

# What Do Language Models Learn in Context? The Structured Task Hypothesis.

Jiaoda Li\* Yifan Hou\* Mrinmaya Sachan Ryan Cotterell

{jiaoda.li, yifan.hou, mrinmaya.sachan, ryan.cotterell}@inf.ethz.ch

ETH zürich

## Abstract

Large language models (LLMs) exhibit an intriguing ability to learn a novel task from in-context examples presented in a demonstration, termed in-context learning (ICL). Understandably, a swath of research has been dedicated to uncovering the theories underpinning ICL. One popular hypothesis explains ICL by *task selection*. LLMs identify the task based on the demonstration and generalize it to the prompt. Another popular hypothesis is that ICL is a form of *meta-learning*, i.e., the models learn a learning algorithm at pre-training time and apply it to the demonstration. Finally, a third hypothesis argues that LLMs use the demonstration to select a *composition of tasks learned during pre-training* to perform ICL. In this paper, we empirically explore these three hypotheses that explain LLMs’ ability to learn in context with a suite of experiments derived from common text classification tasks. We invalidate the first two hypotheses with counterexamples and provide evidence in support of the last hypothesis. Our results suggest an LLM could learn a novel task in context via composing tasks learned during pre-training.

 [https://github.com/eth-lre/LLM\\_ICL](https://github.com/eth-lre/LLM_ICL)

## 1 Introduction

In-context learning (ICL) is a learning paradigm where a pre-trained large language model (LLM) learns to perform a certain task by extrapolating beyond a demonstration of the task in the form of example prompt–response pairs, given to the model as input. In-context learning does *not* require an update to the model’s parameters (Radford et al., 2019; Brown et al., 2020). Conditioned on the demonstration, the LLM is then tasked with generating responses to additional related prompts. Pre-trained large language models have exhibited an impressive ability to learn in context across various domains, e.g., code generation (Chen et al., 2021), education (Kasneci et al., 2023), and

\* Equal contribution

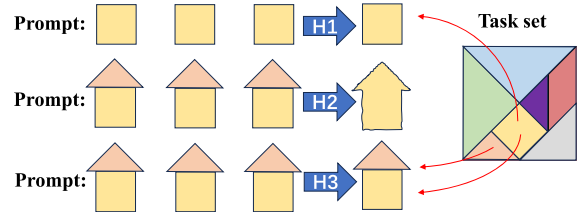


Figure 1: The illustration of three hypotheses.

medicine (Thirunavukarasu et al., 2023). However, there is still no consensus on when or how ICL works. We taxonomize existing candidate theories into three competing hypotheses (Fig. 1), which we summarize below.

**Hypothesis 1** (Informal; Task Selection). *During pre-training, an LLM learns a set of tasks  $\overline{\mathcal{T}}$ . At inference time, the LLM identifies the task  $\tau \in \overline{\mathcal{T}}$  given the user-provided demonstration of the task and generalizes to the prompt.*

Under Hypothesis 1, the demonstration merely allows the model to *recognize* a task, and no actual learning takes place. Min et al. (2022) offers empirical support for Hypothesis 1; they show that randomly shuffling the responses in the demonstration hardly has any effect on ICL performance, suggesting the demonstration of the task only serves to enable the LLM to look up a task. In other words, Hypothesis 1 asserts that no *learning* takes place during ICL. Some authors (Xie et al., 2022; Wang et al., 2023; Wies et al., 2023) have also argued for Hypothesis 1 from a theoretical angle, contending that if an LLM is pre-trained on a corpus that is generated from a mixture model over tasks, it will be able to infer the task that generated the demonstrations to be able to generalize to a prompt not in the demonstration.

The next hypothesis revolves around meta-learning (Schmidhuber, 1987; Thrun and Pratt, 1998; Vilalta and Drissi, 2001; Finn et al., 2017).

**Hypothesis 2** (Informal; Meta-Learning). *During pre-training, an LLM learns certain learning algorithms. During ICL, the LLM learns a task  $\tau$  directly from the demonstration using one of the*

learned learning algorithms.

Hypothesis 2 suggests that the pre-training stage prepares the parameters in an LLM in such a way that various learning algorithms, e.g., gradient descent and least squares regression, can be implicitly deployed during ICL (Von Oswald et al., 2023; Akyürek et al., 2023; Dai et al., 2023) to learn a new task from a demonstration. However, the setting assumed in the above-cited theoretical development is usually over-simplified, e.g., the assumption that the attention mechanism is linear. Moreover, the empirical evidence for Hypothesis 2 is mostly derived from experiments on small transformers trained from scratch on synthetic data (Garg et al., 2022; Raventos et al., 2023; Akyürek et al., 2024).

The final hypothesis may be viewed as a mixture of Hypothesis 1 and Hypothesis 2.

**Hypothesis 3** (Informal; Structured Task Selection). *During pre-training, an LLM learns a set of tasks  $\overline{\mathcal{T}}$ . At inference time, the LLM uses the demonstration to compose a sequence of learned tasks  $\tau_1, \tau_2, \dots \in \overline{\mathcal{T}}$  and uses this composition for prediction. The composition itself may result in a novel task not seen during pre-training.*

Hypothesis 3 extends Hypothesis 1 by allowing ICL to not only index into the pre-learned task set  $\overline{\mathcal{T}}$ , but also compose tasks to obtain a novel task  $\tau = \tau_1 \circ \tau_2 \circ \dots \notin \overline{\mathcal{T}}$ .<sup>1</sup> Hahn and Goyal (2023) lay the theoretical groundwork for Hypothesis 3; under their framework, they argue that the compositional structure of linguistic pre-training data gives rise to the task composition ability of ICL.

In this paper, we conduct a comprehensive battery of experiments to examine the above-stated three hypotheses. Given a demonstration of a task, i.e., a sequence of prompt–response pairs from which an LLM can successfully learn the task in context, we create a response-altered (RA) task by altering the responses such that the new task is unlikely to have occurred during pre-training. Our results confirm that LLMs can learn such an RA task in context, which rejects Hypothesis 1. We then create prompt-altered (PA) tasks, which instead alter the prompts. If Hypothesis 2 is true, this modification should not affect the performance of ICL. However, we find that the LLMs yield a substantially worse performance on PA tasks than on RA tasks, which contradicts Hypothesis 2. Lastly, in support of Hypothesis 3, we identify a sequence of simple tasks that the LLMs can compose

<sup>1</sup>The definition of task composition  $\circ$  will be given in § 2.2.

to obtain unobserved RA tasks. It offers a possible explanation for ICL’s ability to perform novel tasks.

## 2 Preliminaries

### 2.1 Language Models

A language model  $p$  is a distribution over  $\Sigma^*$  where  $\Sigma$  is an alphabet. The elements of  $\Sigma$  are tokens. A string  $\mathbf{w} = w_1 \cdots w_N$  of length  $N$  is a finite sequence of tokens  $w_n \in \Sigma$ . Most modern LLMs are defined in an autoregressive manner, i.e.,

$$p(\mathbf{w}) = p(\text{EOS} \mid \mathbf{w}) \prod_{n=1}^N p(w_n \mid \mathbf{w}_{<n}), \quad (1)$$

and each local conditional distribution  $p(\cdot \mid \mathbf{w}_{<n})$  is defined over  $\overline{\Sigma} \stackrel{\text{def}}{=} \Sigma \cup \{\text{EOS}\}$ . We denote  $\mathbf{w}_{<n} \stackrel{\text{def}}{=} w_1 \cdots w_{n-1}$  and  $\mathbf{w}_{<1} \stackrel{\text{def}}{=} \varepsilon$ . Note that not all models defined as in (1) are distributions over  $\Sigma^*$ . However, in the context of our paper, we assume  $p$  is.

### 2.2 A Restatement of the Hypotheses

In this section, we offer a more formal version of each of the three hypotheses discussed in § 1. Note that our treatment is decidedly *not* a formalization. Nevertheless, we do find it useful to build up some level of formal notation to discuss the three hypotheses more precisely; we save a true formalization for future work.

We start with a concrete definition of a task. In this paper, a **task**  $\tau \in \mathcal{T}$  is taken to be a pair of language models  $\langle \pi_\tau, \rho_\tau \rangle$  where both  $\pi_\tau$  and  $\rho_\tau$  are distributions over  $\Sigma^*$ . We interpret  $\pi_\tau$  as a distribution over prompts for task  $\tau$  and  $\rho_\tau$  as a distribution over responses *conditioned* on a prompt. Let  $\mathcal{T}$  be a countable set of tasks. A **demonstration** of a task  $\tau \in \mathcal{T}$  is an interwoven sequence of prompt–response pairs. We denote a demonstration of length  $L$  as  $\mathbf{d} = \mathbf{p}_1 \mathbf{r}_1 \mathfrak{h} \cdots \mathfrak{h} \mathbf{p}_L \mathbf{r}_L$ , where  $\mathfrak{h} \in \Sigma$  is a distinguished delimiter. We assume  $\mathbf{p}_\ell \sim \pi_\tau(\cdot)$  and  $\mathbf{r}_\ell \sim \rho_\tau(\cdot \mid \mathbf{p}_\ell)$  for  $1 \leq \ell \leq L$ . With  $\overline{\mathcal{T}} \subset \mathcal{T}$ , we denote the set of tasks observed at the pre-training time, i.e., those tasks where there exist demonstrations  $\mathbf{p}_1 \mathbf{r}_1 \mathfrak{h} \cdots \mathfrak{h} \mathbf{p}_L \mathbf{r}_L$  of the task in the pre-training data.

At inference time, given a demonstration  $\mathbf{d} = \mathbf{p}_1 \mathbf{r}_1 \cdots \mathbf{p}_L \mathbf{r}_L$ , we say that a language model  $p$  has (approximately) learned a task if

$$p(\mathbf{r} \mid \mathbf{d} \mathfrak{h} \mathbf{p}) \approx \sum_{\tau \in \overline{\mathcal{T}}} \rho_\tau(\mathbf{r} \mid \mathbf{p}) p(\tau \mid \mathbf{d} \mathfrak{h} \mathbf{p}), \quad (2)$$

and  $p(\tau \mid \mathbf{d} \mathfrak{h} \mathbf{p})$  is sufficiently low entropy for large  $L$ . In words, (2) says that the task to be performed

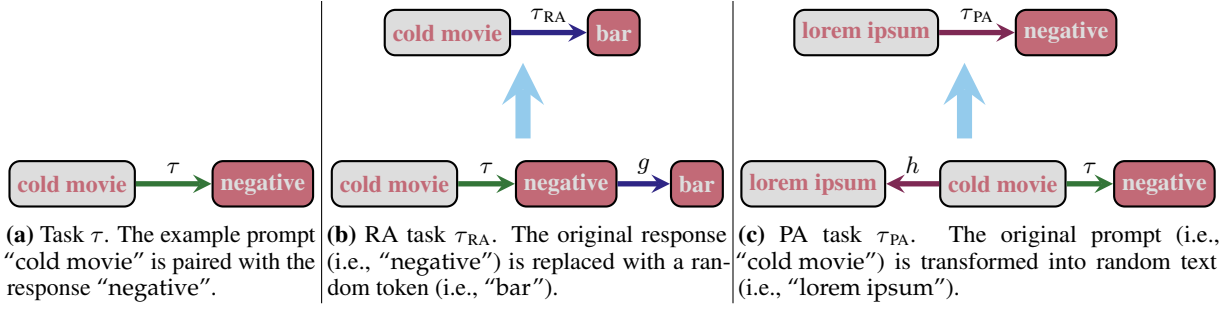


Figure 2: Illustrations of a sentiment classification task, a response-altered (RA) task, and a prompt-altered (PA) task.

by the in-context learning may fruitfully be viewed as a latent variable (Xie et al., 2022). And, moreover, when the task-selection distribution  $p(\tau | \mathbf{d}; \mathbf{p})$  is low entropy for large enough  $L$ , i.e., when we observe a large enough demonstration, the sum is dominated by a single summand. This means the language model has succeeded at identifying the task with high probability. Xie et al. (2022) present a more detailed theoretical framework to explore this scenario in a more precise manner.

We now restate the hypotheses in our notation.

**Hypothesis 1** (Task Selection). *The task-selection distribution  $p(\tau | \mathbf{d}; \mathbf{p})$  is only well-calibrated for tasks in  $\bar{\mathcal{T}}$ , i.e., the finite set of tasks observed at pre-training time.*

**Hypothesis 2** (Meta-Learning). *The task-selection distribution  $p(\tau | \mathbf{d}; \mathbf{p})$  generalizes to some tasks in  $\mathcal{T} \setminus \bar{\mathcal{T}}$ , i.e., tasks not observed at pre-training time.*

To explain our third hypothesis, let  $T$  be a set of **primitive tasks**. We define the notion of task composition as follows. Given two tasks  $\tau_1 = \langle \rho_{\tau_1}, \pi_{\tau_1} \rangle$  and  $\tau_2 = \langle \rho_{\tau_2}, \pi_{\tau_2} \rangle$ , we define

$$\tau_1 \circ \tau_2 \stackrel{\text{def}}{=} \langle \rho_{\tau_1 \circ \tau_2}, \pi_{\tau_2} \rangle, \quad (3)$$

where we further define

$$\rho_{\tau_1 \circ \tau_2}(\mathbf{r} | \mathbf{p}) \stackrel{\text{def}}{=} \sum_{\tilde{\mathbf{r}} \in \Sigma^*} \rho_{\tau_1}(\mathbf{r} | \tilde{\mathbf{r}}) \rho_{\tau_2}(\tilde{\mathbf{r}} | \mathbf{p}), \quad (4)$$

It is easy to see that  $\circ$  is associative.<sup>2</sup> Then, consider the semigroup<sup>3</sup>  $(T^*, \circ)$  where  $\circ$  is as defined above. In other words, any composition of primitive tasks  $\tau_1 \circ \dots \circ \tau_K \in T^*$  results in a new task. This semigroup structure encodes a primitive (non-hierarchical) notion of task composition. We then take  $\mathcal{T} = T^*$ . Finally, let  $\bar{\mathcal{T}}$  be the set of observed primitive tasks. Consider  $(\bar{T}^*, \circ)$ , a

<sup>2</sup>See App. B.

<sup>3</sup>A semigroup is a set endowed with an associative operator, under which the set is closed.

subsemigroup of  $(T^*, \circ)$ . Note that  $\bar{T}^*$  is larger than  $\bar{\mathcal{T}}$ , as it includes tasks *not* observed during the pre-training time, but whose composite primitive tasks were. With this notation, we now present the third hypothesis.

**Hypothesis 3** (Structured Task Selection). *The task-selection distribution  $p(\tau | \mathbf{d})$  is only well-calibrated for  $(\bar{T}^*, \circ)$ , i.e., compositions of primitive tasks observed at pre-training time.*

### 3 Testing Hypothesis 1

According to Hypothesis 1, ICL selects the task it needs to perform from the demonstration. However, it is only able to select among the finite set of tasks observed at the pre-training time, denoted as  $\bar{\mathcal{T}}$ . It follows from Hypothesis 1, then, that if a novel task that has never been seen during pre-training is presented to a pre-trained model as a demonstration, ICL should not be able to perform it. We construct such a novel task as follows. We first create a string-to-string function  $g: \Sigma^* \rightarrow \Sigma^*$ . Note that such a function  $g$  is a special case of a response distribution that places probability 1 on a specific output string for every input string. Then, given a task  $\langle \pi_{\tau}, \rho_{\tau} \rangle$ , we obtain a new task  $\tau_{RA}$  by applying  $g$  to the responses of  $\tau$ , i.e.,

$$\pi_{\tau_{RA}}(\mathbf{p}) \stackrel{\text{def}}{=} \pi_{\tau}(\mathbf{p}) \quad (5a)$$

$$\rho_{\tau_{RA}}(\mathbf{r} | \mathbf{p}) \stackrel{\text{def}}{=} \rho_{\tau}(g^{-1}(\mathbf{r}) | \mathbf{p}). \quad (5b)$$

Note that the definition in (5) is no more than task composition, i.e.,  $\langle g, \bullet \rangle \circ \langle \rho, \pi \rangle$  where  $\bullet$  is a stand-in for an arbitrary prompt distribution. We define  $\tau_g \stackrel{\text{def}}{=} \langle g, \bullet \rangle$  for the remainder of the paper, i.e., a task induced by the string-to-string function  $g$  with a stand-in prompt distribution; we also write  $\tau_{RA} = \tau_g \circ \tau$ . The new task  $\tau_{RA}$  is almost certainly not observed in the pre-training data, i.e.,  $\tau_{RA} \notin \bar{\mathcal{T}}$ . We call  $\tau_{RA}$  a response-altered task (**RA**) and the ICL setting with RA tasks  **$\tau_{RA}$ -ICL**.

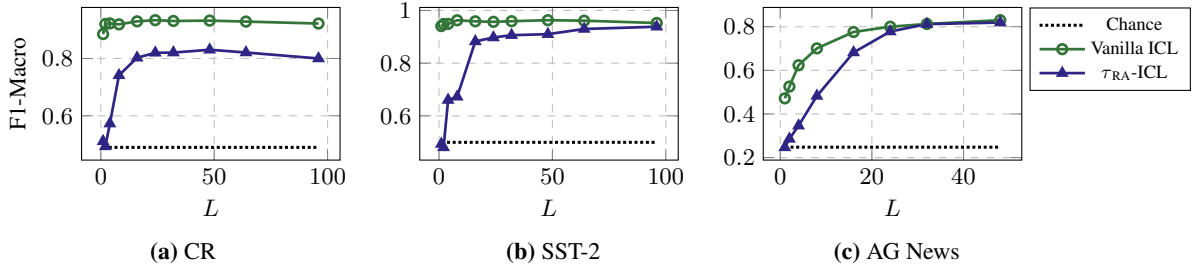


Figure 3: Performance of vanilla ICL and  $\tau_{RA}$ -ICL on the 3 datasets with different demonstration lengths  $L$ . LLaMA2-70B is used. The LLM is able to learn RA tasks as  $L$  grows.

This setting resembles the semantically unrelated label ICL setting of Wei et al. (2023) and the abstract formalization of Pan et al. (2023).

We examine the following logical consequence of Hypothesis 1.

**Prediction 1.** *If Hypothesis 1 is true, then an LLM’s performance in the  $\tau_{RA}$ -ICL setting should be similar to random guessing.*

### 3.1 Experimental Setup

**Tasks.** As shown in Tab. 3 (App. C.1), we select 3 commonly used text classification datasets for ICL: Customer Reviews (CR; Hu and Liu, 2004), Stanford Sentiment Treebank with binary sentiments (SST-2; Socher et al., 2013), and AG News (Zhang et al., 2015); see App. D for the license details.

**Experimental Setup.** Each text classification dataset contains a set of pairs  $\{(\mathbf{x}^{(\ell)}, y^{(\ell)})\}_{\ell=1}^L$  where  $\mathbf{x}^{(\ell)} \in \Sigma^*$  is an input string and  $y^{(\ell)} \in Y$  is  $\mathbf{x}^{(\ell)}$ ’s classification label drawn from  $Y$ , a finite, task-dependent label set. To encode a classification problem as ICL, we map each element of  $Y$  to a string in  $\Sigma^*$  by means of a function  $o: Y \rightarrow \Sigma^*$ . Additionally, we convert the input  $\mathbf{x}$  into a prompt  $\mathbf{p}$  through a templating function  $t: \Sigma^* \rightarrow \Sigma^*$ . The exact templating function we use for each dataset is listed in Tab. 3. Then, we construct a delimiter-separated demonstration of size  $L$ :  $\mathbf{d} = t(\mathbf{x}^{(1)})o(y^{(1)})\dagger \dots \dagger t(\mathbf{x}^{(L)})o(y^{(L)})$ . To perform classification on a test prompt  $t(\mathbf{x})$ , we select the highest-probability class as follows

$$y^* = \operatorname{argmax}_{\tilde{y} \in Y} p(o(\tilde{y}) \mid \mathbf{d}\dagger t(\mathbf{x})). \quad (6)$$

**Settings.** We consider three settings: (1) Chance: random guessing uniformly across different classes;<sup>4</sup> (2) Vanilla ICL: the standard ICL setting; (3)  $\tau_{RA}$ -ICL: The responses  $\mathbf{r}$  in the

<sup>4</sup>All the datasets are largely class-balanced.

demonstrations are replaced by  $g(\mathbf{r})$ . The prompts  $\mathbf{p}$  are left unchanged.

**Implementation Details.** We conduct experiments on a publicly available LLM: LLaMA2 (Touvron et al., 2023) with three sizes: 7B, 13B, and 70B. Each experiment is repeated 20 times with different random seeds and the average F1-Macro score is reported. In each experiment, we construct a distinct test set consisting of 256 prompt–response pairs, and for each element of this test set, we sample  $L$  prompt–response pairs from the training set, which serve as its demonstration.

### 3.2 Results

**ICL Example Number.** We present results on LLaMA2-70B with varying demonstration lengths (labeled as  $L$ ) in Fig. 3. The  $\tau_{RA}$ -ICL’s performance is near chance when the number of prompt–response pairs in the demonstration is small, but it quickly grows to above 80% as  $L$  increases and matches the performance of vanilla ICL on SST-2 and AG News when  $L$  is large.

**Model Size.** In Fig. 4, we report the ICL’s performance ( $L = 32$ ) of models of different sizes. The performance of both vanilla ICL and  $\tau_{RA}$ -ICL generally improves as the model size increases, but even the smallest model (LLaMA2-7B) yields a performance well above chance in  $\tau_{RA}$ -ICL setting.

**Summary.** These results contradict Prediction 1 and demonstrate that an LLM *can* learn RA tasks  $\tau_{RA}$  in context, which are highly unlikely to belong to the set of observed tasks  $\overline{\mathcal{T}}$  during pre-training. These experiments speak against Hypothesis 1.

## 4 Testing Hypothesis 2

We have shown in the previous section that an LLM can learn a novel task in context, but it is still unclear *what type* of tasks can be learned in context.



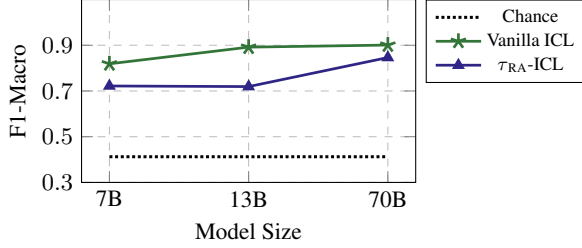


Figure 4: Average performance of **vanilla ICL** and  $\tau_{RA}$ -ICL across 3 datasets (CR, SST-2, AG News). Demonstration length  $L = 32$ . LLaMA2-70B yields the best performance but LLaMA2-7B is not far behind.

If we accept Hypothesis 2 as true, certain learning algorithms are learned by the pre-trained LLM, so learning from a demonstration in context happens on the fly at inference time. This hypothesis implies that there need not be knowledge of the prompted task in the pre-training data. And indeed, on this view, the role of the pre-training is merely to prepare the parameters in the LLM in such a way that the LLM architecture encodes various learning algorithms. For instance, some authors (Von Oswald et al., 2023; Akyürek et al., 2023; Dai et al., 2023) argue that training a linear model with gradient descent can be encoded as in-context learning under certain simplifying assumptions. This leads to the following prediction.

**Prediction 2.** *If Hypothesis 2 is true, then ICL should behave similarly to a model trained with a certain learning algorithm, e.g., gradient descent.*

A caveat of Prediction 2 is, of course, that we do not know *a priori* which learning algorithm the LLMs learn at pre-training time.

## 4.1 Experiment 1

We first propose a prompt-altered (PA) task, where instead of transforming the responses as  $\tau_{RA}$  does, we transform the prompts, i.e., we create a string-to-string function  $h: \Sigma^* \rightarrow \Sigma^*$  and apply it to the prompts of  $\tau$ . This results in the following novel task  $\tau_{PA} = \langle \pi_{\tau_{PA}}, \rho_{\tau_{PA}} \rangle$  where distributions are defined as

$$\pi_{\tau_{PA}}(\mathbf{p}) = \pi_{\tau}(h^{-1}(\mathbf{p})) \quad (7a)$$

$$\rho_{\tau_{PA}}(\mathbf{r} | \mathbf{p}) = \rho_{\tau}(\mathbf{r} | h^{-1}(\mathbf{p})). \quad (7b)$$

The ICL setting with PA tasks is named  $\tau_{PA}$ -ICL. We choose a  $h$  such that the PA task can be learned given the demonstration. If the RA task  $\tau_{RA}$  in the  $\tau_{RA}$ -ICL setting is indeed learned through a meta-learning-esque procedure,  $\tau_{PA}$ -ICL should be just

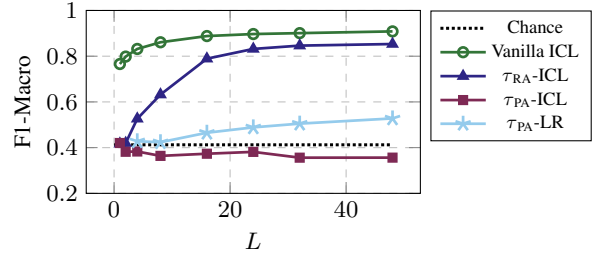


Figure 5: Performance of various settings across 3 text classification tasks. LLaMA2-70B is used.  $\tau_{PA}$ -ICL performs worse than  $\tau_{PA}$ -LR and chance.

as easily learnable in the  $\tau_{PA}$ -ICL setting because  $\tau_{RA}$  and  $\tau_{PA}$  are essentially the same task. And, moreover, performing both tasks in context should exhibit similar performance to a logistic regression classifier if the LLM encodes the ability to learn a linear model implicitly in its parameters.

### 4.1.1 Experimental Setup

Our experimental setup follows § 3.1. However, we introduce two more settings in addition to the ones we consider in § 3.1:

- $\tau_{PA}$ -ICL: As defined in (7), the prompts  $\mathbf{p}$  (including the template tokens such as “Review” and “Sentiment”) in the demonstrations are replaced with  $h(\mathbf{p})$ . The responses  $\mathbf{r}$  are left unchanged.
- $\tau_{PA}$ -LR: To make sure a PA task  $\tau_{PA}$  is learnable from a demonstration, we fit a logistic regression (LR) model as a baseline. Because a variable-length string cannot be easily fed into a logistic regressor, we first tokenize a string using the LLaMA2 tokenizer and then convert it into bag-of-words (BoW) representations.<sup>5</sup> The classifier is then trained using a BoW representation of exactly the same  $L$  prompt–response pairs as used in a demonstration of  $\tau_{PA}$ -ICL. We use this baseline to gauge the performance of a model with minimum learning ability.

### 4.1.2 Results

The average accuracy of LLaMA2-70B across 3 datasets are in Fig. 5. More detailed results can be found in App. C.3.

$\tau_{PA}$ -LR. We find that the logistic regressor is able to learn the tasks to some degree, achieving an F1-Macro score of around 50%. In contrast, a random

<sup>5</sup>More accurately, a bag of tokens.

| Dataset    | $\tau_g$ -Linear (F1-Macro %) | $\tau_g$ -ICL (F1-Macro %) | $\gamma_p$   | $\gamma_s$  |
|------------|-------------------------------|----------------------------|--------------|-------------|
| CR / SST-2 | 100.0 $\pm$ 0.0               | 92.7 $\pm$ 15.2            | N.A.         | N.A.        |
| AG News    | 100.0 $\pm$ 0.0               | 93.8 $\pm$ 10.8            | N.A.         | N.A.        |
| DBPedia    | 99.9 $\pm$ 0.8                | 59.8 $\pm$ 19.7            | -0.02 (0.59) | 0.01 (0.83) |

Table 1: Means and variances of the performance of  $\tau_g$ -Linear and  $\tau_g$ -ICL.  $\gamma_p$  is the Pearson correlation coefficient between  $\tau_g$ -Linear and  $\tau_g$ -ICL and  $\gamma_s$  is the Spearman’s rank correlation coefficient. In the parentheses are  $p$ -values. No significant correlation is observed.

guesser achieves an F1-Macro score of 41%. This experiment shows the task is indeed learnable to a certain extent given the demonstration as training data. And, as expected, the performance improves steadily as  $L$  increases.

**$\tau_{PA}$ -ICL.** However, in the  $\tau_{PA}$ -ICL setting, the LLM always performs near or below chance regardless of the size of the demonstration. Indeed, the performance does not improve even when the demonstration length  $L$  reaches the maximum number of tokens allowed for LLaMA2 (Fig. 9).

**$\tau_{RA}$ -ICL.** In stark contrast to the  $\tau_{PA}$ -ICL setting, as shown in Fig. 5, the LLM has an average score above 80% in the  $\tau_{RA}$ -ICL setting.

**Summary.** The enormous performance gap between  $\tau_{RA}$ -ICL and  $\tau_{PA}$ -ICL does not concord with Prediction 2, and, thereby gives us evidence against Hypothesis 2. Specifically, our results imply that even though the RA tasks  $\tau_{RA}$  are novel, they are not learned with some learning algorithm on the fly at inference time. This is in line with the observations of Kossen et al. (2024) and Shen et al. (2024).

## 4.2 Experiment 2

In our second experiment, we focus on one specific theoretical claim—specifically, that LLMs may implicitly learn a linear regression using gradient descent during ICL (Von Oswald et al., 2023; Akyurek et al., 2023; Dai et al., 2023).

### 4.2.1 Experimental Setup

**Tasks.** In addition to CR, SST-2, and AG News, we also consider DBpedia (Zhang et al., 2015), which has many more classes (Tab. 3).

**Experimental Setup.** We consider the task  $\tau_g$  induced by the string-to-string function  $g$ , as introduced in § 3. Given string-label pairs  $\{(x^{(\ell)}, y^{(\ell)})\}_{\ell=1}^L$ , we construct a demonstration for task  $\tau_g$  as follows:  $o(y^{(1)})g(o(y^{(1)}))\dagger \cdots \dagger o(y^{(L)})g(o(y^{(L)}))$ , i.e.,  $\mathbf{p}_\ell = o(y^{(\ell)})$  and  $\mathbf{r}_\ell = g(o(y^{(\ell)}))$ . In this

formulation, both prompts and responses consist of a single token, i.e.,  $\mathbf{p}, \mathbf{r} \in \Sigma$ . We denote the set of distinct responses as  $R \subset \Sigma$ .

The single-token construction allows us to model the task using linear regression. The embedding layer (0<sup>th</sup> layer) of an LLM is a matrix  $\mathbf{E} \in \mathbb{R}^{|\Sigma| \times D}$ , where  $D$  is the dimensionality of embeddings. We retrieve the row vector  $\mathbf{p}_\ell \stackrel{\text{def}}{=} \mathbf{E}_{\mathbf{p}_\ell, :} \in \mathbb{R}^{1 \times D}$  for each prompt  $\mathbf{p}_\ell$ , where  $\mathbf{E}_{\mathbf{p}_\ell, :}$  denotes the row vector in  $\mathbf{E}$  that corresponds to the token  $\mathbf{p}_\ell$ . We vertically stack the row vectors to create the embedding matrix  $\mathbf{P} \in \mathbb{R}^{L \times D}$  of the prompts in the demonstration.<sup>6</sup> Also, let  $\mathbf{R} \in \mathbb{R}^{L \times D}$  be a similarly constructed embedding matrix of the responses, i.e., each response  $\mathbf{r}_\ell$  is embedded as a row vector  $\mathbf{r}_\ell \in \mathbb{R}^{1 \times D}$ . Additionally, for a test prompt–response pair  $\langle \mathbf{p}, \mathbf{r} \rangle = \langle o(y), g(o(y)) \rangle$ , we embed the prompt  $\mathbf{p}$  as a row vector  $\mathbf{p} \in \mathbb{R}^{1 \times D}$ . Thus, learning  $\tau_g$  is reduced to performing a multiple linear regression, i.e., learning a parameter matrix  $\mathbf{W}^* \in \mathbb{R}^{D \times D}$  that minimizes the following (non-strictly) convex objective

$$\mathbf{W}^* \in \underset{\mathbf{W}}{\operatorname{argmax}} \|\mathbf{R} - \mathbf{P}\mathbf{W}\|^2. \quad (8)$$

Moreover, the data are constructed such that the test prompt–response pair  $\langle \mathbf{p}, \mathbf{r} \rangle$  has appeared at least once in  $\{\langle \mathbf{p}_\ell, \mathbf{r}_\ell \rangle\}_{\ell=1}^L$ , so the task does not require generalization at all.

**Settings.** We compare the following two settings:

- **$\tau_g$ -ICL:** We construct a demonstration  $\mathbf{d} = \mathbf{p}_1 \mathbf{r}_1 \dagger \cdots \dagger \mathbf{p}_L \mathbf{r}_L$  and perform classification as follows:

$$\mathbf{r}^* = \underset{\tilde{\mathbf{r}} \in R}{\operatorname{argmax}} p(\tilde{\mathbf{r}} \mid \mathbf{d} \dagger \mathbf{p}). \quad (9)$$

The classification is correct if  $\mathbf{r}^* = \mathbf{r}$ .

- **$\tau_g$ -Linear:** We train a linear regression with gradient descent. At inference time, we use a minimizer  $\mathbf{W}^*$  to compute the predicted embedding

<sup>6</sup>The rows of  $\mathbf{P}$  are linearly independent.

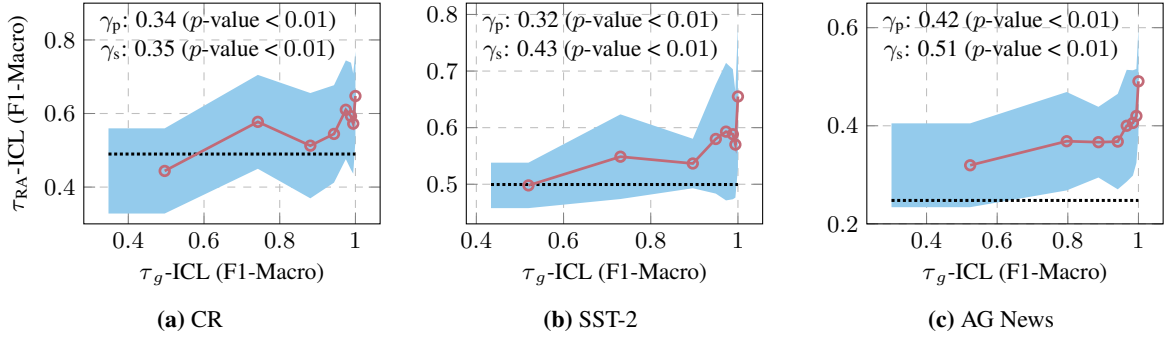


Figure 6: Compare the performance of  $\tau_{RA}$ -ICL ( $y$ -axis) against  $\tau_g$ -ICL ( $x$ -axis). The **dots** represent the mean values, and the **error bars** represent standard deviations. The dashed horizontal line represents the performance of random guessing (i.e., Chance). The Pearson correlation coefficients  $\gamma_p$  and Spearman correlation coefficients  $\gamma_s$  are also reported. Significant correlations ( $p$ -value  $< 0.01$ ) are observed.

vector  $\mathbf{r}^*$  for  $\mathbf{r}$  as follows

$$\mathbf{r}^* = \mathbf{p}\mathbf{W}^* \in \mathbb{R}^{1 \times D}. \quad (10)$$

In order to evaluate its performance against ICL, we utilize the transposed embedding layer  $\mathbf{E}^\top \in \mathbb{R}^{D \times |\Sigma|}$  to project  $\mathbf{r}^*$  into  $\mathbb{R}^{1 \times |\Sigma|}$  and perform classification as follows:

$$\mathbf{r}^* = \underset{\tilde{r} \in R}{\operatorname{argmax}} \left( \mathbf{r}^* \mathbf{E}^\top \right)_{\tilde{r}}. \quad (11)$$

where  $(\mathbf{r}^* \mathbf{E}^\top)_{\tilde{r}}$  denotes the entry of the vector  $(\mathbf{r}^* \mathbf{E}^\top)$  that corresponds to the token  $\tilde{r}$ . We judge classification correct if  $\mathbf{r}^* = \mathbf{r}$ . Note that the embedding layer  $\mathbf{E}$  is taken directly from the LLM and not trained with the linear model.

**Implementation Details.** For all the following experiments in § 4.2 and § 5, we experiment on the largest model (LLaMA2-70B) and set the demonstration length  $L = 32$ . We randomly sample 500 functions  $g$  for  $\tau_g$ . Each mapping  $g$  is constructed by randomly selecting a token from  $\Sigma$  as the image  $g(o(y))$  of a  $o(y)$ . We train  $\tau_g$ -Linear with also 32 examples for 80 epochs which corresponds to the 80 layers of LLaMA2-70B. We choose a learning rate of 1000, which we find yields the best performance. The correlation between the F1-Macro scores of  $\tau_g$ -ICL and  $\tau_g$ -Linear is computed.

#### 4.2.2 Results

As shown in Tab. 1,  $\tau_g$ -Linear can learn most of the functions perfectly. On the other hand,  $\tau_g$ -ICL has a much lower average and higher variance. In the same table, we also list the correlation between the performance of  $\tau_g$ -Linear and  $\tau_g$ -ICL. In contrast to Prediction 2, there does not exist any significant correlation, which again gives us evidence against Hypothesis 2. It is worth mentioning that the high

variance of  $\tau_g$ -ICL’s performance also goes against the claim of Olsson et al. (2022) that there exists a special kind of heads called induction heads that copy any abstract pattern.

### 5 Testing Hypothesis 3

Because neither Hypothesis 1 nor Hypothesis 2 matches our experimental findings, we now turn to Hypothesis 3 to explain the empirical facts.

#### 5.1 Experiment 1

We first examine the following prediction.

**Prediction 3.** Consider  $\tau_{RA} = \tau_g \circ \tau$ . Then, an LLM’s ability to learn  $\tau_{RA}$  in context correlates with its ability to learn  $\tau_g$  in context.

We verify it by comparing the performance of  $\tau_{RA}$ -ICL and  $\tau_g$ -ICL.

##### 5.1.1 Experimental Setup

The experimental setup of  $\tau_{RA}$ -ICL and  $\tau_g$ -ICL follow that of § 3.1 and § 4.2, respectively. The correlation between the F1-Macro scores of  $\tau_{RA}$ -ICL and  $\tau_g$ -ICL is computed.

##### 5.1.2 Results

For each dataset in CR, SST-2, and AG News, we bucket the 500 data points for 500 functions into 8 bins to better visualize the results. The first group contains all functions  $g$  with an F1-Macro of 100% on  $\tau_g$ -ICL. The rest of the data points are put into  $8 - 1 = 7$  bins evenly distributed and by increasing  $\tau_g$  performance. The mean and the standard deviation of  $\tau_{RA}$ -ICL’s performance of each group are reported in Fig. 6. We compute the Pearson correlation coefficient (denoted as  $\gamma_p$ ) and Spearman’s rank correlation coefficient (denoted as  $\gamma_s$ ) between the  $\tau_g$ -ICL scores and the  $\tau_{RA}$ -ICL scores of all the data points. There exists a modest

positive correlation with  $0.34 \leq \gamma_p \leq 0.42$  and  $0.35 \leq \gamma_s \leq 0.51$ , but it is statistically significant with  $p$ -value smaller than 0.01 under Student’s  $t$ -test. We take it as evidence supporting that  $\tau_{RA}$  is learned via composing  $\tau_g$  and  $\tau$ .

## 5.2 Experiment 2

Next, rather than creating string-to-string functions  $g$  randomly, we construct hand-crafted functions that are intuitive and likely to have been in the pre-training data. We call such functions **natural**. One example of such a natural function is  $g_{\text{syn}}$ , where the prompts are mapped to synonyms. We compare ICL performance on these hand-crafted functions against that on random functions. If Hypothesis 3 were true,  $\tau_g$ -ICL should have a significantly higher performance on such natural functions.

### 5.2.1 Experimental Setup

We consider three types of natural functions: (1) synonym: Each prompt  $o(y)$  is mapped to one of its synonyms, (2) antonym: Each prompt  $o(y)$  is mapped to one of its antonyms, and (3) keyword: Each prompt  $o(y)$  is mapped to a keyword in its genre. Synonyms and antonyms are selected using PyMultiDictionary library.<sup>7</sup> Keywords are obtained for each genre by querying GPT-4 (OpenAI, 2023). In contrast to random, we cannot create a large number of natural functions as easily. Therefore, we manually choose a candidate set of 10 possible synonyms (resp. antonyms and keywords) for each prompt. Thus, we can create a natural function by sampling a synonym (resp. antonym and keyword), from the candidate sets for every input. We create 500 such functions.

### 5.2.2 Results

We plot the mean F1 Macro scores as well as the standard deviations of  $\tau_g$ -ICL across different functions in Tab. 2. We observe that the LLM can learn synonym and keyword in context almost perfectly. The function antonym appears to be more difficult to learn, but still clearly much easier than random, which is most evident in DBPedia where the LLM has an F1-Macro score of 84.5% on antonym and only 59.8% on random. We also perform a one-sided Welch’s  $t$ -test between each of antonym, synonym, keyword and random. The  $t$ -values are all greater than 2 and the  $p$ -values are all smaller

<sup>7</sup><https://github.com/ppizarror/PyMultiDictionary>. The library aggregates information from [educalingo.com](http://educalingo.com), [synonym.com](http://synonym.com), and WordNet (Miller, 1994)

| Dataset    | Mapping | F1-Macro (%) | $t$ -test  |            |
|------------|---------|--------------|------------|------------|
|            |         |              | $t$ -value | $p$ -value |
| CR / SST-2 | random  | 93.2 ± 13.6  | N.A.       | N.A.       |
|            | antonym | 97.0 ± 8.5   | 4.35       | < 0.01     |
|            | synonym | 100.0 ± 0.1  | 10.71      | < 0.01     |
|            | keyword | 100.0 ± 0.0  | 10.75      | < 0.01     |
| AG News    | random  | 93.8 ± 10.8  | N.A.       | N.A.       |
|            | antonym | 99.9 ± 0.3   | 12.59      | < 0.01     |
|            | synonym | 100.0 ± 0.0  | 12.72      | < 0.01     |
|            | keyword | 100.0 ± 0.0  | 12.73      | < 0.01     |
| DBPedia    | random  | 59.8 ± 19.7  | N.A.       | N.A.       |
|            | antonym | 84.5 ± 20.5  | 19.42      | < 0.01     |
|            | synonym | 95.8 ± 1.7   | 40.67      | < 0.01     |
|            | keyword | 93.3 ± 4.9   | 36.90      | < 0.01     |

Table 2: The performance of  $\tau_g$ -ICL with different types of functions  $g$ . One-sided  $t$ -tests are performed between the natural functions (antonym, synonym, keyword and the random functions). The LLM learns the natural functions significantly better.

than 0.01. In other words, the natural functions are indeed significantly easier to learn in context, which is in line with our prediction.

## 5.3 Experiment 3

For Hypothesis 3 to be a good hypothesis, however, we need to additionally show why seemingly arbitrary tasks  $\tau_g$  would likely be elements of  $\overline{T^*}$ . We offer a tentative explanation. While we cannot show how to construct an arbitrary string-to-string function  $g$  out of natural functions, we can exhibit compositions of natural functions that appear arbitrary. We do so by composing natural functions, e.g.,  $g_{\text{syn}}$ , repeatedly as follows

$$g_{\text{syn}}^m(\cdot) = \underbrace{g_{\text{syn}}(g_{\text{syn}}(\dots g_{\text{syn}}(\cdot)))}_{\times m}. \quad (12)$$

We call  $g_{\text{syn}}^m$  the  $m^{\text{th}}$  power of  $g_{\text{syn}}$ .

### 5.3.1 Experiment Setup

The powers of the synonym functions are created the same way as in § 5.2.1. Because the candidate sets of synonyms are small (of size 10), to demonstrate how it is possible to create seemingly arbitrary functions, we adversarially sample functions such that for every  $\mathbf{p}$ ,  $g_{\text{syn}}^m(\mathbf{p}) \in S^m(\mathbf{p}) \setminus S^{m-1}(\mathbf{p})$ , where  $S^m(\mathbf{p})$  denotes the set of all the  $m^{\text{th}}$  order synonyms of  $\mathbf{p}$ , i.e., we only keep those that are *not* synonyms of lower orders. As an example of this strategy, we may have the following mapping: “company” is synonymous with “firm”, which is synonymous with “association” (as a noun) and “adamant” (as an adjective). Thus, we would map “adamant” to “company”, a seemingly unrelated word.



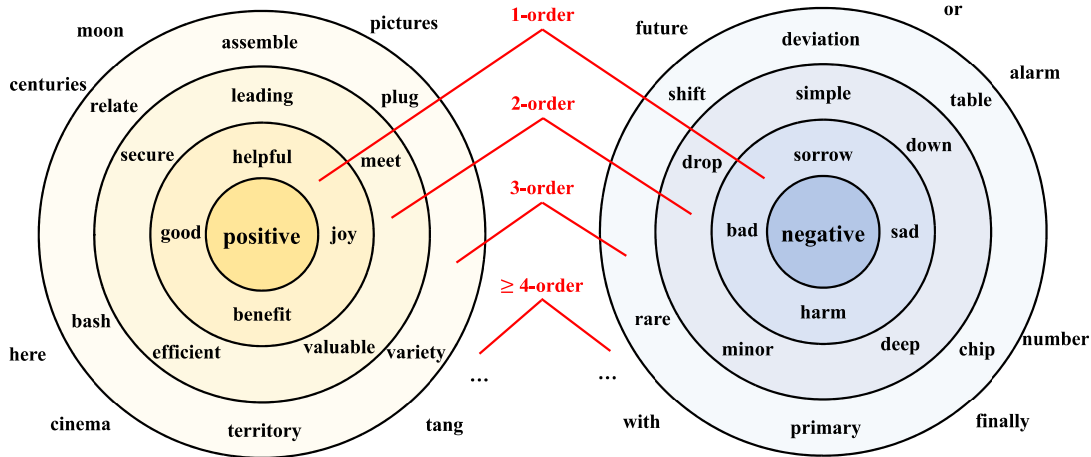


Figure 7: Synonyms of “positive” and “negative”. The words between concentric circles represent elements of the candidate sets of high-order synonyms. As the order gets higher, the synonyms become less related to the seed word.

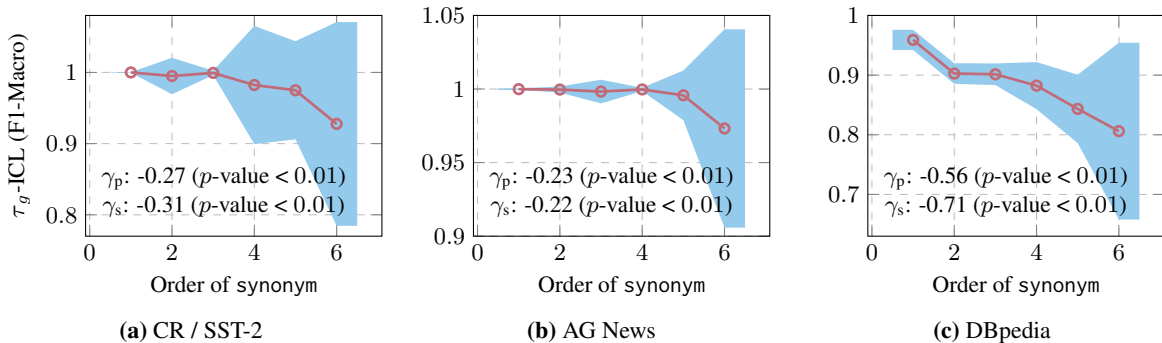


Figure 8: The performance of LLaMA2-70B learning synonym of different orders. The performance in general decreases as the order increases.

### 5.3.2 Results

We visualize a small subset of the synonyms of the two  $o(y)$  of CR/SST in Fig. 7. The concentric circles are the candidate sets, with radii representing their synonym orders. As can be seen, taking a natural function to higher power results in functions that appear as if they were arbitrary string-to-string mappings, e.g.,

“positive”  $\rightarrow$  “tang”  
 “negative”  $\rightarrow$  “or”.

Yet, as our results demonstrate, ICL is still able to learn these functions well—above 90% F1-score—as shown in Fig. 8b. The same also holds true for the AG News and DBpedia text classification datasets. With a large task alphabet  $T$ , we expect an LLM to be able to learn many seemingly arbitrary string-to-string functions. More evidence demonstrating that these functions  $g$  are learned via composition comes from the noticeable performance decrease as the task composition becomes more complex. As shown in Fig. 8, there exists a

negative correlation ( $p$ -value  $< 0.01$ ) between the performance of  $\tau_g$ -ICL and the order of synonym.

## 6 Conclusion

In this paper, we examine the ability of pre-trained LLMs to learn various tasks in context. We find that LLMs *can* learn text classification tasks with corrupted responses ( $\tau_{RA}$ -ICL), but when the prompts are corrupted in the same manner ( $\tau_{PA}$ -ICL), ICL performs significantly worse than a logistic regression model. A closer look suggests the tasks in the  $\tau_{RA}$ -ICL may have been learned via composing primitive tasks learned during pre-training. Overall, our paper provides insights into the nature, abilities, and limitations of ICL.

### Acknowledgements

This publication was made possible by an ETH AI Center doctoral fellowship to Jiaoda Li. Yifan Hou is supported by the Swiss Data Science Center PhD Grant (P22-05).

## Limitations

One limitation is that we only experiment on one family of LLMs—LLaMA2. Our results might not hold on larger models, e.g., GPT-4. Another limitation is that our formalization is not complete and is, sadly, a bit more vibes-based than we would have liked. We hope to achieve a more concrete formalization in future work.

## Ethical Considerations

The datasets and pre-trained LLMs that we use are all publicly available. Our paper focuses on model interpretation, and we thereby do not foresee any ethical issues originating from this work.

## References

- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. [What learning algorithm is in-context learning? investigations with linear models](#). In *International Conference on Learning Representations*.
- Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. 2024. [In-context language learning: Architectures and algorithms](#). *Computing Research Repository*, arXiv:2401.12973. Version 2.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. [Evaluating large language models trained on code](#). *Computing Research Repository*, arXiv:2107.03374.
- Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. [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*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. [Model-agnostic meta-learning for fast adaptation of deep networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.
- Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. [What can transformers learn in-context? A case study of simple function classes](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 30583–30598.
- Michael Hahn and Navin Goyal. 2023. [A theory of emergent in-context learning as implicit structure induction](#). *Computing Research Repository*, arXiv:2303.07971. Version 1.
- Minqing Hu and Bing Liu. 2004. [Mining and summarizing customer reviews](#). In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, et al. 2023. [ChatGPT for good? On opportunities and challenges of large language models for education](#). *Learning and individual differences*, 103:102274.
- Jannik Kossen, Yarin Gal, and Tom Rainforth. 2024. [In-context learning learns label relationships but is not conventional learning](#). In *International Conference on Learning Representations*.
- George A. Miller. 1994. [WordNet: A lexical database for English](#). In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [Rethinking the role of demonstrations: What makes in-context learning work?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann,

- Amanda Aspell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. [In-context learning and induction heads](#). *Computing Research Repository*, arXiv:2209.11895.
- OpenAI. 2023. [GPT-4 technical report](#). *Computing Research Repository*, arXiv:2303.08774.
- Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. [What in-context learning “learns” in-context: Disentangling task recognition and task learning](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8298–8319, Toronto, Canada. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). Github Repository.
- Allan Raventos, Mansheej Paul, Feng Chen, and Surya Ganguli. 2023. [Pretraining task diversity and the emergence of non-bayesian in-context learning for regression](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jürgen Schmidhuber. 1987. [Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook](#). Ph.D. thesis, Technische Universität München.
- Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. 2024. [Do pretrained transformers learn in-context by gradient descent?](#) *Computing Research Repository*, arXiv:2310.08540. Version 5.
- Aaditya K Singh, Stephanie C.Y. Chan, Ted Moskowitz, Erin Grant, Andrew M Saxe, and Felix Hill. 2023. [The transient nature of emergent in-context learning in transformers](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.
- 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.
- Sebastian Thrun and Lorien Pratt. 1998. [Learning to Learn: Introduction and Overview](#), pages 3–17. Springer US, Boston, MA.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiohu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Computing Research Repository*, arXiv:2307.09288. Version 2.
- Ricardo Vilalta and Youssef Drissi. 2001. [A perspective view and survey of meta-learning](#). *Artificial Intelligence Review*, 18.
- Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. [Transformers learn in-context by gradient descent](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR.
- Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023. [Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning](#). In *Workshop on Efficient Systems for Foundation Models @ ICML2023*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. [Emergent abilities of large language models](#). *Transactions on Machine Learning Research*. Survey Certification.
- Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. [Larger language models do in-context learning differently](#). *Computing Research Repository*, arXiv:2303.03846. Version 2.
- Noam Wies, Yoav Levine, and Amnon Shashua. 2023. [The learnability of in-context learning](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. [An explanation of in-context learning as implicit Bayesian inference](#). In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. [Character-level convolutional networks for text classification](#). In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.



## A Related Work

Many papers attempt to provide a theoretical grounding for ICL’s emergence, hypothesizing that ICL emerges from pre-training on documents that are drawn from a mixture of latent concepts (Xie et al., 2022; Wang et al., 2023; Wies et al., 2023) or a compositional attribute grammar (Hahn and Goyal, 2023). After pre-training, the LLM can recognize the concept or the generative process that the prompt is sampled from and use it for next-token prediction. Another stream of research endeavors to show how LLMs can learn new tasks in context. Von Oswald et al. (2023); Akyürek et al. (2023); Dai et al. (2023) show by construction that the transformer can implicitly implement various learning algorithms, such as gradient descent and least squares regression. Nevertheless, there is a gap between the settings assumed in the papers and the ones used in practice, e.g. the attention mechanism needs to be linear, or only linear regression problems are considered. Empirically, it has been shown that transformers can learn various function classes in context, e.g., linear regression (Raventos et al., 2023), multi-layer perceptrons and decision trees (Garg et al., 2022), and regular languages (Akyürek et al., 2024), but the models are trained from scratch on synthetic data derived from functions in the same class, which do not conform with how LLMs are pre-trained in practice. It is argued that their performance might not be extrapolated to models with a larger size (Wei et al., 2022) or longer training time (Singh et al., 2023). Thus, we fix our attention on LLMs that are pre-trained on large natural text corpus and are practically used. Most related to our work are Kossen et al. (2024); Shen et al. (2024), which also find that ICL’s behavior is different from models trained with conventional learning algorithms such as gradient descent. Our work is distinct in the investigation on the possibility of ICL learning via task composition.

## B Associativity of Task Composition

**Proposition 1.** *Task composition is associative, i.e.,  $(\tau_1 \circ \tau_2) \circ \tau_3 = \tau_1 \circ (\tau_2 \circ \tau_3)$ .*

*Proof.* The proof follows by simple manipulation:

$$\begin{aligned}
 \rho_{\tau_1 \circ (\tau_2 \circ \tau_3)}(\mathbf{r} \mid \mathbf{p}) &= \sum_{\tilde{\mathbf{r}}_1 \in \Sigma^*} \rho_{\tau_1}(\mathbf{r} \mid \tilde{\mathbf{r}}_1) \left( \sum_{\tilde{\mathbf{r}}_2 \in \Sigma^*} \rho_{\tau_2}(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{r}}_2) \rho_{\tau_3}(\tilde{\mathbf{r}}_2 \mid \mathbf{p}) \right) \\
 &= \sum_{\tilde{\mathbf{r}}_2 \in \Sigma^*} \left( \sum_{\tilde{\mathbf{r}}_1 \in \Sigma^*} \rho_{\tau_1}(\mathbf{r} \mid \tilde{\mathbf{r}}_1) \rho_{\tau_2}(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{r}}_2) \right) \rho_{\tau_3}(\tilde{\mathbf{r}}_2 \mid \mathbf{p}) \\
 &= \rho_{(\tau_1 \circ \tau_2) \circ \tau_3}(\mathbf{r} \mid \mathbf{p}).
 \end{aligned} \tag{13}$$

$$\begin{aligned}
 \pi_{\tau_1 \circ (\tau_2 \circ \tau_3)} &= \pi_{\tau_2 \circ \tau_3} \\
 &= \pi_{\tau_3} \\
 &= \pi_{(\tau_1 \circ \tau_2) \circ \tau_3}
 \end{aligned} \tag{14}$$

■

## C Supplementary Results

### C.1 Dataset Details

The prompt templating functions  $t(\mathbf{x})$  and the responses  $(o(y))$  for CR, SST-2, and AG News are in Tab. 3. The altered responses are tokens from the *Lorem Ipsum* generator.

### C.2 (First-Order) Synonym

The 10 candidate synonyms for each prompt are in Tab. 4.

### C.3 Detailed Results

We provide detailed results on CR, SST-2, and AG News in Fig. 9. We report LLaMA2 with three model sizes. The results on LLaMA2-7B and LLaMA2-13B are consistent with those on LLaMA2-70B that we report in the main text.

| Dataset | Prompt                       | Response                        |                   |
|---------|------------------------------|---------------------------------|-------------------|
|         | Template                     | $\tau$                          | $\tau_{RA}$       |
| CR      | Review: $x$ \n<br>Sentiment: | positive, negative              | por, Ne           |
| SST-2   | Review: $x$ \n<br>Sentiment: | positive, negative              | por, Ne           |
| AG News | News: $x$ \n<br>News type:   | word, sports, business, science | Mag, Am, Num, Lab |

Table 3: Dataset information, templates, and responses used for  $\tau$  and  $\tau_{RA}$ .

| Dataset    | Prompt  | Synonyms   |
|------------|---|--|
| CR / SST-2 | positive<br>negative  | good, bright, happy, cheer, benefit, fortune, helpful, joy, help, favorite<br>bad, dark, dire, sorrow, harm, down, dim, bitter, sad, blue  |
| AG News    | world<br>sports<br>business<br>science  | earth, planet, universe, sphere, creation, domain, environment, habitat, society, system<br>game, play, exercise, competition, activity, challenge, contest, match, training, racing<br>trade, commerce, industry, company, market, operation, firm, establishment, production, organization<br>research, technology, knowledge, experiment, investigation, theory, discovery, analysis, discipline, learning  |
| DBpedia    | company<br>transport<br>player<br>politics<br>artist<br>animal<br>school<br>plant<br>village<br><br>book<br>nature<br>album<br>building<br>film | firm, business, startup, establishment, operator, producer, chain, brand, office, Agency<br>vehicle, car, train, bus, plane, ship, tram, cab, Metro, carriage<br>runner, footballer, basketball, race, box, golf, cycle, sky, board, sail<br>government, policy, state, nation, election, party, assembly, council, republic, leader<br>painter, writer, composer, singer, actor, designer, director, producer, poet, photograph<br>creature, pet, bird, fish, insect, species, habitat, conservation, wild, migration<br>university, college, prep, primary, secondary, high, middle, grammar, technical, night<br>flower, tree, Fern, grass, leaf, bud, root, branch, seed, growth<br>Township, settlement, community, district, parish, cluster, region, municipality, neighborhood, rural<br>novel, volume, text, manual, guide, reference, edition, journal, cover, series<br>terrain, forest, mountain, river, sea, lake, ocean, beach, desert, garden<br>record, release, compilation, single, track, score, collection, edition, session, live<br>structure, house, stad, tower, hall, temple, palace, castle, fort, shed<br>movie, picture, cinema, feature, animation, drama, comedy, western, mystery, horror |

Table 4: The first-order synonyms used for constructing  $g_{syn}$ .

## D Experimental Setup

We implement our experiments on A100-80G. Each experiment takes around 1-2 GPU hours. For LLaMA2, We use the implementation and pre-trained weights provided by HuggingFace (Wolf et al., 2020). CR (Hu and Liu, 2004) and SST (Socher et al., 2013) are under the CC-BY license. AG News (Zhang et al., 2015) and DBpedia (Zhang et al., 2015) are under the BSD-3-Clause license. We use these datasets consistently with their intended use.

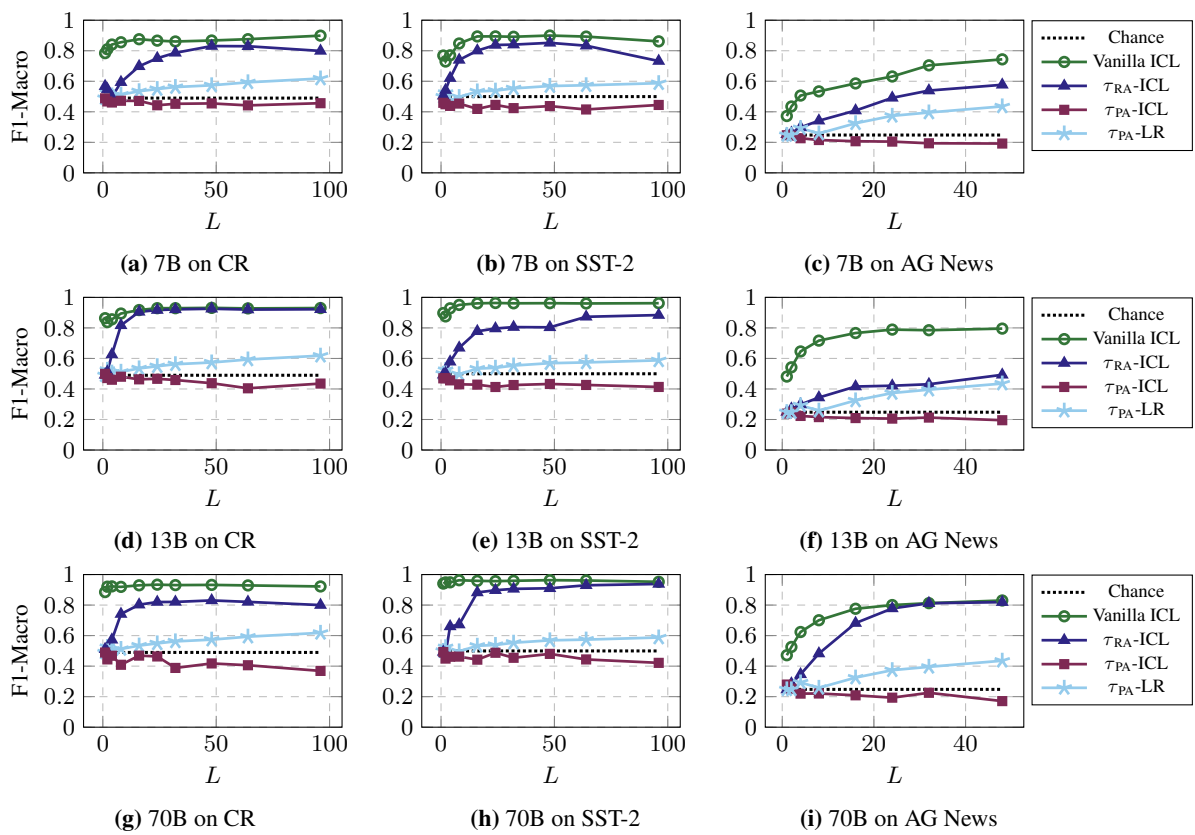


Figure 9: ICL performance with different demonstration lengths  $L$ .