.81	↓15.38
Finance	1	100% Specific Domain	↑20.59	↓7.94	↑5.45	↓10.29	↓12.65	↓6.47	↑5.45	↓16.76
		Uniform Distribution of Non-Target Domains	↑12.05	↓7.36	↑8.21	↓6.74	↑19.41	↓8.53	↑8.21	↑8.83
		VersaTune	↑15.07	↓4.12	↑10.46	↑3.24	↑17.06	↓8.55	↑10.46	↑22.70
	2	100% Specific Domain	↑18.24	↓9.71	↑34.97	↓9.41	↑5.29	↓8.82	↑34.97	↓4.41
		Uniform Distribution of Non-Target Domains	↑20.59	↓7.94	↑31.08	↓4.71	↓7.69	↑4.98	↑31.08	↑5.23
		VersaTune	↑24.70	↑7.35	↑33.92	↓5.02	↑7.08	↑10.58	↑33.92	↑44.69
	3	100% Specific Domain	↑23.53	↓9.41	↑55.87	↓17.35	↓14.71	↓7.94	↑55.87	↓25.88
		Uniform Distribution of Non-Target Domains	↑15.02	↓8.24	↑52.41	↓12.06	↓10.30	↓4.98	↑52.41	↓20.56
		VersaTune	↑24.71	↓11.18	↑53.04	↑5.29	↑4.44	↑8.83	↑53.04	↑32.09
	4	100% Specific Domain	↑5.00	↓9.12	↑62.89	↓20.29	↓12.94	↓21.76	↑62.89	↓59.11
		Uniform Distribution of Non-Target Domains	↑5.88	↓5.29	↑56.13	↓14.09	↓13.23	↓20.87	↑56.13	↓47.60
		VersaTune	↑14.19	↓8.24	↑58.47	↓13.24	↓8.52	↓9.41	↑58.47	↓25.22
Science	1	100% Specific Domain	↓10.29	↑17.06	↓3.82	↑6.78	↓4.71	↓7.35	↑6.78	↓9.11
		Uniform Distribution of Non-Target Domains	↓6.76	↑17.64	↓9.18	↑5.37	↑7.05	↑4.73	↑5.37	↑13.48
		VersaTune	↓3.53	↑17.35	↓7.64	↑8.35	↑16.67	↑6.98	↑8.35	↑29.83
	2	100% Specific Domain	↓11.47	↑12.35	↓4.71	↑40.84	↓5.88	↓12.94	↑40.84	↓22.65
		Uniform Distribution of Non-Target Domains	↓8.24	↑20.59	↑5.68	↑32.78	↑9.12	↓5.85	↑32.78	↑21.30
		VersaTune	↓12.36	↑16.17	↑9.43	↑36.97	↑16.89	↑12.65	↑36.97	↑42.78
	3	100% Specific Domain	↓21.76	↑7.94	↓8.82	↑63.20	↓4.41	↓10.00	↑63.20	↓37.05
		Uniform Distribution of Non-Target Domains	↓9.98	↑12.06	↓15.01	↑55.78	↓4.11	↓14.13	↑55.78	↓31.17
		VersaTune	↓11.47	↑11.18	↓6.76	↑55.40	↑16.49	↑6.81	↑55.40	↑16.25
	4	100% Specific Domain	↓27.94	↑2.35	↓11.47	↑66.15	↓12.65	↓12.59	↑66.15	↓62.30
		Uniform Distribution of Non-Target Domains	↓13.82	↑13.53	↓11.18	↑58.46	↓6.57	↓6.47	↑58.46	↓24.51
		VersaTune	↓10.12	↑10.00	↓4.21	↑61.30	↓5.30	↓6.74	↑61.30	↓16.37
Code	1	100% Specific Domain	↓3.82	↑7.35	↓17.35	↑9.12	↑10.46	↓7.29	↑10.46	↓11.99
		Uniform Distribution of Non-Target Domains	↓3.76	↑5.09	↓7.80	↑7.68	↑5.23	↑5.65	↑5.23	↑6.86
		VersaTune	↓5.06	↑11.82	↓8.76	↑8.96	↑5.98	↑8.19	↑5.98	↑15.15
	2	100% Specific Domain	↓9.71	↓6.47	↓7.94	↑5.29	↑47.28	↓6.18	↑47.28	↓25.01
		Uniform Distribution of Non-Target Domains	↓7.90	↑13.49	↓9.05	↑12.03	↑38.31	↓17.76	↑38.31	↓9.19
		VersaTune	↓12.22	↑15.04	↑6.78	↑16.28	↑39.77	↑6.14	↑39.77	↑32.02
	3	100% Specific Domain	↓22.35	↓8.82	↓14.12	↑7.94	↑61.95	↓10.02	↑61.95	↓47.37
		Uniform Distribution of Non-Target Domains	↓9.96	↓15.07	↓8.46	↑5.33	↑55.62	↓10.04	↑55.62	↓38.20
		VersaTune	↓17.39	↑17.25	↓9.09	↑12.31	↑56.12	↑6.19	↑56.12	↑9.27
	4	100% Specific Domain	↓26.18	↓7.06	↓8.82	↓3.24	↑64.76	↓22.65	↑64.76	↓67.95
		Uniform Distribution of Non-Target Domains	↓14.01	↓13.86	↓6.57	↑6.66	↑58.06	↓13.93	↑58.06	↓41.71
		VersaTune	↓5.66	↑4.42	↓10.10	↑9.95	↑59.71	↓13.57	↑59.71	↓14.96

Table 8: Results of VersaTune on flexible domain expansion, we computed the average percentage change across various models for each method. "Sum. (%)" denotes the total percentage of performance variations across all target and non-target domain tasks. Symbols ↑ and ↓ indicate an increase or decrease in the percentage of scores (%) compared to the *initial state* before supervised fine-tuning. The current target domain is highlighted using the corresponding domain color in Figure 1, which includes law, medicine, finance, science, code, and general fields.

D.2.2 Importance of Proportion Thresholds

Here we describe the significance of establishing proportion thresholds for specific domains during domain expansion in detail. We compare VersaTune with those implementing an *unconditional dynamic increase of the specific domain*, where we remove the implementation of Line 8 in Algorithm 3, to ablate the component of criteria for determining the upper limit of domain expansion. In Figure 14, we present the trends in the overall multi-domain performance of the models under specific domain expansion for each domain. It can be observed that, for the majority of domains, the gap in average multi-task performance between models trained with VersaTune and those without an upper limit on domain proportion becomes increasingly pronounced after the second or third epoch. We deduce that this occurs due to the fact that as training progresses, the models' ability to learn within the target domain becomes nearly maximized. Enhancing the emphasis on the current domain beyond this point yields marginal benefits and may even result in a substantial degradation of performance in other domains. Notably, between the *second* and *third* epochs of supervised fine-tuning, the model reaches a balance where the efficiency of improvement in the target domain is matched by the rate of performance degradation in non-target domains. The finding shows that the criteria for determining the upper limit on the proportion of a specific domain during domain expansion, has mitigated the loss of capabilities in other domains experienced by the target model M_θ during the fine-tuning process. Moreover, it ensures gains in the capacity for the current domain of interest.

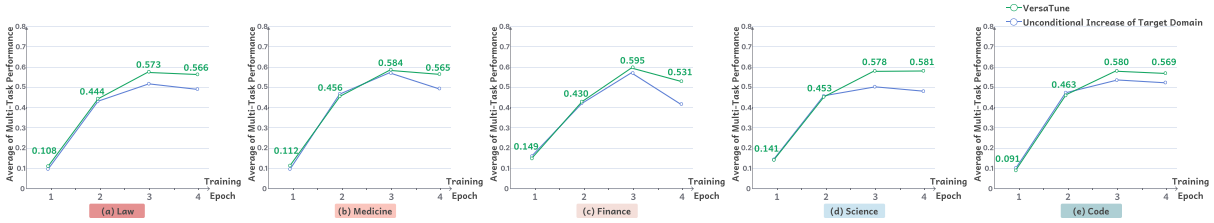

Figure 14: Line chart of the multi-task performances of models across different domains during the domain expansion process. We calculated the average percentage change for both target and non-target domains comparing to the *initial state*. Additionally, we highlighted the performance changes of the VersaTune at various checkpoints using green numerical annotations.

In summary, VersaTune exhibits the following properties:

- **Efficient.** VersaTune employs distribution consistency training of the domain knowledge proportion during models' SFT stage, providing an efficient data composition strategy for enhancing versatile capabilities (for ***C2***).
- **Flexible.** VersaTune can be flexibly adapted to scenarios that expand performance on specific domain tasks while minimizing the degradation of the model's capabilities in other non-target domains (for ***C1***, ***C3***).
- **Robust.** Our strategy achieves significant performance improvements in open-sourced models with parameter sizes ranging from 7B-32B, adding to the effectiveness of VersaTune (for ***C1***, ***C2*** and ***C3***).

E Prompts

We present the prompts that are employed throughout our pipeline in VersaTune . Only the English version is presented due to LaTeX compilation issues with non-English languages.

Prompt: Domain Probability Inference

You are a data domain annotation expert, and you currently have the following six data domains: law, medical && health care, finance, science, code, and other. Please classify the following text fragment based on their topic and structure by providing the probability distribution of its belonging to each category, where the sum of probabilities across all domain categories equals 1, without additional commentary:

 # Text
{text_content}

Output Format:

```
```json
{
    "Law": "",
    "Medicine": "",
    "Finance": "",
    "Sciencee": "",
    "Code": "",
    "Other": ""
}
'''
```
```