ICL CIPHERS: Quantifying "Learning" in In-Context Learning via Substitution Ciphers

Anonymous ACL submission

Abstract

Recent works have suggested that In-Context Learning (ICL) operates in dual modes, i.e. task retrieval (remember learned patterns from pre-training) and task learning (inference-time "learning" from demonstrations). However, disentangling these the two modes remains a challenging goal.

We introduce ICL CIPHERS, a class of task reformulations based on *substitution ciphers* borrowed from classic cryptography. In this approach, a subset of tokens in the in-context inputs are substituted with random tokens, rendering English sentences less comprehensible to human eye. However, by design, *there is a latent, fixed pattern to this substitution, making it reversible*. This ensures that the task remains a well-defined task in some abstract sense, despite the transformations. It is a curious question if LLMs are capable of solving ICL CIPHERS, which essentially would require deciphering the latent cipher, providing a novel approach to quantify "learning" in ICL.

Evaluations on four datasets and four models shows LLMs' ability to solve ICL CIPHERS is consistently above our random baseline, however, this gap remains small across the board (e.g., datasets, models, scale). Additionally, using a logit lens to examine LLMs' internal representations reveals promising, albeit weak, trends in their ability to decode the ciphered inputs.

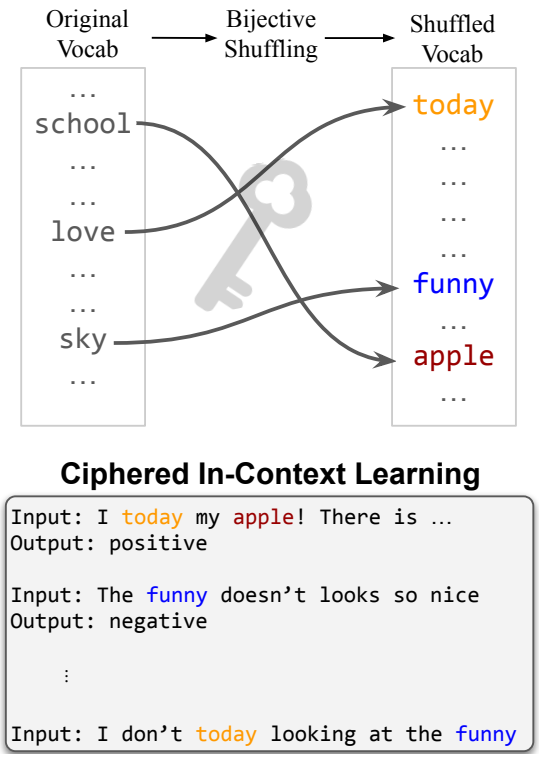

Figure 1: An example of ICL CIPHERS, a cryptographic task reformulations framework where a subset of tokens are ciphered (replaced with other words in the lexicon) via a bijective map (e.g., each instance of "school" replaced with "apple".) Since this cipher is a bijection, one can recover the original format of the ICL instance, ensuring the well-defined task upon the transformations.

1 Introduction

In-Context Learning (ICL) is an emergent behavior in Large Language Models (LLMs) that allows them to identify patterns in demonstrations given as prompts and apply these patterns to similar tasks (Brown et al., 2020). This intriguing real-time learning ability has spurred numerous studies to better understand its dynamics. Despite recent efforts (Xie et al., 2021; Min et al., 2022; Srivastava et al., 2023; Shin et al., 2022; Razeghi et al., 2022; Von Oswald et al., 2023; Shen et al., 2024), the literature's understanding of the functional aspects of ICL remains elusive and contentious.

Most pertinent to our study, Pan et al. (2023); Lin and Lee (2024); Wang et al. (2024) propose ICL's dual behavior: ***task retrieval*** (TR), which involves recalling a previously encountered task from pre-training data through its demonstrations, and ***task learning*** (TL), which refers to the ability to grasp new input-label mappings that were *not* seen during pre-training. Although these two mechanisms

are not necessarily separate in practice, examining them independently may help researchers better understand their strengths and limitations. Specifically, Pan et al. (2023) measure TL by assessing task performance when labels are substituted with abstract symbols (such as numbers or letters) that have never co-occurred with the inputs during pre-training. However, it remains unclear whether this label mapping is sufficient to ensure pure TL (i.e., no partial influence of TR). It is conceivable that LLMs could still use the human-readable inputs and prompt structure to deduce the task, thereby performing implicit task retrieval. This consideration motivates the exploration of alternative approaches for quantifying task learning.

In this study, we introduce ICL CIPHERS, a cryptographic framework that uses substitution ciphers to encode task inputs. For example, in a sentiment classification task where sentences are assigned to target classes, we apply bijective shuffling to the LLM's original vocabulary, ensuring a one-to-one correspondence between tokens in the shuffled and original vocabularies. This random bijective shuffling is done before the experiments and remains constant throughout. We then replace tokens in the input text with their corresponding tokens based on this mapping (e.g., every instance of "love" is replaced with "today"), as shown in Figure 1.

The outcome of substitution ciphers is generally not easily interpretable by humans (see Fig.1 for examples), resembling a random shuffling of words. However, since ICL ciphers are *reversible*, the original tasks can be reconstructed from the encoded version, ensuring that the task, although not easily understood by human eyes, still represents a valid task. This lack of interpretability is a design feature (rather than a flaw) here as it greatly reduces the likelihood that our prompts have been encountered in the pre-training data. As a results, our working hypothesis is that any gains above the random (non-bijective shuffles should be indicative of TL (as opposed to TR) within ICL.

We evaluate ICL ciphers using three pre-trained models across four well-known benchmarks. Our empirical results demonstrate that ICL achieves better-than-random performance on ciphered tasks (§5.1). For example, on the bijective ciphered HellaSwag, Llama3.1 (8B) averages 3.6% higher accuracy than random shuffling, across various demonstration counts (Table 2). This suggests that LLMs can learn and decode these random bijections, enabling them to solve ICL Ciphers. Furthermore, we provide additional results with the number of demonstrations, model scale, shuffling rate. Finally, we perform an interpretability analysis (§5.2) which reveals promising, albeit weak, trends in their ability to decode the ciphered inputs. .

Unlike previous work by (Pan et al., 2023; Wang et al., 2024) that intervenes in task outputs through label shuffling, our approach modifies task inputs. This creates instances less likely to have been encountered in pre-training data, offering an alternative TL indicator by necessitating the LLM to decode ciphers as part of task solving. These perspectives can be seen as complementary, each assessing different aspects of "learning" in ICL.

In conclusion, we propose an alternative method for quantifying "learning" in ICL through the use of substitution ciphers. We establish a framework for evaluating the performance of ICL within this experimental context. Our findings demonstrate evidence of task learning in ICL, both in terms of downstream performance and through probing analysis. We hope these insights inspire future research aimed at gaining a deeper understanding of the emergence of ICL in LLMs.

2 Related Work

Dual operating modes of ICL: Previous research has identified two primary modes of operation for In-Context Learning (ICL): *task retrieval* (TR), which involves recalling patterns previously encountered in pre-training data, and *task learning* (TL), which involves learning new patterns on-the-fly that were not seen during pre-training. Some studies emphasize TR by exploring the factual recall capabilities of ICL (Sun et al., 2022; Golchin et al., 2024; Han et al., 2023; Zhao, 2023; Reddy, 2023; Dankers and Titov, 2024), providing insights into how large language models (LLMs) memorize pre-training data, thus facilitating TR. Other studies (Lin and Lee, 2024; Song et al., 2024; Nafar et al., 2024; Anand et al., 2024) focus on simplified datasets (e.g., linear regression) or architectures (e.g., shallow transformers), which differ from our focus. Additionally, Pan et al. (2023); Wang et al. (2024) have attempted to separate TR and TL through *output* intervention by replacing labels with abstract symbols like numbers or letters. However, it remains uncertain whether using abstract labels effectively eliminates the influence of TR in ICL. Many human-readable tasks may have inherent priors embedded in the pre-training

datasets, suggesting that LLMs might still use inputs and prompt structures to infer the task, thereby engaging in implicit task retrieval. Our approach proposes an alternative method for quantifying TL by intervening in the *input* space.

Ciphers and their use in AI: The problem of deciphering substitution ciphers is studied in NLP as it may provide automatic ways to decipher lost languages without any parallel corpus (Knight et al., 2006; Ravi and Knight, 2008, 2011; Dou and Knight, 2012; Berg-Kirkpatrick et al., 2013; Pourdamghani and Knight, 2017; Nuhn et al., 2013; Berg-Kirkpatrick and Klein, 2011; Corlett and Penn, 2010; Aldarrab and May, 2020, *inter alia*). For instance, Ravi and Knight (2011) introduces a Bayesian approach for deciphering substitution ciphers, combining information from letter n-gram language models and word dictionaries to perform efficient sampling-based inference for decipherment results. We also note various optimization-based and heuristic-based computational frameworks that are deterministic in nature for deciphiring substitution ciphers (Peleg and Rosenfeld, 1979; Ganesan and Sherman, 1993; Olson, 2007).

We also note the work of Yuan et al. (2023) which is the only work (that we know of) applying ciphers on LLMs (GPT-4, specifically) in the context of safety problems, which is a different focus than ours.

Alternative explanations of ICL: Since the discovery of ICL (Brown et al., 2020), numerous studies have explored it across various contexts (Zhao et al., 2021; Min et al., 2022; Mishra et al., 2022; Han et al., 2023; Wang et al., 2023; Sia et al., 2024; Vacareanu et al., 2024; Mueller et al., 2024). For example, Perez et al. (2021); Lu et al. (2022); Mishra et al. (2022) demonstrated ICL's sensitivity to the selection and sequence of demonstrations, while Shin et al. (2022); Razeghi et al. (2022) highlighted its sensitivity to the frequency and size of the relevant pre-training corpus. Another research direction seeks to elucidate the mechanisms behind ICL. Xie et al. (2021) described ICL as implicit Bayesian inference, where ICL demonstrations are mapped to a latent concept (task) learned during pre-training. Other works have attempted to explain ICL as a form of implicit optimization (gradient descent and its variants) (Garg et al., 2022; Zhang et al., 2023; Dai et al., 2022; Akyürek et al., 2022; Von Oswald et al., 2023; Li et al., 2023), though the applicability of these formalisms to real LLMs is debated (Shen et al., 2024). A few studies aim to understand how ICL emerges in LLMs. Hahn and Goyal (2023) suggested that the compositional structure of natural language leads to emergent in-context learning, while other works (Chan et al., 2022) propose that certain distributional properties in the pre-training data may give rise to ICL. Although many of these studies explain certain aspects of ICL, they fall short in others. The precise origins of ICL in LLMs remain an active area of research.

3 Problem Setup

In-Context Learning: Let f_θ denote a pre-trained language model parameterized by θ. This model performs ICL by conditioning on an ordered set of n-many input-output pairs $D_{\text{demo}} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$. To measure this model's competence, we evaluate it on a collection of input-output pairs $D_{\text{test}} = \{(x_i, y_i)\}$. Specifically, for instance $(x_{\text{test}}, y_{\text{test}}) \sim D_{\text{test}}$, from an LM conditioned on the demonstrations with an appropriate encoding: $y_{\text{pred}} \sim f_\theta(D_{\text{demo}}, x_{\text{test}})$ we extract a predicted label y_{pred} which is then compared against the gold label y_{gold}.

Substitution cipher: A simple substitution cipher is a technique for encoding messages in which each letter in the plaintext is substituted with a different letter from the alphabet, usually according to a predetermined rule or key. Formally, we can define a cipher function $c : V \to V$ that maps each [sub]word in the lexicon $V = \{t_j\}_{j=1}^{|V|}$ to another [sub]word. In the special case where this cipher function is the identity function (i.e., $c(t_j) = t_j$ for all j), no changes occur in the text. However, if for certain letters or [sub]words the function c maps them to different counterparts, the plaintext is transformed into a ciphertext that conceals the original message. We define *shuffle rate* $r \in [0, 1]$ as the proportion of [sub]words in the lexicon that are mapped to a different [sub]word. If the lexicon has 100 [sub]words, a shuffle rate of 0.1 will randomly select 10% of these [sub]words $S \subseteq V$ and create a substitution key (c) for this subset. Next, we describe how this key is generated.

BIJECTION cipher: We induce the substitution cipher in the inputs $(x_1, \ldots, x_n)$ of ICL demos. Given a set S of tokens (sub-words) to be shuffled, we create a bijection mapping between two permuted orders of S. For example, say the to-

ken "school" is mapped to "apple", as illustrated in Figure 1. Let the input x_i be constituted of K_i tokens, i.e., x_i is the ordered sequence of tokens $(t_1, \ldots, t_{K_i})$. For all t_j = school $\in x_i$ or $x_{\text{test}}, c(t_j)$ = apple. This results in corresponding ciphered inputs x'_i or x'_{test}. Moreover, as c is a bijection, $\exists\, c^{-1}$ such that for all t_j = apple $\in x'_i$ or $x'_{\text{test}}, c^{-1}(t_j)$ = school.

Decipherability of BIJECTION cipher: Let the actual function between all (x_i, y_i) pairs be h, i.e. $h(x_i) = y_i, \forall (x_i, y_i) \in D_{\text{demo}} \cup D_{\text{test}}$. Using ICL, the model f_θ employs both TR and TL to approximate $h' \approx h$ such that $h'(x_i) \approx y_i$. This original function h can not be expected to work on ciphered (or shuffled) inputs x'_i. However, there is a corresponding function $g = h(c^{-1}(x'_i))$ that is equivalent to $h(x_i)$. This shows that h is still recoverable from the ciphered inputs. In natural language, replacing a word with another fixed but randomly decided word can completely change the meaning of its context. Any TR capabilities are expected to be severely hurt with ciphered inputs. To perform well on D_{test}, the model has to rely heavily on TL to learn and perform this composite function.

RANDOM cipher: For comparison, we also create a random substitution cipher which replaces all tokens ($\in S$) from the inputs x_i or x_{test} with tokens picked uniformly at random from S, i.e., $c(x_i) \sim \texttt{uniform}(S)$.

Indecipherability of RANDOM cipher: As c is not surjective anymore, c^{-1} does not exist. This implies that a composite function through which h can be recovered also does not exist. In this case, TL can no longer work.

Measuring "learning": Our two ciphers can be used to study the TL capabilities of LLMs. As the RANDOM cipher removes learnable patterns in the ICL Demos, the performance of LLMs on RANDOM ciphered text can act as a baseline to measure any remaining impact of TR. The gap between the performance of LLMs on RANDOM and BIJECTION ciphered text can be a practical measure of TL.

4 Experimental Setup

We evaluate a range of pretrained LLMs on four datasets to study ICL CIPHERS.

Overview: As shown in Figure 1, and described in §3, we apply BIJECTION and RANDOM Ciphers on the vocabulary of multiple LLMs for a variety of shuffle rates. We then use the ciphered ICL prompts without any additional instructions to evaluate the performance, and in turn quantify the TL capabilities of these LLMs on various tasks. We averaged the metrics across 3 runs of experiments for a more reliable evaluation.

4.1 Empirical Considerations for Implementing ICL CIPHERS

Here, we describe some nuances for shuffling vocabulary and sampling ICL examples.

Zipfian shuffling: Literature has shown a strong correlation between token frequency in the pretraining corpus and model performance (Razeghi et al., 2022). To diminish the influence of frequency, we constrain the shuffling between tokens of similar frequency. Inspired by zipfian shuffling (Piantadosi, 2014), we divide all the tokens into k ($k = 10$ in our experiments) chunks and shuffle the tokens within each chunk. Since the pre-training corpora are usually not accessible for LLMs, we use the wikipedia corpus (Foundation) to calculate token frequency instead, which is an approximation to the actual token frequency.

Informative sampling of ICL demos: To create n-shot ICL demo set, we employ an informative sampling method to prioritize examples that demonstrate the substituted tokens of the test case input. Specifically, given a input $x_i = \{t_1, t_2 \ldots t_{K_i}\}$, where m of the K_i tokens $\in S$ and are to be substituted, we sample m examples from the pool of demos, each contain at least one unique substituted token. See more details in Appendix A.3. The reason we use informative sampling instead of random sampling is to more reliably expose LLMs to relevant demos, from where they can learn to decipher. For all the scenaiors in the main text we use informative sampling. However, in §D.1 we compare these two sampling strategies and show that they perform comparably.

Preserved tokens: LLMs usually have a list of special tokens that help the model understand the prompt and task (e.g. next token prediction). For example, Llama3.1 models use `<|begin_of_text|>` and `<|end_of_text|>` to denote the start of input and end of generation. We preserve special and punctuation tokens from getting ciphered to avoid hurting models' basic functionality. Full lists of preserved tokens are in Ap-

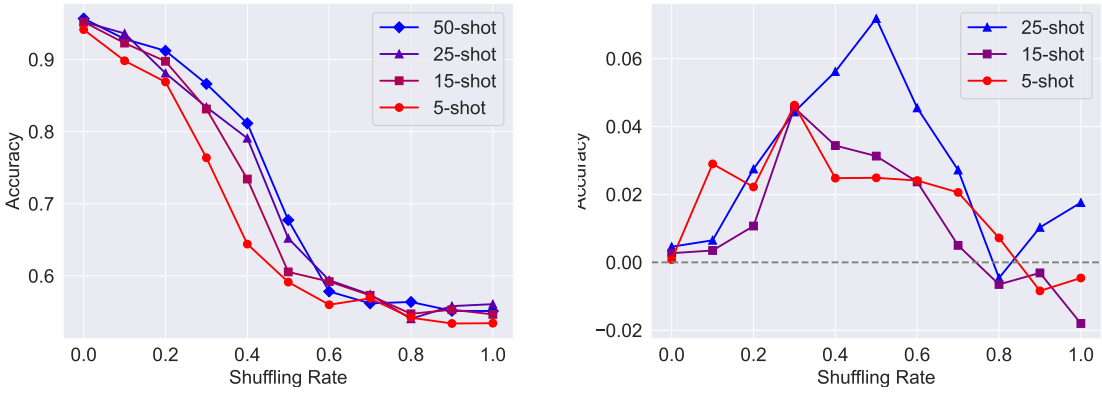

Figure 2: **Left:** Peformance of LLaMa 3.1 8B on SST-2 dataset with informative sampling. The accuracy of the BIJECTIVE cipher changes smoothly as the shuffling rate varies. With more demonstrations, accuracy increases, suggesting that the model's ability to solve ICL CIPHERS improves with additional examples. **Right:** The y-axis displays the accuracy gap between BIJECTIVE and RANDOM ciphers. This gap, though small, is consistently positive, indicating that **LLMs can decipher randomly generated ICL CIPHERS**.

pendix A.1.

Handling of white spaces: We carefully design our shuffling method to avoid disturbing spaces in the original text. See details in Appendix A.2

4.2 Models

We focus on pretrained LLMs in our experiments, including Llama 3.1 (Dubey et al., 2024, `Llama-3.1-8B`), QWen 2.5 (Team, 2024b, `Qwen2.5-7B`), OLMo (Groeneveld et al., 2024, `OLMo-7B-0724-hf`) and Gemma 2 (Team, 2024a, `Gemma-2-9b`). We don't do experiments on aligned (instruction-tuned or RLHF-ed) models as prior work shows that alignment trades typically hurts in-context learning performance (Fu et al., 2022).

4.3 Datasets

We conduct experiments on four datasets. SST-2 (Socher et al., 2013) and Amazon (Hou et al., 2024, Amazon Reviews 2023) are for binary sentiment classification task. HellaSwag (Zellers et al., 2019) is for sentence completion task, formatted as four-choices QAs. WinoGrande (Sakaguchi et al., 2020) is for pronoun resolution task, formatted as binary-choice QAS. For each dataset, we curate a demo pool for sampling ICL demos and a test set contain to-be-tested cases. Further details on our detasets are included in Appendix A.5.

4.4 Prompt Template

We don't include any instructions in our prompt. For SST-2 and Amazon, we use the following prompt template:

Input: {*input_demo*}
Output: {*label_demo*}
...
Input: {*input_test*}

where {*input_demo*} and {*label_demo*} are the input text and sentiment labels of demonstrations, and {*input_test*} is the input text of test case.

For HellaSwag and WinoGrande, we use the following prompt template:

Question: {*question_demo*}
Options: {*options_demo*}
Answer: {*answer_demo*}
...
Question: {*question_test*}
Options: {*options_test*}

where *question_demo*}, *options_demo*} and {*answer_demo*} are the questions, options and correct answers of demos, and *question_test*} and *options_test*} are the question and option of the test case.

Appendix B shows example inputs/outputs for all the datasets.

5 Empirical Findings

5.1 LLMs Shows Decipherability

LLMs can decipher ICL CIPHERS Figure 2 illustrates the performance of Llama 3.1 on SST-2 dataset with informative sampling. We can observe a clear gap between bijective ciphers and random ciphers across all the shuffling rate, indicating the model's decipherability (to some extent) of ICL ciphers. Moreover, Table 1 shows the performance of

Models → Datasets ↓	Cipher	20-shot			
		LLaMA	Qwen	Olmo	Gemma
SST-2 ($r = 0.6$)	BIJECTION	58.7 +3.0	67.4 +1.7	54.5 -2.4	62.6 +3.5
	RANDOM	55.7	65.7	56.9	59.1
Amazon ($r = 0.6$)	BIJECTION	74.9 +11.2	76.8 +5.5	75.9 +3.7	83.4 +4.4
	RANDOM	63.7	71.3	72.2	79.1
HellaSwag ($r = 0.3$)	BIJECTION	32.0+3.3	60.5 +7.5	27.0 +0.7	36.2 +4.0
	RANDOM	28.7	53.0	26.3	32.2
WinoGrande ($r = 0.1$)	BIJECTION	55.6 +1.9	63.4 +2.1	54.8 +1.3	63.1 +0.4
	RANDOM	53.7	61.3	53.5	62.7

Table 1: LLM Performance(%) with 20-shot under bijection and random cipher. In the majority of cases, models demonstrate better performances under bijection cipher than random cipher.

Shots → Datasets ↓	Cipher	Model: Llama 3.1 8B					
		5-shot	10-shot	15-shot	20-shot	25-shot	50-shot
SST-2 ($r = 0.6$)	BIJECTION	56.0 +2.4	57.6 +0.3	59.2 +2.4	58.7 +3.0	59.4 +4.5	57.8 +1.2
	RANDOM	53.6	57.3	56.8	55.7	54.9	56.6
Amazon ($r = 0.6$)	BIJECTION	65.7 +4.9	74.6 +6.7	75.6 +9.1	74.9 +11.2	74.6 +8.9	77.3 +10.3
	RANDOM	60.8	67.9	66.5	63.7	65.7	67.0
HellaSwag ($r = 0.3$)	BIJECTION	33.8 +2.6	33.1 +3.6	32.8 +2.7	32.0 +3.3	32.7 +3.5	32.0 +2.7
	RANDOM	31.2	29.5	30.0	28.7	29.3	29.3
WinoGrande ($r = 0.1$)	BIJECTION	55.8 +0.1	57.8 +3.3	55.6 +0.8	55.6 +1.9	55.1 +2.7	54.4+2.4
	RANDOM	55.7	54.5	54.8	53.7	52.4	52.0

Table 2: Task accuracy with varying numbers of ICL examples under bijection and random substitution. **Bijection vs. Random**: In all cases, model gets a higher accuracy under bijection substitution compared to random substitution. Moreover the performance gap generally increases as more examples are shown.

fours LLMs on four datasets with 20-shot prompting. It presents consistent gaps between bijection ciphers and random ciphers, except for on SST-2. This (to some extent) indicates that decipherability is universal, instead of being limited to a certain model on a certain dataset.

We have also explored the effect of Model size. As shown in Figure 3, as the size of model increases, performance for both bijection cipher and random cipher improves, while the gaps between them remain existing.

Effect of demos As evident from Table 2, the gap in performance between BIJECTION and RANDOM ciphers has an increasing trend with more demonstrations until it reaches a certain point, especially for SST-2 and Amazon. This suggests that with more demonstrations, we see more learning through TL. The effect is more pronounced with small number of demos, and the increment flattens with high number of demos.

5.2 Probing Representations

So far, our experiments have shown that LLMs are capable of deciphering substitution ciphers via ICL on multiple downstream tasks. To understand the model's internal processing of ciphered inputs, we use Logit Lens (nostalgebraist, 2020) to probe LLMs's intermediate layer representationsr. Logit Lens uses embeddings of a token from some intermediate layer, and uses the final LM head to decode it as the next token (representing what the model understands at that position).

In this section, we focus on probing on the sentiment classification task and use the same prompt in 4.4. Specifically, we first pick 600 most frequent tokens in the demo set after filtering[1], then randomly sample 30 tokens from them as the "original tokens". We then randomly sample another 30 tokens from the remaining 570 tokens as the "substitution tokens", each of which has a one-to-one correspondence with the original tokens. We apply two kinds of substitution strategies to the

[1]Filtering details are in §A.4

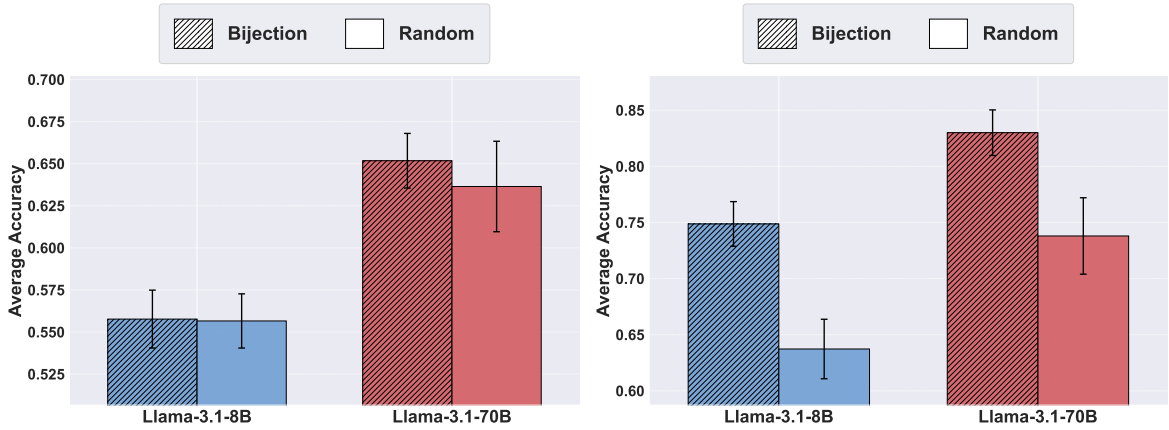

Figure 3: Performance comparison of Llama-3.1-8B and Llama-3.1-70B models on SST-2 (left) and Amazon (right) datasets under Bijection and Random ciphers. The experimental setting is 20-shot with $r = 0.6$. Larger model consistently outperforms smaller one under the same substitution strategy. Bijection substitution yields higher accuracy than random substitution for both models.

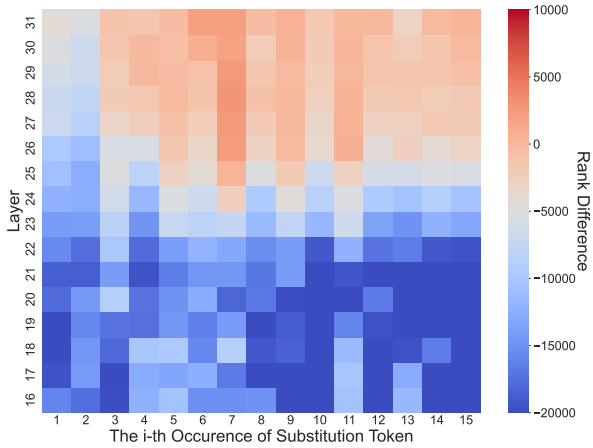

Figure 4: Rank difference (original token rank - substitution token rank) for Llama 3.1 with bijective substitution on Amazon. Positive values (red) indicate the model's preference for substituted tokens over original ones. The visualization reveals that **as layers deepen, the model gradually deciphers and favors substituted tokens**.

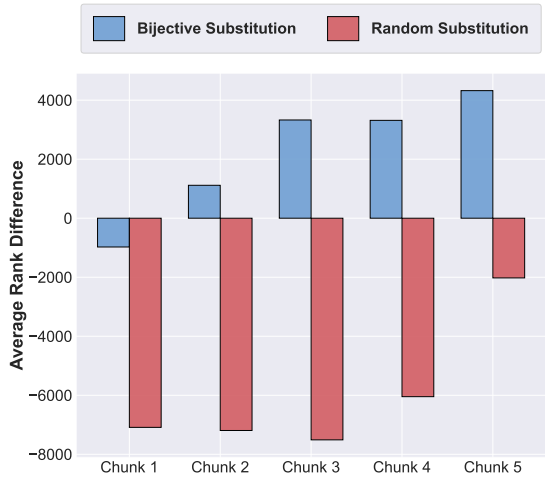

Figure 5: Average rank differences (original token rank - substitu- tion token rank) in SST-2 for BIJECTION (blue) and RANDOM (red) Cipher strategies over 15 occurrences, divided into 5 chunks of size 3. Rank difference serves as a proxy for the model's deciphering ability. Under bijective substitution, this ability improves with more exposure to substituted tokens, while random substitution shows no clear pattern.

original tokens: bijective and random. For **bijective substitution**, we substitute the original tokens with corresponding substitution tokens. For **random substitution**, we substitute each occurrence of original tokens by a randomly sampled token from the remaining 570 tokens.

For each original token, we sample 15 examples that contain it from the demo pool, applying different substitution strategies to build the ciphered prompt. Given the positions of original tokens $P = \{p_1, p_2, ..., p_n\}$, we apply logit lens to embeddings at $P' = \{p_1 - 1, p_2 - 1, ..., p_n - 1\}$, observing the ranks of original tokens and substitution tokens in the computed logits.

Figure 4 illustrates the rank difference (original token rank - substitution token rank) for Llama 3.1 with bijective substitution on Amazon dataset. As the layer gets deeper, the rank difference become positive from negative (red from blue), which means the model are ranking higher for the substitution tokens and the contrary for original tokens. This suggests model's evolving understanding for the substitution through layers.

Moreover, we also observe a trend of increasing as the the model sees more substitution tokens in

Figure 4. To get a clearer vision, we extract the rank difference from the last layer on SST-2, dividing them equally into 5 chunks, as shown in Figure 5.2. For random substitution, there is not much change for rank difference. For bijective substitution, rank difference increase as the chunk number gets bigger. This suggests that as LLM sees more occurence of the substitution token, it learns to use substitution token as the original token, namely soloving ICL CIPHERS.

6 Conclusion

We introduced ICL CIPHERS, a class of cryptography text transformations designed to evaluate novel task learning capabilities of LLMs. To perform above baseline on these ciphered tasks, the LLM has to decipher a randomly generated but reversible key. We show that LLMs exhibit the capacity to decipher these novel tasks during inference. This evidence consolidates LLMs' ability to learn novel tasks outside of their pre-training corpus. The exact mechanism of this "learning" remains an active area of study. Understanding this mechanism holds the potential to make LLMs innovate solutions to unsolved problems.

Limitations

Our work has several limitations that merit discussion:

Isolating TL: Although ICL CIPHERS aim to remove and practically hurt TR capabilities of LLMs, it is unclear if they are sufficient to isolate purely TL capabilities. We defer this exploration to future work.

Deviation from Natural Language: Ciphered text generated using ICL CIPHERS diverges from natural language. While this is useful to assess LLMs' TL capabilities, it may also make the task excessively challenging for them. Therefore, we focus on the performance gaps (between BIJECTION and RANDOM) instead of their absolute values.

More models and datasets: Although we evaluated 16 settings (four models $\times$ four datasets), expanding our study to include more and larger models would strengthen our findings. The largest model we tested was Llama 3.2 70B, due to lack of more compute resources. Additionally, we did not evaluate aligned models such as GPT-4, O1, or Gemini. Anecdotal evidence suggests that aligned models may lose their ability to follow in-context demonstrations (Fu et al., 2022), a crucial aspect of our task definition. However, we acknowledge that our task could potentially be adapted into a task description or instruction format suitable for aligned models, which deviates from our current setting and could be explored in future work. It would also be interesting to evaluate ICL CIPHERS on various pre-training checkpoints to better understand how ICL "learning" emerges through pre-training.

More sophisticated interpretability analysis: In terms of interpretability analysis, we experimented with several approaches but found success only with the simplest method, the logit lens. More advanced interpretability analyses could provide deeper insights into the underlying mechanisms, offering a clearer understanding of the processes involved.

We recognize these as areas for further exploration and encourage future research to address these limitations.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. In *International Conference on Learning Representations* (ICLR).

Nada Aldarrab and Jonathan May. 2020. Can sequence-to-sequence models crack substitution ciphers? *arXiv preprint arXiv:2012.15229*.

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

Taylor Berg-Kirkpatrick, Greg Durrett, and Dan Klein. 2013. Unsupervised transcription of historical documents. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 207–217.

Taylor Berg-Kirkpatrick and Dan Klein. 2011. Simple effective decipherment via combinatorial optimization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 313–321.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* (NeurIPS).

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

Eric Corlett and Gerald Penn. 2010. An exact a* method for deciphering letter-substitution ciphers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1040–1047.

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*.

Verna Dankers and Ivan Titov. 2024. Generalisation first, memorisation second? memorisation localisation for natural language classification tasks. *arXiv preprint arXiv:2408.04965*.

Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 266–275.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Wikimedia Foundation. Wikimedia downloads.

Yao Fu, Hao Peng, and Tushar Khot. 2022. How does gpt obtain its ability? tracing emergent abilities of language models to their sources. *Yao Fu's Notion*.

Ravi Ganesan and Alan T Sherman. 1993. Statistical techniques for language recognition: An introduction and guide for cryptanalysts. *Cryptologia*, 17(4):321–366.

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* (NeurIPS), 35:30583–30598.

Shahriar Golchin, Mihai Surdeanu, Steven Bethard, Eduardo Blanco, and Ellen Riloff. 2024. Memorization in in-context learning. *arXiv preprint arXiv:2408.11546*.

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

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

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. 2023. Understanding in-context learning via supportive pretraining data. In *Annual Meeting of the Association for Computational Linguistics* (ACL), pages 12660–12673.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.

Juno Kim and Taiji Suzuki. 2024. Transformers learn nonlinear features in context: Nonconvex mean-field dynamics on the attention landscape. *arXiv preprint arXiv:2402.01258*.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 499–506.

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

Ziqian Lin and Kangwook Lee. 2024. Dual operating modes of in-context learning. In *International Conference on Machine Learning* (ICML).

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Annual Meeting of the Association for Computational Linguistics* (ACL).

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? *arXiv preprint arXiv:2202.12837*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to gptk's language. In *Annual Meeting of the Association for Computational Linguistics* (ACL) - *Findings*.

Aaron Mueller, Albert Webson, Jackson Petty, and Tal Linzen. 2024. In-context learning generalizes, but not always robustly: The case of syntax. In *Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL), pages 4761–4779.

Aliakbar Nafar, Kristen Brent Venable, and Parisa Kordjamshidi. 2024. Learning vs retrieval: The role of in-context examples in regression with llms. *arXiv preprint arXiv:2409.04318*.

nostalgebraist. 2020. Interpreting gpt: The logit lens.

Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. Beam search for solving substitution ciphers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1576.

Edwin Olson. 2007. Robust dictionary attack of short simple substitution ciphers. *Cryptologia*, 31(4):332–342.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Shmuel Peleg and Azriel Rosenfeld. 1979. Breaking substitution ciphers using a relaxation algorithm. *Communications of the ACM*, 22(11):598–605.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems* (NeurIPS).

Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21:1112–1130.

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2513–2518.

Sujith Ravi and Kevin Knight. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 812–819.

Sujith Ravi and Kevin Knight. 2011. Bayesian inference for zodiac and other homophonic ciphers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 239–247.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP) - *Findings*.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WINOGRANDE: an adversarial winograd schema challenge at scale. In *Conference on Artificial Intelligence* (AAAI).

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. 2024. Do pretrained transformers learn in-context by gradient descent? In *International Conference on Machine Learning* (ICML).

Seongjin Shin, Sang Woo Lee, Hwijeen Ahn, Sungdong Kim, Hyoung Seok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung Woo Ha, et al. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL).

Suzanna Sia, David Mueller, and Kevin Duh. 2024. Where does in-context translation happen in large language models. *arXiv preprint arXiv:2403.04510*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 1631–1642.

Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. 2024. Out-of-distribution generalization via composition: a lens through induction heads in transformers. *arXiv preprint arXiv:2408.09503*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research* (TMLR).

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*.

Gemma Team. 2024a. Gemma.

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

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. 2024. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. In *Conference on Language Modeling* (COLM).

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 Learning Representations* (ICLR), pages 35151–35174.

Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Investigating the pre-training dynamics of in-context learning: Task recognition vs. task learning. *arXiv preprint arXiv:2406.14022*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Model with Self Generated Instructions. In *Annual Meeting of the Association for Computational Linguistics* (ACL).

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*.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. 2023. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. *arXiv preprint arXiv:2308.06463*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *ACL*.

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

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

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning* (ICML), pages 12697–12706.

Supplemental Material

A Additional Experimental Details

A.1 Preserved Tokens

For Llama 3.1, we preserve the tokens whose ids range from 0 to 255, 128000 to 128256. For Qwen 2.5, we preserve the tokens whose ids range from 0 to 255, 151643 to 151664. For OLMo, we preserve the tokens whose ids range from 0 to 244, 50254 to 50279. For Gemma 2, we preserve the tokens whose ids range from 0 to 472, 255968 to 255999. For all the models, we preserve the spaces and underlines to ensure the framework of each task. For example, in the WinoGrande dataset, LLMs are asked to predict the pronouns in a sentence, where the original pronouns are replaced by a underline.

A.2 Handling of White Space

LLMs encode the spaces between words differently depending on their tokenization. Gemma 2 uses a special underline to represent a space, while Llama 3.1 , QWen 2.5 and OLMo uses 'Ġ'. There are usually two versions of the same word – with or without a space before it, which corresponds to two different tokens. Take Llama 3.1 for example, the encoded id of "is" is 285 while that of "Ġis" is 374. We name tokens containing a space at the beginning as "space tokens" and the others as "non-space tokens". To avoid disturbing spaces in the original text, which may confuse the model, we constrain the shuffling to be within their space/non-space sets.

A.3 Informative Sampling

In informative sampling, we sample n examples for text that has m substituted tokens. If $m < n$, we randomly sample $n - m$ more examples other than the m examples. If $m > n$, we randomly pick n examples from the m sampled examples.

A.4 Filtering

We use nltk (Bird et al., 2009) to filter tokens other than verbs, nouns and adjectives.

A.5 Datasets

SST-2: We use its validation set as our test set, which has size of 872. Its training set, which contains 67.3k examples, is used as the demo pool.

Amazon: To fit the Amazon dataset into binary sentiment classification framework, we filter ratings 4-5 as positive and 1-2 as negative (discard rating 3). We focus on reviews under the the "All_Beauty" category in our experiments. We sample 144k positive and negative samples to build the demo pool; and 500 other positive and negative examples as the test set.

HellaSwag: We use its validation set as our test set, which contains 444 positive examples and 428 negative examples (872 examples in total). Its training set, which contains 38K positive examples and 30k negative examples, is used as the demo pool.

We randomly sample 1k examples from the validation set as our test set. We use its training set as the demo pool, which contains 40k examples.

WinoGrande: We use its develop set as the test set, which contains 1267 examples. Its xl training set is used as demo pool, which has 40k examples.

B Example Inputs/Outputs

Here we display the example inputs/outputs on all the four datasets. Note that in our experiments the original inputs are not included in the prompts.

Dataset: SST-2; Model: QWen 2.5 ; Cipher: bijection; Shuffle Rate: 0.6

Ciphered Input: been sc Mil Swift the Inch for pen Venezuela Moody
Original Input: been sent back to the tailor for some major alterations
Output: negative

Ciphered Input: is born Slovenia of an Platform San sitcom involved also Sr implementedecture embarrassed Swift Malay you reach for the tissues Confederate
Original Input: is born out of an engaging storyline , which also is n't embarrassed to make you reach for the tissues .
Output: positive
...

Ciphered Test Input: allows us Swift hope Esc implementedolan Sr poised Swift cheating a Venezuela career Mr a assembled Kann steak filmmaker Confederate
Original Test Input: allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker .

Dataset: Amazon ; Model: Gemma 2 ; Cipher: bijection; Shuffle Rate: 0.6

Ciphered Input: didnSUwell really notice anything mob. I sink it householder substance Woodward Bean Simple Woodward Senior Caldwell Snowyyn Ato was instance.
Original Input: didn't really notice anything special. I bought it because of the reviews and the price but honestly, I was disappointed.
Output:negative

Ciphered Input: Item arrived regions principle unrest neighbours']modern /><modern urchatosyn Woodward item was calcium steamer principle Counter cap rendering Woodward cover ent since it periodsSUwell Fam Arch anymore Simple iconicBer bottom Simple consequently']modern /><modern urchofficial was wrapped dentist regions principle padded envelope.
Original Input: Item arrived in a quick manner.

However, the item was received with a damaged cap rendering the cover useless since it won't snap on anymore and dented bottom and top.

It was wrapped tightly in a padded envelope.
Output:negative
...

Ciphered Test Input: tried it for cosmetic qualifications perimeter a day spaŏ0f2 didnPervers Tehran workil
Original Test Input: tried it for cosmetic procedures in a day spa; didn't really work.

Dataset: Hellaswag; Model: OLMo ; Cipher: bijection; Shuffle Rate: 0.3

Ciphered Question: Ter Back sits million titled with his Board effective on the keys. the Back
Original Question: A man sits a piano with his hands placed on the keys. the man
Ciphered Options: (1) begins playing the titled.(2) Carlos the keys with million malorn.(3) beats the titled in million benefitedmic thought.(4) increases the play for playing.
Original Options: (1) begins playing the piano.(2) hits the keys with a mallet.(3) beats the piano in a rhythmic beat.(4) increases the volume for playing.
Answer: (1)
...
Ciphered Question: People are noted on the street. million Back
Original Question: People are running on the street. a man
Ciphered Options: (1) is wearing poetilts.(2) limited million drink out Wars million After presidents.(3) negotiating into million encourages Wars fire.(4) limited million high jump in million Chris competition.
Original Options: (1) is wearing stilts.(2) takes a drink out of a water bottle.(3) jumps into a pile of fire.(4) takes a high jump in a bar competition.

Dataset: WinoGrande ; Model: Llama 3.1 ; Cipher: bijection; Shuffle Rate: 0.3

```
Ciphered Question: Estonia ferry that my parents story tied I permanent in Johnston permanent
Stadium partners bla than my house now because the _ permanent anchored.
Original Question: The home that my parents had when I was in school was a lot nicer than my
house now because the _ was sophisticated.
Ciphered Options: (1) ferry, (2) house
Original Options: (1) home, (2) house
Answer:(1)
...
Ciphered Question: Sarah permanent Stadium much better Chart than Maria so _ always got the
easier cases.
Original Question: Sarah was a much better surgeon than Maria so _ always got the easier cases.
Ciphered Options: (1) Sarah, (2) Maria
Original Options: (1) Sarah, (2) Maria
```

C Additional Related Work

Empirical understanding of ICL: Ever since In-Context Learning was discovered (Brown et al., 2020), multiple works have studied it under diverse settings (Zhao et al., 2021; Min et al., 2022; Mishra et al., 2022; Han et al., 2023; Wang et al., 2023; Sia et al., 2024; Vacareanu et al., 2024; Mueller et al., 2024). For instance, Srivastava et al. (2023) benchmarked ICL under multiple tasks and models; Perez et al. (2021); Lu et al. (2022) showed the sensitivity of ICL to the choice of demonstrations and their orderings; Shin et al. (2022); Razeghi et al. (2022) showed the sensitivity of ICL performance to the frequency and size of the relevant pre-training corpus. These works have made useful observations that allow us to better use this elusive quality of LLMs.

Functional nature of ICL: A more recent line of study aims to understand how ICL actually works in LLMs. Multiple works have compared ICL with implicit optimization (specifically gradient descent) (Garg et al., 2022; Zhang et al., 2023; Dai et al., 2022; Akyürek et al., 2022; Von Oswald et al., 2023; Li et al., 2023; Kim and Suzuki, 2024). This line of work claims that Transformers can meta-learn to perform optimization of internal models given a set of demonstrations. However, their study setup with toy transformers does not align with how LLMs are trained as shown by Shen et al. (2024). Moreover, this line of study does not explain the TR capabilities of LLMs.

Forces that lead to ICL: Few works try to understand *how ICL emerges in LLMs*. Xie et al. (2021) explained ICL as implicit Bayesian inference, which maps a ICL demonstrations to a latent concept (task) learned via pre-training. Hahn and Goyal (2023) posited that compositional structure in natural language gives rise to emergent in-context learning. Other works (Chan et al., 2022) theorize more distributional properties in the pre-training data, that may give rise to ICL. Many of these works explain some properties of ICL, but fail at others. The exact origin of ICL in LLMs still remains an active area of study.

D Additional Experiment Results

D.1 Informative vs Random Sampling

Figure 6 shows peformance of LLaMa 3.1 8B on SST-2 dataset with random sampling. Comparing with Figure 2, they demonstrate similar trend but there are more variance in random sampling due to the its random nature. Therefore, we use informative sampling throughout our experiments for more steady results.

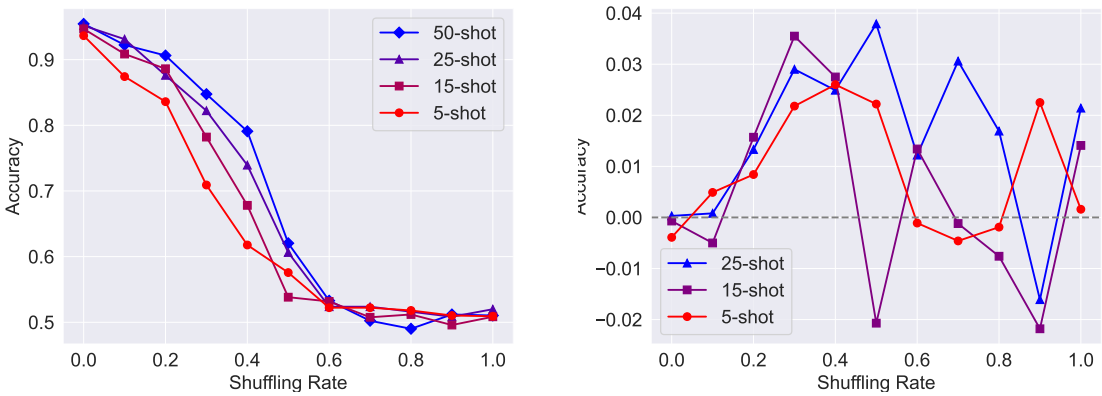

Figure 6: **Left:** Peformance of LLaMa 3.1 8B on SST-2 dataset with random sampling. **Right:** The y-axis displays the accuracy gap between Bijection and Random ciphers.

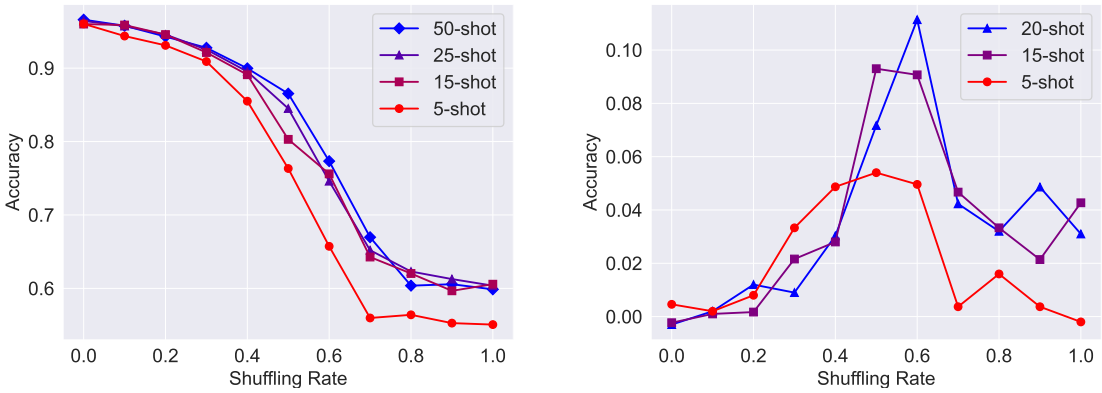

Figure 7: LLama3 performance on Amazon dataset with informative sampling (similar to Fig.2). The left plot shows the performance change of bijective substitution. The right plot shows the gap between bijective and random substitution.

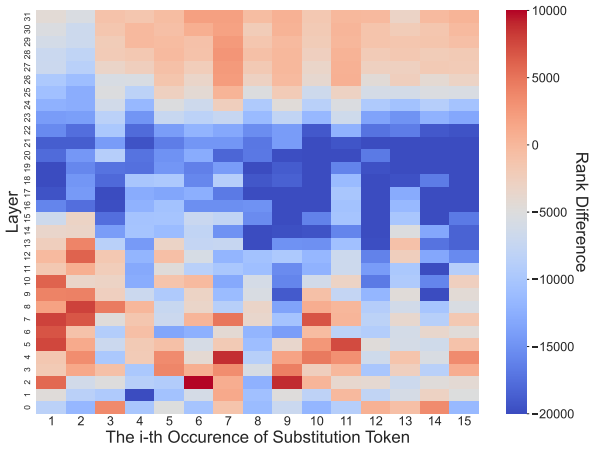

Figure 8: Whole heatmap of original token rank minus substitution token rank on Amazon. We show the partial heatmap for a subset of the layers in Fig.4.

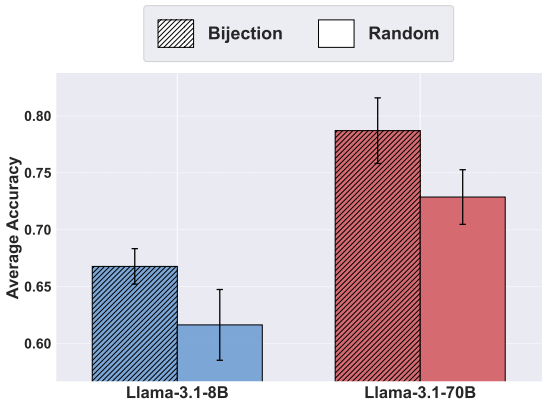

Figure 9: Performance comparison of Llama-3.1-8B and Llama-3.1-70B models on Amazon datasets(random sampling) under Bijection and Random substitution strategies.

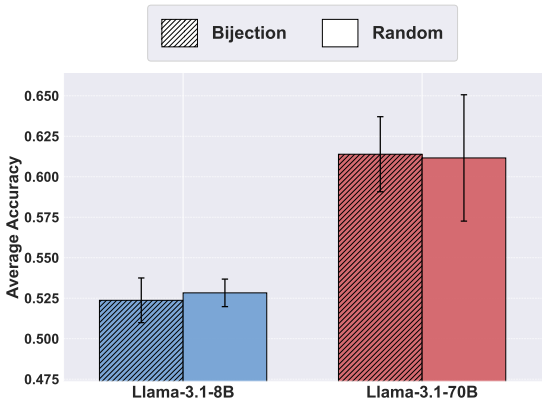

Figure 10: Performance comparison of Llama-3.1-8B and Llama-3.1-70B models on SST_2 datasets(random sampling) under Bijection and Random substitution strategies.

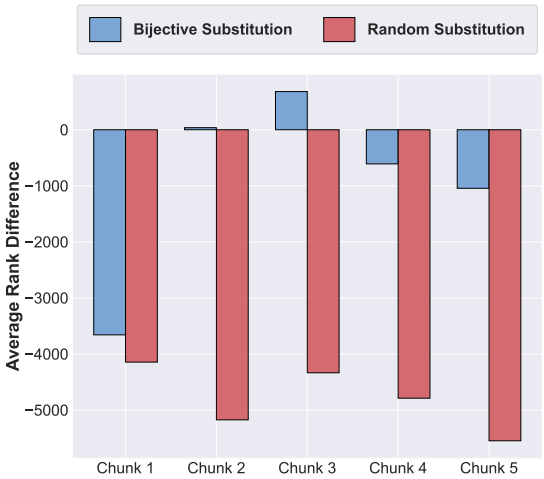

Figure 11: Average rank differences in Amazon for bijective (blue) and random (red) substitution strategies over 15 occurrences, divided into 5 chunks of size 3.