Identifying and Analyzing Task-Encoding Tokens in Large Language Models

Anonymous ACL submission

Abstract

In-context learning (ICL) has become an effective solution for few-shot learning in natural language processing. However, our understanding of ICL's working mechanisms is limited, specifically regarding how models learn to perform tasks from ICL demonstrations. For example, unexpectedly large changes in performance can arise from small changes in the prompt, leaving prompt design a largely empirical endeavour. In this paper, we investigate this problem by identifying and analyzing *task-encoding* tokens on whose representations the task performance depends. Using experiments that ablate the representations of different token types, we find that template and stopword tokens are the most prone to be task-encoding. In addition, we demonstrate experimentally that lexical meaning, repetition, and text formatting are the main distinguishing characteristics of these tokens. Our work sheds light on how large language models (LLMs) learn to perform a task from demonstrations, deepens our understanding of the varied roles different types of tokens play in LLMs, and provides insights for avoiding instability from improperly utilizing task-encoding tokens.

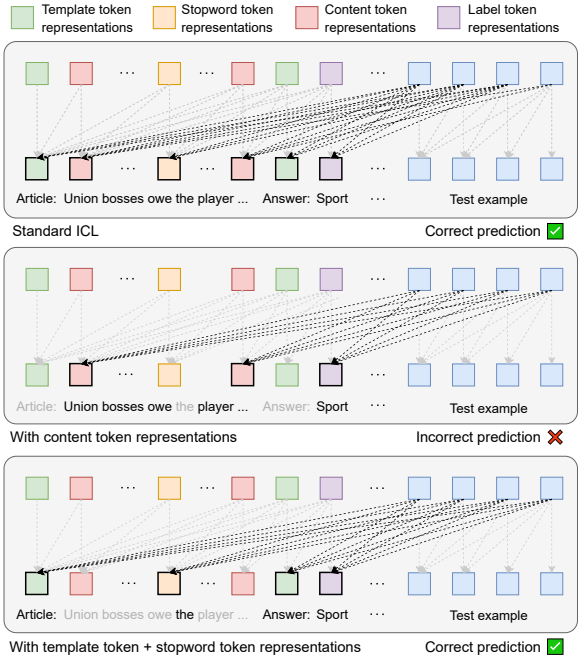

Figure 1: An illustration of the 4-way text classification on AGNews with different parts of its 4-shot ICL demonstrations masked with respect to the attention of the test example. Masking the representations of what we call the template and stopword tokens from the attention of the test example leads to a significant drop in performance while masking representations of the content tokens leaves the performance relatively unchanged.

1 Introduction

In-context learning (ICL) has become a popular technique employed with large language models (LLMs) (Brown et al., 2020). However, ICL has been shown to be unstable in that slight changes to the in-context prompts (e.g., reordering of demonstrations) can lead to substantial differences in performance (Lu et al., 2022; Zhang et al., 2022). This circumstance is difficult to control due to a lack of understanding of the model's working mechanisms, leaving us uncertain about the exact process by which LLMs learn to perform a task from demonstrations. Previous papers have begun to explore this issue, focusing on specific aspects such as the label space (Min et al., 2022) and the hidden states of the last prompt token (Hendel et al., 2023; Todd et al., 2023), but have been limited in scope. In this work, we conduct a more comprehensive study on how LLMs extract information valuable to improving task performance from demonstrations, by characterizing the model's *task-encoding tokens*.

Conceptually, task-encoding tokens are defined as tokens whose representations encode the task-solving process. Since this cannot be extracted directly from the representations constructed by the model, as a practical proxy, we take task-encoding tokens to be those tokens whose representations LLMs depend on to achieve high-level per-

formance. In other words, removing the representations of these tokens should lead to diminished task performance. In this paper, we identify and analyze the characteristics of task-encoding tokens, motivated by two main reasons: 1) The identification of task-encoding tokens is likely to provide new insights into how large language models leverage critical information useful for performing a task. 2) Analyzing the characteristics of task-encoding tokens helps us better understand how they are different from the tokens whose representations do not explicitly affect the ICL performance, which would provide insights into avoiding possible instability caused by improperly utilizing these tokens.

In order to identify the task-encoding tokens, we divide tokens into three structural categories: template tokens, stopword tokens (including punctuations, conjunctions, etc.), and content tokens. Intuitively, template tokens structure the entire ICL prompt into organized text by indicating the task input and output, while stopword tokens contribute to the overall structure of the plain text. With these structural categories in mind, we ablate the representations of tokens of different types from the attention of ICL test examples, masking partial information during the model's solving of the task. We use the observed changes in task performance to draw conclusions about which types of tokens are likely task-encoding tokens, as shown in Figure 1. Results of these experiments provide evidence that template tokens and stopword tokens are the most prone to be task-encoding tokens as ablating their representations together significantly decreases performance. In contrast, content tokens have a negligible impact on performance. We further investigate every token in the template with the same ablation method, confirming that most of these tokens have representations necessary for preserving the task performance.

Beyond identifying task-encoding tokens, we investigate other ways in which they may differ from other tokens. We find the following three distinguishing characteristics: the **lexical meaning** of tokens as it relates to the task being solved, the **repetition** of tokens throughout the prompt, and the **text formatting** which the tokens provide to the prompt. Our research indicates that the lexical meaning, repetition, and text formatting brought by task-encoding tokens contribute to task performance across all model sizes, suggesting that these characteristics should not be disrupted in order to avoid performance fluctuation, therefore maintaining its stability.

Our work reveals that we can identify and characterize the types of tokens with representations that are most needed for an LLM to perform well on downstream tasks during the ICL process. We investigate the characteristics of lexical meaning, repetition, and text formatting related to task-encoding tokens which allow us to partially explain the presence of task-encoding tokens and help us better avoid the instability caused by them. Our findings deepen the understanding of the roles different types of tokens play in large language models, suggesting future work based on leveraging specific representations of different token types. Code and data will be released in the future.

2 Related Work

2.1 Working mechanisms of in-context learning

Since the proposal of in-context learning (Brown et al., 2020), its working mechanisms have been extensively studied by the research community (Min et al., 2022; Liu et al., 2021; Olsson et al., 2022; Bhattamishra et al., 2023). Min et al. (2022) suggest that demonstrations primarily provide the label space, the distribution of the input text, and the format of the sequence for the test example. They argue that the precise ground truth labels do not have significant importance. Conversely, Yoo et al. (2022) propose a differing view, stating that the impact of the ground truth labels depends on the experimental configuration. Xie et al. (2021) explain ICL as implicit Bayesian inference, while Akyürek et al. (2022) explore ICL learning process using linear models. Theoretical explanations (Guo et al., 2023; Bai et al., 2023; Li et al., 2023) and gradient descent explanations have also been proposed.

Additional analyses exploring different aspects of ICL have also been studied. For instance, order sensitivity where task performance fluctuates based on the order of the same ICL demonstrations has been identified as a limitation of ICL (Lu et al., 2022). Yan et al. (2023) propose that repetitive patterns in the prompt could affect the ICL performance in both positive and negative ways. Pan et al. (2023) analyze the ICL process by disentangling it into task recognition and task learning.

Our work investigates the execution of ICL in LLMs at inference time, demonstrating that certain specific tokens are more likely to possess representations that could affect the processing of the final

test sample, improving the task performance.

2.2 Function vectors of in-context learning

In a similar line of work to ours, Todd et al. (2023) and Hendel et al. (2023) provide evidence of function vectors that store information used to solve a task in ICL. They probe and extract the hidden representations of the final tokens in the prompt. These vectors can then be added to, or used to replace, the corresponding vectors in a zero-shot example, achieving results comparable to those obtained when the model uses all demonstrations as context. In addition, Liu et al. (2023) also propose using an in-context vector to represent the target task and applying feature shifting to query examples. They first feed each input and its corresponding target separately into an LLM, then concatenate all the latent states. A PCA method is applied to derive a vector that is more closely aligned with the task. Finally, Wang et al. (2023) proposes that label words in the demonstration examples function as information anchors by aggregating the information from previous demonstrations and providing it to the test example. This finding suggests that we may view label tokens as satisfying our definition for task-encoding tokens.

All these previous studies either solely focus on a single token (i.e., the last prediction prompt token or label token) of the ICL prompt or treat the entire demonstration as a single unit, neglecting the other individual tokens within it. Our research focuses on all the tokens in the prompt and reveals that there are additional tokens with specific characteristics whose representations significantly affect the final ICL performance.

3 Preliminaries

3.1 Notation

In-context learning (ICL) is a technique that enables large language models (LLMs) to perform tasks in a few-shot manner by placing task demonstrations (e.g., input-output pairs) in the context fed to a large language model (Brown et al., 2020). In ICL, these demonstrations are leveraged to construct a structured prompt that guides the model in predicting the final answer. Formally, the structural prompt consists of the following components: the instruction $\mathbf{I}$, the templates $\mathbf{T}^{\text{in}}$, $\mathbf{T}^{\text{out}}$, and the demonstrations $\mathbf{D}_i^{\text{in}}$, $\mathbf{D}_i^{\text{out}}$, where i denotes the ith demonstration while in and out refer to the input text and output labels, respectively. These

Component notation	Component example
$\mathbf{I}$	Classify the news articles into the categories of World, Sports, Business, and Technology.\n\n
$\mathbf{T}^{\text{in}}$	Article: {$\mathbf{D}^{\text{in}}$}\n
$\mathbf{T}^{\text{out}}$	Answer: {$\mathbf{D}^{\text{out}}$}\n\n
$\mathbf{D}_1^{\text{in}}$	Radio veteran Karmazin joins Sirius. Sirius Satellite Radio Inc. named former Viacom Inc. president Mel...
$\mathbf{D}_1^{\text{out}}$	Business
$\mathbf{D}_2^{\text{in}}$	Numbers point to NY. NEW YORK - The New York Yankees can achieve two milestones with one more victory...
$\mathbf{D}_2^{\text{out}}$	Sports
ICL Prompt	Classify the news articles into the categories of World, Sports, Business, and Technology. Article: Radio veteran Karmazin joins Sirius. Sirius Satellite Radio Inc. named former Viacom Inc. president Mel... Answer: Business Article: Numbers point to NY. NEW YORK - The New York Yankees can achieve two milestones with one more victory... Answer: Sports

Table 1: An example of the components of a 2-shot ICL prompt in the AGNews dataset.

prompt component are concatenated to form the ICL prompt, as shown in Table 1. During inference, the templated version of the test example (without an answer) is appended to the ICL prompt and then sent to the large language model to predict the corresponding answer.

3.2 Experimental Settings

In this section, we describe the experimental setup for all of our experiments.

For the datasets, we consider the most widely used text classification datasets used by previous studies (Zhao et al., 2021). For topic classification, we use the 4-way and 14-way datasets AGNews and DBPedia (Zhang et al., 2015). For textual entailment, we use the 3-way CB (De Marneffe et al., 2019) and 2-way RTE dataset (Dagan et al., 2005). We also use SST2 (Socher et al., 2013) and TREC (Voorhees and Tice, 2000) for sentiment and question classification tasks.

For each dataset, we randomly select 4 training demonstrations from the training set using 15 different random seeds limited by the computational cost of the inference stage of LLMs. For testing, we evaluate each setting on 500 randomly selected test examples. Instruction prompt I is retained in all the different kinds of ablations since it is essential for enhancing the classification performance of the model (Yin et al., 2023).

For the LLMs, we utilize the 7B, 13B, and 30B versions of the Llama model and a 3B OpenLlama model. All the experiments are conducted using a single A100 80G GPU. For the 13B and 30B models, we apply 8-bit quantization to ensure the model fits into a single GPU. The results are inferred using Huggingface Transformers (Wolf et al., 2020).

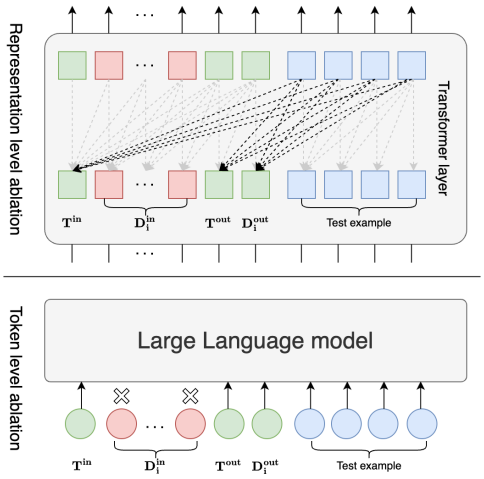

Figure 2: An illustrative example of the representation and token-level ablation methods we use to test if template and stopword tokens are task-encoding tokens. The test example can only access partial information (shown in green) from the demonstrations.

4 Methodology

In this section, we aim to find the task-encoding tokens in the ICL prompt. We first structurally categorize all the tokens in the prompt into three types: template, stopword, and content tokens. Then, we provide supporting evidence from the view of task performance to show that the representations of template and stopword tokens are the most prone to be task-encoding tokens.

4.1 Token types

We categorize ICL tokens based on the way the ICL prompt is structured according to our notation:

Template tokens (TEMP) In defining template tokens, we include all the tokens which serve as templates for the ICL prompt. This collection of tokens includes the tokens in $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$.

Stopword tokens (STOP) In defining stopword tokens, we include punctuation and conjunction word, such as [,], [.], etc., in the ICL prompt. We use the stopword tokens appear in the instructions[1]. The stopword token list is shown in Appendix B.

Content tokens (CONT) In defining content tokens, we include all the tokens from $\mathbf{D}^{\text{in}}$ except for the ones that are already stopword tokens. We use the term "content" as they convey the meaningful information found in the demonstrations.

[1]Ablation experiments with a more complete NLTK (Loper and Bird, 2002) stopwords list are conducted in Appendix B.

Given this categorization, we conduct ablation experiments to determine which tokens are more likely to be task-encoding tokens.

4.2 Ablation on token types

To determine which token types are more likely to be task-encoding tokens whose representations affect the final performance largely, we design two experiments. The first involves keeping and masking the different kinds of token representations from the attention of the test example. The second involves dropping the various kinds of tokens from the ICL prompt. Illustrations of these two methods which we refer to as representation-level and token-level ablations are shown in Figure 2.

4.2.1 Representation-level ablation

Our first ablation stems from the intuition that if LLMs essentially rely on the representations of certain token types to achieve high-level performance, then the model should perform the target task adequately with only these representations. Meanwhile, performance should decrease significantly if we remove them from the attention of the test example. Hence, we first pass the entire ICL prompt to the LLM and then restrict the attention of the test example such that the LLM may only attend to the representations of tokens of a particular type (or types)[2] during its solving of the task. This representation-level ablation is illustrated in the upper part of Figure 2. We compute task performances with every possible ablation combination, removing the representations of one (e.g., Standard ICL − TEMP) or two token types (e.g., Zero-shot + CONT[3]) from the attention of the test example[4]. All the task performances and the averaged relative performance changes are reported, shown in Table 2 and Table 3.

Overall, these results demonstrate that the representations of template tokens and stopword tokens are more likely to be task-encoding tokens than content tokens. On the one hand, template token representations are crucial for LLMs' task-solving ability via ICL, achieving an average performance 39.8% higher than the zero-shot baseline by only utilizing these representations at inference time. In

[2]Since $\mathbf{D}^{\text{out}}$ tokens have been shown to significantly impact performance (Wang et al., 2023), we always preserve the attention on the representations of the $\mathbf{D}^{\text{out}}$ tokens.

[3]Removing two types of tokens from Standard ICL is equivalent to adding the other type of tokens to Zero-shot.

[4]Both STOP and TEMP include the "\n" token; therefore, we preserve the "\n" as long as one of them is not ablated.

Models	Setting	AGNews	SST2	TREC	DBPedia	RTE	CB	△Avg.
OpenLlama 3B	Zero-shot	22.0	20.0	23.6	5.4	44.4	1.8	19.5
	+ CONT	26.2	52.1	30.1	7.4	51.9	37.9	+14.8
	+ STOP	36.7	82.9	**32.0***	52.4	**58.8**	**56.2**	+33.7
	+ TEMP	**56.5**	**86.7**	27.1	**62.2**	56.4	52.3	**+37.4**
Llama 7B	Zero-shot	25.0	29.2	41.4	0.0	54.2	3.6	25.6
	+ CONT	32.4	57.9	42.5	12.5	55.5	46.1	+15.6
	+ STOP	57.3	83.7	49.8	43.0	55.9	50.7	+31.1
	+ TEMP	**70.8**	**90.2**	**58.4**	**66.2**	**66.3**	**73.5**	**+45.3**
Llama 13B	Zero-shot	59.0	18.0	37.0	0.0	0.0	0.0	19.0
	+ CONT	27.7	52.4	33.5	10.9	61.7	41.7	+19.0
	+ STOP	72.2	73.5	46.8	50.7	58.6	30.6	+36.4
	+ TEMP	**80.0**	**92.3**	**58.6**	**76.9**	**68.5**	**47.7**	**+51.7**
Llama 30B	Zero-shot	70.2	88.6	60.6	30.2	58.1	19.6	54.6
	+ CONT	24.4	61.7	62.1	10.5	65.2	63.6	−6.7
	+ STOP	72.9	92.7	**66.7***	69.1	69.6	63.0	+17.7
	+ TEMP	**80.5**	**95.2**	65.2	**75.2**	**79.0**	**80.0**	**+24.6**

Table 2: The accuracy results of the representation-level ablation study where, for example, + TEMP refers to allowing attention only to template tokens. All values are presented as percentages. Except where noted with *, all test statistics reported correspond to p-values < 0.05. The best results are in bold.

Models	Setting	AGNews	SST2	TREC	DBPedia	RTE	CB	△Avg.
OpenLlama 3B	Standard ICL	63.7	91.2	21.9	61.9	57.4	52.0	58.0
	− CONT	58.2	86.9*	27.6	61.9	56.5*	51.7	−0.9
	− STOP	62.3	91.0	24.8*	62.9	57.1	51.1*	+0.2
	− TEMP	41.9	87.2	26.0	56.3	58.5	57.4	−5.4
Llama 7B	Standard ICL	82.4	94.3	63.5	68.7	68.6	71.3	74.8
	− CONT	77.9	91.5	58.5	66.5	67.8	74.4	−2.0
	− STOP	80.4	94.6	61.1	68.0	67.2	72.0	−0.9
	− TEMP	64.5	84.1	54.0	58.0	56.8	54.3	−12.8
Llama 13B	Standard ICL	81.6	94.3	60.0	76.1	70.6	39.9	70.4
	− CONT	81.4	93.1	58.9	75.7	69.6	45.1	+0.2
	− STOP	81.2	94.1	59.3	76.9	69.2	40.6	−0.2
	− TEMP	74.1	80.0	46.5	30.6	58.3	25.4	−17.9
Llama 30B	Standard ICL	85.0	96.5	68.1	78.4	78.5	83.3	81.6
	− CONT	82.3	95.4	64.9	76.1	80.4	82.0	−1.4
	− STOP	84.3	95.6	65.7	77.6	78.6	81.8	−1.0
	− TEMP	76.6	93.9*	61.2	72.7	70.3	59.6	−9.2

Table 3: The accuracy results of the representation-level ablation study where, for example, − TEMP refers to allowing attention only to content and stopword tokens. All values are presented as percentages. Except where noted with *, all test statistics reported correspond to p-values < 0.05. The results showing the greatest decrease from ablation are underlined.

contrast, content token representations only bring an average improvement of 10.7%. If the representations of stopword tokens are further included (i.e., Standard ICL−CONT), the performance is nearly equivalent to that of the Standard ICL. On the other hand, the performance decreases the most with Standard ICL−TEMP, highlighting the significance of template tokens again.

Rare exception cases appear when performance is relatively poor with Standard ICL (e.g., OpenLlama 3B in TREC). In some cases, masking the representations of the content tokens brings even better performance than the Standard ICL method, which is possibly due to the elimination of noisy information in the demonstration content. Another interesting observation is that the performance results of Standard ICL−STOP and Standard ICL−CONT where the attention to the content and stopword tokens is ablated respectively are close, with an average difference of only 0.7%. This indicates that the representation of stopword tokens may contain overlapping information with their preceding content tokens. We believe that this could enable LLMs to model long sequences without significant architectural changes (e.g., using stopword token representations as synthesis checkpoints) and leave the verification of this hypothesis to future work.

4.2.2 Token-level ablation

In this section, we modify the ICL prompt by removing certain types of tokens from the ICL prompt[5] to further determine the extent to which final performance relies on different token types.

[5]This includes *both* the demonstration tokens as well as the tokens in the test example.

Models	Settings	AGNews	SST2	TREC	DBPedia	RTE	CB	Avg.
OpenLlama 3B	Standard ICL	63.7	91.2	21.9	61.9	57.4	52.0	58.0
	− CONT	31.5	63.0	40.6	25.4	56.1	48.9	44.3
	− STOP	64.4	91.5	20.9	62.3	57.8	52.6	58.3
Llama 7B	Standard ICL	82.4	94.3	63.5	68.7	68.6	71.3	74.8
	− CONT	55.2	67.2	42.6	50.8	57.4	56.3	54.9
	− STOP	82.3	93.8	64.1	69.7	66.5	70.0	74.4
Llama 13B	Standard ICL	81.6	94.3	60.0	76.1	70.6	39.9	70.4
	− CONT	78.8	81.7	45.3	75.1	55.1	54.5	65.1
	− STOP	82.5	92.5	61.5	76.5	69.6	40.5	70.5
Llama 30B	Standard ICL	85.0	96.5	68.1	78.4	78.5	83.3	81.6
	− CONT	74.0	89.6	67.0	73.0	69.8	49.0	70.4
	− STOP	85.3	96.4	66.9	77.9	77.7	81.3	80.9

Table 4: Results of the token-level ablation where, for example, −STOP refers to the ablation where stopword tokens are dropped from the ICL prompt. Models without template tokens consistently yielded an accuracy of 0% and are thus omitted from this table.

The token-level ablation is illustrated in the lower part of Figure 2. When we ablate the template tokens, we preserve the answer and next-line tokens in the templates to maintain a basic separator between the demonstration inputs and outputs. Results are presented in Table 4.

Our main finding from this ablation is that removing template tokens causes the LLMs to completely lose their ability to solve tasks via ICL with an overall task accuracy performance of 0% for all sizes and all tasks. We hypothesize that this is because the model no longer has an explicit cue to generate the target label. In this case, if we add back the last prompt token after the next-line token, the results return to their original level due to the introduction of a template token. This finding confirms previous claims that preserving the format of ICL prompts plays a significant role in retaining the task performance (Min et al., 2022).

In additon, these experiments further demon-

Models	Settings	AGNews		SST2		TREC		DBPedia		RTE		CB	
		with ":"	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**
OpenLlama 3B	TEMP with $\mathbf{D}^{\text{out}}$	56.5	47.4	86.7	83.7	27.1	26.5	62.2	59.8	56.4	56.0	52.3	56.1
	$-\mathbf{T}^{\text{in}}$	50.3	47.1	85.7	84.4	28.9	24.4	57.7	57.7	56.5	56.1	53.2	55.2
	$-\mathbf{T}^{\text{out}}$	34.6	32.7	86.9	82.3	28.2	31.2	55.5	54.1	58.3	59.2	55.4	58.3
Llama 7B	TEMP with $\mathbf{D}^{\text{out}}$	70.8	57.3	90.2	87.1	58.4	46.7	66.2	63.8	66.3	59.5	73.5	69.6
	$-\mathbf{T}^{\text{in}}$	62.7	55.1	91.6	87.1	52.8	43.3	61.6	61.8	58.9	56.5	59.2	55.7
	$-\mathbf{T}^{\text{out}}$	50.8	48.6	84.9	82.8	46.0	50.2	57.9	55.2	56.7	56.7	66.2	64.5
Llama 13B	TEMP with $\mathbf{D}^{\text{out}}$	80.0	76.2	92.3	89.1	58.6	54.0	76.9	71.4	68.5	59.8	47.7	35.0
	$-\mathbf{T}^{\text{in}}$	79.9	76.3	91.5	88.9	55.1	47.8	75.8	70.7	65.5	59.0	35.7	24.5
	$-\mathbf{T}^{\text{out}}$	72.0	72.1	81.1	75.9	47.1	48.3	60.3	35.5	61.2	58.4	36.2	36.0
Llama 30B	TEMP with $\mathbf{D}^{\text{out}}$	80.5	75.0	95.2	93.3	65.2	66.7	75.2	73.5	79.0	77.1	80.0	70.7
	$-\mathbf{T}^{\text{in}}$	78.7	71.5	95.2	92.8	68.1	67.7	75.1	73.8	77.4	75.4	73.3	62.3
	$-\mathbf{T}^{\text{out}}$	69.2	69.5	93.9	92.9	62.1	66.2	71.3	70.1	72.8	70.0	67.4	63.5

Table 5: Ablation for different template token representations, presented as percentages. The results showing the greatest impact from ablation are underlined.

Models	Settings	AGNews		SST2		TREC		DBPedia		RTE		CB	
		with ":"	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**	**with ":"**	**w/o ":"**
OpenLlama 3B	TEMP w/o $\mathbf{D}^{\text{out}}$	41.5	54.6	14.3	73.2	36.5	42.0	29.4	21.7	24.7	45.7	0.7	3.5
	$-\mathbf{T}^{\text{in}}$	42.2	52.2	18.5	79.9	39.7	42.2	22.6	22.5	49.8	57.1	3.1	6.5
	$-\mathbf{T}^{\text{out}}$	36.3	35.5	83.0	83.4	43.2	41.9	16.3	18.7	54.4	56.8	1.2	4.2
Llama 7B	TEMP w/o $\mathbf{D}^{\text{out}}$	50.4	56.6	68.2	56.1	55.3	48.5	0.2	1.3	40.7	42.5	28.5	18.8
	$-\mathbf{T}^{\text{in}}$	46.1	50.6	61.9	55.1	43.5	44.7	0.0	0.2	43.7	49.9	27.1	26.5
	$-\mathbf{T}^{\text{out}}$	21.4	12.7	86.2	66.5	54.4	55.6	0.0	0.0	56.0	55.6	39.4	35.1
Llama 13B	TEMP w/o $\mathbf{D}^{\text{out}}$	66.9	77.0	65.6	87.9	51.8	53.1	0.1	0.1	57.5	53.7	16.7	21.9
	$-\mathbf{T}^{\text{in}}$	72.9	76.6	83.0	89.5	45.5	48.4	0.0	0.0	53.6	52.8	16.0	20.1
	$-\mathbf{T}^{\text{out}}$	79.2	77.7	77.5	47.5	56.8	43.2	0.0	0.0	54.8	53.7	4.3	2.4
Llama 30B	TEMP w/o $\mathbf{D}^{\text{out}}$	77.3	78.2	17.3	88.9	65.4	69.3	31.0	41.7	71.8	65.8	23.8	23.0
	$-\mathbf{T}^{\text{in}}$	72.9	72.4	29.2	87.4	65.6	70.9	14.9	37.9	70.6	67.8	19.9	21.1
	$-\mathbf{T}^{\text{out}}$	69.5	74.3	92.6	92.8	70.0	70.8	42.0	20.3	67.0	61.3	23.1	18.5

Table 6: Ablation for different template token representations without the answer label token representations. All values are presented as percentages. The results showing the greatest decrease during the ablation are underlined.

strate the importance of the template tokens. Notably, even without stopword or content tokens, the model can still acquire limited predictive ability.

5 Analyses of task-encoding tokens

In this section, we provide analyses of the tokens whose representations we believe mainly store information that affects the performance of a task drastically. We focus on the template tokens since, as evidenced by the findings in Section 4.2.1 and 4.2.2, their representations are the most important to maintaining task performance. Our analyses include the effects of different template tokens on the final performance and the distinguishing characteristics of these tokens.

5.1 Effects of different task-encoding tokens

In Section 4.2.1, we assume that the label token $\mathbf{D}^{\text{out}}$ is needed for ICL to achieve performance results on par with Standard ICL, as suggested by previous work (Wang et al., 2023). This circumstance raises the question: are the last prompt token and the label token $\mathbf{D}^{\text{out}}$ the only active components for affecting the task performance, or do other tokens in the template also matter? To answer this question, we examine the performance of the model when the test example is allowed to attend only to specific tokens of the ICL template.

5.1.1 Representation ablation with $\mathbf{D}^{\text{out}}$

First, we examine the performance effect of $\mathbf{T}^{\text{in}}$, $\mathbf{T}^{\text{out}}$, and the last prompt token ":" when $\mathbf{D}^{\text{out}}$ is always preserved, by removing or retaining their representations from the attention of the test example, similar to Section 4.2.1. The results of this experiment are presented in Table 5.

We find that for most of the cases, the indicator and the $\mathbf{D}^{\text{out}}$ provide the majority of the information needed for the final prediction, which is consistent with previous work (Todd et al., 2023; Wang et al., 2023). Moreover, we observe that other parts of the template text, especially $\mathbf{T}^{\text{out}}$, could also bring a large improvement in performance in most of the cases. This shows that all of the tokens in the template contribute to the final performance.

5.1.2 Representation ablation without $\mathbf{D}^{\text{out}}$

To further investigate the influence of the answer label token $\mathbf{D}^{\text{out}}$ while clarifying the effect of $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$, we include another set of experiments where all the $\mathbf{D}^{\text{out}}$ tokens are removed from the attention of test examples, shown in Table 6.

In this case, the performance becomes less stable, where adding back a template token (e.g., ":") does not always bring performance improvements. However, we find that in AGNews, TREC, and RTE datasets, models can still achieve relatively high performance while the model outperforms the

Settings	Notations	Examples
$\text{Random}_{\text{fixed}}$	$\mathbf{T}^{\text{in}}$	dsafjkldafdsajk: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}^{\text{out}}$	reqwiorewsdafjl: {$\mathbf{D}^{\text{out}}$}\n\n
Swap	$\mathbf{T}^{\text{in}}$	Answer: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}^{\text{out}}$	Article: {$\mathbf{D}^{\text{out}}$}\n\n
$\text{Random}_{\text{nonfixed}}$	$\mathbf{T}_1^{\text{in}}$	dsafjkldaasdfjkl: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_1^{\text{out}}$	xiadfjdsalgfweqrjl: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	ewqroudajfsdafq: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_2^{\text{out}}$	yufoufgaddavfdnsl: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	vcxnkfgahvczxkl: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_t^{\text{out}}$	dafhglajfdvcaol: {$\mathbf{D}^{\text{out}}$}\n\n

Table 7: An example of the ICL template with random strings used in AGNews.

Models	Settings	AGNews	SST2	TREC	DBPedia	RTE	CB	Avg.
OpenLlama 3B	Standard ICL	63.7	91.2	21.9	61.9	57.4	52.0	58.0
	Swap	64.4	86.8	<u>21.7</u>	58.7	60.6	54.6	57.8
	$\text{Random}_{\text{fixed}}$	<u>57.5</u>	<u>71.4</u>	32.4	<u>51.2</u>	<u>53.3</u>	<u>49.8</u>	<u>52.6</u>
Llama 7B	Standard ICL	82.4	94.3	63.5	68.7	68.6	71.3	74.8
	Swap	70.2	<u>11.4</u>	44.3	58.2	64.5	50.1	49.8
	$\text{Random}_{\text{fixed}}$	<u>19.5</u>	11.4	<u>13.2</u>	<u>7.4</u>	<u>19.7</u>	<u>21.7</u>	<u>15.5</u>
Llama 13B	Standard ICL	81.6	94.3	60.0	76.1	70.6	39.9	70.4
	Swap	81.5	<u>67.4</u>	36.4	75.9	69.1	52.1	63.7
	$\text{Random}_{\text{fixed}}$	<u>52.1</u>	76.8	<u>27.7</u>	<u>48.9</u>	<u>55.7</u>	<u>34.5</u>	<u>49.3</u>
Llama 30B	Standard ICL	85.0	96.5	68.1	78.4	78.5	83.3	81.6
	Swap	84.5	94.9	60.8	<u>75.5</u>	<u>68.0</u>	55.5	73.2
	$\text{Random}_{\text{fixed}}$	<u>78.7</u>	<u>92.5</u>	<u>52.2</u>	75.8	68.9	<u>41.1</u>	<u>68.2</u>

Table 8: Results validating the effect of lexical meanings of template tokens. The results showing the greatest decrease during the disruption are underlined.

zero-shot baselines in other datasets (except in DBPedia). These results show that representations of other template tokens may also be seen as information anchors whose representations aggregates and serve information to the final prediction of LLMs, broadening the conclusions of Wang et al. (2023) who claim that answer tokens serve as information anchors. This result also emphasizes that representations of tokens besides $\mathbf{D}^{\text{out}}$ and the last prompt token in the ICL template can also have a major effect on the final task performance.

5.2 Characteristics of task-encoding tokens

With the task-encoding tokens identified, we still lack insight into what distinguishes them as task-encoding from other tokens. Since they are the most important for LLMs' task performance, improper use of them (e.g., bad designs of the template) could cause instability in the final performance. Hence, to gain a deeper understanding of the presence of task-encoding tokens and avoid the potential instability issues, we further investigate the distinguishing characteristics of task-encoding tokens. We focus on the characteristics of **lexical meaning** referring to the task-related lexical meaning of a task-encoding token, **repetition** referring to the multiple appearances of the task-encoding tokens in the prompt, and **text formatting** referring to how task-encoding tokens format the ICL prompt into structured text.

We design several experiments to test whether these characteristics affect the presence of task-encoding tokens by disrupting each characteristic in the ICL prompts. A characteristic is related if there is a performance drop after the disruption. We examine each characteristic in the following sections by disrupting an ICL prompt with different kinds of random string templates. We use 5 different random string templates and average all the results for each setting. An example of the templates we used can be seen in Table 7. All the templates we used are attached to Appendix D.

5.2.1 Lexical meaning

A token might serve as a task-encoding token depending on its specific lexical meaning. One possible hypothesis is that if the token carries specific task-related meanings like "Article" and "Answer", it is more likely to serve as a task-encoding token.

To verify if lexical meanings could affect the formation of task-encoding tokens, we 1) Replace the tokens from $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$ with the same random strings across the different demonstrations (**Random**$_{\text{fixed}}$), thus completely disrupting the lexical characteristic of these tokens; 2) Swap $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$ (**Swap**), thus partially disrupting the lexical characteristic of these tokens. Shown in Table 8, we observe that for smaller models (OpenLlama 3B) disrupting the lexical meaning of tokens would slightly impact task performance. For larger models, the disruption causes more significant drops in performance. Specifically, Llama 7B is particularly sensitive to the lexical meaning of tokens and demonstrates poorer performance when semantics are disturbed via random strings or swapping. Therefore, the lexical meaning of tokens is likely to play a role in their task-encoding nature, especially in the case of larger models.

5.2.2 Repetition

The presence of task-encoding tokens could also be influenced by their repetition throughout the ICL prompt. Intuitively, via the attention mechanism, repetitive patterns are more likely to propagate information through the processing of text. Yan et al. (2023) propose self-reinforcement in in-context learning, also suggesting that repetition could be a significant factor in in-context learning.

We experiment with the repetition characteristic by comparing the results of the previously discussed **Random**$_{\text{fixed}}$ experiment with an experiment replacing each $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$ with different

Models	Settings	AGNews	SST2	TREC	DBPedia	RTE	CB	Avg.
OpenLlama 3B	$\text{Random}_{\text{fixed}}$	**57.5**	**71.4**	**32.4**	**51.2**	**53.3**	**49.8**	**52.6**
	$\text{Random}_{\text{nonfixed}}$	30.2	71.4	17.1	18.6	47.9	47.7	38.8
Llama 7B	$\text{Random}_{\text{fixed}}$	**19.5**	11.4	**13.2**	**7.4**	**19.7**	21.7	**15.5**
	$\text{Random}_{\text{nonfixed}}$	15.5	**11.6**	10.4	1.8	4.6	**25.6**	11.6
Llama 13B	$\text{Random}_{\text{fixed}}$	**52.1**	**76.8**	**27.7**	**48.9**	**55.7**	**34.5**	**49.3**
	$\text{Random}_{\text{nonfixed}}$	32.1	34.5	19.2	6.0	21.0	32.8	24.3
Llama 30B	$\text{Random}_{\text{fixed}}$	**78.7**	**92.5**	**52.2**	**75.8**	**68.9**	41.1	**68.2**
	$\text{Random}_{\text{nonfixed}}$	78.5	87.5	46.3	63.1	63.6	**46.1**	64.2

Table 9: Experimental results validating the effect of repetitive patterns. We bold the highest accuracy for each classification task and model size.

Models	Settings	AGNews	SST2	TREC	DBPedia	RTE	CB	Avg.
OpenLlama 3B	$\text{Random}_{\text{fixed}}$	47.5	51.8	32.6	19.4	51.8	42.4	40.9
	Zero-shot+TEMP	39.5	49.8	27.7	13.3	49.8	44.9	37.5
	Zero-shot + ":"	<u>31.5</u>	<u>35.9</u>	<u>23.8</u>	<u>8.0</u>	<u>35.9</u>	<u>33.8</u>	<u>28.2</u>
Llama 7B	$\text{Random}_{\text{fixed}}$	3.9	16.9	<u>3.5</u>	9.6	16.9	10.4	10.2
	Zero-shot+TEMP	<u>2.1</u>	15.5	7.6	3.7	15.5	<u>5.4</u>	8.3
	Zero-shot + ":"	3.6	<u>7.5</u>	14.6	<u>3.0</u>	<u>7.5</u>	6.8	<u>7.2</u>
Llama 13B	$\text{Random}_{\text{fixed}}$	46.1	47.5	<u>25.0</u>	50.8	47.5	21.4	39.7
	Zero-shot+TEMP	29.2	48.9	36.1	35.7	48.9	<u>14.0</u>	35.5
	Zero-shot + ":"	<u>14.3</u>	<u>22.4</u>	25.4	<u>22.5</u>	<u>22.4</u>	28.9	<u>22.7</u>
Llama 30B	$\text{Random}_{\text{fixed}}$	69.7	53.0	37.8	72.8	53.0	<u>37.6</u>	54.0
	Zero-shot+TEMP	61.2	56.3	41.1	69.2	56.3	43.0	54.5
	Zero-shot + ":"	<u>43.3</u>	<u>41.8</u>	<u>37.4</u>	<u>65.0</u>	<u>41.8</u>	39.5	<u>44.8</u>

Table 10: One-shot representation masking experiments conducted to verify if structural template formats could influence the effectiveness of the task-encoding tokens. $\mathbf{D}^{\text{out}}$ is preserved in all the settings. The results showing the greatest decrease during the ablation are underlined.

random strings (**Random**$_{\text{nonfixed}}$), thus breaking the repetition of template tokens present in ICL demonstrations. We see from Table 9 that without consistent repetition of the task-encoding tokens, the performance for most models decreases. This decrease in performance suggests that information necessary for maintaining the performance of the task may not have been properly accumulated and stored in the representations of the template tokens. These experiments demonstrate that repetitive patterns significantly influence the presence of task-encoding tokens.

5.2.3 Text formatting

Beyond lexical meaning and repetition, the occurrence of task-encoding tokens may also be influenced by the formatting of ICL prompts. Similar to our definition of template and stopword tokens, ICL prompts are often formatted with structural cues that assist the model in differentiating between elements with distinct roles, such as task inputs and target labels, within a demonstration. For instance, template tokens (i.e., $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$) delimit the presentation of demonstration examples and labels in ICL prompts. These structural cues are similar to those found in an LLM's pretraining data (e.g., column names in SQL tables). As a result, we suspect that pretraining on such data enables the structuring nature of the task-encoding tokens to be recognized, causing its representations to store higher-level information.

An intuitive method to verify the effect of the text formatting would be using the same random strings to replace $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$, making it harder for a model to parse the structure of the text. However, this would bring the factor of repetition into the process, potentially confounding the results. Hence, we instead conduct a one-shot **Random**$_{\text{fixed}}$ experiment. The one-shot **Random**$_{\text{fixed}}$ setting allows us to control both the characteristics of lexical meaning and repetition since the templates are made up of random strings and there is only one training demonstration. With these two characteristics controlled, we use the masking ablation method from Section 4.2.1 to confirm to what extent these random string tokens can function effectively as delimiters between inputs and outputs in ICL prompts. Specifically, we include results from the Zero-shot + ":", Zero-shot + TEMP scenarios, as well as the standard results of one-shot **Random**$_{\text{fixed}}$, for a more comprehensive analysis, shown in Table 10.

We observe that adding the attention to random template token representations in the one-shot setting often leads to performance increases while masking the attention to the template tokens and only attending to ":" $+\mathbf{D}^{\text{out}}$ leads to performance decreases. This indicates that the presence of these tokens is critical to maintaining task performance. With all other characteristics being controlled, this leads us to believe that the delimiting nature of template tokens is likely to be an important characteristic in their role as task-encoding tokens.

6 Conclusion

In this paper, we have provided a fine-grained characterization of task-encoding tokens. Through a series of comprehensive experiments, we have examined the roles of template tokens and stopword tokens within ICL as potential task-encoding tokens. Our findings suggest more subtlety exists in previous claims made about ICL, for example, that tokens other than label words could also provide valuable information affecting the performance. Overall, our results demonstrate that model performance depends directly on the presence of these tokens and that their lexical meaning, their repetition throughout the ICL prompt, and their formatting of ICL demonstrations are likely to play a role in how effectively they allow an LLM to recover the critical information to perform a task.

Limitations

In this paper, the token categorization is performed manually, leaving room for further refinement. While the results provide robust support to our categorization, the identification process itself lacks precision. For instance, stopwords may only represent a subset of all in-context task-encoding tokens. The manual nature of our categorization limits our ability to comprehensively track these tokens. Moreover, our experiments are limited to classification datasets, suggesting that our conclusions should be further validated for generation tasks. Additionally, our focus on task-encoding tokens, whose representations could impact task performance, may overlook other tokens responsible for other possible functions.

Ethics Statement

This work focuses on analyzing the working mechanisms of large language models and, as such, does not present any increased risks of harm beyond the existing norms of natural language processing or computational linguistics research. The associated risks include using a model trained on vast amounts of text, which may inadvertently contain biases. Another concern is the potential misuse of the model for generating misleading or harmful content. However, such a scenario is unlikely in our work, as we concentrate on classification tasks with fixed outputs.

References

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

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*.

Satwik Bhattamishra, Arkil Patel, Phil Blunsom, and Varun Kanade. 2023. Understanding in-context learning in transformers and llms by learning to learn discrete functions. *arXiv preprint arXiv:2310.03016*.

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*, 33:1877–1901.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

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.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19565–19594. PMLR.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, L. Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *Workshop on Knowledge Extraction and Integration for Deep Learning Architectures; Deep Learning Inside Out*.

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

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

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*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads.

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, Toronto, Canada, July 9-14, 2023*, pages 8298–8319. Association for Computational Linguistics.

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 *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *Conference on Empirical Methods in Natural Language Processing*.

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

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

Jianhao Yan, Jin Xu, Chiyu Song, Chenming Wu, Yafu Li, and Yue Zhang. 2023. Understanding in-context learning from repetitions.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3063–3079, Toronto, Canada. Association for Computational Linguistics.

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*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Datasets	Notations	Examples
AGNews	$\mathbf{I}$	Classify the news articles into the categories of World, Sports, Business, and Technology.\n\n
	$\mathbf{T}^{\text{in}}$	Article: $\{\mathbf{D}^{\text{in}}\}$\n
	$\mathbf{T}^{\text{out}}$	Answer: $\{\mathbf{D}^{\text{out}}\}$\n\n
SST2	$\mathbf{I}$	Classify the reviews into the categories of Positive and Negative.\n\n
	$\mathbf{T}^{\text{in}}$	Review: $\{\mathbf{D}^{\text{in}}\}$\n
	$\mathbf{T}^{\text{out}}$	Sentiment: $\{\mathbf{D}^{\text{out}}\}$\n\n
RTE	$\mathbf{I}$	Classify the entailment of the hypothesis and the premise into the categories of True and False.\n\n
	$\mathbf{T}^{\text{in}}$	Hypothesis: $\{\mathbf{D}^{\text{inA}}\}$\n Premise: $\{\mathbf{D}^{\text{inB}}\}$\n
	$\mathbf{T}^{\text{out}}$	Answer: $\{\mathbf{D}^{\text{out}}\}$\n\n
CB	$\mathbf{I}$	Classify the entailment of the hypothesis and the premise into the categories of true, neither and false.\n\n
	$\mathbf{T}^{\text{in}}$	Hypothesis: $\{\mathbf{D}^{\text{inA}}\}$\n Premise: $\{\mathbf{D}^{\text{inB}}\}$\n
	$\mathbf{T}^{\text{out}}$	Answer: $\{\mathbf{D}^{\text{out}}\}$\n\n
TREC	$\mathbf{I}$	Classify the questions based on whether their answer type is a Number, Location, Person, Description, Entity, or Abbreviation.\n\n
	$\mathbf{T}^{\text{in}}$	Question: $\{\mathbf{D}^{\text{in}}\}$\n
	$\mathbf{T}^{\text{out}}$	Answer Type: $\{\mathbf{D}^{\text{out}}\}$\n\n
DBPedia	$\mathbf{I}$	Classify the documents based on whether they are about a Company, School, Artist, Athlete, Politician, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, or Book.\n\n
	$\mathbf{T}^{\text{in}}$	Article: $\{\mathbf{D}^{\text{in}}\}$\n
	$\mathbf{T}^{\text{out}}$	Answer: $\{\mathbf{D}^{\text{out}}\}$\n\n

Table 11: An example of the ICL template used in all of our experiments.

Datasets	Stopwords
AGNews	"the", "into", "of", "and", "," , ".", "\n"
SST2	"the", "into", "of", "and", ".", "\n"
RTE	"the", "of", "into", "and", "into", ".", "\n"
CB	"the", "of", "and", "into", "," ,".", "\n"
TREC	"the", "based", "on", "whether", "their", "is", "a", ",", "or", ".", "\n"
DBPedia	"the", "based", "on", "whether", "they", "are", "about", "a", ",", "or", ".", "\n"

Table 12: The stopwords used in our experiments. Words rarely exist in the task demonstrations are omitted.

A In-context Learning Templates

In this section, we present all the in-context learning templates used in this paper. The examples are provided in Table 1. For the RTE and CB datasets, there are two distinct inputs in the demonstrations (i.e., the hypothesis and the premise), which we denote as $\mathbf{D}^{\text{inA}}$ and $\mathbf{D}^{\text{inB}}$, respectively.

B Stopword Tokens

For the results shown in the main paper, we used the stopword token list shown in Table 12. This list only includes the stopword tokens from the task instruction, aiming to minimize their presence. We made this choice under the assumption that task-affecting information should be stored densely in a few tokens. Hence, the number of tokens whose representations affect the final task performance significantly should be small.

Nevertheless, one might be curious about the results if we used a more complete stopword list. In this case, we utilize a more comprehensive stopword token list of NLTK[6] shown in Table 13 and conduct the representation-level ablation once more. The results are presented in Table 14. It can be observed that all the conclusions from Section 4.2.1 are still well established. A few results are different from Table 2 and Table 3 because we accidentally masked the representations of the "</s>" token in this set of experiments. We claim that this accidental masking does not impact the main findings of these experiments.

NLTK Stopwords List
"i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your", "yours", "yourself", "yourselves", "he", "him", "his", "himself", "she", "her", "hers", "herself", "it", "its", "itself", "they", "them", "their", "theirs", "themselves", "what", "which", "who", "whom", "this", "that", "these", "those", "am", "is", "are", "was", "were", "be", "been", "being", "have", "has", "had", "having", "do", "does", "did", "doing", "a", "an", "the", "and", "but", "if", "or", "because", "as", "until", "while", "of", "at", "by", "for", "with", "about", "against", "between", "into", "through", "during", "before", "after", "above", "below", "to", "from", "up", "down", "in", "out", "on", "off", "over", "under", "again", "further", "then", "once", "here", "there", "when", "where", "why", "how", "all", "any", "both", "each", "few", "more", "most", "other", "some", "such", "no", "nor", "not", "only", "own", "same", "so", "than", "too", "very", "s", "t", "can", "will", "just", "don", "should", "now", "", "", "#", "$", "%", "^", "&", "*", "(", ")", "-", "_", "+", "=", "[", "]", "{", "}", "|", "\", ";", "'", "'", "<", ">", ",", "." , "?", "/", "\n"

Table 13: The whole stopword list from NLTK. We add the punctuation tokens in this case.

C Significance test for the representation-level ablation

In this section, we report the p-value of all the pair-wise comparisons in the representation-level ablation experiments in Table 2 and Table 3. Results are shown in Table 15

D Template used for the random string experiments

In this section, we present all the in-context learning templates used for the random experiments in Section 5.2. In the $\textbf{Random}_{\text{fixed}}$ scenario, the $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$ are consistent across all demonstrations. For the $\textbf{Random}_{\text{nonfixed}}$ scenario, we employ different random string templates for each demonstration. We use 5 random string templates for each setting, shown in Table 16, Table 17, Table 18, Table 19, and Table 20. The results in Section 5.2 are averaged over the results with all the different random string templates.

[6] https://gist.github.com/sebleier/554280

Models	Setting	AGNews	SST2	TREC	DBPedia	RTE	CB
OpenLlama 3B	Zero-shot	22.0	20.0	23.6	5.4	44.4	1.8
	+ CONT	26.2	52.1	30.1	7.4	51.9	37.9
	+ STOP	38.0	85.1	**31.6**	54.6	**58.8**	**55.7**
	+ TEMP	**56.5**	**86.7**	27.1	**62.2**	56.4	52.3
	Standard ICL	63.7	91.2	21.9	61.9	57.4	52.0
	- TEMP	42.1	87.2	25.9	56.3	**58.3**	**57.4**
	- CONT	57.1	88.4	**27.1**	**62.6**	56.8	52.4
	- STOP	**61.6**	**90.7**	24.8	62.2	56.7	51.9
Llama 7B	Zero-shot	25.0	29.2	41.4	0.0	54.2	3.6
	+ CONT	32.4	57.9	42.5	12.5	55.5	46.1
	+ STOP	59.9	85.9	51.7	28.9	56.0	52.7
	+ TEMP	**70.8**	**90.2**	**58.4**	**66.2**	**66.3**	**73.5**
	Standard ICL	82.4	94.3	63.5	68.7	68.6	71.3
	- TEMP	64.7	84.1	54.0	56.7	56.1	48.2
	- CONT	75.4	93.8	59.8	67.5	66.8	**74.8**
	- STOP	**81.4**	**94.2**	**60.5**	**67.9**	**67.6**	72.1
Llama 13B	Zero-shot	59.0	18.0	37.0	0.0	0.0	0.0
	+ CONT	30.6	52.4	43.8	13.0	60.2	45.5
	+ STOP	72.7	78.7	49.2	27.4	58.5	27.1
	+ TEMP	**78.5**	**92.3**	**59.0**	**74.2**	**67.4**	**52.3**
	Standard ICL	81.6	94.3	60.0	76.1	70.6	39.9
	- TEMP	71.7	80.1	56.2	8.7	56.5	29.3
	- CONT	**79.3**	93.4	**60.1**	**74.1**	68.4	**47.6**
	- STOP	79.2	**94.1**	59.3	73.8	**68.9**	44.6
Llama 30B	Zero-shot	70.2	88.6	60.6	30.2	58.1	19.6
	+ CONT	27.8	61.7	61.9	10.8	64.2	68.1
	+ STOP	74.7	93.6	**66.9**	70.8	69.1	63.8
	+ TEMP	**80.6**	**95.2**	63.1	**71.9**	**78.7**	**84.0**
	Standard ICL	85.0	96.5	68.1	78.4	78.5	83.3
	- TEMP	79.5	93.8	58.5	62.8	68.0	68.0
	- CONT	82.7	95.9	**62.9**	**74.1**	**79.6**	**83.1**
	- STOP	**84.4**	**96.1**	61.8	72.8	79.4	82.1

Table 14: The accuracy results of the representation level ablation study where we use the more complete stopword token list of NLTK. All values are presented as percentages. The best results presented by the number of ablated token types are in bold.

Models	Settings	AGNews		SST2		TREC		DBPedia		RTE		CB	
		P-value	p < 0.05	P-value	p < 0.05	P-value	p < 0.05	P-value	p < 0.05	P-value	p < 0.05	P-value	p < 0.05
OpenLlama 3B	temp <-> cont	0.0000581	T	0.0000000	T	0.1952165	F	0.0000000	T	0.0042427	T	0.0027293	T
	temp <-> stop	0.0001605	T	0.0571278	F	0.0242797	T	0.0000663	T	0.0319815	T	0.0985942	F
	cont <-> stop	0.0023957	T	0.0000001	T	0.1940792	F	0.0000000	T	0.0000073	T	0.0000698	T
	temp_cont <-> cont_stop	0.0000065	T	0.0385760	T	0.3206221	F	0.0000000	T	0.1237570	F	0.0544049	F
	temp_stop <-> cont_stop	0.0001166	T	0.4514005	F	0.2549225	F	0.0000001	T	0.0545474	F	0.0534521	F
	temp_cont <-> temp_stop	0.0000507	T	0.0096752	T	0.0005775	T	0.0000000	T	0.1208140	F	0.3193696	F
Llama 7B	temp <-> cont	0.0000000	T	0.0000001	T	0.0000020	T	0.0000001	T	0.0000004	T	0.0000001	T
	temp <-> stop	0.0000083	T	0.0101283	T	0.0002883	T	0.1438193	F	0.0000000	T	0.0000031	T
	cont <-> stop	0.0000060	T	0.0000001	T	0.0019529	T	0.0000001	T	0.3392237	F	0.0016487	T
	temp_cont <-> cont_stop	0.0000115	T	0.0030175	T	0.0005950	T	0.0000000	T	0.0000000	T	0.0001649	T
	temp_stop <-> cont_stop	0.0002004	T	0.0227328	T	0.0015468	T	0.0000000	T	0.0000001	T	0.0000094	T
	temp_cont <-> temp_stop	0.0086396	T	0.0003632	T	0.0089932	T	0.0000007	T	0.1637608	F	0.1553081	F
Llama 13B	temp <-> cont	0.0000000	T	0.0000000	T	0.0000082	T	0.0000001	T	0.0006445	T	0.1060226	F
	temp <-> stop	0.0003841	T	0.0000012	T	0.0034370	T	0.0002018	T	0.0000000	T	0.0010178	T
	cont <-> stop	0.0000000	T	0.0000202	T	0.0002820	T	0.0000000	T	0.0098209	T	0.0022848	T
	temp_cont <-> cont_stop	0.0010838	T	0.0000730	T	0.0004557	T	0.0000048	T	0.0000001	T	0.0002364	T
	temp_stop <-> cont_stop	0.0007763	T	0.0000310	T	0.0016544	T	0.0000000	T	0.0000000	T	0.0000888	T
	temp_cont <-> temp_stop	0.4411518	F	0.1158895	F	0.3323328	F	0.0000000	T	0.3148253	F	0.0144961	T
Llama 30B	temp <-> cont	0.0000000	T	0.0000003	T	0.1534319	F	0.0000000	T	0.0000000	T	0.0002244	T
	temp <-> stop	0.0007359	T	0.0048547	T	0.1797405	F	0.0000023	T	0.0000002	T	0.0008789	T
	cont <-> stop	0.0000000	T	0.0000003	T	0.0204911	T	0.0000000	T	0.0002626	T	0.4319440	F
	temp_cont <-> cont_stop	0.0001365	T	0.0788756	F	0.0032131	T	0.0000000	T	0.0000098	T	0.0003242	T
	temp_stop <-> cont_stop	0.0006045	T	0.0609501	F	0.0165374	T	0.0000011	T	0.0000009	T	0.0003821	T
	temp_cont <-> temp_stop	0.0012936	T	0.3583931	F	0.1415489	F	0.0001034	T	0.0009055	T	0.3979685	F

Table 15: The pair-wise t-test significance results. "T" means True while "F" means False. In this table, "temp" means only keeping temp, which is zero-shot + TEMP. "temp_cont" means ablating the stopword token representations, which is Standard ICL − STOP.

Datasets	Notations	Examples
		$\text{Random}_{\text{fixed}}$
CB & RTE	$\mathbf{T}^{\text{in}}$	fdafdasjklfdadf: {$\mathbf{D}^{\text{inA}}$}\n zcxvnmxcjkfdas: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}^{\text{out}}$	reqwiorewsdafjl: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}^{\text{in}}$	dsafjkldafdsajk: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}^{\text{out}}$	reqwiorewsdafjl: {$\mathbf{D}^{\text{out}}$}\n\n
		$\text{Random}_{\text{nonfixed}}$
CB & RTE	$\mathbf{T}_1^{\text{in}}$	fdafdasjklfdadf: {$\mathbf{D}^{\text{inA}}$}\n zcxvnmxcjkfdas: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_1^{\text{out}}$	xiadfjdsalgfweqrjl: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	gfhdajkgfhdasfj: {$\mathbf{D}^{\text{inA}}$}\n cvxhlkdadsajfk: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_2^{\text{out}}$	yufoufgaddavfdnsl: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	rrqetrizxcsdafq: {$\mathbf{D}^{\text{inA}}$}\n vncmxasdgfadsl: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_3^{\text{out}}$	afdgvcxjlzxnvxzla: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	mvfvxadfawewqro: {$\mathbf{D}^{\text{inA}}$}\n lkajsdfopsadfp: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_4^{\text{out}}$	fgsgfskjvcdafds: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	sdsajfjdsaczvvv: {$\mathbf{D}^{\text{inA}}$}\n hkljfdiabasdfj: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_t^{\text{out}}$	dafhglajfdvcaol: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}_1^{\text{in}}$	dsafjkldaasdfjkl: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_1^{\text{out}}$	xiadfjdsalgfweqrjl: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	ewqroudajfsdafq: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_2^{\text{out}}$	yufoufgaddavfdnsl: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	eqdashcxzlreqguio: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_3^{\text{out}}$	afdgvcxjlzxnvxzla: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	cxzvadeqrczxdsa: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_4^{\text{out}}$	fgsgfskjvcdafds: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	vcxnkfgahvczxkl: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_t^{\text{out}}$	dafhglajfdvcaol: {$\mathbf{D}^{\text{out}}$}\n\n
		Swap
CB & RTE	$\mathbf{T}^{\text{in}}$	Answer: {$\mathbf{D}^{\text{inA}}$}\n Hypothesis: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}^{\text{out}}$	Premise: {$\mathbf{D}^{\text{out}}$}\n\n

Table 16: Example #1 of the ICL template used in all of our random experiments.

Datasets	Notations	Examples
		$\text{Random}_{\text{fixed}}$
CB & RTE	$\mathbf{T}^{\text{in}}$	eszycidpyopumzg: {$\mathbf{D}^{\text{inA}}$}\n sgrlobvqgthjpwz: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}^{\text{out}}$	zbyygcrmzfnxlsu: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}^{\text{in}}$	eszycidpyopumzg: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}^{\text{out}}$	zbyygcrmzfnxlsu: {$\mathbf{D}^{\text{out}}$}\n\n
		$\text{Random}_{\text{nonfixed}}$
CB & RTE	$\mathbf{T}_1^{\text{in}}$	eszycidpyopumzg: {$\mathbf{D}^{\text{inA}}$}\n sgrlobvqgthjpwz: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_1^{\text{out}}$	zbyygcrmzfnxlsu: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	cwknayjkywwvpty: {$\mathbf{D}^{\text{inA}}$}\n muzprouhvtidhqe: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_2^{\text{out}}$	lnlgffeurextxme: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	pdnizszmpkfjzvo: {$\mathbf{D}^{\text{inA}}$}\n ujulhuzkkqlfwkl: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_3^{\text{out}}$	gflemobnbdjngii: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	gvsrxbdoxmpablo: {$\mathbf{D}^{\text{inA}}$}\n ujulhuzkkqlfwkl: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_4^{\text{out}}$	gflemobnbdjngii: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	gvsrxbdoxmpablo: {$\mathbf{D}^{\text{inA}}$}\n xipddzrshrhprrb: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_t^{\text{out}}$	npkxdzaipdpkbrs: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}_1^{\text{in}}$	eszycidpyopumzg: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_1^{\text{out}}$	zbyygcrmzfnxlsu: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	cwknayjkywwvpty: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_2^{\text{out}}$	lnlgffeurextxme: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	pdnizszmpkfjzvo: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_3^{\text{out}}$	gflemobnbdjngii: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	gvsrxbdoxmpablo: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_4^{\text{out}}$	npkxdzaipdpkbrs: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	dgldzypdptzcekq: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_t^{\text{out}}$	xobxfpnzsfzipol: {$\mathbf{D}^{\text{out}}$}\n\n

Table 17: Example #2 of the ICL template used in all of our random experiments.

Datasets	Notations	Examples
		$\text{Random}_{\text{fixed}}$
CB & RTE	$\mathbf{T}^{\text{in}}$	bcclfxzvjitgtbs: {$\mathbf{D}^{\text{inA}}$}\n evtlfrwvtfmjtns: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}^{\text{out}}$	qtnheeipeustcwn: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}^{\text{in}}$	bcclfxzvjitgtbs: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}^{\text{out}}$	qtnheeipeustcwn: {$\mathbf{D}^{\text{out}}$}\n\n
		$\text{Random}_{\text{nonfixed}}$
CB & RTE	$\mathbf{T}_1^{\text{in}}$	bcclfxzvjitgtbs: {$\mathbf{D}^{\text{inA}}$}\n evtlfrwvtfmjtns: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_1^{\text{out}}$	qtnheeipeustcwn: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	ymupnggvmbnoobq: {$\mathbf{D}^{\text{inA}}$}\n rrrnpgbmmgqymky: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_2^{\text{out}}$	xleuwtyqnnfgzjx: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	pdnizszmpkfjzvo: {$\mathbf{D}^{\text{inA}}$}\n qlfulxzxwfnwbum: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_3^{\text{out}}$	jpnvgbnjjlawqfo: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	mfkqxjoxtpmzdrs: {$\mathbf{D}^{\text{inA}}$}\n yyzdeayigwzjosn: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_4^{\text{out}}$	pdsqooqrhvydszp: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	rerlkjfvlvyzpmc: {$\mathbf{D}^{\text{inA}}$}\n iuumpcsevursgqe: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_t^{\text{out}}$	tuaqblysbipihsv: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}_1^{\text{in}}$	bcclfxzvjitgtbs: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_1^{\text{out}}$	qtnheeipeustcwn: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	ymupnggvmbnoobq: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_2^{\text{out}}$	xleuwtyqnnfgzjx: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	pdwunmjronsmuvu: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_3^{\text{out}}$	jpnvgbnjjlawqfo: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	mfkqxjoxtpmzdrs: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_4^{\text{out}}$	pdsqooqrhvydszp: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	rerlkjfvlvyzpmc: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_t^{\text{out}}$	tuaqblysbipihsv: {$\mathbf{D}^{\text{out}}$}\n\n

Table 18: Example #3 of the ICL template used in all of our random experiments.

Datasets	Notations	Examples
		$\text{Random}_{\text{fixed}}$
CB & RTE	$\mathbf{T}^{\text{in}}$	hsreltpusctapir: {$\mathbf{D}^{\text{inA}}$}\n woxwxgwctxdumok: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}^{\text{out}}$	prlhxooromawkcp: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}^{\text{in}}$	hsreltpusctapir: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}^{\text{out}}$	prlhxooromawkcp: {$\mathbf{D}^{\text{out}}$}\n\n
		$\text{Random}_{\text{nonfixed}}$
CB & RTE	$\mathbf{T}_1^{\text{in}}$	hsreltpusctapir: {$\mathbf{D}^{\text{inA}}$}\n woxwxgwctxdumok: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_1^{\text{out}}$	prlhxooromawkcp: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	cbptgaytithxayh: {$\mathbf{D}^{\text{inA}}$}\n bhxgcstisqmfnpz: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_2^{\text{out}}$	mvpvoeuvgczfemz: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	htkbzfizxwpeqrm: {$\mathbf{D}^{\text{inA}}$}\n felxgmjeuabznwd: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_3^{\text{out}}$	glfwilpyrwnsujg: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	frskoasvqybxcob: {$\mathbf{D}^{\text{inA}}$}\n bkepuhnckdaqmhx: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_4^{\text{out}}$	ljttiywadveyzah: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	dfpqndhxehhtser: {$\mathbf{D}^{\text{inA}}$}\n bvucjofrggmmcsh: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_t^{\text{out}}$	koesxfmmjjjjvmp: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}_1^{\text{in}}$	hsreltpusctapir: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_1^{\text{out}}$	prlhxooromawkcp: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	cbptgaytithxayh: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_2^{\text{out}}$	mvpvoeuvgczfemz: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	htkbzfizxwpeqrm: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_3^{\text{out}}$	glfwilpyrwnsujg: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	frskoasvqybxcob: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_4^{\text{out}}$	ljttiywadveyzah: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	dfpqndhxehhtser: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_t^{\text{out}}$	koesxfmmjjjjvmp: {$\mathbf{D}^{\text{out}}$}\n\n

Table 19: Example #4 of the ICL template used in all of our random experiments.

Datasets	Notations	Examples
		$\text{Random}_{\text{fixed}}$
CB & RTE	$\mathbf{T}^{\text{in}}$	hjdxmpeccamrjzy: {$\mathbf{D}^{\text{inA}}$}\n agxyhmkawezafde: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}^{\text{out}}$	ndxtrwvqugyygku: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}^{\text{in}}$	hjdxmpeccamrjzy: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}^{\text{out}}$	ndxtrwvqugyygku: {$\mathbf{D}^{\text{out}}$}\n\n
		$\text{Random}_{\text{nonfixed}}$
CB & RTE	$\mathbf{T}_1^{\text{in}}$	hjdxmpeccamrjzy: {$\mathbf{D}^{\text{inA}}$}\n agxyhmkawezafde: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_1^{\text{out}}$	ndxtrwvqugyygku: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	mcsgenpkdwsfknc: {$\mathbf{D}^{\text{inA}}$}\n egnqobhzvxjhsxh: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_2^{\text{out}}$	ijkdikcmiskofsg: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	cmaqcvtdkemdauv: {$\mathbf{D}^{\text{inA}}$}\n oslzaygbefxlwqt: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_3^{\text{out}}$	mumrjhndwmidwmj: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	cgmylzvslxmojvq: {$\mathbf{D}^{\text{inA}}$}\n tlwxsjmnfkolffl: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_4^{\text{out}}$	mitaowjyibjwwol: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	pvockachyflybtk: {$\mathbf{D}^{\text{inA}}$}\n wtjqmtwxbnpyqbp: {$\mathbf{D}^{\text{inB}}$}\n
	$\mathbf{T}_t^{\text{out}}$	ydediotfezhfnbx: {$\mathbf{D}^{\text{out}}$}\n\n
Other tasks	$\mathbf{T}_1^{\text{in}}$	hsreltpusctapir: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_1^{\text{out}}$	prlhxooromawkcp: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_2^{\text{in}}$	cbptgaytithxayh: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_2^{\text{out}}$	mvpvoeuvgczfemz: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_3^{\text{in}}$	htkbzfizxwpeqrm: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_3^{\text{out}}$	glfwilpyrwnsujg: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_4^{\text{in}}$	frskoasvqybxcob: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_4^{\text{out}}$	ljttiywadveyzah: {$\mathbf{D}^{\text{out}}$}\n\n
	$\mathbf{T}_t^{\text{in}}$	dfpqndhxehhtser: {$\mathbf{D}^{\text{in}}$}\n
	$\mathbf{T}_t^{\text{out}}$	koesxfmmjjjjvmp: {$\mathbf{D}^{\text{out}}$}\n\n

Table 20: Example #5 of the ICL template used in all of our random experiments.