A Survey to Recent Progress Towards Understanding In-Context Learning

Anonymous ACL submission

Abstract

In-Context Learning (ICL) empowers Large Language Models (LLMs) with the ability to learn from a few examples provided in the prompt, enabling downstream generalization without the requirement for gradient updates. Despite encouragingly empirical success, the underlying mechanism of ICL remains unclear. Existing research remains ambiguous with various viewpoints, utilizing intuition-driven and ad-hoc technical solutions to interpret ICL. In this paper, we leverage a data generation perspective to reinterpret recent efforts from a systematic angle, demonstrating the potential broader usage of these popular technical solutions. For a conceptual definition, we rigorously adopt the terms of *skill recognition* and *skill learning*. Skill recognition selects one learned data generation function previously seen during pre-training while skill learning can learn new data generation functions from in-context data. Furthermore, we provide insights into the strengths and weaknesses of both abilities, emphasizing their commonalities through the perspective of data generation. This analysis suggests potential directions for future research.

1 Introduction

LLMs have revolutionized Natural Language Processing (NLP) (Achiam et al., 2023) and other relevant areas such as multi-modal tasks over vision and language (Liu et al., 2023a), accelerating numerous challenging research directions, e.g., AI agent (Durante et al., 2024), reasoning (Wei et al., 2022b), and story telling (Xie et al., 2023). These amazing applications display LLMs' emerging capabilities, which can be formally defined as new abilities that are not present in small models but arise in larger ones (Zhao et al., 2023). Among them, the emerging ICL ability serves as an important foundation of other capabilities. Notably, small models also have the capability to perform

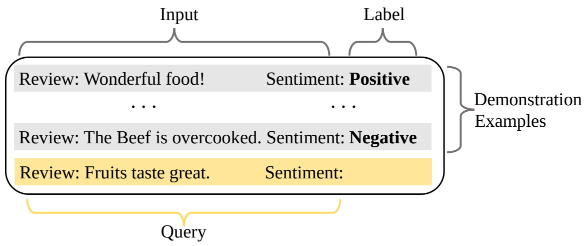

Figure 1: Illustration of ICL for Sentiment Analysis. The upper instances (with background color gray) are the labeled in-context demonstrations, while the last line is the query for which LLMs infer the sentiment label.

ICL, but the level of capability is different from that of larger models, wherein people can easily observe more in-depth displays of understanding for the given context of inputs, e.g., identify long-term dependency and abstract concept comprehension. For instance, Ganguli et al. (2023) demonstrates that only LLMs over 22B parameters can understand the moral concepts, being able to generate unbiased answers.

ICL, a fundamental and emerging capability serving as the pre-requisite for many complicated abilities, is the process of leveraging a few selected labeled demonstrations with the format *(input, label)*[1], before the query input, for making predictions in a few-/one-shot manner. An example of ICL is illustrated in Figure 1.

Despite the empirical success of various ICL prompting strategies for downstream applications (Mavromatis et al., 2023; Ye et al., 2022), the mechanism of ICL remains unclear, leading to unexplainable observations, e.g., sensitivity to the sample order (Lu et al., 2021), or being robust to human-crafted yet irrational input-label mapping. Increasing attention has been paid to understand ICL from various perspectives. However, this area is still growing, with many open research questions are actively being explored. Due to the complexity of LLMs, most existing works only take one indi-

[1] In this paper, we focus on classification tasks as most works on theoretical side of ICL leverages them with well-defined mathematical tools and clear evaluation metrics.

Table 1: A summarization table of representative works. SR and SL stand for skill recognition and skill learning, respectively. Function approximation revolves on how effectively ICL can fit different generalize functions. The Internal Mechanism describes how LLMs learn through various gradient descent algorithms.

Literature	Ability	Analysis View	Date Generation Function	Characteristics
Xie et al. (2021); Zhang et al. (2023c)	SR	Theoretical & Empirical	HMM	Internal Mechanism
Wang et al. (2023)	SR	Empirical	LDA	Generalization
Zhao (2023)	SR	Theoretical	Hopfield Network	Internal Mechanism
Raventos et al. (2023)	SL	Theoretical	linear regression	Generalization
Wu et al. (2023a)	SL	Empirical	linear regression	Generalization
Garg et al. (2022)	SL	Empirical	linear regression, decision tree, NN	Function Approximation
Bai et al. (2023); Fu et al. (2023a)	SL	Theoretical	linear regression, decision tree, NN	Generalization
Yadlowsky et al. (2023); Ahuja et al. (2023)	SL	Empirical	linear regression, polynomial regression	Generalization
Von Oswald et al. (2023); Zhang et al. (2023b) (Mahankali et al., 2023; Ahn et al., 2023a)	SL	Theoretical	linear regression	Internal Mechanism
Akyürek et al. (2022)	SL	Theoretical	linear regression	Internal Mechanism
Li et al. (2023a); Ren and Liu (2023) Cheng et al. (2023); Guo et al. (2023)	SL	Theoretical	non-linear regression	Internal Mechanism
Hahn and Goyal (2023)	SR&SL	Theoretical	context-free grammar	Generalization

vidual factor into account, e.g., the pre-training data distribution (Chan et al., 2022a), model scale (Wei et al., 2023), or difficulty level of the in-context task (Raventos et al., 2023). Moreover, existing works focusing the same factor may adopt different experimental settings (Yoo et al., 2022; Min et al., 2022), leading to potentially conflicting conclusions. Typically, Pan (2023) categorizes ICL into two abilities: task recognition and task learning.

In this paper, we propose the data generation perspective as a principled angle to comprehend existing studies towards understanding ICL. Following this perspective, the pretraining stage can be interpreted as learning the data generation function classes underlying pretraining corpus, where the masked language modeling objective (Devlin et al., 2019) and the next token prediction objective (Radford et al., 2018) are both objectives that allow us learn the data generation functions. Similarly, the ICL stage can be considered as a label generation process given the query inputs. Therefore, adopting this data generation perspective enables a unified framework through which we can cohesively analyze both pretraining and ICL stages, offering a holistic approach to understanding the foundations of LLMs.

Guided by the data generation perspective, we introduce a more principled and rigorous understanding framework on *skill learning* and *skill recognition*, distinguished by whether LLMs can learn a new data generation function in context. The skill learning ability is to learn a new data generation function in context, which is unseen in the pretraining stage. The skill recognition ability selects one learned data generation function previously seen during pre-training. To analyze the mechanism of abilities, the function learning statistical framework (Garg et al., 2022) and the Bayesian inference statistical framework (Xie et al., 2021) are representative works for skill learning and skill recognition ability, respectively.

Organization: Section 2 introduces previous studies of ICL and Section 3 presents the terminology. Key contributions lie in Section 4 and 5, which systematically review the skill recognition with the Bayesian inference framework and the skill learning with the function learning framework, respectively. We outline the challenges and potential directions in Section 6, aiming to offer a valuable guide for newcomers to the field while also illuminating pathways for future research.

2 Related works

Comparison with existing relevant literature. As far as we know, this paper is the first to provide a comprehensive discussion on existing studies about the mechanism of ICL and advocating a principled data generation perspective. This paper distinguishes itself from existing surveys like those by Dong et al. (2022); Zhao et al. (2023); Wei et al. (2022a), which predominantly primarily adopt on a broad, application-oriented perspective, instead of dedicating on the mechanism understanding.

Distinguish skill learning from skill recognition. The skill can be regarded as a data generation function, referring to the underlying hypothesis on the textual data generation. To determine whether the utilized skill is from the pre-training function class or is a new function, an empirical method is to validate whether LLMs can fit a set of data generated with a ground-truth function which is outside the pre-training function class.

Distinguish skill recognition/learning from task recognition/learning (Pan, 2023). We distinguish our proposed skill recognition/learning from a data generation perspective with previous task recognition/learning proposed in (Pan, 2023). Task recognition/learning is a narrower aspect of our skill recognition/learning as they majorly focus on the empirical performance variation under the label permutation on in-context data. Task learning is recognized as performance degradation, indicating ICL learns the permuted in-context data. In contrast, the task recognition corresponds to the unchanged performance, indicating ICL only relies on pre-training knowledge. The key advantages of our proposed skill recognition/learning definition are shown as follows: (1) Thanks to the mathematical description with a data generation function, skill learning/recognition enables both theoretical analysis and empirical evidence, instead of only focusing on the empirical one. (2) Task recognition/learning can only emphasize the performance of a classification task in complicated real-world applications. Instead, skill learning/recognition can utilize different existing data generation functions in the NLP domain, e.g., HMM, and LDA, rather than merely input-label mapping for classification. Moreover, the data generation enables to conduct synthetic analyses in a systematic and controllable setting.

3 Terminology

The prompt sequence of In-Context Learning consists of two parts: (1) The demonstration is illustrated as an *(input, label)* pair, denoted as (x_i, y_i); These demonstrations provide the basic description of the intended task. (2) The query is the test input after a few demonstrations. ICL aims to provide the correct prediction for the query based on the in-context demonstrations and the prior knowledge of a pre-trained LLM. The *data generation function* in this paper refers to the underlying hypothesis on language data generation. It serves as the data assumption in the theoretical understanding and the simulation data generator for the synthetic experimental analysis. Each data generation function obtained by the LLM can be recognized as a skill.

4 Skill Recognition

Skill recognition ability is the ability of an LLM to select the most proper data generation function from the function class obtained during pre-training. And this selection process is driven by the in-context demonstrations. A Bayesian inference framework (Xie et al., 2021) is introduced to explain the skill recognition. The ICL inference can be instantiated as a Bayesian inference process as follows:

$$p(\text{y}|\text{prompt}) = \int_{\text{concept}} p(\text{y}|\text{concept,prompt})p(\text{concept}|\text{prompt})d(\text{concept}) \tag{197}$$

where $p(\text{y}|\text{prompt})$ is the conditional probability of the output generation y given the prompt. It can be marginalized with pre-training concepts and *each concept corresponds to a pre-training data generation function*. $p(\text{concept}|\text{prompt})$ is the probability of locating the latent concept aligned with in-context demonstrations. After locating the aligned concept, $p(\text{y}|\text{concept}, \text{prompt})$ utilizes the selected data generation function for the output generation.

This approach to modeling latent concepts is widely used in the field of NLP, as language data is inherently compositional, involving underlying concepts—such as sentiment, topics, and syntactic structures—that are not explicitly observable in the raw text (Chung et al., 2015; Zhou et al., 2020). Latent variable models can specify prior knowledge and structural dependencies for language data which enjoys the characteristics of high compositionality. Deep latent variable models are popularly utilized to improve various tasks such as alignment in statistical machine translation, topic modeling, and text generation (Kim et al., 2018; Fang et al., 2019; Wang et al., 2023).

Though there are various definitions of latent concepts, any latent information that can help ICL can be considered as a good choice for the *concept* in the Bayesian inference process above. We summarize the existing concept definitions as follows: (1) Xie et al. (2021) defines the concept as the transition matrix θ of a Hidden Markov Model (HMM) (Baum and Petrie, 1966), which assumes to be the underlying distribution of the real-world language data. The concept helps to state a transition distribution over observed tokens. A concrete example of the concept is the transition between name (Albert Einstein) $\rightarrow$ nationality (German) $\rightarrow$ occupation (physicist) in wiki bios. (2) Wang et al. (2023) simplifies the transition between tokens, modeled by HMM, with LDA topic models where each topic corresponds to one latent concept (Blei et al., 2003). (3) Despite the above mathematical interpretations, Todd et al. (2023) and Liu et al. (2023b) empirically establish the connection between the latent concept and the downstream task, e.g., supervised classification and question-answering, where the particular latent representation in the LLM can capture essential information about the task.

The Bayesian inference framework is firstly proposed by Xie et al. (2021), interpreting how obtained pre-training data functions are activated by in-context demonstrations. Key challenges in this framework are: (1) In the pre-training stage, how the model obtains the latent concepts from the pre-training corpus; and (2) In the ICL inference stage, how in-context demonstrations can locate the most relevant concept to generate the desired output.

The pre-training stage aims to obtain various concepts from the large pre-training corpora if each pre-training document is generated from an individual HMM model. In such cases, the next token prediction objective can converge if and only if the LLM can successfully generate the correct next token matching the HMM transitions. The transitions are dominated by the underlying concept (Xie et al., 2021). Different documents can be generated from various concepts sampled from the concept set denoted as Θ.

The ICL inference stage conducts an implicit Bayesian inference to locate an appropriate concept $\theta^* \in \Theta$ which shows the optimal likelihood to generate the given in-context demonstrations. The format of the prompt is shown below:

$$\begin{aligned}&[S_n, x_{\text{test}}]\\&= \left[x_1, y_1, o^{\text{del}}, \ldots, x_n, y_n, o^{\text{del}}, x_{\text{test}}\right] \sim p_{\text{prompt}}\end{aligned} \quad (1)$$

where p_{prompt} is a data generation process implemented with HMM parameterized by θ^*. x_i, y_i and o^{del} are the input, label, and delimiter, respectively. The difficulty in locating θ^* is due to low probability for all the pre-training concepts to generate the in-context demonstrations. The key reason is that token transition patterns of the in-context demonstrations are of three types: (1) the input to the label $x_i \rightarrow y_i$, (2) the label to the delimiter, and (3) the delimiter to the input. The latter two patterns hardly appear in the pre-training data due to different delimiter usages.

To address the above issue of low probability, Xie et al. (2021) proposes some assumptions. One example is the located concept θ^* enjoys a higher probability transiting to delimiters than that of other concepts. Equipped with those assumptions, we are able to locate the aligned pre-training concept to implement Bayesian inference. The model can locate the corrrect concept with $p(\theta^*|\text{prompt}) = 1$ and $p(\theta|\text{prompt}) = 0$ for all $\theta \in \Theta \setminus \theta*$. Even though we cannot locate the aligned concept, Xie et al. (2021) provides the theoretical guarantee on the effectiveness of the ICL in such cases, where the ICL performance improves along with the increasing number of in-context examples.

Inspired by the above Bayesian inference framework, more methods towards understanding skill recognition are proposed, e.g., the PAC-Bayesian framework (Alquier et al., 2024) and Hopfield Network (Hopfield, 2007). Zhang et al. (2023c) analogizes ICL inference to a Bayesian model averaging algorithm. Wies et al. (2023) presents a PAC-based generalization framework exhibiting satisfying generalization bound on the ICL where a transformer trained on multi-task can match the ICL performance of a transformer trained solely on the downstream task. Zhao (2023) analogizes the latent concept location as memory retrieval with the Hopfield Network. More recently, a novel information-theoretic framework (Jeon et al., 2024) has been introduced, decomposing the ICL prediction error into three distinct terms: irreducible error, meta-learning error, and intra-task error. This decomposition helps aligning ICL with existing studies hypothesizing ICL as an instance of meta-learning.

Nonetheless, existing studies are based on either synthetic data or pure theoretical analysis. It could be a promising direction to investigate how LLMs retrieve concepts and how to interpret the retrieved concept through natural language.

5 Skill Learning

Through the skill learning ability, LLMs can inference a new data generation function which has not been seen during pre-training. The function learning framework[2] is utilized to interpret the skill learning ability. Specifically, pre-training is considered as a process to learn a class of functions that can fit the pre-training corpora, and the ICL inference is to learn a new data generation function via fitting the ICL demonstrations.

Discussions on the skill learning ability are organized as follows. In Section 5.1, we first provide

[2] We refer to algorithm learning as function learning with an emphasis on the approximated functions by algorithms and, in this way, it is easier to analyze ICL.

a clear description of the function learning framework and illustrate its benefits and drawbacks. In Section 5.2, we investigate: (1) whether LLMs can learn new functions in context, and (2) if yes, the generalization performance of the learned function. In Section 5.3 illustrates ICL can implement different learning algorithms, e.g., gradient descent. More discussions on the robustness of ICL can be found in Appendix E.

5.1 The Function Learning Framework

Previous research reformulates the pre-training objective of next-token prediction into an input-label mapping objective during the ICL inference stage. One limitation of the function learning framework is that it has to pre-train the model from scratch as the pre-training objective is different from the next token prediction. Due to computational resource limitations, most works utilize transformers with less than 6 layers. These conclusions may not be generalizable to larger scale models. Garg et al. (2022) has been the only work to utilize a relative larger-scale model, reaching a similar scale as GPT-2.

Denoting $\mathbf{x} \sim \mathcal{P}_{\mathcal{X}}, \mathbf{x} \in \mathbb{R}^d$ where $\mathcal{P}_{\mathcal{X}}$ is a distribution, a function class $\mathcal{F}$ where for each $f \in \mathcal{F}, f : \mathbb{R}^d \to \mathbb{R}$. Given a sequence $(\mathbf{x}_1, \cdots \mathbf{x}_i)$ $(i > 1)$ sampled from $\mathcal{P}_{\mathcal{X}}$ sequentially, and a sampled function $f \sim \mathcal{F}$, the learning objective aims to correctly predict $f(x_i)$ based on the sequence $(\mathbf{x}_1, f(\mathbf{x}_1), \cdots, \mathbf{x}_{i-1}, f(\mathbf{x}_{i-1}), \mathbf{x}_i)$ with both in-context examples and the query input $\mathbf{x}_i$.

$$\underset{\substack{\mathbf{x}_1 \ldots \mathbf{x}_n \sim \mathcal{P}_{\mathcal{X}} \\ f \sim \mathcal{F}}}{\mathbb{E}} \left[\sum_{i=2}^{n} \mathcal{L}\left(f(\mathbf{x}_i), T_\omega\left([\mathbf{x}_1, f(\mathbf{x}_1) \ldots \mathbf{x}_i]\right)\right) \right] \tag{2}$$

Eq. (2) describes the learning objective, where $\mathcal{L}$ is the loss function. T_ω denotes the transformer model, ω is the parameter of the transformer.

Notably, the model is pre-trained on the above ICL objective instead of the original next-token prediction objective. The function learning framework enables us to: (1) arbitrarily generate data with desired properties from the pre-defined function class $\mathcal{F}$; (2) clearly examine the function-approximation ability and the generalization of skill learning in ICL; and (3) utilize well-developed statistical learning theory to understand ICL.

5.2 Function Approximation and Generalization of ICL

In this subsection, we investigate the function approximation and generalization behavior of ICL. *Function approximation* indicates to what extent transformers can approximate the ground-truth function underlying a given input, in the ICL inference stage. *Generalization*, on the other hand, measures the gap between the approximated function and the ground-truth data generation function. Notably, the function learning framework investigates ICL in the function space, rather than the token space.

To explore the function approximation ability, Raventos et al. (2023) leverages different linear functions to generate pre-training data and in-context demonstrations. When pre-training on a small set of linear functions, ICL acts as a Bayesian optimal estimator, illustrating the skill recognition ability (Raventos et al., 2023). If enlarging the set of pre-training linear functions, ICL can act as an optimal least squares estimator with better function approximation, illustrating the skill learning ability (Raventos et al., 2023). Wu et al. (2023a) provides a theoretical explanation to support the above empirical observations.

Beyond the linear function class, Garg et al. (2022) observes that the ICL is expressive enough to approximate more complicated functions, including sparse linear functions, two-layer neural networks, and decision trees. The only requirement is that the same function class must be encountered during both pre-training and the ICL stage. Bai et al. (2023) and Fu et al. (2023a) propose theoretical explanations with a generalization bound between the prediction error of the transformer model and that of the target function. However, two essential questions remain unsolved: (1) Why do transformers suddenly obtain the skill learning ability with significant performance increase once the number of pre-training data generation functions reaches a certain threshold? (2) Why is the learned data generation function of ICL demonstrations from the same class as the pre-training data generation function?

The *generalization* of ICL is validated by comparing the ground-truth data generation function of in-context demonstrations and the approximated one through ICL inference. A more complicated experimental setting is considered where pre-training involves data generation functions from multiple function classes simultaneously, rather than being restricted to a single function class, as in the above function approximation experiments. Assuming pre-training data generation functions cover decision trees and linear functions, the ground-truth data generation function of ICL demonstrations is a linear function. The ICL generalization is strong if and only if the predicted function of ICL demonstrations is a linear one.

Bai et al. (2023); Ahuja et al. (2023); Vasudeva et al. (2024); Tripuraneni et al. (2023) indicate that transformers can achieve the Bayesian optimal selection, choosing the best-fitting function class with the minimum description length, from those function classes seen during the pre-training stage. Such Bayesian optimal selection helps a transformer pre-trained with multiple function classes reach comparable ICL performance as one pre-trained with only the ground-truth function class. Notably, such Bayesian optimal on the synthetic dataset may not fully explain all the experimental observations. Yadlowsky et al. (2023) generates each pre-training instance with functions from multiple function classes, e.g., $0.7f_1(x) + 0.3f_2(x)$ where f_1 and f_2 are from different function classes. The ICL can still achieve Bayesian optimal selection, holding the same conclusion. Notably, the above works focus on the scenario where the ground-truth data function is within pre-training function classes. Skill learning fails if the ground-truth data function is out of the pre-training function class (Yadlowsky et al., 2023); ICL degrades to skill recognition with Bayesian optimal estimator.

In summary, skill learning emerges if the number of pre-training data generation functions is sufficiently large. ICL can learn a function that lies in the same function class of the pre-training data. Moreover, ICL would implement a Bayesian optimal selection to select the function best-fitting on ICL demonstrations, from pre-training function classes.

5.3 The Internal Mechanisms of ICL

In this subsection, we explore *how ICL can learn an unseen function in context*. Notably, there are two common assumptions generally utilized in existing works: (1) The data generation functions for both pre-training data and in-context demonstrations are linear. (2) The toy transformer model is linearized by removing feed-forward layers and the softmax activation function in the attention layer. This linearized simplification may generalize to the standard transformer, as Ahn et al. (2023b) illustrates that the training dynamic of the linearized version is similar to the standard transformer.

Previous works analogize ICL to meta-learning (Finn et al., 2017). The pre-training stage corresponds to the outer-loop optimization, and the ICL inference stage is an instance of the inner-loop optimization, implementing fast adaptation on new novel tasks. Rather than a real inner gradient update, ICL inference mimics gradient update via a forward process with in-context demonstrations (Hubinger et al., 2019; von Oswald et al., 2023; Zheng et al., 2024).

Based on the dual view that *the backward process on a linear neural layer is equivalent to the forward process on a linear attention layer*, Irie et al. (2022); Dai et al. (2022) proves the mathematical equivalence, illustrating the implicit gradient descent implementation with a linear attention. However, such an analogy is only limited to mathematical equivalence. It remains unclear why ICL can learn a function since such an analogy overlooks many practical details, including the choice of the learning objective, pre-training weights, and the training data distribution (Mahdavi et al., 2024).

To address the gap between theoretical models and real-world implementation, the following works consider the construction of pre-training weights. Von Oswald et al. (2023) first demonstrate that ICL on the single-layer transformer can implement one-step gradient descent with a linear regression objective. Bai et al. (2023) further show that ICL inference can implement ridge regression, least square, lasso, and even gradient descent on a two-layer Neural Network. Nonetheless, those strong assumptions about the attention weights may be not practically reasonable. For instance, Von Oswald et al. (2023) construct the key, query, value matrices W_K, W_Q, W_V with $W_K = W_Q = \begin{pmatrix} I_x & 0 \\ 0 & 0 \end{pmatrix}, W_V = \begin{pmatrix} 0 & 0 \\ W_0 & -I_y \end{pmatrix}$, where I_x and I_y are two different identity matrices and W_0 is the initialized parameters of the transformer model. Nonetheless, it is unclear why a pre-trained transformer would have such type of weights, and it has been reported that this is not easily achieved in practice (Shen et al., 2023).

Instead of explicit attention weight construction, Zhang et al. (2023a); Mahankali et al. (2023); Ahn et al. (2023a) analyze the *converged weights* obtained after pre-training. Von Oswald et al. (2023) observes the ICL on the one-layer linear transformer can implement gradient descent or preconditioned gradient descent algorithm (Ahn et al., 2023a) given a linear regression objective. Given a two-layer transformer, ICL can implement a gradient descent with adaptive step size and special sparsity regularization (Ahn et al., 2023a). Moreover, Ahn et al. (2023a); Von Oswald et al. (2023) reveal that multiple-layered transformers can implement a GD++ algorithm. For larger-scale transformers, Akyürek et al. (2022) empirically illustrates that, instead of performing GD, large-scale transformers show emergent ability directly approximating the closed-form solution of ridge-regression, while there is still a gap on why this ability emerges as the model-scale increases.

Beyond the linear activation for attention heads, recent researches take the softmax activation function into consideration. Von Oswald et al. (2023) demonstrates there exists a transformer that performs GD to solve more complicated nonlinear regression tasks. Li et al. (2023a); Ren and Liu (2023) identify the nonlinear regression task as the softmax regression and contrastive learning objective, respectively. Cheng et al. (2023) further takes non-linear data generation functions into consideration, elucidating a transformer can implement gradient descent and converge to the Bayes optimal predictor. Wibisono and Wang (2023) theoretically finds that the softmax can help to find the correct data pair from the unstructured data which the input-output pair is permuted.. Guo et al. (2023); Zhang et al. (2024) further studies a more challenging but practical setting of representation learning, in which predictions depend on inputs through the MLP. The theoretical evidence in Guo et al. (2023) indicates that the ICL inference can implement ridge regression in context with the input of neural representations.

Practical usage of mechanism analysis. The above section has indicated that ICL implements a gradient descent vector to achieve successful function learning. From a practical perspective, Todd et al. (2023); Liu et al. (2023b) find the existence of compressed task vectors[3] in transformers with specific functionality. More recently, Li et al. (2024) attempts to connect the gradient vector with the compressed task vector, utilizing inner and momentum optimization towards a better task vector. Success of the new optimized task vector can be found on multiple tasks.

6 Insights & Future Directions

In this section, we delve into key insights from the data mechanism perspective of ICL and identify open questions that remain to be addressed in this evolving field.

The uniformity of the two frameworks. Our new data generative perspective suggests the researcher find a suitable statistical framework as the starting point for analysis. We exhibit the potential that both frameworks can be easily utilized to understand the mechanism of both abilities. Such extension enables the future mechanism analysis to select the suitable analysis framework, by referring to their strengths and weaknesses. The original function learning framework for the skill learning ability also implements an implicit Bayesian optimal selection (Ahuja et al., 2023). Moreover, Swaminathan et al. (2023) extends the Bayesian inference framework to learn new in-context data generate functions. A comprehensive discussion can be found in Appendix A.

The unique strengths and weaknesses of skill learning/recognition ability Skill learning effectively updates knowledge from in-context data. However, it may be distracted by irrelevant information (Shi et al., 2023). The skill recognition is robust to in-context noise (Webson and Pavlick, 2021) but less adaptable to new patterns, which leads to the failure on the specification-heavy task (Peng et al., 2023). Therefore, careful evaluation of each ability is recommended to select the most suitable one for specific downstream tasks. A comprehensive discussion can be found in Appendix B.3

Emergent Skill Composition Ability. We majorly focus on the skill recognition/learning ability in our paper. More recently, new skill composition ability is found on larger model with specialized ICL prompts like Chain-of-Thought (CoT) (Wei et al., 2022b). The skill composition ability combines multiple data generation functions to create a more complicated data generation function. This ability, supported theoretically by Arora and Goyal (2023), shows that complex tasks can exhibit performance gains when decomposed skills improve linearly. More analyses on the effectiveness of skill composition ability can be found in Appendix C.

Application of Skills. After acquiring skill learning and skill recognition abilities during pretraining, we examine how the LLM utilizes both abilities to achieve satisfactory performance on downstream tasks during the ICL inference stage. Generally, the LLM's behavior aligns more with the skill recognition mechanism on challenging tasks, while skill learning is more frequently observed on easier tasks. Min et al. (2022) first ob-

[3] Similar task vectors (Hojel et al., 2024) can also be found in the computational vision domain.

serves that the corrupted mapping does not necessarily lead to the overall performance degradation, indicating an overall skill recognition behavior. Instead of examining the overall performance across tasks, Yoo et al. (2022) conducts a more careful evaluation of each task individually where the ICL shows different behaviors on tasks with different difficulties. The relatively easy tasks exhibit performance degradation on the wrong input-label mapping while the robust performance appears on those difficult tasks. Such observation indicates that the skill learning ability is more applicable to relatively easy tasks while the skill recognition ability dominates on the difficult ones. A more detailed discussion can be found in B.2

How the skill learning ability emerges during pre-training. The emergence of the skill learning ability can be partially attributed to the skewed rank-frequency distribution of pre-training corpora. (Chan et al., 2022a), and (Reddy, 2023) highlight the role of the induction head (Olsson et al., 2022), a particular attention head which explicitly searches for a prior occurrence of the current token in-context and copying the suffix as predictions. Moreover, the function class-based analysis (Raventos et al., 2023) illustrates that the transition from skill recognition to skill learning only happens given diverse enough tasks in pre-training corpora. It is interesting to explore how these factors collaboratively influence the emergence of skill learning.

Why does ICL only learn the data generation function that appeared during pre-training? In Section 5, we provide a comprehensive discussion on what function can be learned in context. Observations indicate that ICL can only learn the function within the pre-training data generation function class. Nonetheless, the causality of the pre-training data generation function to ICL remains unclear. Garg et al. (2022) proposes the research question as: *Can we train a model to in-context learn a certain function class* but overlooks the effect of the pre-training data generation function class. Once we have a certain clue about causality, we can leverage the skill-learning ability in a more controllable and safe manner.

Another line of research is to conduct analyses on more realistic scenarios. Recently, Chen et al. (2024) finds the parallel structures in pre-training data-pairs of phrases following similar templates in the same context window is the key to the emergence of the ICL capability. We conjecture that the underlying reason can be the formulation of the induction head with repeat patterns.

Data generation functions aligned with real-world scenarios. One major concern on the statistical framework is that the correspondence with real-world scenarios is unknown and overly simplified. Recently, Akyürek et al. (2024) proposes a new approach for generating data functions that are more aligned with real-world scenarios. The framework allows for more accurate simulations and testing of machine learning models by integrating domain-specific knowledge and constraints into the data generation process. This alignment enhances the applicability and reliability of existing conclusions to the real-world scenarios. We advocate for theoretical analyses focused on real-world data generation functions, moving beyond traditional statistical frameworks. More empirical analysis on skill learning and skill recognition abilities are illustrated in Appendix B.

Extending existing findings to other capabilities of LLMs. more ICL capabilities are observed except for classification tasks, e.g., step-by-step reasoning ability (Wei et al., 2022b) for reasoning and self-correction (Ganguli et al., 2023). A critical question is how we can extend the understanding frameworks introduced in this paper, particularly the data generation perspective, to more complicated LLMs' capabilities. Some pioneering research has been done; Prystawski and Goodman (2023) extends the Bayesian inference framework to understand the effectiveness of the CoT prompt. Kadavath et al. (2022) focuses on the self-evaluation prompt showing that LLMs can accurately examine the correctness of their statements. We believe the introduced data generation perspective and two main understanding frameworks on ICL serve as the milestone to explore more intrinsic capabilities of LLMs.

7 Conclusion

In this study, we introduce a novel data generation perspective to understand the underlying mechanism driving the current success of ICL. We primarily focus on understanding the LLM's ability of skill learning and skill recognition, and investigate whether ICL inference is capable of learning new data generation functions in context. Our work makes a step forward to enhancing our understanding of underlying mechanisms.

8 Limitations

In this paper, we provide a mechanism understanding of the ICL from a data generation perspective, We systematically consider the limitations from various perspectives such as fairness, security, harm to people, and so on, and we do not find any apparent social risk related to our work. However, there is a notable technical limitation in our study. The current statistical frameworks with controlled experimental settings may not fully capture complexities present in real-world scenarios. This gap between the theoretical framework and practical applications suggests that further research is needed to adapt and refine the mechanism analysis to align with real-world application.

References

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

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. 2023a. Transformers learn to implement preconditioned gradient descent for in-context learning. *arXiv preprint arXiv:2306.00297*.

Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. 2023b. Linear attention is (maybe) all you need (to understand transformer optimization). *arXiv preprint arXiv:2310.01082*.

Kabir Ahuja, Madhur Panwar, and Navin Goyal. 2023. In-context learning through the bayesian prism. *arXiv preprint arXiv:2306.04891*.

Kartik Ahuja and David Lopez-Paz. 2023. A closer look at in-context learning under distribution shifts. *arXiv preprint arXiv:2305.16704*.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. 2024. In-context language learning: Arhitectures and algorithms. *arXiv preprint arXiv:2401.12973*.

Pierre Alquier et al. 2024. User-friendly introduction to pac-bayes bounds. *Foundations and Trends® in Machine Learning*, 17(2):174–303.

Sanjeev Arora and Anirudh Goyal. 2023. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. 2023. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*.

Leonard E Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. 2023. Birth of a transformer: A memory viewpoint. *arXiv preprint arXiv:2306.00802*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022a. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–188x91.

Stephanie CY Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K Lampinen, and Felix Hill. 2022b. Transformers generalize differently from information stored in context vs in weights. *arXiv preprint arXiv:2210.05675*.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2024. Parallel structures in pretraining data yield in-context learning. *arXiv preprint arXiv:2402.12530*.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Ifeoluwa Adelani, Pontus Stenetor, Sebastian Riedel, and Mikel Artetx. 2023. Improving language plasticity via pretraining with active forgetting. *arXiv preprint arXiv:2307.01163*.

Xiang Cheng, Yuxin Chen, and Suvrit Sra. 2023. Transformers implement functional gradient descent to learn non-linear functions in context. *arXiv preprint arXiv:2312.06528*.

Noam Chomsky and Marcel P Schützenberger. 1959. The algebraic theory of context-free languages. In *Studies in Logic and the Foundations of Mathematics*, volume 26, pages 118–161. Elsevier.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*.

Antoine Dedieu, Nishad Gothoskar, Scott Swingle, Wolfgang Lehrach, Miguel Lázaro-Gredilla, and Dileep George. 2019. Learning higher-order sequential structure with cloned hmms. *arXiv preprint arXiv:1905.00507*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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

Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, et al. 2024. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. 2023. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3946–3956.

Guhao Feng, Yuntian Gu, Bohang Zhang, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: a theoretical perspective. *arXiv preprint arXiv:2305.15408*.

Jiahai Feng and Jacob Steinhardt. 2023. How do language models bind entities in context? *arXiv preprint arXiv:2310.17191*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Hengyu Fu, Tianyu Guo, Yu Bai, and Song Mei. 2023a. What can a single attention layer learn? a study through the random features lens. *arXiv preprint arXiv:2307.11353*.

Jingwen Fu, Tao Yang, Yuwang Wang, Yan Lu, and Nanning Zheng. 2023b. How does representation impact in-context learning: A exploration on a synthetic task. *arXiv preprint arXiv:2309.06054*.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598.

Dileep George, Rajeev V Rikhye, Nishad Gothoskar, J Swaroop Guntupalli, Antoine Dedieu, and Miguel Lázaro-Gredilla. 2021. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature communications*, 12(1):2392.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Tianyu Guo, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai. 2023. How do transformers learn in-context beyond simple functions? a case study on learning with representations. *arXiv preprint arXiv:2310.10616*.

Michael Hahn and Navin Goyal. 2023. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*.

Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. 2024. Finding visual task vectors. *arXiv preprint arXiv:2404.05729*.

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. 2023. A closer look at the self-verification abilities of large language models in logical reasoning. *arXiv preprint arXiv:2311.07954*.

John J Hopfield. 2007. Hopfield network. *Scholarpedia*, 2(5):1977.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023a. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Qian Huang, Eric Zelikman, Sarah Li Chen, Yuhuai Wu, Gregory Valiant, and Percy Liang. 2023b. Lexinvariant language models. *arXiv preprint arXiv:2305.16349*.

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. 2019. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. 2022. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *International Conference on Machine Learning*, pages 9639–9659. PMLR.

Samy Jelassi, Stéphane d'Ascoli, Carles Domingo-Enrich, Yuhuai Wu, Yuanzhi Li, and François Charton. 2023. Length generalization in arithmetic transformers. *arXiv preprint arXiv:2306.15400*.

Hong Jun Jeon, Jason D Lee, Qi Lei, and Benjamin Van Roy. 2024. An information-theoretic analysis of in-context learning. *arXiv preprint arXiv:2401.15530*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491*.

Yoon Kim, Sam Wiseman, and Alexander M Rush. 2018. A tutorial on deep latent variable models of natural language. *arXiv preprint arXiv:1812.06834*.

Dongfang Li, Zhenyu Liu, Xinshuo Hu, Zetian Sun, Baotian Hu, and Min Zhang. 2024. In-context learning state vector with inner and momentum optimization. *arXiv preprint arXiv:2404.11225*.

Shuai Li, Zhao Song, Yu Xia, Tong Yu, and Tianyi Zhou. 2023a. The closeness of in-context learning and weight shifting for softmax regression. *arXiv preprint arXiv:2304.13276*.

Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak. 2023b. Dissecting chain-of-thought: Compositionality through in-context filtering and learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yuxuan Li and James McClelland. 2023. Representations and computations in transformers that support generalization on structured tasks. *Transactions on Machine Learning Research*.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2022. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Sheng Liu, Lei Xing, and James Zou. 2023b. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.

Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. 2023. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*.

Sadegh Mahdavi, Renjie Liao, and Christos Thrampoulidis. 2024. Revisiting the equivalence of in-context learning and gradient descent: The impact of data distribution. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7410–7414. IEEE.

Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. 2023. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*.

Sewon Min, Xinxi 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.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. 2023. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Jane Pan. 2023. *What In-Context Learning "Learns" In-Context: Disentangling Task Recognition and Task Learning*. Ph.D. thesis, Princeton University.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yunjia Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, et al. 2023. When does in-context learning fall short and why? a study on specification-heavy tasks. *arXiv preprint arXiv:2311.08993*.

Ben Prystawski and Noah D Goodman. 2023. Why think step-by-step? reasoning emerges from the locality of experience. *arXiv preprint arXiv:2304.03843*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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*.

Gautam Reddy. 2023. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. *arXiv preprint arXiv:2312.03002*.

Ruifeng Ren and Yong Liu. 2023. In-context learning with transformer is really equivalent to a contrastive learning pattern. *arXiv preprint arXiv:2310.13220*.

Frieda Rong. 2021. Extrapolating to unnatural language processing with gpt-3's in-context learning: The good, the bad, and the mysterious.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. 2023. Do pretrained transformers really learn in-context by gradient descent? *arXiv preprint arXiv:2310.08540*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Aaditya K Singh, Stephanie CY Chan, Ted Moskovitz, Erin Grant, Andrew M Saxe, and Felix Hill. 2023. The transient nature of emergent in-context learning in transformers. *arXiv preprint arXiv:2311.08360*.

Sivaramakrishnan Swaminathan, Antoine Dedieu, Rajkumar Vasudeva Raju, Murray Shanahan, Miguel Lazaro-Gredilla, and Dileep George. 2023. Schema-learning and rebinding as mechanisms of in-context learning and emergence. *arXiv preprint arXiv:2307.01201*.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. *arXiv preprint arXiv:2305.17256*.

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2023. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*.

Nilesh Tripuraneni, Lyric Doshi, and Steve Yadlowsky. 2023. Can transformers in-context learn task mixtures? In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.

Bhavya Vasudeva, Deqing Fu, Tianyi Zhou, Elliott Kau, Youqi Huang, and Vatsal Sharan. 2024. Simplicity bias of transformers to learn low sensitivity functions. *arXiv preprint arXiv:2403.06925*.

Max Vladymyrov, Johannes von Oswald, Mark Sandler, and Rong Ge. 2024. Linear transformers are versatile in-context learners. *arXiv preprint arXiv:2402.14180*.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Johannes von Oswald, Eyvind Niklasson, Maximilian Schlegel, Seijin Kobayashi, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Max Vladymyrov, Razvan Pascanu, et al. 2023. Uncovering mesa-optimization algorithms in transformers. *arXiv preprint arXiv:2309.05858*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

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

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

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2021. Thinking like transformers. In *International Conference on Machine Learning*, pages 11080–11090. PMLR.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Kevin Christian Wibisono and Yixin Wang. 2023. On the role of unstructured training data in transformers' in-context learning capabilities. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*.

Noam Wies, Yoav Levine, and Amnon Shashua. 2023. The learnability of in-context learning. *arXiv preprint arXiv:2303.07895*.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. 2023a. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023b. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2023. The next chapter: A study of large language models in storytelling. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 323–351.

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

Steve Yadlowsky, Lyric Doshi, and Nilesh Tripuraneni. 2023. Pretraining data mixtures enable narrow model selection capabilities in transformer models. *arXiv preprint arXiv:2311.00871*.

Haotong Yang, Fanxu Meng, Zhouchen Lin, and Muhan Zhang. 2023. Explaining the complex task reasoning of large language models with template-content structure. *arXiv preprint arXiv:2310.05452*.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. 2021. Self-attention networks can process bounded hierarchical languages. *arXiv preprint arXiv:2105.11115*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2022. Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892*.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. *arXiv preprint arXiv:2205.12685*.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. 2023a. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*.

Ruiqi Zhang, Jingfeng Wu, and Peter L Bartlett. 2024. In-context learning of a linear transformer block: Benefits of the mlp component and one-step gd initialization. *arXiv preprint arXiv:2402.14951*.

Shizhuo Dylan Zhang, Curt Tigges, Stella Biderman, Maxim Raginsky, and Talia Ringer. 2023b. Can transformers learn to solve problems recursively? *arXiv preprint arXiv:2305.14699*.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. 2023c. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*.

Jiachen Zhao. 2023. In-context exemplars as clues to retrieving from large associative memory. *arXiv preprint arXiv:2311.03498*.

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

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*.

Chenyu Zheng, Wei Huang, Rongzhen Wang, Guoqiang Wu, Jun Zhu, and Chongxuan Li. 2024. On mesa-optimization in autoregressively trained transformers: Emergence and capability. *arXiv preprint arXiv:2405.16845*.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. 2023. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*.

Wangchunshu Zhou, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. *Advances in Neural Information Processing Systems*, 33:6803–6814.

A Insights on the Bayesian Inference and the Function Learning Framework

The core idea from the data-generative perspective is to (1) construct a data generation function hypothesis with one specific statistical framework and (2) analyze the data generation capability of the LLM with ICL instances with a focus on either skill learning/recognition mechanism. The existing pipelines on skill recognition and skill learning abilities are comprehensively discussed with the statistical frameworks of the Bayesian inference and function learning in Section 4 and 5, respectively. However, most existing analysis follows one-to-one correspondence which explains one ability with one specific statistical framework, serving as a solution for skill learning.

Our new data generative perspective suggests the researcher find a suitable statistical framework as the starting point for analysis. We exhibit the potential that both frameworks can be easily utilized to understand the mechanism of both abilities. Such extension enables the future mechanism analysis to select the suitable analysis framework, by referring to their strengths and weaknesses. The function learning framework provides an elegant description of the data generation process with more comprehensive conclusions. However, it is over-simplified with an unclear relevance to the real-world scenario. The Bayesian inference framework provides a more concrete and detailed description of the data generation process through an HMM model, e.g., the delimiter is taken into consideration, while the theoretical analysis on the role of delimiters is hard since it requires several assumptions over statistical modeling.

We provide a comprehensive discussion on extending one framework to the other statistical framework. The function learning framework can be easily extended to understand skill recognition by simply replacing the data generation function from a mixture of HMMs with linear functions. In this section, we focus on how to utilize the Bayesian inference framework to model the mechanism of skill learning. We first show that the original function learning framework for the skill learning ability also implements an implicit Bayesian optimal selection in Section A.1. We then extend the Bayesian inference framework to learn new in-context data generate functions in Section A.2. the Bayesian inference framework can also serve as a solution for skill learning.

A.1 Bayesian Selection in the Function Learning Framework

The Bayesian perspective can be found in the function learning framework originally utilized for the skill learning mechanism. Typically, we illustrate the underlying Bayesian selection in the function learning framework, indicating the intrinsic connection between the two statistical frameworks. According to Ahuja et al. (2023), the transformers pre-trained on the data generated from diverse function classes exhibit improved function-fitting ability across all the pre-training function classes. To identify the best-fit solution among the whole function class, the function selection process implements a Bayesian optimal selection. More details can be found in Section 5.2. Notably, instead of the original Bayesian inference framework only selecting pre-training data generation functions, the function selection scope is enlarged, including all the unseen functions from the same function class with the pre-training functions.

A.2 Extending the Bayesian Inference Framework for Skill Learning

We then illustrate the possibility of extending the Bayesian inference framework to understand the skill learning mechanism to capture new data generation functions from the in-context data via relaxing the particular assumption. One important assumption in the Bayesian inference framework (Xie et al., 2021) is that all ICL demonstrations should be generated with the same latent concept. Nonetheless, this strong assumption may not be held in practice. For instance, one demonstration sample discusses the topic of sociology but another one is relevant to cardiology, the data generation function for these two domains should be rather different. Inspired by the high compositionality nature of language data, Hahn and Goyal (2023) came up with an information-theoretic bound showing that ICL performance can be improved given more unique compositional structures in pre-training data, therefore skill learning ability can appear by combining compositionality structures, in pre-training data, to infer the data generation function of ICL demonstrations.

Empirical evidence shows that, given an input-label pair of two semantically unrelated concepts, e.g., mapping sports to animals, Rong (2021); Wei et al. (2023) still observe a satisfactory performance with the increasing model scale, indicating that the LLM can retrieve multiple concepts and combine them as a new data generation process. Feng and Steinhardt (2023) interpret the combination with a binding mechanism with an internal function vector to recognize the input feature and bind it to the corresponding label.

Swaminathan et al. (2023) proposes another way to extend the existing Bayesian framework for skill learning via replacing the original HMM model into the clone-structured causal graph (CSCG) (George et al., 2021; Dedieu et al., 2019). The major difference is that the CSCG considers a learnable emission matrix, which determines the probability of observing a particular output given each hidden state in the model. A relevant transition matrix as the concept is retrieved, similar to the Bayesian inference (Xie et al., 2021). The hidden states for each token can then be obtained given the particular relevant template. The LLM then learns the suitable emission matrix, providing the best-fit mapping from the hidden states to the observed token.

B Empirical Investigation On Skill Recognition and Skill Learning

In this section, we exhibit more empirical analyses revolving around skill recognition and skill learning abilities. In contrast to the mechanism analysis that focuses on whether the ICL can learn new in-context data generation functions or not, empirical evidence in this section indicates that it is highly likely that LLMs exhibit both skill recognition and skill learning abilities of various levels, instead of an all-or-nothing conclusion. We first discuss how the LLM jointly obtains both abilities during the pre-training stage in Section B.1. Specifically, the origin of both abilities is determined by the pre-training data distribution (Chan et al., 2022a) and the model scale (Wei et al., 2023; Pan, 2023). We then investigate how the LLM effectively utilizes the obtained abilities during the ICL inference stage in Section B.2. Typically, the LLM exhibits varying degrees of usage on those two abilities according to tasks with different difficulties. The unique strengths and weaknesses of each ability are shown in Section B.3.

B.1 Origin of Skills

In this subsection, we carefully examine how well the LLM obtains the skill learning and the skill recognition abilities during the pre-training stage, with a focus on the impact of the pre-training data distribution and model scale. Roughly speaking, the skill recognition ability is easy to achieve while the skill learning ability develops much slower and only emerges when the model scale is sufficiently large.

Analyses are first conducted focusing on how those abilities are developed along the pre-training procedure. (Bietti et al., 2023) observe that the skill recognition ability is obtained early in the pre-training procedure, while the skill learning ability is developed much later. However, Singh et al. (2023) shows that the obtained skill learning ability gradually vanishes after over-training and is replaced by the skill recognition ability. Such observation indicates that skill learning is a transient ability that may disappear when the model is over-trained rather than a persistent one which can be kept once obtained. The reason can be attributed to the pre-training data distribution (Chan et al., 2022a) where the task learning ability degrades if the pre-training data follows a uniform, i.i.d distribution. Nonetheless, such degradation may not happen when the pre-training data follows a properly skewed Zipfian distribution. Chan et al. (2022a) further emphasizes that the skill learning ability emerges when the pre-training data meets the following properties: (1) Skewed rank-frequency distributions: Dynamic contextual meaning does not uniform across data, instead, only a few meanings dominate with the long tail of other infrequent meanings. (2) Burstiness: Dynamic contextual meaning is not uniform across time, but appears in clusters. The reason why ICL ability can be obtained on such data distribution remains unclear. A potential explanation could be that the pre-training weight can only obtain the head meaning frequently appears while the long tail knowledge can only be obtained via ICL.

Analyses are then conducted with a focus on the impact of the model scale. Pan (2023) illustrates that the skill recognition ability can be found across LLMs with different scales. In contrast, LLMs obtain better skill learning ability along with an increasingly larger scale. Similar observations can be found in (Wei et al., 2023) that the LLM can learn the flipped input-label mapping and override pre-training knowledge when the model scale is sufficiently large. (Fu et al., 2023b) provides the potential explanation where the good skill recognition ability serves as a necessity for developing the skill learning ability.

B.2 Application of Skills

After the LLM obtained the skill learning and skill recognition abilities during pre-training, we then investigate how the model utilizes both abilities for achieving satisfactory downstream task performance during the ICL inference stage. Overall, the behavior of the LLM is more consistent with the skill recognition mechanism on difficult tasks while observations aligned with skill learning are more common to see on easy tasks.

Empirical analyses are conducted on the well-trained LLM, focusing on the ICL behavior on downstream tasks with various difficulties. Typically, we examine whether the model behavior aligns with the skill recognition ability or the skill learning one via the performance sensitivity on corrupting in-context data with incorrect input-label mapping. If the LLM takes advantage of the skill learning ability more, the LLM can learn the corrupted in-context mapping, leading to performance degradation compared with the origin setting. In contrast, if the LLM follows the skill recognition ability more, the LLM should be robust to the correctness of the input-label mapping, since the skill recognition ability only implements the pre-training data generation function with correct input-label mapping. Min et al. (2022) first observes that the corrupted mapping does not necessarily lead to the overall performance degradation, indicating an overall skill recognition behavior. Instead of examining the overall performance across tasks, Yoo et al. (2022) conducts a more careful evaluation of each task individually where the ICL shows different behaviors on tasks with different difficulties. The relatively easy tasks exhibit performance degradation on the wrong input-label mapping while the robust performance appears on those difficult tasks. Such observation indicates that the skill learning ability is more applicable to relatively easy tasks while the skill recognition ability dominates on the difficult ones.

B.3 Advantages and Disadvantages of Skills

Considering the intricate interplay of both abilities on different tasks, we further illustrate the strengths and weaknesses inherent in each ability. Skill learning ability can obtain new knowledge from the in-context data, and even over-ride the pre-training knowledge. It provides an easy way to update the knowledge on the specific application without requiring computationally heavy fine-tuning. Such ability has been successfully utilized in different LLM applications, e.g. model editing with ICL (Zheng et al., 2023). Nonetheless, the skill learning ability may fail as it can be easily distracted by irrelevant context (Shi et al., 2023). Skill recognition ability is insensitive to the new in-context pattern leading to the failure on the specification-heavy task (Peng et al., 2023) while it exhibits robustness to the incorrectness of label-demonstrations and other in-context noise (Webson and Pavlick, 2021). Based on the above discussion, we suggest a careful evaluation of LLM about each ability and select a desired one for the downstream task.

C Skill Composition

We primarily focus on the skill learning ability where the ICL can learn a new data generation function, and skill recognition ability where the ICL utilizes the data generation function from pre-training data. Instead of focusing on the single data generation function, combining multiple data generation functions together can lead to a complicated data generation function. We named such capability as skill composition capability, helping the LLM to achieve a complicated task by combining a sequence of simple and basic steps. Arora and Goyal (2023) theoretically indicates the effectiveness of skill composition where the complicated task can exhibit emergent performance gain when all the decomposed basic skills improve linearly.

The discussions on skill composition are organized as follows. In Section C.1, we investigate the effectiveness of skill composition ability. In Section C.2, we analyze when the skill composition capability can work. In Section C.3, we further illustrate more discussion and real-world applications on the skill composition ability. Notably, the skill composition ability is complicated without a general data generation function framework so far. The skill-composition ability often requires to be elicited by specific-designed ICL prompts, e.g., Chain-of-Thought prompting (CoT) (Wei et al., 2022b), Tree-of-thought (Yao et al., 2023), and Graph-of-Thought (Besta et al., 2023), which generates multiple intermediate steps before the final answer. Most following literature conducts analysis on the CoT prompt.

C.1 Effectiveness of Skill Composition

In this section, we investigate the effectiveness of skill composition ability. Feng et al. (2023) indicates that if the skill decomposition is applied, the LLM can be more expressive to describe more complicated problems, e.g., mathematical and decision-making problems. Li et al. (2023b); Yang et al. (2023) further demonstrate the data efficiency where the skill composition facilitates can learn complicated functions with a reduced sample complexity. Prystawski and Goodman (2023) attributes the above expressiveness and efficiency with the local structures in the training data generation function. Such locality enables to accurate inference on each intermediate step supported by the similar pre-training data generation function. In contrast, direct inference as a whole instead of each local steps are likely to fail requiring since such complicated data generation function does not appear during the pre-training stage. In summary, the skill composition ability of LLMs enhances their expressiveness and data efficiency for modeling complicated data generation function, building on the basis of locality data generation function from the pre-training data.

C.2 When Skill Composition Works

We demonstrate the effectiveness of the composition in Section C.1, however, it remains unknown whether the decomposed intermediate steps are well-organized aligning with human cognition. To examine the correctness of the LLM decomposition, the literature focuses on formal deductive reasoning tasks like math reasoning (Ahn et al., 2024). It enables to conducting systematic and controllable analysis on each reasoning step with the unique correct answer.

LLMs are able to conduct correct decomposition on particular tasks, aligning with the ideal human reasoning process. Zhou et al. (2023) finds a theoretical criterion to identify when the LLM can implement the ideal decomposition. Typically, when the task can be described by a short RASP program (Weiss et al., 2021), a programming language designed for the computational model of a Transformer, the LLM can achieve the correct decomposition. Similarly, Yao et al. (2021) demonstrates that the transformer can process correct decomposition on particular formal languages with hierarchical structure, e.g., Dyck$_k$ (Chomsky and Schützenberger, 1959). With a suitable decomposition, LLMs can easily solve arbitrary complicated problems (Jelassi et al., 2023; Li and McClelland, 2023).

Beyond those identified tasks, it remains many tasks where LLMs cannot conduct an ideal decomposition. The key underlying reason (McCoy et al., 2023) is the gap between human cognition and the next-token prediction pre-training task, requiring to tackle problems sequentially greedily. Instead of a proper decomposition, a greedy shortcut can be obtained from standard training, which skips the particular step instead of a formal decomposition. Theoretical evidence on the existence of shortcuts can be found in (Liu et al., 2022) on the semi-automaton reasoning task. Saparov and He (2022) indicates that the shortcut can easily select the wrong step, leading to an incomplete planning and subsequently an incorrect answer, leading to failure on complicated tasks (Dziri et al., 2023). Such inherent failure is unavoidable as the transformer always finds a shortcut solution (Liu et al., 2022) while impossibile to find the exact implementation of the semi-automaton reasoning requiring recurrent models of computation with shallow and non-recurrent architecture. On the contrary, the shortcut also shows its benefits, converting the original complicated reasoning problem with multiple hops into a simpler one with less hops (Wu et al., 2023b; Saparov and He, 2022), alleviating the performance degradation along with the increased hop.

In summary, the shortcut solution of LLMs can be a double-side sword to solve a compositional problem. Nonetheless, it remains no existing study on how the LLM acquires the decomposition capability from pre-training data. Notably, we focus on whether the LLM composition aligns with the human decomposition while the manually-conducted deduction rules may not be optimal. The optimal decomposition remains unknown.

C.3 More Discussions

Despite the above comprehensive understanding, there are more empirical studies on the skill composition ability from various perspectives as follows. Madaan and Yazdanbakhsh (2022) divides the CoT prompt into three key components: symbols, patterns, and text with distinct roles as follows: (1) The exact type of symbols does not matter. (2) The patterns are the template serving as a trigger helping to locate the correct concept (3) Text contains commonsense knowledge and meaning, leading to the ultimate success. Similarly, Wang et al. (2022) divides the CoT prompt into two key components: bridging objects (the key and necessary objects) and language templates. Interestingly, neither of them matters. In contrast, the relevance to the query and correct reasoning ordering matters.

More recently, Xu et al. (2024) challenges the skill compositional capability of LLMs, pointing out the failure on the sequential reasoning tasks. On the contrary, LLMs can perform well on simple composite tasks that can be easily separated into sub-tasks based on the inputs solely. The skill composition ability remains mysterious, requiring further analyses.

D Discussions

D.1 The Emergence Phenomenon On the ICL Generalization

Chan et al. (2022b) proposes an interesting perspective to characterize how the ICL generalizes to the test data based on the in-context samples. Observations exhibit that the larger LLMs can achieve rule-based generalization similarly with the SVM. The rule-based generalization makes decisions using a minimal set that is central to the category definition, disregarding less essential data, Nonetheless, induction heads mechanism with prefix match and copy are more aligned with examplar-based generalization like KNN. The reason why LLM can achieve rule-based generalization still remains unclear.

D.2 Advantages And Disadvantages of Skill Learning And Skill Recognition

Skill learning mechanism can obtain new knowledge from the in-context pattern, and even over-ride the pre-training knowledge. It provides an easy way to update the knowledge on the specific application without requiring computational-heavy fine-tuning. Such ability has been successfully utilized in different LLM applications, e.g. model editing with ICL (Zheng et al., 2023). Nonetheless, the skill learning mechanism may fail as it can be easily distracted by irrelevant context (Shi et al., 2023). The failure reason found in (Tang et al., 2023) is that the input-label mapping is more to be the shortcut as the model scale increases. Skill recognition mechanism is insensitive to the new in-context pattern leading to the failure on the specification-heavy task (Peng et al., 2023) while it exhibits robustness to the incorrectness of label-demonstrations and other in-context noise (Webson and Pavlick, 2021). For instance, the skill recognition mechanism can perform well in a noisy setting as it can only locate the origin ability developed during the training procedure. The LLM cannot learn the new in-context information with noisy labels. Instead, it only helps to locate the most similar concept seen during the pre-training stage. Despite the labels being noisy, ICL may still be able to locate the correct concept with the input text information. Empirical evidences (Min et al., 2022) indicates that even random permute the model label can lead to a satisfying performance.

D.3 Abstraction Ability of LLMs

Despite the success of LLM based in the natural language, (Webb et al., 2023; Mirchandani et al., 2023; Huang et al., 2023b; Chen et al., 2023) indicate the effectiveness on abstract symbol without knowing semantic meanings of any individual symbol. Webb et al. (2023) exhibits the emergence ability of LLM for abstract pattern induction while (Mirchandani et al., 2023) suggest that LLM is a general pattern machine extrapolating sequences of numbers that represent states over time to complete simple motions. Huang et al. (2023b) achieves comparable performance using random Gaussian vectors instead of the original token embedding when context is sufficient. Chen et al. (2023) indicates such abstraction with randomizing embeddings can help LLM learn multiple languages.

D.4 Discussion On the Self-correction

The self-correction (Pan et al., 2023; Kim et al., 2023; Gou et al., 2023; Welleck et al., 2022) is an advanced ICL technique iteratively revise the outputs of LLM utilizing feedbacks, aiming to mitigate undesired and inconsistent behaviors, e.g., lexically constrained generation and toxic reduction. Despite its effectiveness, the underlying mechanism remains an open question. The initial observations can be found as follows. Kadavath et al. (2022) illustrates positive evidence where LLM can accurately examine the correctness of their statements, serving as the necessary condition for self-correction. Nonetheless, Huang et al. (2023a) observes that self-correction cannot improve the performance since the added feedback may bias the model away from producing an optimal response to the initial prompt. Hong et al. (2023) provides more detailed evaluation setting and identifies that (1) LLMs perform much worse at identifying fallacies related to logical structure than those related to content. (2) LLMs cannot classify different types of fallacies. Despite the above phenomenons, there is still no understanding of the underlying mechanism of self-correction so far.

D.5 How The Data-generating Functions Are Different Than Arbitrary Functions

We first emphasize the importance of the data generation function. The strong generative capability is an essential ability for LLMs. Most successful applications and usage of the LLM revolve around the generative capability. Therefore, the data generation perspective is essential to understand the LLM.

The data-generating function is generally utilized to understand the data-generation capability of LLMs. It can be defined as 'the underlying hypothesis on textual data generation'. Technically, the data-generation function can be any function that can model the probability over a potential token given a sequence of tokens, after being trained with text data. The main difference between the data-generation function and arbitrary function is whether the function can be used to generate reasonable natural language sequences. Understanding the data-generation process is a core problem in natural language processing, particularly for natural language generation tasks.

More concretely, N-gram, HMM, and Recurrent Neural Networks are three straightforward data-generation functions but they cannot model long contexts, and the first two are non-parameterized data-generation functions. On the other hand, we can have a linguistic-driven data generation function, e.g. probabilistic context-free grammar (Hahn and Goyal, 2023), to introduce some priors of syntax. Since the complicated and hierarchical nature of human languages, LLMs are great in terms of incorporating contextual information through a powerful function approximation ability. Honestly speaking, we can claim that the impressive results of LLMs depend on the ability to approximate the unknown data-generation function underlying the pre-training corpora.

Notably, the statistical framework, which utilized the input-label mapping as the data generate function is a simplified setting. Such a simplified setting enables to conduct of more theoretical analysis. Therefore, we can qualitatively analyze the expressiveness, generalization, and internal mechanisms of the ICL. For instance, with the function abstraction, we can analyze the generalization within the same function class and between different function classes. However, how to take advantage of it in a real-world scenario remains unclear.

D.6 Whether Different Demonstrations Represent Different Data Generation Functions

Whether different demos represent different data generation functions depends on the hypothesis of the data generative function. It is possible for different demonstrations to share the same data generation function. On the contrary, it is also possible for different orders of the demonstrations to correspond to different data generation functions.

D.7 Whether there is the connection between skill learning/recognition and model under/overfitting?

The ICL procedure does not have any backward learning process, i.e. gradient descent, generally utilized in deep learning. Therefore, the ICL procedure is not explicitly related to the model under-/overfitting without an explicit fitting procedure.

Both skill learning and skill recognition can achieve a certain generalization, without explicit under-fitting or over-fitting. The skill recognition is not directly memorization. Given the train data $(\mathbf{x}, \mathbf{y})$ generated from the function $\mathbf{y} = \mathbf{k}\mathbf{x}$, the pre-training data can be within the input interval $\mathbf{x} \in [0, 1]$, while the ICL test data can be within the input interval $\mathbf{x} \in [1, 2]$. In such a case, the ICL can still achieve satisfying performance, indicating the generalization ability. It indicates the ICL with skill recognition can achieve generalization when test data are within the same function. A more comprehensive discussion when meeting out-of-distribution scenarios can be found in Appendix E.

The difference between skill learning and recognition is the different extent of the generalization. The skill recognition generalizes through seeking an existing function within the same function class but skill learning can come up with a new function within this function class.

D.8 The real-world correspondence of data generation functions

Our paper focuses on whether the ICL can learn a new data generation function in context. From a practice scenario, the new data generation function can be defined as the n-gram does not appear in the training stage. Such compositional generalization is a key concept in the NLP domain. For instance, such out-of-distribution can happen when LLMs read the news. The skill learning mechanism can learn the new n-gram and knowledge in context, while skill recognition tries to map the pre-training knowledge with the news.

E The Robustness of ICL On the Statistical Framework

We primarily analyze the skill-learning mechanism when (1) data generation functions during the pre-training and ICL inference stages are from the same function class, and (2) input features are sampled from the same distribution in Section 5. In this section, we provide a further discussion of how the skill-learning mechanism works when distribution shifts happen, indicating the robustness of the ICL. The robustness of the ICL is evaluated in different out-of-distribution scenarios, which can be roughly divided into the following categories: (1) Task shift, where the pre-training and in-context labels are generated from different function classes, is discussed in Appendix E.2. (2) Corvariate shift, where the pre-training and in-context inputs are sampled from different distributions, is discussed in Appendix E.3. (3) Query shift, where the in-context training inputs and the query sample input are sampled from different distributions, is discussed in Appendix E.4. Notably, all the above out-of-distribution scenarios are conducted on the statistical framework while it remains an unclear correspondence to the real-world LLM system pre-training on the massive corpus. More recently, Vladymyrov et al. (2024) focuses on the corrupted training data scenario with noises on different extend. Both empirical and theoretical results indicate the robustness of transformers in such scenario.

E.1 Preliminary

To formally describe different out-of-distribution scenarios, we first provide a rigorous description of the pre-training and prompt data from a distribution perspective. The pre-training data is defined as $(\mathbf{x}_1, \mathbf{h}(\mathbf{x}_1), \cdots, \mathbf{x}_N, \mathbf{h}(\mathbf{x}_N), \mathbf{x}_{\text{query}})$ where $\mathbf{x}_i \sim \mathcal{D}^{\text{train}}_{\mathbf{x}}$, $\mathbf{x}_{\text{query}} \sim \mathcal{D}^{\text{train}}_{\mathbf{x}}$ and $\mathbf{h} \sim \mathcal{D}^{\text{train}}_{\mathcal{H}}$. The test prompt is defined similarly but drawing from a different distribution where $\mathbf{x}_i \sim \mathcal{D}^{\text{test}}_{\mathbf{x}}$ and $\mathbf{x}_{\text{query}} \sim \mathcal{D}^{\text{test}}_{\mathbf{x}}$. We then describe different out-of-distribution scenarios and how the LLM behaves on them differently in the following sections.

E.2 Task Shift

Task shift (Zhang et al., 2023a) is a concept shift which be formally defined as $\mathcal{D}^{\text{train}}_{\mathcal{H}} \neq \mathcal{D}^{\text{test}}_{\mathcal{H}}$. It describes that the pre-training and in-context labels are generated from different function groups. Existing literature demonstrates two different task shifts, i.e., noise shift (Zhang et al., 2023a), and regression vector shift (Raventos et al., 2023).

Noise shift (Zhang et al., 2023a) corresponds to the scenario where the shift is induced by the random Gaussian noise. Typically, the pre-training data generation function is $\mathbf{y} = \langle \mathbf{w}, \mathbf{x} \rangle$ where in-context data generation function is from noisy linear function $\mathbf{y}_i = \langle \mathbf{w}, \mathbf{x} \rangle + \epsilon$. Zhang et al. (2023a) observes satisfying performance under such shift, indicating the robustness under such Gaussian noise.

Regression vector shift (Raventos et al., 2023) corresponds to the scenario where pre-training data generation functions are a limited group $\mathcal{F}_{\text{train}}$ of linear functions $\mathbf{f_i} : \mathbf{y} = \langle \mathbf{w}_i, \mathbf{x} \rangle + \mathbf{b}_i$, where $\mathbf{f_i} \in \mathcal{F}_{\text{train}}$ The in-context data generation function is from all the possible linear functions covering the entire function space $\mathbf{f}_i \in \mathcal{F}_{\text{context}}$, where $\mathcal{F}_{\text{train}} \subseteq \mathcal{F}_{\text{context}}$. The task shift appears on the unseen data generation function during training. Raventos et al. (2023) observes that ICL exhibits the generalization gap with insufficient pre-training data. The emergence happens when the number of pre-training functions increases with satisfying out-of-distribution performance.

E.3 Covariate Shift

Covariate shift (Zhang et al., 2023a) can be formally defined as $\mathcal{D}_{\mathbf{x}}^{\text{train}} \neq \mathcal{D}_{\mathbf{x}}^{\text{test}}$. It describes that the pre-training inputs and the in-context inputs are sampled from different distributions. Existing literature demonstrates different covariate shifts including low-dimensional subspace shift, skewed covariance shift, mean shift, and random covariate shift.

Low-dimensional subspace shift (Garg et al., 2022) samples prompt input feature from random 10-dimensional subspace from the pre-training input feature. Garg et al. (2022) empirically observes the robustness over such covariate shift.

Skewed covariance shift (Garg et al., 2022) samples in-context features from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma}$ is a skewed covariance matrix with eigen-basis chosen uniformly at random and i^{th} eigenvalue proportional to $1/i^2$. Empirically observations (Garg et al., 2022) indicate the performance degradation when the input feature dimension is larger than 10.

Mean shift (Ahuja and Lopez-Paz, 2023) samples train and test inputs from $\mathcal{N}(\mu_{\text{train}}, \mathbf{\Sigma})$ and $\mathcal{N}(\mu_{\text{test}}, \mathbf{\Sigma})$ where $\mathcal{N}(\mu_{\text{train}} \neq \mathcal{N}(\mu_{\text{test}})$. Despite performance degradation to a certain extend, the transformer backbone shows better generalization than the MLP backbone with both empirical observations and theoretical evidence.

Random covariate shift (Zhang et al., 2023a) corresponds to that pre-training training prompts and in-context prompts are sampled from distributions with different covariates. The ICL performance degradation (Von Oswald et al., 2023; Zhang et al., 2023c) drops to 0 quickly with theoretical explanation (Zhang et al., 2023c). The larger transformer with non-linearity serves as the solution to random covariate shift, while the reason underlying the emergent ability remains unclear.

E.4 Query Shift

Query shift (Zhang et al., 2023a) is the covariate shift, which can be formally defined as $\mathcal{D}_{\text{query}}^{\text{test}} \neq \mathcal{D}_{\mathbf{x}}^{\text{test}}$. It describes the distribution shift within the in-context training samples and test samples are sampled from different distributions. Different from the task shift focusing on the distribution shift between pre-training data and prompt data, query shifts describe the distribution shift within the prompt data, where the training prompt data distribution is different from the prompt query distribution. Existing literature demonstrates two different query shifts as follows.

The orthants shift changes the positive or negative signs to each coordinate of in-context features, ensuring both prompt data and prompt query fall within the same orthant, distinct from the query input's orthant. Garg et al. (2022) observes the robustness to this shift when differences between orthants are not large.

The orthogonal shift maps the the prompt query to the orthogonal space of prompt data, which is an extreme case of the formal one. Garg et al. (2022) shows empirical evidence where the prediction will be zero and the error will be significantly large. Zhang et al. (2023c) further theoretically underpins the underlying reason while no solution is found currently.