Investigating the Pre-Training Dynamics of In-Context Learning: Task Recognition vs. Task Learning

Xiaolei Wang[1,3*], **Xinyu Tang**[1,3*], **Junyi Li**[4], **Wayne Xin Zhao**[1,3†] and **Ji-Rong Wen**[1,2,3]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
[4]Department of Computer Science, National University of Singapore
wxl1999@foxmail.com, txy20010310@163.com, batmanfly@gmail.com

Abstract

The emergence of in-context learning (ICL) is potentially attributed to two major abilities: *task recognition* (TR) for recognizing the task from demonstrations and utilizing pre-trained priors, and *task learning* (TL) for learning from demonstrations. However, relationships between the two abilities and how such relationships affect the emergence of ICL is unclear. In this paper, we take the first step by examining the pre-training dynamics of the emergence of ICL. With carefully designed metrics, we find that these two abilities are, in fact, *competitive* during pre-training. Moreover, we observe a negative correlation between the competition and the performance of ICL. Further analysis of common pre-training factors (*i.e.,* model size, dataset size, and data curriculum) demonstrates possible ways to regulate the competition. Based on these insights, we propose a simple yet effective method to better integrate these two abilities for ICL at inference time. Through adaptive ensemble learning, the performance of ICL can be significantly boosted, enabling two small models to outperform a larger one with more than twice the parameters. The code is available at https://github.com/RUCAIBox/Competitive-ICL.

1 Introduction

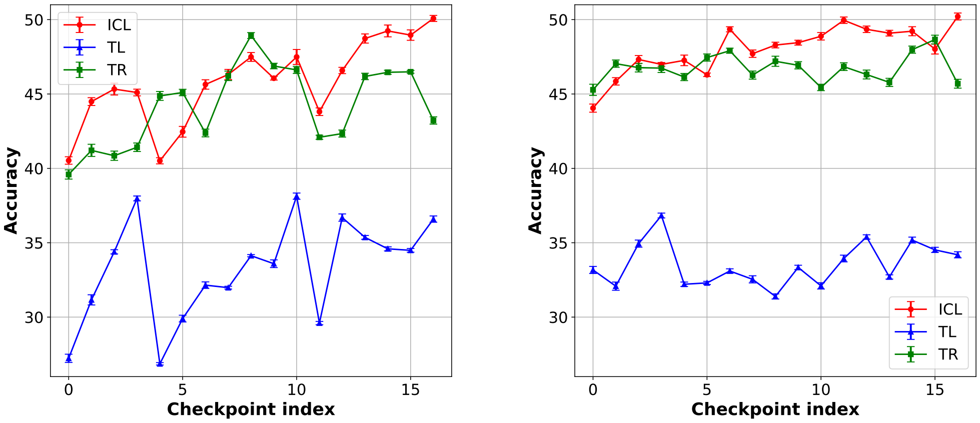
In-context learning (ICL) (Brown et al., 2020) represents a significant advancement in the capabilities of large language models (LLMs) (Zhao et al., 2023; Hu et al., 2024b). It allows models to rapidly adapt to new tasks without updating the parameters by adding only a few examples as demonstrations to the input. This capability has profound applications on a wide range of tasks (Dong et al., 2022; Lin et al., 2023).

To explore the underlying mechanism, existing work (Pan et al., 2023; Wei et al., 2023) mainly focuses on how LLMs perform ICL during inference. For the underlying mechanisms of ICL, two major abilities are considered to play important roles: *task recognition (TR)*, which recognizes the target task from demonstrations and utilizes the prior knowledge learned from pre-training to solve, and *task learning (TL)*, which directly learns from demonstrations to deal with the task. Furthermore, recent research (Pan et al., 2023) has found that TR is relatively easier to obtain and can be observed in smaller models with only 350M parameters, while TL would often emerge in larger models with billions of parameters. Based on this observation, Wei et al. (2023) further explore the relationships between these two abilities and reach the same conclusion that TR takes the dominant in smaller LLMs while TL is more emphasized in larger LLMs. However, how these two abilities *quantitatively* affect the emergence of ICL is under-explored.

In this work, we take the first step towards unraveling the mystery, *i.e.,* **relationships between TR and TL and how such relationships affect the emergence of ICL**, by examining the performance

[*] Equal contribution.

[†] Corresponding author.



(a) MiniCPM-2B

(b) Amber-7B

Figure 1: The performance of ICL and its two abilities (*i.e.,* task recognition (TR) and task learning (TL)). The emergence of ICL encounters many fluctuations, where the performance of task recognition and task learning changes in the opposite direction.

changes of ICL during pre-training. To achieve this goal, we first disentangle the two abilities by manipulating the input-label pairs following previous settings (Pan et al., 2023), upon which we can easily measure the performance of TR and TL individually. As illustrated in Figure 1, we can observe that the emergence of ICL encounters many fluctuations, along with *competition* between its two abilities (*i.e.,* their performance actually changes in the opposite direction). To quantify such a competitive relationship between TR and TL, we propose *competition intensity*, a new measurement based on the performance change of TR and TL between two adjacent checkpoints to reflect the extent to which one ability surpasses another.

With the proposed metric, we find that the competitive relationship is prevalent across many LLMs with various training settings. More importantly, it demonstrates a negative correlation with ICL. First, during pre-training, the competition exhibits a "stable–rise" pattern, and simultaneously, the performance of ICL improves or fluctuates in correspondence. Second, with respect to the entire pre-training process, the average competition intensity is negatively correlated with the final ICL performance: the less the competition, the better the ICL performance. These findings suggest that regulating the competition between TR and TL would be crucial for the emergence of ICL. We further investigate the influence of common pre-training factors (*i.e.,* model size, dataset size, and data curriculum) on the competition. We conclude that: (1) scaling model size can lead to the early appearance of competition but effectively reduce the average intensity of competition; (2) scaling dataset size can postpone the competition; and (3) specific data curricula can adjust the intensity of competition for the enhancement of LLMs.

Our analysis reveals that with effective regulation of the competition between TR and TL, LLMs could achieve better ICL performance. To this end, we propose a simple yet effective method to fuse the two abilities for better ICL performance at inference time. Specifically, we first select two checkpoints from the pre-training process with the best abilities of TR and TL, respectively. Then, they are fused with adaptive ensemble learning, where the contribution of each one is adaptively determined by its performance. To validate the effectiveness of our approach, we conduct experiments on extensive datasets and LLMs with various training settings. Experimental results demonstrate that this simple method can effectively enhance ICL performance, outperforming several competitive baselines, even with less than half the parameters of a larger LLM.

Our contributions can be summarized as follows:

• To the best of our knowledge, we are the first to study the relationship between the two abilities of ICL (*i.e.,* TR and TL) and its effect on the emergence of ICL. With newly proposed measurements, we conduct an empirical analysis of the pre-training dynamics of ICL, and discover a competitive relationship between TR and TL and its negative correlation with the emergence of ICL.

• We conduct a fine-grained analysis of common pre-training factors (*i.e.,* model size, dataset size, and data curriculum) to understand their influence on the competition between TR and TL, *e.g.,* scaling model size can effectively decrease the average competition intensity.

• We propose a simple but effective approach to better integrate TR and TL at inference time. By fusing two smaller models through adaptive ensemble learning, the performance of ICL can outperform a larger model with much more parameters than the smaller models.

2 Background and Measurement

In this section, we introduce the background of task recognition (TR) and task learning (TL) and further propose our measurements to quantify the competitive relationship between them.

2.1 Task Recognition and Task Learning

Typically, an LLM performs ICL by using input-label pairs from the target task as demonstrations, *i.e.,* $D_k = \{(x_1, y_1), \ldots, (x_k, y_k)\}$, to predict the label for the test input. In existing literature (Pan et al., 2023; Lin & Lee, 2024; Jang et al., 2024), it has been widely recognized that ICL can be attributed to two major underlying abilities, namely *task recognition (TR)* and *task learning (TL)*. Specifically, TR refers to the ability of an LLM to recognize the target task from demonstrations and only utilize its own knowledge obtained from pre-training to solve the task, while TL refers to the ability of an LLM to solve the target task solely based on demonstrations.

To disentangle the two main abilities from ICL, existing studies (Pan et al., 2023; Lin & Lee, 2024) are mainly developed based on an important assumption: *the mapping information between the input and the label in demonstrations is more important for TL than TR*. Under this assumption, three settings can be used to evaluate these two abilities and ICL:

• *Gold*: It refers to the standard ICL setting, where we use the correct input-label pairs. This setting is used to evaluate the ICL performance of LLMs.

• *Random*: To evaluate TR ability, we randomly sample labels from the label space of the target task for each input in demonstrations.

• *Abstract*: To evaluate TL ability, we map the original correct labels in demonstrations to semantically unrelated tokens (*e.g.,* numbers, letters, or symbols).

The above settings are widely used in existing work (Pan et al., 2023; Lin & Lee, 2024; Jang et al., 2024). To align with them, we follow the same settings to quantify the performance of TL and TR.

2.2 Competition Measurement

In our prior analysis shown in Figure 1, we discover that there exists a certain degree of competition between TR and TL during pre-training. To quantify this, we propose *competition intensity*, a new measurement based on the performance changes of TR and TL. As a prerequisite, we assume that the intermediate checkpoints of an LLM are available, denoted as $\mathcal{M}_\theta = \{M_{\theta_1}, M_{\theta_2}, \ldots, M_{\theta_N}\}$, where the index increases with the number of training steps and N is the total number of checkpoints. Based on this, to track the pre-training dynamics of TR and TL, we can calculate their performance changes as follows:

$$\Delta \mathrm{TR}_i = \mathrm{Acc}_{i+1}^{\mathrm{rand}} - \mathrm{Acc}_{i}^{\mathrm{rand}} \qquad \Delta \mathrm{TL}_i = \mathrm{Acc}_{i+1}^{\mathrm{abs}} - \mathrm{Acc}_{i}^{\mathrm{abs}}, \tag{1}$$

where $\mathrm{Acc}_{i}^{\mathrm{rand}}$ and $\mathrm{Acc}_{i}^{\mathrm{abs}}$ denote the accuracy of the intermediate checkpoint M_{θ_i} under the *random* and *abstract* settings introduced in Section 2.1. To determine whether there exists competition between TR and TL, one feasible way is to check whether their performance changes in the *opposite* direction since competition typically results in a trade-off between these two abilities. To indicate such an existence of competition, we compute a competition indicator C_i^H as follows:

$$C_i^H = \mathbb{I}(\Delta \mathrm{TR}_i \cdot \Delta \mathrm{TL}_i < 0) \cdot \mathbb{I}(|\Delta \mathrm{TR}_i| > \epsilon) \cdot \mathbb{I}(|\Delta \mathrm{TL}_i| > \epsilon), \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function and we add two additional indicator functions to reduce the influence of inaccurate performance estimation. ϵ is set to 0.01 in the experiment. Furthermore, based on the competition indicator, we propose *competition intensity* to quantify the degree of competition at different pre-training stages. We denote competition intensity as C_i^S, which is represented as the ratio between the performance changes of TR and TL:

$$C_i^S = C_i^H \cdot \left[\mathbb{I}(\Delta\mathrm{TR}_i < 0) \cdot \left| \frac{\Delta\mathrm{TR}_i}{\Delta\mathrm{TL}_i} \right| + \mathbb{I}(\Delta\mathrm{TL}_i < 0) \cdot \left| \frac{\Delta\mathrm{TL}_i}{\Delta\mathrm{TR}_i} \right| \right]. \tag{3}$$

Here, if the competition exists (*i.e.,* $C_i^H = 1$), an increase in the performance of one ability will lead to a decline in the performance of another (*i.e.,* $\Delta\mathrm{TR}_i \cdot \Delta\mathrm{TL}_i < 0$ in Eq. 2). Thus, a larger value of C_i^S suggests higher competition intensity, as it implies a greater decrease in the performance of one ability (*i.e.,* numerator in Eq. 3) for a given increase in the performance of another (*i.e.,* denominator in Eq. 3). Moreover, to track the dynamics of competition intensity, we calculate the accumulative intensity R_i as follows:

$$R_i = \frac{\sum_{j=1}^{i} C_j^S}{\sum_{j=1}^{N} C_j^S}. \tag{4}$$

This measure tracks the cumulative proportion of the intensity up to the i-th intermediate checkpoint, providing insight into how competition evolves over time.

3 Empirical Analysis

In this section, we present the empirical analysis of the competition between TR and TL.

3.1 Experimental Setup

Tasks and Datasets. Following Pan et al. (2023), we select 16 datasets across four types of tasks for the experiment: sentiment analysis, topic/state classification, toxicity detection, and natural language inference/paraphrase detection. Details about the datasets are depicted in Appendix A. Due to computational constraints, we sample 1000 examples from each dataset for evaluation.

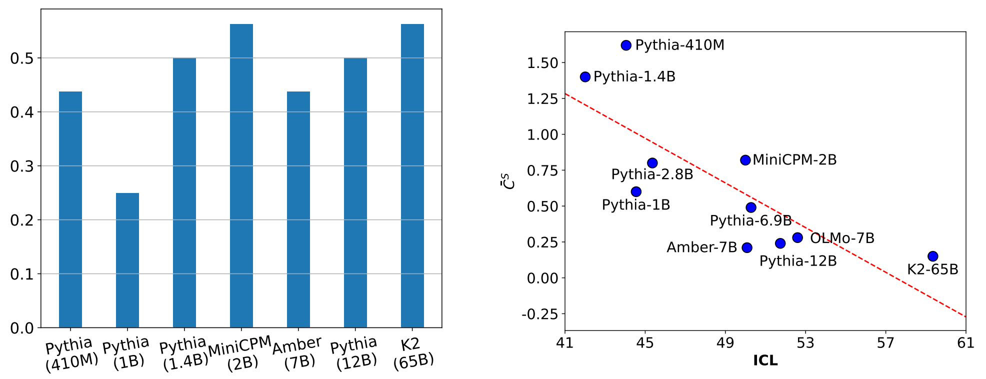
Models. Since our work focuses on the pre-training dynamics of ICL, we select LLMs with more than 350M parameters and access their intermediate checkpoints: the Pythia suite (6 model sizes ranging from 410M to 12B) (Biderman et al., 2023), MiniCPM-2B (Hu et al., 2024a), Amber-7B (Liu et al., 2023), CrystalCoder-7B (Liu et al., 2023), OLMo-7B (Groeneveld et al., 2024), Baichuan2-7B (Yang et al., 2023), and K2-65B (Liu et al., 2024). Due to computational constraints, we sample 16 checkpoints for each model, which are evenly distributed in the pre-training process. Experiments with other numbers of checkpoints yield similar results, which are shown in Appendix B.1. To make the output as deterministic as possible, we set `temperature=0` when sampling.

Other Details. We randomly sample 16 examples as demonstrations by default across the paper following Min et al. (2022). The discussion about the number of examples can be seen in Appendix B.2. We use minimal templates to construct demonstrations following Pan et al. (2023). Specifically, we use a single newline character (*i.e.,* \n) to connect each input-label pair and three ones to separate examples. We utilize symbols as labels in the abstract setting. Other kinds of abstract labels yield similar results as discussed in Appendix B.3. The results are averaged across five random seeds.

3.2 Task Recognition and Task Learning Are Competitive During Pre-Training

Figure 1 shows that the emergence of ICL is along with competition between its two abilities (*i.e.,* TR and TL). In this section, we delve into this competition and unveil its relationship with ICL.

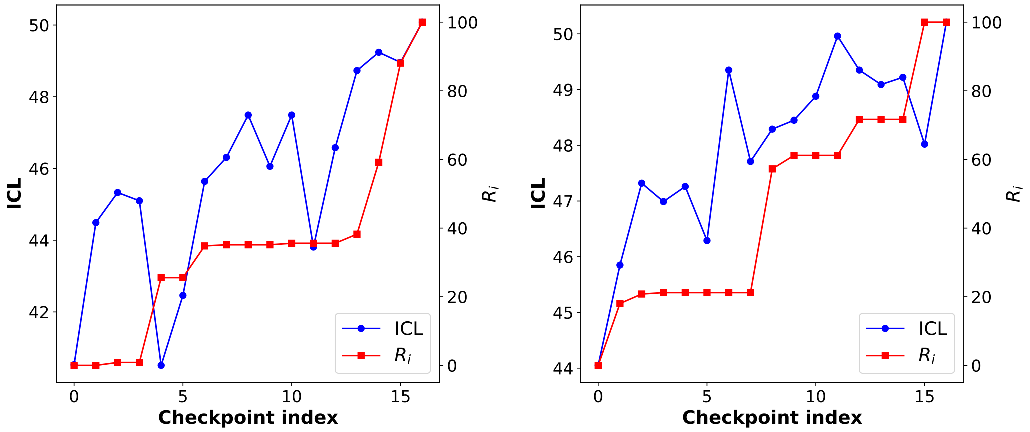
The Existence of Competition. To confirm the existence of competition between TR and TL, we investigate the pre-training process of LLMs with various training settings. Specifically, we calculate the average occurrence of competitions according to the indicator C_i^H (Eq. 2). As illustrated in



(a) Existence of competition

(b) Relationship between competition and ICL

Figure 2: (Left) Average occurrence of competition among seven LLMs. (Right) Average competition intensity $\bar{C}^S$ *w.r.t.* ICL performance of the final checkpoint.



(a) MiniCPM-2B

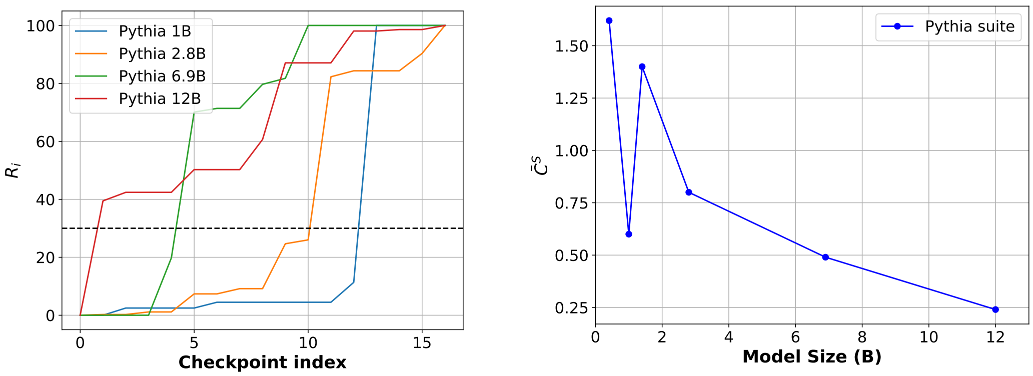
(b) Amber-7B

Figure 3: ICL performance *w.r.t.* accumulative competition intensity R_i during pre-training.

Figure 2a, all the LLMs exhibit certain levels of competition during pre-training. For some LLMs, there exists competition for more than half of the time (*i.e.,* over 0.5). It means that the competition between TR and TL is a widespread phenomenon during pre-training.

The Dynamic of Competition. We further explore the dynamics of competition during pre-training. Specifically, we choose MiniCPM-2B and Amber-7B, which are trained with over a trillion tokens with different amounts of parameters. We show the accumulative competition intensity (Eq. 4) with respect to ICL performance in Figure 3. We can observe that the accumulative competition intensity displays a "stable–rise" pattern. When the accumulative curve flattens (*i.e.,* the intensity C_i^S is low), ICL performance rises in fluctuation. Conversely, when the accumulative curve rises (*i.e.,* the intensity C_i^S is high), ICL performance shows no obvious improvement and may even decline. Such an interesting phenomenon inspires us to further examine the relationship between the competition and ICL.

The Relationship Between Competition and ICL. In this part, we delve into the relationship between the competition and ICL performance based on their pre-training dynamics. As shown in Figure 3, when there exists competition (*i.e.,* the red curve rises), the performance of ICL tends to decrease or does not improve. However, in the absence of competition (*i.e.,* the red curve flattens),



(a) Evolving process

(b) Average competition intensity

Figure 4: The evolution (R_i) and average intensity ($\bar{C}^S$) of competition with different model sizes.

the performance of ICL rises in fluctuation. To demonstrate the global impact of competition, we depict the average competition intensity $\bar{C}^S$ with respect to the ICL performance of the final checkpoint. As shown in Figure 2b, with the increase of $\bar{C}^S$, the ICL performance tends to drop. To further verify their correlation, we calculate the Pearson correlation coefficient. The result is -0.714, validating their negative correlation. This finding suggests that regulating the competition between TR and TL could be crucial for enhancing the ICL ability.

3.3 How Do Factors of Pre-Training Influence the Competition?

As discussed in Section 3.2, the competition between TR and TL during pre-training demonstrates a negative correlation with the final ICL performance. This motivates us to investigate the influence of pre-training factors on the degree of competition. Specifically, we investigate three common factors, *i.e.,* model size, dataset size, and data curriculum.

3.3.1 Model Size

We investigate the effect of model size on the level of competition between TR and TL. Specifically, we use the Pythia suite (Biderman et al., 2023) for experimentation since this family of models share the same training setting but only differ in the number of parameters.

We first pay attention to their differences in the evolution of competition. We can observe from Figure 4a that as the model size increases, the evolving curve of competition keeps moving to the left. This means that scaling up model size could make the appearance of competition earlier. The main reason is that the learning ability of larger LLMs is stronger (Kaplan et al., 2020), and they can possess TR and TL more quickly, thus causing the competition between them to occur earlier.

Then, we focus on the changes in the average competition intensity. As shown in Figure 4b, the average competitive intensity sharply decreases with the increase of model size, with the exception of Pythia-1B. This means that scaling up the model size is helpful in reducing the overall competition. This can be attributed to the fact that an LLM with more parameters has a larger capacity, where TR and TL can be allocated with more exclusive resources (*e.g.,* neurons). As a result, although the competition becomes earlier in larger LLMs, the average intensity of competition becomes lower.

3.3.2 Dataset Size

In this part, we explore the impact of dataset size on the competition between TR and TL. We conduct experiments using models with roughly the same amount of parameters but trained with different dataset sizes. Specifically, we make the comparison with two sets of LLMs: (Pythia-2.8B and MiniCPM-2B) and (Pythia-6.9B, Amber-7B, and OLMo-7B).

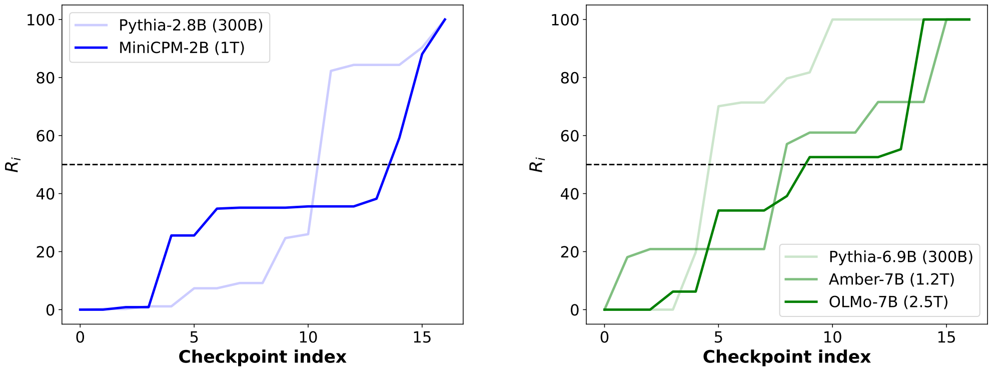
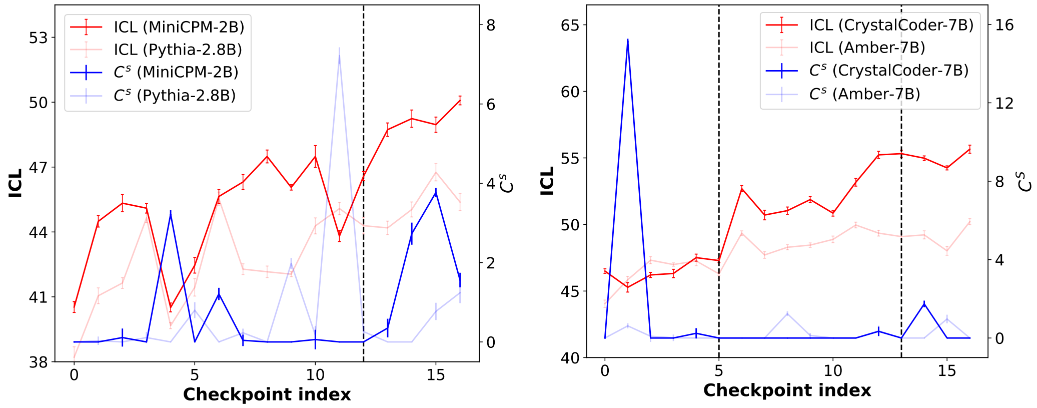


Figure 5: Competition evolving process (R_i) of LLMs trained with different dataset sizes.



(a) MiniCPM-2B vs. Pythia-2.8B

(b) CrystalCoder-7B vs. Amber-7B

Figure 6: Comparisons in ICL performance and competition intensity (C_i^S). Dashed lines are used to distinguish between different training phases in data curriculum.

Figure 5 illustrates the evolution of competition during pre-training for these two sets of LLMs. It can be observed that, for both sets, the evolving curve keeps moving to the right with the increasing dataset size. This means that scaling up the dataset size could postpone the competition. The possible reason behind this is that, when pre-trained on a small dataset, LLMs can quickly develop the TR ability since the pre-training knowledge for them to memorize is limited. Meanwhile, the TL ability is also easy to acquire, as it primarily involves direct utilization of the information in context (Singh et al., 2024). As a result, the competition between TR and TL occurs in the early stage of pre-training. With the increase in dataset size, the development of the TR ability becomes slower since there is more pre-training knowledge to memorize. Therefore, more competition happens at the later stage of pre-training, making the evolving curve shift to the right.

3.3.3 Data Curriculum

In this part, we explore the influence of data curriculum on the competition between TR and TL. Here, we consider two representative strategies for scheduling data curriculum: (1) quality curriculum, which makes arrangements for data with different levels of quality, and (2) domain curriculum, which makes arrangements for data from different domains.

We first pay attention to the influence of quality curriculum on the competition. Specifically, we use MiniCPM-2B for the experiment, which utilizes coarse-quality unlabeled data in the first phase and mixes high-quality labeled data in the second phase. We compare its pre-training dynamics with Pythia-2.8B, which has a similar model size and dataset size but lacks any data curriculum.

As illustrated in Figure 6a, the ICL performance of MiniCPM-2B boosts in the second phase with a decline of competition intensity at the end. In contrast, the ICL performance of Pythia-2.8B traps in fluctuations in the final stage of pre-training with an increase in competition intensity. From the perspective of competition, the success of quality curriculum can be attributed to the reduced intensity of competition, resulting from less noise in high-quality labeled data.

We then focus on the influence of domain curriculum. Specifically, we use CrystalCoder-7B for the experiment, which utilizes general domain data (*i.e.,* SlimPajama (Soboleva et al., 2023)) in the first phase, then mixes with code domain data (*i.e.,* StarCoder (Li et al., 2023a)) in the second phase, and mainly uses specific programming language data (*i.e.,* Python and web-related data sampled from StarCoder) in the final phase. Similar to quality curriculum, we compare its pre-training dynamics with Amber-7B, which has a similar model size and dataset size but lacks any data curriculum. As shown in Figure 6b, compared with Amber-7B, CrystalCoder-7B exhibits less competition in the second and third phases, along with a larger overall performance improvement. This is due to domain duplication since part of the training data in each phase shares the same domain with the previous phase. Domain duplication can make the training process smoother, which helps reduce competition intensity and leads to better ICL performance.

4 From Competition to Collaboration at Inference Time

As discussed in Section 3.2, the competition between TR and TL would lead to a decrease in the performance of ICL. To alleviate this, our idea is to facilitate their collaboration at inference time. In this section, we first introduce the proposed method and then demonstrate the experimental results.

4.1 Adaptive Ensemble Learning

Our analysis in Section 3.2 and 3.3 has shown that regulating the competition between TR and TL can help the model achieve better ICL performance. One potential solution to balancing the TR and TL abilities during pre-training is to design special loss functions and train models from scratch. However, this method is costly and cannot be applied to trained LLMs. To save the cost and be applicable to trained LLMs, we propose an *adaptive ensemble learning* approach to fuse TR and TL abilities at inference time.

Specifically, we first select two checkpoints with the best ability of TR and TL respectively, and then integrate their probability distributions by ensemble learning:

$$\arg\max_{y\in\mathcal{Y}} \left[w_r \mathrm{Pr}_r^{\text{rand}}(y|x) + w_l \mathrm{Pr}_l^{\text{abs}}(y|x)\right], \tag{5}$$

where $\mathrm{Pr}_r^{\text{rand}}(y|x)$ and $\mathrm{Pr}_l^{\text{abs}}(y|x)$ denote the probability for the TR and TL models to predict the label y under the random and abstract settings respectively, and w_r and w_l denote their weights respectively. Here, considering that the contribution of TR and TL abilities to ICL usually are not equal (Lin & Lee, 2024), we propose adaptive weights based on their performance as follows:

$$w_r = \frac{\mathrm{Acc}_r^{\text{rand}} - b}{(\mathrm{Acc}_r^{\text{rand}} - b) + (\mathrm{Acc}_l^{\text{abs}} - b)} \qquad w_l = \frac{\mathrm{Acc}_l^{\text{abs}} - b}{(\mathrm{Acc}_r^{\text{rand}} - b) + (\mathrm{Acc}_l^{\text{abs}} - b)}, \tag{6}$$

where $\mathrm{Acc}_r^{\text{rand}}$ is the performance of the TR model under the random setting, $\mathrm{Acc}_l^{\text{abs}}$ is the performance of the TL model under the abstract setting, and b is the performance of random guessing.

4.2 Experimental Setting

To comprehensively validate the effectiveness of our method, we consider three different combinations of TR and TL models for fusion: (1) only the backbone model (*e.g.,* Pythia-1B), (2) two models with a similar training setting (*e.g.,* Pythia-1B and Pythia-2.8B), and (3) two models with different training settings (*e.g.,* Pythia-1B and MiniCPM-2B). We select checkpoints with the best performance for the required ability. Other settings are the same as Section 3.1.

In Table 1, we compare our method with four types of baselines: (1) the backbone TR or TL model for fusion ("*Small models*"), (2) LLMs with more parameters than the sum of TR and TL models ("*Large models*"), and (3) model parameter fusion with fixed and adaptive weights calculated using the same method as ours ("*Parameter fusion*"). All the baselines are tested in the gold setting.

Table 1: Averaged accuracy and TFLOPs across 16 datasets. "Fixed" and "adaptive" denote fixed and adaptive fusion weights. We highlight the best performance among various model selections for TR and TL. Numbers marked with * indicate that the improvement is statistically significant compared with baselines (t-test with p-value < 0.05). More results are shown in Table 2.

Model	**# Parameters**	**TFLOPs**	**Accuracy**
Large models			
Amber-7B$_{\text{ICL}}$	7B	$9.98_{\pm 0.35}$	$50.08_{\pm 0.18}$
OLMo-7B$_{\text{ICL}}$	7B	$8.79_{\pm 0.32}$	$52.10_{\pm 0.19}$
Baichuan2-7B$_{\text{ICL}}$	7B	$8.18_{\pm 0.41}$	$52.77_{\pm 0.16}$
CrystalCoder-7B$_{\text{ICL}}$	7B	$10.17_{\pm 0.45}$	$55.66_{\pm 0.31}$
Pythia-12B$_{\text{ICL}}$	12B	$15.51_{\pm 0.57}$	$52.77_{\pm 0.21}$
Small models			
Pythia-1B$_{\text{ICL}}$	1B	$1.24_{\pm 0.15}$	$44.55_{\pm 0.33}$
Pythia-2.8B$_{\text{ICL}}$	2.8B	$3.60_{\pm 0.20}$	$45.38_{\pm 0.32}$
MiniCPM-2B$_{\text{ICL}}$	2.7B	$3.84_{\pm 0.21}$	$50.08_{\pm 0.16}$
Parameter fusion of small models			
Pythia-1B$_{\text{TR}}$+ Pythia-1B$_{\text{TL}}$ (fixed)	1B	$1.24_{\pm 0.15}$	$46.63_{\pm 0.26}$
Pythia-1B$_{\text{TR}}$+ Pythia-1B$_{\text{TL}}$ (adaptive)			$45.61_{\pm 0.29}$
Pythia-2.8B$_{\text{TR}}$+ Pythia-2.8B$_{\text{TL}}$ (fixed)	2.8B	$3.60_{\pm 0.20}$	$47.57_{\pm 0.23}$
Pythia-2.8B$_{\text{TR}}$+ Pythia-2.8B$_{\text{TL}}$ (adaptive)			$47.18_{\pm 0.13}$
MiniCPM-2B$_{\text{TR}}$+ MiniCPM-2B$_{\text{TL}}$ (fixed)	2.7B	$3.84_{\pm 0.21}$	$52.18_{\pm 0.26}$
MiniCPM-2B$_{\text{TR}}$+ MiniCPM-2B$_{\text{TL}}$ (adaptive)			$52.06_{\pm 0.27}$
Logit fusion of small models (Ours)			
Pythia-1B$_{\text{TR}}$+ Pythia-1B$_{\text{TL}}$ (fixed)	2B	$2.48_{\pm 0.21}$	$56.16_{\pm 0.40}$
Pythia-1B$_{\text{TR}}$+ Pythia-1B$_{\text{TL}}$ (adaptive)			$\mathbf{56.25}^{*}_{\pm 0.38}$
Pythia-1B$_{\text{TR}}$+ Pythia-2.8B$_{\text{TL}}$ (fixed)	3.8B	$4.84_{\pm 0.25}$	$56.62_{\pm 0.41}$
Pythia-1B$_{\text{TR}}$+ Pythia-2.8B$_{\text{TL}}$ (adaptive)			$\mathbf{56.83}^{*}_{\pm 0.41}$
Pythia-1B$_{\text{TL}}$+ Pythia-2.8B$_{\text{TR}}$ (fixed)			$55.23_{\pm 0.47}$
Pythia-1B$_{\text{TL}}$+ Pythia-2.8B$_{\text{TR}}$ (adaptive)			$55.39_{\pm 0.40}$
Pythia-1B$_{\text{TR}}$+ MiniCPM-2B$_{\text{TL}}$ (fixed)	3.7B	$5.08_{\pm 0.26}$	$55.21_{\pm 0.43}$
Pythia-1B$_{\text{TR}}$+ MiniCPM-2B$_{\text{TL}}$ (adaptive)			$55.31_{\pm 0.34}$
Pythia-1B$_{\text{TL}}$+ MiniCPM-2B$_{\text{TR}}$ (fixed)			$54.31_{\pm 0.63}$
Pythia-1B$_{\text{TL}}$+ MiniCPM-2B$_{\text{TR}}$ (adaptive)			$\mathbf{55.85}^{*}_{\pm 0.75}$

4.3 Results

As shown in Table 1, our proposed method can significantly boost performance compared to the single model. In addition, such an improvement is consistent across various model combinations, demonstrating that our method is widely applicable. To our surprise, by using our method, two small models together can even outperform larger models with lower inference overhead, despite their total parameters being less than half of the larger ones. It suggests that our method can effectively and efficiently fuse the abilities of TR and TL to achieve better ICL performance.

Furthermore, to verify the effectiveness of each component in our method, we conduct the ablation study. We substitute the best checkpoints with random/empty ones (Table 3) and set the weights of TR and TL to the same ("fixed" in Table 1), respectively. We can observe that removing any design would lead to a decrease in performance. It demonstrates the effectiveness of all the components of our approach. In addition, the selection of checkpoints with the best TR/TL ability seems to be more important, which yields a larger performance drop after being removed. Checkpoints with the best TR/TL ability are more diverse in their predictions, which is important for successful fusion.

5 Discussion

Why does the competition between TL and TR occur? According to our observations and existing literature, we think that the competition between TL and TR may originate from the limited resources in models (*e.g.,* neurons or layers). As shown in existing work (Kaplan et al., 2020; Tirumala et al., 2022), larger language models can memorize more training data and obtain smaller test loss with the same training steps. This indicates that the amount of resources (*e.g.,* neurons or layers) is important for the capacity of LLMs. We also investigate the influence of model size on the competition in Section 3.3.1, the results demonstrate that the average competitive intensity sharply decreases with the increase of model size. Thus, we infer that the competition could result from the limited resources in models, such as neurons or layers, being dynamically allocated between TL and TR during pre-training.

Limitations. When investigating the influence of a specific pre-training factor in Section 3, we do our best to keep other factors as consistent as possible. However, it is very challenging to run carefully controlled experiments due to the lack of pre-trained models in many settings. For example, when investigating the impact of dataset size, we can only find that models that have roughly the same amount of parameters, but cannot control other factors like data mixture and optimization. As future work, we will try to collect enough computational resources and conduct fully controlled pre-training from scratch to ensure that only one factor differs to investigate its influence.

6 Related Work

Our work is closely related to the studies on the mechanisms of ICL and model fusion.

The Mechanism of ICL. Existing work primarily explores the mechanisms of ICL from the pre-training and inference stages of LLMs. Some work discusses how ICL emerges from pre-training by conducting analysis on pre-training factors like data (Chan et al., 2022; Reddy, 2023) and optimization (Singh et al., 2024; Anand et al., 2024). Other work (Pan et al., 2023; Min et al., 2022; Dai et al., 2023) studies the operating mechanism of ICL at inference time. Researchers empirically find two main abilities in ICL: task recognition (TR) for recognizing the task and utilizing pre-trained priors of LLMs (Min et al., 2022) and task learning (TL) for learning from demonstrations (Dai et al., 2023). In this paper, we explore how TR and TL affect the emergence of ICL by examining the pre-training dynamics of LLMs.

Model Fusion. Model fusion aims to enhance performance by combining the strengths of multiple models (Li et al., 2023b). One line of work aims to reduce the difference among different models from perspectives like mode connectivity (Nagarajan & Kolter, 2019) and alignment (Tatro et al., 2020). Another line of work studies how to leverage the diversity among models through techniques like weight average (Wang et al., 2019) and ensemble learning (Sagi & Rokach, 2018). In this paper, we propose adaptive ensemble learning to fuse TR and TL and achieve better ICL performance.

7 Conclusion

In this paper, we presented the first study of the competitive relationship between TR and TL, and quantified its effect on the emergence of ICL. With specially designed metrics, we found that this competition widely exists in existing LLMs, and the competition intensity is negatively correlated with the ICL performance. Then, we conducted a detailed analysis of several pre-training factors (*i.e.,* model size, dataset size, and data curriculum) to demonstrate possible ways to regulate the competition. Furthermore, we proposed a simple yet effective method to better integrate TR and TL at inference time. Through adaptive ensemble learning, the performance of ICL can be significantly boosted, enabling two small models to outperform a larger one with more than twice the parameters.

Overall, our work provides novel approaches and insights to study and understand the underlying mechanism of ICL, which is worth deep exploration for improving the capacity of LLMs.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 92470205 and 62222215, Beijing Municipal Science and Technology Project under Grant No. Z231100010323009, Beijing Natural Science Foundation under Grant No. L233008, and the Outstanding Innovative Talents Cultivation Funded Programs 2022 of Renmin University of China. Xin Zhao is the corresponding author.

References

Suraj Anand, Michael A Lepori, Jack Merullo, and Ellie Pavlick. Dual process learning: Controlling use of in-context vs. in-weights strategies with weight forgetting. *arXiv preprint arXiv:2406.00053*, 2024.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *SemEval@NAACL-HLT*, pp. 54–63. Association for Computational Linguistics, 2019.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, pp. 632–642. The Association for Computational Linguistics, 2015.

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

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891, 2022.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *IWP@IJCNLP*. Asian Federation of Natural Language Processing, 2005.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. Olmo: Accelerating the science of language models. *CoRR*, abs/2402.00838, 2024.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zhen Leng Thai, Kai Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies. *CoRR*, abs/2404.06395, 2024a.

Yiwen Hu, Huatong Song, Jia Deng, Jiapeng Wang, Jie Chen, Kun Zhou, Yutao Zhu, Jinhao Jiang, Zican Dong, Wayne Xin Zhao, et al. Yulan-mini: An open data-efficient language model. *arXiv preprint arXiv:2412.17743*, 2024b.

Joonwon Jang, Sanghwan Jang, Wonbin Kweon, Minjin Jeon, and Hwanjo Yu. Rectifying demonstration shortcut in in-context learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4294–4321, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *KR*. AAAI Press, 2012.

Raymond Li, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, LI Jia, Jenny Chim, Qian Liu, et al. Starcoder: may the source be with you! *Transactions on Machine Learning Research*, 2023a.

Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. Deep model fusion: A survey. *arXiv preprint arXiv:2309.15698*, 2023b.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*, 2023.

Ziqian Lin and Kangwook Lee. Dual operating modes of in-context learning. *arXiv preprint arXiv:2402.18819*, 2024.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. LLM360: towards fully transparent open-source llms. *CoRR*, abs/2312.06550, 2023.

Zhengzhong Liu, Bowen Tan, Hongyi Wang, Willie Neiswanger, Tianhua Tao, Haonan Li, Fajri Koto, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Liqun Ma, Liping Tang, Nikhil Ranjan Ranjan, Zhuang Yonghao, He Guowei, Renxi Wang, Mingkai Deng, Robin Algayres, Yuanzhi Li, Zhiqiang Shen, Preslav Nakov, and Eric Xing. Llm360 k2-65b: Scaling up fully transparent open-source llms, 2024. URL `https://www.llm360.ai/paper2.pdf`.

Pekka Malo, Ankur Sinha, Pekka J. Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*, 65(4): 782–796, 2014.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *LREC*, pp. 216–223. European Language Resources Association (ELRA), 2014.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*, pp. 11048–11064. Association for Computational Linguistics, 2022.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *SemEval@NAACL-HLT*, pp. 1–17. Association for Computational Linguistics, 2018.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. ETHOS: an online hate speech detection dataset. *CoRR*, abs/2006.08328, 2020.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning ”learns” in-context: Disentangling task recognition and task learning. In *ACL (Findings)*, pp. 8298–8319. Association for Computational Linguistics, 2023.

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *The Twelfth International Conference on Learning Representations*, 2023.

Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 8(4):e1249, 2018.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: contextualized affect representations for emotion recognition. In *EMNLP*, pp. 3687–3697. Association for Computational Linguistics, 2018.

Emily Sheng and David C. Uthus. Investigating societal biases in a poetry composition system. *CoRR*, abs/2011.02686, 2020.

Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, June 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pp. 1631–1642. ACL, 2013.

Norman Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. *Advances in Neural Information Processing Systems*, 33:15300–15311, 2020.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. In *SIGIR*, pp. 200–207. ACM, 2000.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2019.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. Baichuan 2: Open large-scale language models. *CoRR*, abs/2309.10305, 2023.

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

(a) Different numbers of intermediate checkpoints.

# Checkpoint	8	16	32
Pythia-6.9B	12.50	37.50	53.12
OLMo-7B	50.00	37.50	43.75
MiniCPM-2B	62.50	56.25	37.50

(b) Different numbers of examples in the demonstration.

# Examples	4	8	16
Pythia-6.9B	18.75	25.00	37.50
OLMo-7B	5.00	37.50	37.50
MiniCPM-2B	43.75	31.25	56.25

(c) Different types of abstract labels.

Abstract label	Symbols	Numbers	Letters
Pythia-6.9B	37.50	18.75	37.50
OLMo-7B	37.50	50.00	43.75
MiniCPM-2B	56.25	43.75	50.00

A Tasks and Datasets

We conduct experiments on four types of tasks: Sentiment Analysis, Topic/Stance Classification, Toxicity Detection, and Natural Language Inference/Paraphrase Detection. For **Sentiment Analysis**, we use datasets including SST-2 (Socher et al., 2013), financial_phrasebank (Malo et al., 2014), emotion (Saravia et al., 2018), and poem_sentiment (Sheng & Uthus, 2020). For **Topic/Stance Classification**, we utilize TREC (Voorhees & Tice, 2000), tweet_eval_atheist, and tweet_eval_feminist (Mohammad et al., 2018; Basile et al., 2019). For **Toxicity Detection**, we include tweet_eval_hate, ethos_race, ethos_gender, ethos_national_origin, and ethos_religion (Mollas et al., 2020). For **Natural Language Inference/Paraphrase Detection**, we employ SICK (Marelli et al., 2014), SNLI (Bowman et al., 2015), WNLI (Levesque et al., 2012), and MRPC (Dolan & Brockett, 2005).

We follow Min et al. (2022) to select samples from the training set as demonstrations. Additionally, we randomly sample 300 examples as the development set for validation in Section 4 and another 1000 examples as the test set for evaluation in all experiments from the development set.

B More Experiments

B.1 The Number of Intermediate Checkpoints

In the paper, we use 16 checkpoints in addition to the final one. In this part, we conduct experiments using different numbers of checkpoints (*i.e.,* 8 and 32). We report the average competition ratio across 16 datasets and 5 random seeds. Table 7a shows that the number of checkpoints does not affect the experimental results. They consistently demonstrate that there is a competitive relationship between TR and TL during the pre-training process.

B.2 The Numbers of Examples in Demonstration

In the paper, we use 16 randomly sampled examples as demonstrations. To explore the impact of the number of examples, we report the average competition ratio with other numbers (*i.e.,* 4 and 8) of demonstrations. As presented in Table 7b, it can be observed that the number of examples does not affect the competitive relationship during the pre-training process.

B.3 The Type of Abstract Labels

In the paper, we utilize symbols in the abstract setting. In this part, we follow Pan et al. (2023) to use other types of semantically unrelated labels (*i.e.,* numbers and letters). Table 7c shows the average competition ratio by using different labels. It indicates that, regardless of the choice of semantically unrelated labels, the conclusions are consistent with the abstract symbols.

C Related Work

Our work is closely related to the studies on the mechanisms of ICL and model fusion.

The Mechanism of ICL. In-context learning (ICL) has garnered significant interest as a core capability of large language models (LLMs), enabling them to perform tasks without fine-tuning by simply leveraging input demonstrations. Existing studies investigate ICL mechanisms through both pre-training and inference dynamics. Some work discusses how ICL emerges from pre-training by conducting analysis on pre-training factors like data (Chan et al., 2022; Reddy, 2023) and optimization (Singh et al., 2024; Anand et al., 2024). Other work (Pan et al., 2023; Min et al., 2022; Dai et al., 2023) studies the operating mechanism of ICL at inference time. Researchers empirically find two main abilities in ICL: task recognition (TR) for recognizing the task and utilizing pre-trained priors of LLMs (Min et al., 2022) and task learning (TL) for learning from demonstrations (Dai et al., 2023). The interaction of TR and TL is crucial for effective ICL, and understanding this interaction has been a focus of empirical and theoretical research Pan et al. (2023); Wei et al. (2023). While these studies provide valuable insights, they often examine TR and TL in isolation. The interplay between these abilities, particularly how they emerge during pre-training, remains underexplored. In this work, we bridge this gap by investigating how TR and TL affect the emergence of ICL through examining the pre-training dynamics of LLMs.

Model Fusion. Model fusion aims to enhance performance by combining the strengths of multiple models (Li et al., 2023b). One line of research focuses on minimizing discrepancies among models through strategies like mode connectivity (Nagarajan & Kolter, 2019) and alignment techniques (Tatro et al., 2020). Mode connectivity methods (Nagarajan & Kolter, 2019) align model representations by finding low-loss pathways in parameter space, facilitating smoother transitions and improved generalization. Alignment techniques (Tatro et al., 2020) focus on harmonizing model predictions or representations, often through methods like knowledge distillation or contrastive learning. These approaches aim to create more consistent models that are easier to integrate. Another line of research capitalizes on diversity to enhance ensemble performance. Techniques such as weight averaging (Wang et al., 2019) and ensemble learning (Sagi & Rokach, 2018) combine models with varying inductive biases or training histories. By aggregating predictions or interpolating model parameters, these methods harness complementary strengths to achieve robust and adaptive performance. The potential of model fusion to improve ICL performance is a promising but underexplored area. While TR and TL abilities have been analyzed independently in individual models, their integration through fusion strategies remains unaddressed. In this paper, we introduce a novel adaptive ensemble learning approach that combines TR and TL, demonstrating its efficacy in enhancing ICL across diverse tasks.

Table 2: Averaged accuracy and TFLOPs across 16 datasets. "Fixed" and "adaptive" denote fixed and adaptive fusion weights. We highlight the best performance among various model selections for TR and TL. Numbers marked with * indicate that the improvement is statistically significant compared with baselines (t-test with p-value < 0.05).

Model	**# Parameters**	**TFLOPs**	**Accuracy**
Large models			
Amber-7B$_{\texttt{ICL}}$	7B	9.98$_{\pm 0.35}$	50.08$_{\pm 0.18}$
OLMo-7B$_{\texttt{ICL}}$	7B	8.79$_{\pm 0.32}$	52.10$_{\pm 0.19}$
Baichuan2-7B$_{\texttt{ICL}}$	7B	8.18$_{\pm 0.41}$	52.77$_{\pm 0.16}$
CrystalCoder-7B$_{\texttt{ICL}}$	7B	10.17$_{\pm 0.45}$	55.66$_{\pm 0.31}$
Pythia-12B$_{\texttt{ICL}}$	12B	15.51$_{\pm 0.57}$	52.77$_{\pm 0.21}$
Small models			
Pythia-1B$_{\texttt{ICL}}$	1B	1.24$_{\pm 0.15}$	44.55$_{\pm 0.33}$
Pythia-2.8B$_{\texttt{ICL}}$	2.8B	3.60$_{\pm 0.20}$	45.38$_{\pm 0.32}$
MiniCPM-2B$_{\texttt{ICL}}$	2.7B	3.84$_{\pm 0.21}$	50.08$_{\pm 0.16}$
Parameter fusion of small models			
Pythia-1B$_{\texttt{TR}}$+ Pythia-1B$_{\texttt{TL}}$ (fixed)	1B	1.24$_{\pm 0.15}$	46.63$_{\pm 0.26}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-1B$_{\texttt{TL}}$ (adaptive)			45.61$_{\pm 0.29}$
Pythia-2.8B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$ (fixed)	2.8B	3.60$_{\pm 0.20}$	47.57$_{\pm 0.23}$
Pythia-2.8B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$ (adaptive)			47.18$_{\pm 0.13}$
MiniCPM-2B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (fixed)	2.7B	3.84$_{\pm 0.21}$	52.18$_{\pm 0.26}$
MiniCPM-2B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (adaptive)			52.06$_{\pm 0.27}$
Logit fusion of small models (Ours)			
Pythia-1B$_{\texttt{TR}}$+ Pythia-1B$_{\texttt{TL}}$ (fixed)	2B	2.48$_{\pm 0.21}$	56.16$_{\pm 0.40}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-1B$_{\texttt{TL}}$ (adaptive)			**56.25**$^{*}_{\pm 0.38}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$ (fixed)	3.8B	4.84$_{\pm 0.25}$	56.62$_{\pm 0.41}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$ (adaptive)			**56.83**$^{*}_{\pm 0.41}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TR}}$ (fixed)			55.23$_{\pm 0.47}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TR}}$ (adaptive)			55.39$_{\pm 0.40}$
Pythia-1B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (fixed)	3.7B	5.08$_{\pm 0.26}$	55.21$_{\pm 0.43}$
Pythia-1B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (adaptive)			55.31$_{\pm 0.34}$
Pythia-1B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TR}}$ (fixed)			54.31$_{\pm 0.63}$
Pythia-1B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TR}}$ (adaptive)			**55.85**$^{*}_{\pm 0.75}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (fixed)	6.5B	8.68$_{\pm 0.33}$	53.58$_{\pm 0.53}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (adaptive)			55.00$_{\pm 0.75}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TR}}$ (fixed)			53.67$_{\pm 0.44}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TR}}$ (adaptive)			56.32$_{\pm 0.42}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TR}}$ (fixed)			53.03$_{\pm 0.48}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TR}}$ (adaptive)			55.26$_{\pm 0.37}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TR}}$ (fixed)			57.24$_{\pm 0.42}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TR}}$ (adaptive)			57.35$_{\pm 0.28}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (fixed)			55.94$_{\pm 0.58}$
Pythia-1B$_{\texttt{TL}}$+ Pythia-2.8B$_{\texttt{TR}}$+ MiniCPM-2B$_{\texttt{TL}}$ (adaptive)			56.45$_{\pm 0.55}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TL}}$ (fixed)			58.05$_{\pm 0.32}$
Pythia-1B$_{\texttt{TR}}$+ Pythia-2.8B$_{\texttt{TL}}$+ MiniCPM-2B$_{\texttt{TL}}$ (adaptive)			**58.27**$^{*}_{\pm 0.40}$

Table 3: Ablation study for model fusion. We highlight the best performance among various model selections for TR and TL. "Random" means that the checkpoint is randomly selected, while "Best" means that the checkpoint has the best performance for TR/TL. Numbers marked with * indicate that the improvement is statistically significant compared with others (t-test with p-value < 0.05).

Models for fusion	**Model selection for TR**	**Model selection for TL**	**Accuracy**
TR: Pythia-1B TL: Pythia-1B	TR-Best	-	$47.47_{\pm 0.19}$
	-	TL-Best	$44.63_{\pm 0.32}$
	Random	Random	$52.66_{\pm 0.44}$
	TR-Best	Random	$53.52_{\pm 0.32}$
	Random	TL-Best	$54.19_{\pm 0.45}$
	TR-Best	ICL-Best	$55.53_{\pm 0.22}$
	ICL-Best	TL-Best	$54.76_{\pm 0.29}$
	ICL-Best	ICL-Best	$54.56_{\pm 0.40}$
	TR-Best	TL-Best	$\mathbf{56.25}^{*}_{\pm 0.38}$
TR: Pythia-1B TL: Pythia-2.8B	TR-Best	-	$47.47_{\pm 0.19}$
	-	TL-Best	$42.05_{\pm 0.09}$
	Random	Random	$48.10_{\pm 0.14}$
	TR-Best	Random	$50.76_{\pm 0.23}$
	Random	TL-Best	$55.42_{\pm 0.49}$
	TR-Best	ICL-Best	$54.90_{\pm 0.44}$
	ICL-Best	TL-Best	$55.93_{\pm 0.32}$
	ICL-Best	ICL-Best	$54.11_{\pm 0.29}$
	TR-Best	TL-Best	$\mathbf{56.83}^{*}_{\pm 0.41}$
TR: Pythia-2.8B TL: Pythia-1B	TR-Best	-	$44.28_{\pm 0.30}$
	-	TL-Best	$44.63_{\pm 0.32}$
	Random	Random	$52.48_{\pm 0.66}$
	TR-Best	Random	$53.89_{\pm 0.38}$
	Random	TL-Best	$54.39_{\pm 0.43}$
	TR-Best	ICL-Best	$54.56_{\pm 0.43}$
	ICL-Best	TL-Best	$54.90_{\pm 0.49}$
	ICL-Best	ICL-Best	$53.56_{\pm 0.34}$
	TR-Best	TL-Best	$\mathbf{55.39}^{*}_{\pm 0.40}$
TR: Pythia-1B TL: MiniCPM-2B	TR-Best	-	$47.47_{\pm 0.19}$
	-	TL-Best	$47.49_{\pm 0.24}$
	Random	Random	$53.99_{\pm 0.55}$
	TR-Best	Random	$54.79_{\pm 0.36}$
	Random	TL-Best	$54.51_{\pm 0.32}$
	TR-Best	ICL-Best	$54.81_{\pm 0.41}$
	ICL-Best	TL-Best	$55.02_{\pm 0.17}$
	ICL-Best	ICL-Best	$54.55_{\pm 0.25}$
	TR-Best	TL-Best	$\mathbf{55.31}_{\pm 0.34}$
TR: MiniCPM-2B TL: Pythia-1B	TR-Best	-	$45.10_{\pm 0.18}$
	-	TL-Best	$44.63_{\pm 0.32}$
	Random	Random	$52.07_{\pm 0.47}$
	TR-Best	Random	$53.61_{\pm 0.12}$
	Random	TL-Best	$53.73_{\pm 0.34}$
	TR-Best	ICL-Best	$55.76_{\pm 0.33}$
	ICL-Best	TL-Best	$54.88_{\pm 0.70}$
	ICL-Best	ICL-Best	$54.39_{\pm 0.48}$
	TR-Best	TL-Best	$\mathbf{55.85}_{\pm 0.75}$