Inference and Verbalization Functions During In-Context Learning

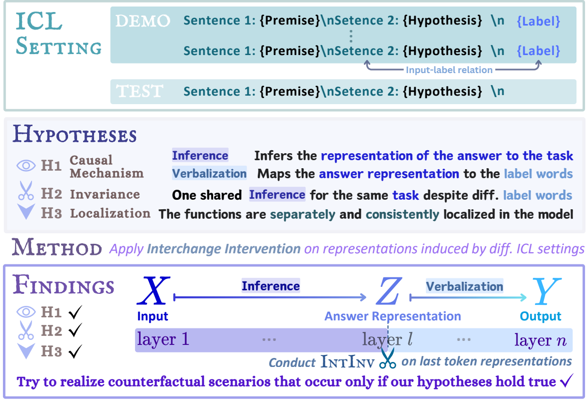
Junyi Tao[*]
Stanford University
junyitao@stanford.edu

Xiaoyin Chen[*]
Mila, University of Montreal
xiaoyin.chen@mila.quebec

Nelson F. Liu
Stanford University
nfliu@cs.stanford.edu

Abstract

Large language models (LMs) are capable of in-context learning from a few demonstrations (example-label pairs) to solve new tasks during inference. Despite the intuitive importance of high-quality demonstrations, previous work has observed that, in some settings, ICL performance is minimally affected by irrelevant labels (Min et al., 2022). We hypothesize that LMs perform ICL with irrelevant labels via two sequential processes: an `inference` function that solves the task, followed by a `verbalization` function that maps the inferred answer to the label space. Importantly, we hypothesize that the `inference` function is invariant to remappings of the label space (e.g., "true"/"false" to "cat"/"dog"), enabling LMs to share the same `inference` function across settings with different label words. We empirically validate this hypothesis with controlled layer-wise interchange intervention experiments. Our findings confirm the hypotheses on multiple datasets and tasks (natural language inference, sentiment analysis, and topic classification) and further suggest that the two functions can be localized in specific layers across various open-sourced models, including GEMMA-7B, MISTRAL-7B-V0.3, GEMMA-2-27B, and LLAMA-3.1-70B.



Sentence 1: {Premise}\nSetence 2: {Hypothesis} \n {Label}
⋮
Sentence 1: {Premise}\nSetence 2: {Hypothesis} \n {Label}

Sentence 1: {Premise}\nSetence 2: {Hypothesis} \n

Inference Infers the representation of the answer to the task
Verbalization Maps the answer representation to the label words
One shared Inference for the same task despite diff. label words
The functions are separately and consistently localized in the model

Apply Interchange Intervention on representations induced by diff. ICL settings

Conduct INTINV on last token representations
Try to realize counterfactual scenarios that occur only if our hypotheses hold true ✓

Figure 1: **Summary of our hypotheses, method, and findings.** We hypothesize that LMs perform ICL via two sequential functions: (1) an `inference` function that uses an input ICL context and returns a representation of the answer and 2) a `verbalization` function that maps the aforementioned answer representation to the output label space specified by the demonstrations. Furthermore, the `inference` function is invariant to remappings of the output label space. We empirically test our hypotheses by conducting interchange intervention experiments (i.e., fixing one of the functions while modifying the other).

1 Introduction

Large language models (LMs) are capable of in-context learning (ICL)—the ability to solve novel tasks from solely a handful of demonstration examples provided in-context during inference (Brown et al., 2020). Previous work has found that, in certain settings, ICL performance is minimally affected by using demonstrations with irrelevant label words (Min et al., 2022). How do LMs manage to perform in-context learning with irrelevant and even misleading label words? This research seeks to offer a *causal* explanation for such model behaviors, going beyond mere summaries of input-output patterns. We hypothesize that the model develops and utilizes certain abstractions to consistently work well on ICL tasks despite remappings of the label spaces. Specifically, we propose and test the following hypotheses:

H_1 **Causal mechanism:** LMs sequentially apply two functions when performing ICL: First, an `inference` function that constructs a representation of the answer to the ICL input; and second, a `verbalization` function that maps this answer representation to the output label space specified by the demonstrations.

H_2 **Invariant to label remapping:** The `inference` function is invariant to remappings of the output label space (e.g., "true"/"false" to "cat"/"dog").

[*]Equal contribution.

Findings of the Association for Computational Linguistics: EACL 2024, pages 16394–16421
November 12-16, 2024 ©2024 Association for Computational Linguistics

To learn about the model's *low-level* implementations of these *high-level* abstractions (distributed across neural activations), we train probes to detect representations of label words. Its results suggest that the two functions are *separately* located in the *similar* layers across different runs (Appendix A). This motivates an additional hypothesis:

H_3 **Separate and consistent localization:** The two functions are located (largely) *separately* in different sets of sequential layers, and the locations of these layers are *consistent* across settings with different remapped label spaces.

This is a working hypothesis that serves as a "ladder", providing protocols for us to intervene on the high-level abstractions through modifying the low-level internal representations of the model.

Methods. Our hypotheses are validated if we can define and realize a counterfactual scenario that occurs *only if* our hypotheses are true. For example, let's imagine a scenario where we "concatenate" an `inference` function and a `verbalization` function, with one induced from a run where the label words are "true"/"false", and another from a run where the label words are "cat"/"dog". Suppose the output of the former run is "true", and the latter run is "false". This concatenation can make the model generate the hypothetical `counterfactual output` "cat" *only if* all of our hypotheses hold true (see more discussion on this in Section 2.1). We realize this hypothetical concatenation by swapping the last token representations created for one input into the model which is processing another input with remapped label words. This operation is known as interchange intervention or activation patching (Geiger et al., 2020; Vig et al., 2020; Finlayson et al., 2021; Meng et al., 2022).

Findings. By experimenting with remapped label spaces that will induce changes in the `verbalization` function (illustrated in Figure 2), we find that our intervention on certain layers *can* achieve a "concatenated' model that produces the `counterfactual output`. This validates H_2. We also conduct a complementary experiment to induce changes in the `inference` function with constructed alternative tasks on MultiNLI, where the example-label relations are changed and the input and label spaces are fixed (illustrated in Figure 7). This is aimed for further validating the localization of the `inference` function. The results from these two experiments together validate H_1. Additionally, we observe that the two functions are *separately* and *consistently* located in similar layers of each model, across various tasks and datasets, including MultiNLI, RTE, ANLI, IMDb, and AGNews. This is aligned with H_3. These findings apply to models of various sizes and families, including GEMMA-7B, MISTRAL-7B-V0.3, GEMMA-2-27B, and LLAMA-3.1-70B.[1]

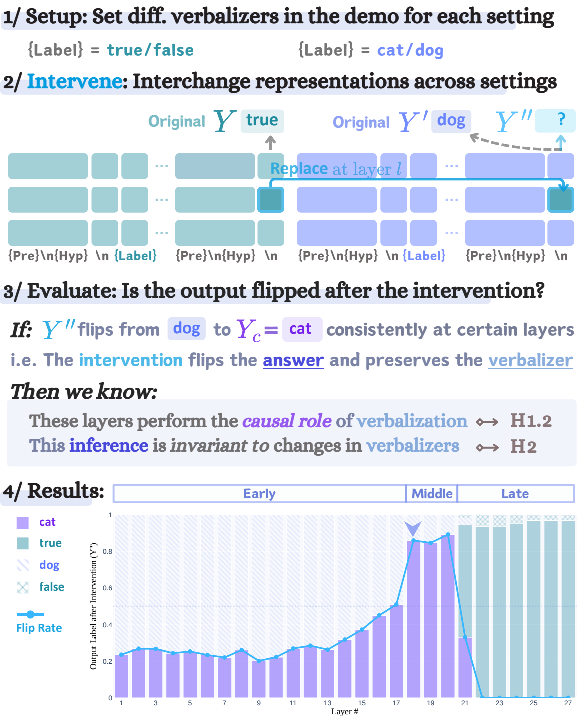


Figure 2: **Intervention experiments with remapped label spaces.** (1, top): First, we induce changes in the `verbalization` function by remapping the output label space in the demonstrations (e.g., "true"/"false" to "cat"/"dog"). (2): Then, we intervene on the `verbalization` function by replacing the representation of the intervened model (prompted with the remapped label space "cat"/"dog") at layer l with the representation from the source model (prompted with the default label space "true"/"false") at the same layer. (3): We evaluate the intervention effects by measuring how often the intervened output Y'' matches the hypothetical `counterfactual output` Y_c. (4, bottom): Distribution of outputs predicted by GEMMA-7B on MultiNLI when intervening with the remapped label space "true"/"false" → "cat"/"dog". For clarity, we visualize the subset examples where the hypothetical `counterfactual output` is "cat". We observe that the rate at which the intervened output matches the `counterfactual output` peaks around layers 18-20, localizing the `verbalization` function and suggesting the `inference` function is invariant to label remapping.

[1] Our code is publicly available at https://github.com/JunyiTao/infer-then-verbalize-during-icl.

2 Methods

We aim for an explanation that reveals the causal structure employed by the model. Successful causal explanation is marked by its ability to allow us to make counterfactual predictions, that is, answering the "what-if-things-had-been-different questions" should we perform certain interventions (Woodward, 2003). Typically, direct observation of outcomes from distinct interventions on the same unit is not possible (Holland, 1986). But, it is not an issue for us, because we can let the model generate outputs under any conceivable intervention on the units. Moreover, we can make counterfactual statements with mere interventional information with deterministic neural network models (Bareinboim et al., 2022), which is also typically not possible (Pearl and Mackenzie, 2018). These enable a concrete operationalization of our hypothesis testing.

2.1 Testing Hypotheses with Counterfactual Scenarios

Framework for testing our hypotheses. Our approach is to (1) define a counterfactual scenario aligned with our hypotheses, wherein certain model behaviors emerge *only if* our hypotheses hold true, (2) employ appropriate intervention methods to realize this scenario, and (3) conduct the intervention-based experiments and interpret their results.

Design a counterfactual scenario. Imagine a scenario corroborating our hypotheses, where we "concatenate" the `verbalization` function and `inference` function induced from different model runs with distinct label words. Assuming H_2 **(invariant to label remapping)** is true, this concatenation should function effectively, enabling the `verbalization` function to successfully decode the answer representation produced by the `inference` function and correctly verbalize it, leading to a hypothetical `counterfactual output` where the answer from the first run appears in the label words of the second run. Their functionalities must remain unaffected by the concatenation for `inference` function to transfer the answer representation to `verbalization` function, collectively producing the counterfactual output.

Operationalize the setting. With our "ladder" hypotheses, we can modify the model's *low-level* internal representation as a protocol for intervening on the *high-level* `inference` function and `verbalization` function. Specifically, the two functions being localized *separately* ($H_{3.1}$ **separate localization**) enables us to isolate their causal effects, and $H_{3.2}$ **(consistent localization)** makes it possible to conduct the intervention with the counterfactual value of the representation—the value it would take under alternative input scenarios *in the same position*—rather than a value the neuron might never *actually* assume. With these, our objective becomes to find a layer (or sequence of layers), l_{mid}, such that taking the last token representation from one run and plugging it into another would change the model's output in a way that reflects the hypothetical counterfactual condition.

Interpreting results. Our hypotheses are supported if the model consistently generates a sufficient proportion of `counterfactual outputs`. Additionally, this will also provide information about the specific (though not exact) locations of the two functions within the model—l_{mid} should occur after the layer where the answer representation is sufficiently developed. This, however, does not tell us the exact starts and ends of the two functions. It is possible that only a *subset* of layers before l_{mid} is actively involved in implementing the `inference` function and after l_{mid} is actively involved in implementing the `verbalization` function, while some others may be redundant.

Our hypotheses can be weakened by the absence of evidence, i.e., no specific layer found to allow effective concatenation of the functions; and they can falsified if our interventions yield consistent results that contradict our predictions, which may indicate that the model systematically employs mechanisms different than those hypothesized.

2.2 Using Interchange Interventions to Realize Counterfactual Scenarios

Interchange intervention. We will first introduce the formulation of interchange intervention (Geiger et al., 2020, 2021, 2022, 2024a,b; Huang et al., 2024) and then apply it to our context. Consider a model $\mathcal{M}$ that takes an input string $\mathbf{x}$ and generates an output string $\mathbf{y}$. We denote the entire set of internal representations of the model $\mathcal{M}$ created during this inference as $\mathcal{M}(\mathbf{x})$ and the predicted token $\mathbf{y} = \tau(\mathcal{M}(\mathbf{x}))$.

Let's conduct interchange intervention on a set of intermediate representations Z of the model $\mathcal{M}$ by replacing them with the value of $\mathbf{z}$ and denote the post-intervention model as $\mathcal{M}_{\mathbf{Z}\leftarrow\mathbf{z}}$. The difference between $\tau(\mathcal{M}(\mathbf{x}))$ and $\tau(\mathcal{M}_{\mathbf{Z}\leftarrow\mathbf{z}}(\mathbf{x}))$ manifests

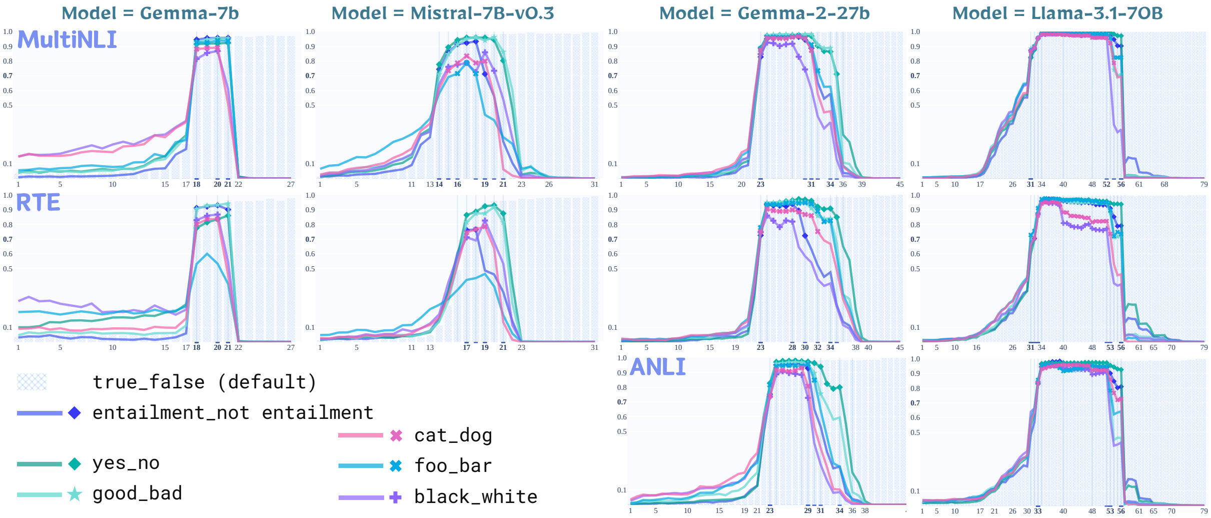


Figure 3: **NLI experiments with remapped label spaces.** The y-axis is the rate of the output label flipped into the hypothetical `counterfactual output` label ("flip rate") and the x-axis is the intervened layer. Each curve represents the flip rates observed when the intervened model is prompted with the corresponding label space. The background bars represent the flip rates of the *default* ← *default baseline*, where the intervened model is also prompted with the default label space ("true"/"false"). Due to their unsatisfactory ICL performance, we do not conduct intervention experiments with the two 7-billion models on ANLI (see more discussion on this in Section 2.2).

the causal contribution of $\mathbf{Z}$ to the model behavior. To get the intended value of $\mathbf{z}$, we use `GetVals` to retrieve the representation values that the variables $\mathbf{Z}$ *would have* taken on if $\mathcal{M}$ processes another input $\mathbf{x}'$, denoted as $\texttt{GetVals}(\mathcal{M}, \mathbf{x}', \mathbf{Z})$. With the two operations, we obtain the post-intervention version of the model, $\mathcal{M}_{\mathbf{Z} \leftarrow \texttt{GetVals}(\mathcal{M}, \mathbf{x}', \mathbf{Z})}$, where the values of $\mathbf{Z}$ set as those obtained by processing $\mathbf{x}'$. We refer to $\mathcal{M}(\mathbf{x}')$ as the **source model** since it provides the desired values of $\mathbf{z}$ and $\mathcal{M}(\mathbf{x})$ as the **intervened model** whose internal representations are to be modified. Thus, we get the output of the intervened model after the interchange intervention: $\texttt{IntInv}(\mathcal{M}, \mathbf{x}, \mathbf{x}', \mathbf{Z}) := \tau\big(\mathcal{M}_{\mathbf{Z} \leftarrow \texttt{GetVals}(\mathcal{M}, \mathbf{x}', \mathbf{Z})}(\mathbf{x})\big)$.

In our study, we want the intervention to reroute the computational graph of the source model to the intervened model during the last l layers, to achieve a "concatenation" of functions from different runs (illustrated in Figure 2). We achieve this by setting the variables $\mathbf{Z}$ to be the representations of the *last token* of the input context. The causal attention mask of the decoder-only architecture of the LMs we study ensures that no intermediate representation of a token is influenced by its successors, allowing interventions on the last token without altering the computational graph of preceding tokens. Thus, the intervened model will apply the same function during the last l layers for both its original representation and the plugged-in representation.

Formulate counterfactual outputs. An ICL prompt $\mathbf{x}$ is composed of (1) a set of demonstrations $\{x_{demo}^{(j)}, y_{demo}^{(j)}\}_{j=1}^{k}$, where the example-label relations are intended to determine the task to be inferred, such as NLI, and $y_{demo}^{(j)}$ in the output label space $\mathcal{Y}$, such as $\{$`"true"`,`"false"`$\}$ where each element corresponds to the positive and negative label respectively, (2) a test example x_{test}, and (3) a prompt template (see Figure 1, Top).

In Section 2.1, we imagine a counterfactual scenario where we concatenate two functions, each induced by $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$. The hypothetical `counterfactual output` here reflects that the answer comes from the model $\mathcal{M}$ inferring $\mathbf{x}'$ and appears in the label words specified in $\mathbf{x}$. Note that when experimenting with **remapped label spaces** (Figure 2 and Section 3.1), we take the *upstream* `inference` function from the source model (with default label words) to pass information to the *downstream* `verbalization` function of the intervened model, *not the reverse*.

To make the causal effects of our intervention *observable* on the behavioral level, we set up (x, x') pairs to ensure that the hypothetical `counterfactual output` always *differs* from the original output of the intervened model. For example (illustrated in Figure 2), consider an input $\mathbf{x}'$ with the default label words "true"/"false" and an input $\mathbf{x}$ with the irrelevant label words "cat"/"dog". If feeding $\mathbf{x}'$ to the source model yields an output

$\tau(\mathcal{M}(\mathbf{x}')) = \texttt{true}$, we will sample the corresponding $\mathbf{x}$ to construct our intervention experiment from the subset $\{\mathbf{x} \mid \tau(\mathcal{M}(\mathbf{x})) = \texttt{dog}\}$ where the intervened model outputs "dog". This enables us to observe whether the output of the intervened model *changes* from "dog" to the `counterfactual output` "cat". This construction requires the test example in $\mathbf{x}$ and $\mathbf{x}'$ to differ from each other, which we will justify later in this section.

One interesting property from the "concatenation" view is that we can interpret the "flip" from two perspectives: (1) the output of the intervened model flips from "dog" to "cat", manifesting the causal effects of `inference` function, or (2) the output of the source model flips from "true" to "cat", manifesting the causal effects of `verbalization` function.

Evaluate causal effects. We measure the effectiveness of our interchange intervention based on the **flip rate**, which is straightforwardly defined as the percentage of cases in which the intervention succeeds in making the intervened model's output flipped to the hypothetical `counterfactual output`. The strength of the evidence supporting our hypotheses is proportional to the flip rate: $\geq 70\%$ is considered adequate, and $\geq 90\%$ is considered very strong.

We further construct a ***default*** $\leftarrow$ ***default baseline*** where the intervention is conducted on the intervened model and the source model that are both prompted with the default label words. The `counterfactual output` is simply a flipped label word: if the intervened model's original output is "true", then the `counterfactual output` is "false". We consider the processes the model implements at certain layers to be *shared* across settings with remapped label spaces if the corresponding flip rates are *close* to this baseline; conversely, if the flip rates *diverge* from this baseline, we conclude that the model uses different functions during these layers. In this case, the diverging processes are very likely to perform causal roles related to the label words.

Use different test examples. Another implication of differentiating the test examples in $\mathbf{x}$ and $\mathbf{x}'$ is it prevents the intervened model from ignoring the plugged-in representations and just solving the task on its own by attending to the test example in its input contexts. We control that only the source model "knows" the test example, ensuring that the concatenated model produces the `counterfactual output` *only if* it correctly verbalizes the answer passed on by the `inference` function.

Filtering out cases with low ICL accuracy. We generally focus on a subset of cases where the model achieves adequate ICL accuracy (above 0.7). This serves dual purposes: first, to verify if these results are reproducible on *base* models, and second, to ensure that the model's successes are not merely due to chance, but rather a result of systematic causal mechanisms developed to address the task. We discuss how we adjust the prompting strategy to improve the ICL performance without explicit instructions in Appendix D and explore the low ICL cases in Appendix C.

3 Experiments

Models. We choose open-sourced base models of various sizes and families, including GEMMA-7B (Team et al., 2024), MISTRAL-7B-V0.3 (Jiang et al., 2023), GEMMA-2-27B (Team et al., 2024), and LLAMA-3.1-70B (Dubey et al., 2024).

Datasets. We first conduct experiments on three NLI datasets: MultiNLI (Williams et al., 2018), RTE (Wang et al., 2018), and ANLI (Nie et al., 2020). We cast MultiNLI and ANLI into binary classification tasks by dropping the "neutral" class to facilitate the remapping of label spaces and to avoid ambiguities inherent in the "neutral" examples. We further test if our findings generalize to other tasks by using two sentence classification datasets: IMDb (Maas et al., 2011) and AGNews (Zhang et al., 2015). Due to computational limits, we only take a subset of 300 test examples that are sampled randomly and fixed across different settings on the dataset (see Table 2 for details).

Implementation details. We conduct intervention on each layer in the model except the last one, since exchanging the last token representation at that layer is equivalent to replacing the output token. ICL settings are summarized in Table 3. All flip rates are averaged over three runs with evenly sampled demonstration examples, i.e., balanced for each class. Model outputs are decoded greedily by always selecting the top predicted token.

3.1 Experiment with Remapped Label Spaces

Modulate the label words. For each task, we experiment with a diverse set of label words. For example, in addition to the default label words

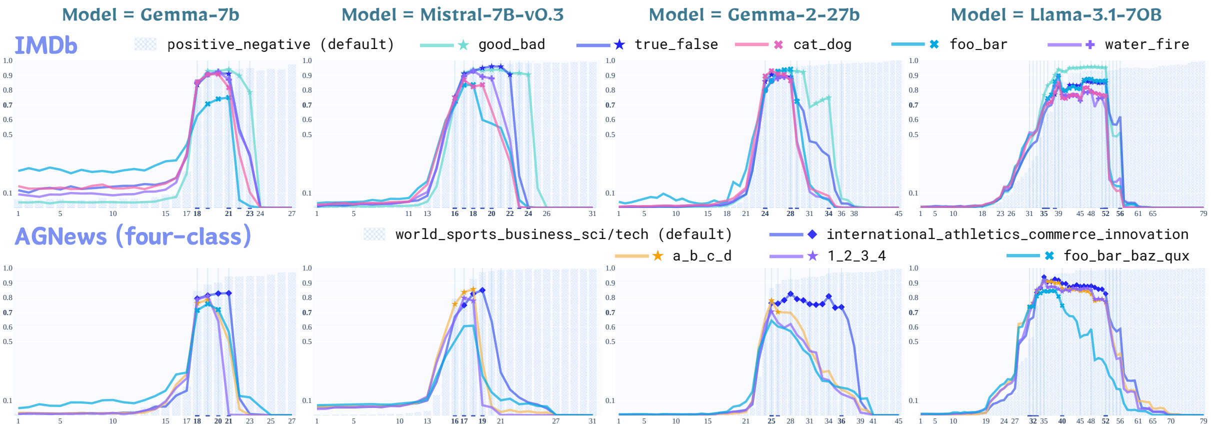


Figure 4: **Experiment with remapped label spaces on IMDb and AGNews.** We perform the same experiments with remapped label spaces on tasks other than NLI. Different sets of label spaces are used for IMDb and AGNews to accommodate the semantic variations of the label words.

used for NLI ("true"/"false"), we select (1) the task-related label words "entailment"/"not entailment" and (2) a set of task-irrelevant label words, including those generally conceived and used as binary pairs ("yes"/"no" and "good"/"bad") and those that are less obviously binary or used in contrasting contexts ("cat"/"dog", "foo"/"bar", and "black"/"white"). We list all label words in Table 1.

Findings of counterfactual outputs. We will examine both the patterns of flip rates (Figure 3) and the distribution of output tokens (Figure 2, bottom; Appendix E). Below, we discuss these results based on a representative setting: the experiments on GEMMA-7B, as illustrated in Figure 2.

During the early layers (1-17), flip rates are generally very low. Such ineffective intervention suggests that the answer representation is not yet sufficiently developed to be decoded by the other `verbalization` function. This can be validated by the distribution of output tokens: the intervened model mostly generates "dog", which is the original output of the intervened model.

During the middle layers (18-20), we observe that nearly all settings achieve a flip rate of at least 70%. This means our interventions on these layers can effectively achieve the counterfactual scenario, that is, create a "concatenated" model that outputs the `counterfactual output` "cat". This success depends on the `verbalization` function of the intervened model correctly decoding representations from the `inference` function of the source model and maintaining a correct mapping function to the label space.

During the late layers (21-27), the model outputs are gradually dominated by the original outputs of the source model ("true" instead of "cat"). It is possible that the intervention may have bypassed the starting point of the `verbalization` function, which means the representations can no longer be successfully transformed into the intended label space.

Findings of generalizability. We experiment with three groups of tasks and datasets, including: the primary setting MultiNLI (Figure 3, Top), other NLI tasks such as RTE and ANLI (Figure 3, Mid and bottom), and other tasks such as sentiment analysis on IMDb and topic classification on AGNews (Figure 4). The patterns' generalization to other NLI datasets means they are not just specific to some constructs in MultiNLI; the generalization to IMDb means that they are not specific to NLI but perhaps to classification tasks in general; and the generalization to AGNews means that they are not specific to binary classification tasks but are shared even in multi-class classifications.

Caveat. An alternative explanation for the results is that the intervened model ignores the plugged-in representations entirely and just re-solves the task on its own, given that the plugged-in representation may fully represent the test example. In this case, what has been transferred in the intervened representations is merely a task-agnostic representation of the test example. To address this concern, we conduct a complementary experiment to induce changes in the `inference` function with constructed alternative tasks on MultiNLI, where the example-label relations are changed and the input and label spaces are fixed (see Section 3.2). So

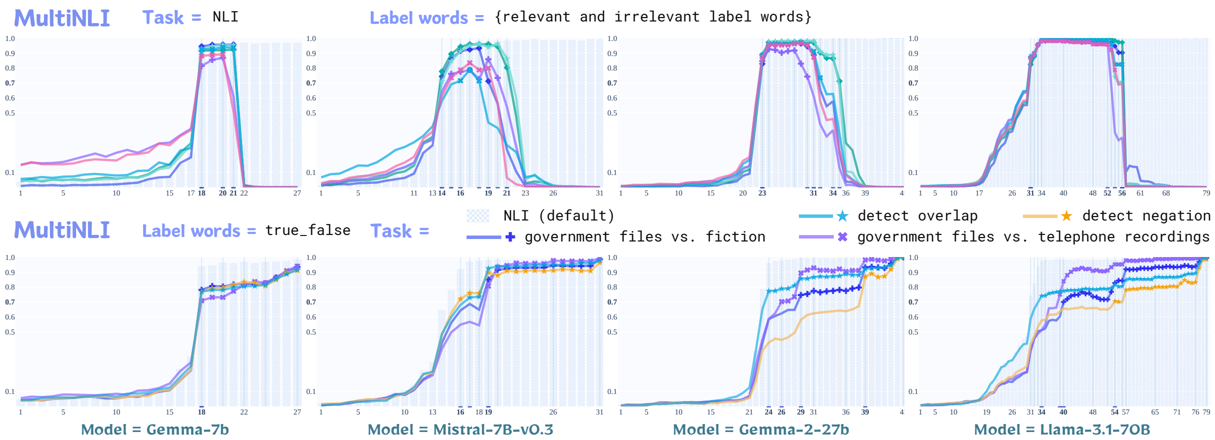


Figure 5: **Experiment with reconstructed tasks on MultiNLI.** Curves represent the flip rates of different settings where the intervened model is prompted with alternative tasks. The intervened model always sees the default task. The background bars represent the flip rates of *default* ← *default baseline*, where both the intervened model and the base model are prompted with the default task (NLI).

far we can only conclude that the computation after the middle layers, e.g., after layer 18 for GEMMA-7B, at least performs `verbalization`.

Further interpretation of the results of multi-class classification. Flip rates sometimes decrease *earlier* on AGNews than in binary classification. Why? We analyze the distribution of output tokens (Appendix E). In binary classification datasets, the output tokens are typically dominated by the `counterfactual output` during middle layers, which we labeled as "flipped" tokens; subsequently, these tokens are increasingly replaced by those from the source model, which we refer to as "overwritten" tokens. However, on AGNews, we observe that during the middle layers: (1) a significant proportion of "flipped" tokens become "overwritten" tokens, and (2) the remainder are tokens not predefined as default or irrelevant, referred to here as "other tokens" (Figure 13 and Figure 14).

This behavior does not contradict our hypotheses. It is possible that the `inference` function is implemented in the same layer as in binary settings, evidenced by the peak in flip rates around similar layers in both binary and AGNews classification; while the `verbalization` function starts in earlier layers in multi-class classification, whose initialization is marked by the increasing dominance of "overwritten" tokens. Meanwhile, a further investigation into the "other tokens" reveals that they seem to still encode the correct information of the answer and be decoded into tokens that are semantically similar to the default ones. For example, "other tokens" includes "science" and "technology", both are synonymous with the default label word "sci/tech", and the token "politics" will pick up examples similar to those labeled by "world" in the default label space. Together, the findings suggest that the intervened model can still *correctly decode* the answer representation transmitted by the `inference` function of the source model, but its implementation of the `verbalization` function is more sensitive to our interventions.

3.2 Experiment with Reconstructed Tasks on MultiNLI

Purpose. To address the potential concern mentioned earlier, we design a complementary experiment to test if the plugged-in representations are task-*agnostic*. We want to control that the intervened model can *only* know about the intended tasks (that can lead to the `counterfactual output`) by decoding a task-*relevant* representation from the source model, i.e., the representations we take from the source model and plug into the intervened model. We conclude that the plugged-in representations are *not task-agnostic* if the model can still achieve high flip rates; and this conclusion can be strengthened if the high flip rates occur in a similar location as in the previous experiments with remapped label spaces, that is, the results validate the location of the `inference` function.

Set desiderata for alternative tasks. We want the alternative tasks that (1) maximally preserve the input space to allow the model to continue employing similar mechanisms, (2) secure a balanced dataset with an adequate number of examples for both positive and negative labels, and (3) allow the model to attain sufficient ICL accuracy, which means the model has been redirected to the new tasks and developed corresponding abstractions. The criterion for ICL accuracy is less stringent than the experiment with remapped label spaces given the challenges of re-purposing a dataset designed for another task.

Construct alternative tasks. To achieve these conditions, we want to find suitable tasks that are typically simple, deterministic functions, especially those based on shallow linguistic features or those that capture the ambiguity or heuristics exploited by the model. Following Si et al. (2023), beyond the default task NLI, we construct two domain classification tasks using existing metadata of the MultiNLI and two lexical classification tasks reflecting the simple syntactic heuristics the model may exploit to achieve high performance on NLI tasks (McCoy et al., 2019). We list them in Table 7.

Apply interchange Intervention. In experiments, we always feed the NLI task to the intervened model and use “true”/“false” for the label space (illustrated in Figure 7). We define different tasks with the same sets of demonstration examples, but use different example-label relations for each task. Specifically, each data point is assigned with two labels, with one for the NLI task and another for the alternative task. In addition, we prepend instructions to each example, adopted from (Si et al., 2023) (summarized in Table 8), since none of the prompting templates without explicit instructions we have tried can lead to adequate ICL performance. The *default* ← *default baseline* is created by using the NLI task for both the intervened and the source model.

Findings of consistent locations. As shown in Figure 5, the flip rates surge at the same layer in both the experiment with remapped label spaces and with reconstructed tasks. Given that the intervention design ensures only the source model has access to the correct task—which leads to the hypothetical `counterfactual output`—a high flip rate is possible only if the plugged-in representation has already encoded information about this `counterfactual output`. This means that the early-stage layers of the upstream source model have already performed the task, i.e., implemented the `inference` function. Thus, the results validate the location of the `inference` function and thus help us address the aforementioned concern.

Caveat. The evidence coming from this experiment will only serve complementary roles, and its strength is weaker than the experiment with remapped label words. First, it is hard to control what has been changed when intervening on the upstream process; and second, the patterns here are less consistent (alternative tasks can have flip rates varying in values and locations where they peak). However, these do not contradict our hypotheses, as they do not require high and consistent flip rates in this setting. The goal of this experiment is to verify the early-stage layers do perform `inference` function, and for this purpose, it is sufficient to observe that in all settings (except one for MISTRAL-7B-V0.3), the flip rates with reconstructed tasks rise significantly above the random baseline (0.5) before the flip rates in the experiment with remapped label spaces drop to 0. The inconsistencies of the locations where the flip rates peak are expected, since changing the task likely causes significant changes in the model’s internal circuits and states used to solve the tasks (even though these tasks might share some sub-processes), making the representations less transferable.

4 Discussion

Validation of H_1: Causal mechanism. This hypothesis is confirmed if we can “concatenate” an `inference` function with another `verbalization` function and observe the generation of the `counterfactual output` that reflects the answer obtained by the `inference` function and the corresponding label space of the `verbalization` function. We achieve such a scenario with the experiment with remapped label spaces, as manifested by the high flip rates we observe during the middle layers of the model.

Concern may be raised that the intervened model just re-solves the task on its own during the downstream processes, that is, both `inference` function and `verbalization` function are performed in the middle and the late layers. Our experiment with alternative tasks shows that this is not true. This experiment ensures that only the source model knows the answer that leads to the hypothetical `counterfactual output`, which means high flip rates can only be achieved if the plugged-in representation involves a sufficiently developed representation of the answer. Therefore, the high flip rates we observe during and after the middle layers support the hypothesis that the `inference` function is implemented by the upstream processes (before the layers with effective interventions).

Validation of H_2: Invariant to label remapping. From the results of the experiment with remapped label spaces, we know that intervening on a sequence of middle layers in the model can successfully achieve the counterfactual scenario, where the `inference` function can be concatenated with another `verbalization` function to produce the hypothetical `counterfactual output`. This indicates that the `inference` function can play its causal role invariant to the downstream `verbalization` function.

Validation of H_3: Separate localization and consistent localization. This hypothesis is a necessary condition for us to use interchange intervention to realize the counterfactual scenario, and thus it gets validated if our interventions are effective. It can be further supported by the observation that for each model, the trend of flip rates is similar across different datasets and tasks. For example, flip rates surge at the same or very close layers (Figure 3 and 4). We summarize the start and the end of the layers enabling effective interventions in Table 5.

5 Related Work

Understanding ICL. A variety of work has sought to better understand how language models perform in-context learning. Min et al. (2022) show that ICL performance on a variety of text classification tasks is minimally affected by randomly changing the demonstration labels, hypothesizing that demonstrations may primarily help with determining the label space (i.e., the `verbalization` function), the distribution of the input examples, and how the output text should be formatted. (Xie et al., 2022) hypothesize that LMs learn latent tasks during pre-training, primarily using ICL demonstrations to identify which latent task is most pertinent to the provided ICL input. Other work (Akyürek et al., 2023; von Oswald et al., 2022; Dai et al., 2022) hypothesizes that LMs perform implicit gradient descent on a latent model during ICL. Wei et al. (2023) and Pan et al. (2023) confirmed these observations and further indicated that this consistency depends on the size of the model. Our study offers a deeper analysis of these behaviors at the representational level. Wei et al. (2023) study how model size affects whether LMs prefer to rely on semantic priors about a task, as opposed to the example-label mappings provided in the demonstrations. They find that smaller models tend to rely on semantic priors during pre-training, since they ignore flipped labels presented in the ICL demonstrations. In contrast, larger models make use these flipped labels, indicating that they rely less on their semantic task priors. Pan et al. (2023) disentangle task learning and task recognition in in-context learning; intuitively, LMs use their priors to perform novel tasks during ICL (since a few demonstrations are unlikely to provide complete information about a complex task), but they also do not completely rely on their task priors (since they can handle ICL settings with arbitrary label mappings).

Implicit functions in language models. Another line of prior work has sought to characterize what functions might be implicitly implemented by LMs. Olsson et al. (2022) identify "induction heads", a unique type of attention head that replicates repeating patterns from prior contexts. Hendel et al. (2023) argue ICL compresses the demonstrations into a task vector, which is then used to generate the output for the ICL input. Merullo et al. (2023) provide evidence that Transformer LMs perform ICL in three layer-wise stages: argument formulation, function application, and saturation. Todd et al. (2024) show that ICL creates function vectors of the demonstrated task, and these function vectors can trigger execution of the task in other settings (e.g., zero-shot prediction or natural text).

6 Conclusion

We hypothesize that when LMs perform ICL with irrelevant or misleading label words, they first apply an `inference` function to obtain a representation of the answer and then apply a `verbalization` function to verbalize the answer as one of the label words specified in the demonstrations (H_1). In addition, the `inference` function is invariant to remappings of label words (H_2), and both the functions can be localized separately and consistently within certain layers of the model (H_3). To validate our hypotheses, we design experiments based on interchange intervention to realize counterfactual scenarios that would *only* occur if all of our hypotheses hold true. Our experiments across a variety of tasks, datasets and models indicate that our hypotheses hold across a variety of settings. Our findings contribute to a growing body of work on mechanistically understanding how language models perform in-context learning.

7 Limitations

Coverage of models and tasks. Since our intervention study requires access to the model's intermediate representations, we can only experiment with open-source models. There is no guarantee that similar findings can be observed in the state-of-the-art close models, and how the MoE architectures will affect our findings remains unclear. In addition, we acknowledged that we only focus on classification tasks on which the notion of "verbalization" or remapping of label spaces can be more naturally defined. Future studies could extend to more complex tasks, such as those that involve answering open-ended questions.

Further specifying the inference function. Does the model *truly* carry out natural language inference when processing datasets like MultiNLI, RTE, and ANLI? Our current evidence does not allow us to definitively answer this question, and the notion of "textual entailment" itself is not without its controversies. We cannot claim about what the model is actually doing during the layers of the `inference` function. Moreover, when experimenting with reconstructed tasks on MultiNLI, we do not know the *true* relationships between the alternative tasks (lexical and domain classification) and NLI. Nonetheless, these ambiguities do not undermine our argument—we do not assert that the model performs an "NLI-inference" function on NLI datasets. By `inference` function we refer to the (general) processes the model uses to derive the answer representation.

Predicting ICL performance. Prior studies have shown that the model's ICL performance can be (sometimes significantly) harmed by the irrelevant labels (Min et al., 2022; Pan et al., 2023; Wei et al., 2023). We do not conduct a systematic analysis and explanation of how different factors—such as prompt templates, label word choices, and the number of shots—causally contribute to the model's performance on ICL tasks with irrelevant labels. A nuanced understanding of the underlying mechanism would require analysis tailored to each model and setting. In particular, our current research does not address which specific features of label words influence ICL performance positively or negatively, which could be an interesting topic to explore.

Unknown training data. Concern arises whether the model's good performance on ICL tasks with irrelevant label words stems from its prior exposure to similar cases. We cannot rule out the possibility of data leakage during pre-training, primarily because we do not have access to the pretraining data of the models we study. We are hopeful, however, that our use of a diverse range of irrelevant labels in our prompts reduces the likelihood that these specific cases were present in the pre-training dataset, thus alleviating potential concerns.

8 Acknowledgements

We would like to thank Atticus Geiger, Jing Huang, Jacqueline Harding, Zhengxuan Wu, Kara Schechtman, Junlin Wang, Jiarui Lu, Zhihan Li, and the anonymous reviewers for feedback and discussions that helped improve this work. This research was enabled in part by the computational resources provided by Mila (https://mila.quebec) and NVIDIA.

References

Ekin Akyürek, D. Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *Proc. of ICLR*.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. ArXiv:1610.01644.

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. 2022. *On Pearl's Hierarchy and the Foundations of Causal Inference*, 1 edition, page 507–556. Association for Computing Machinery, New York, NY, USA.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. ArXiv: 2005.14165.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proc. of BlackboxNLP*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2022. Why can GPT learn in-context? language models implicitly perform gradient descent as meta-optimizers. ArXiv:2212.10559.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir

Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The Llama 3 herd of models. ArXiv:2407.21783.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proc. of ACL*.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. 2024a. Causal abstraction: A theoretical foundation for mechanistic interpretability. ArXiv:2301.04709.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Proc. of NeurIPS*.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proc. of BlackboxNLP*.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah D. Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. ArXiv:2112.00826.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2024b. Finding alignments between interpretable causal variables and distributed neural representations. ArXiv:2303.02536.

Jacqueline Harding. 2023. Operationalising representation in natural language processing. *The British Journal for the Philosophy of Science*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. ArXiv:2310.15916.

Paul W. Holland. 1986. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. 2024. RAVEL: Evaluating interpretability methods on disentangling language model representations. ArXiv:2402.17700.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61(1):907–926.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. ArXiv:2310.06825.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proc. of ACL*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proc. of ACL*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Proc. of NeurIPS*. Curran Associates, Inc.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. A mechanism for solving relational tasks in transformer language models. ArXiv:2305.16130.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proc. of EMNLP*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proc. of ACL*.

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. *arXiv:2209.11895*.

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 ACL*.

Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*, 1st edition. Basic Books, Inc., USA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proc. of EMNLP*.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. 2023. Measuring inductive biases of in-context learning with underspecified demonstrations. In *Proc. of ACL*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Pier Giuseppe Sessa, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv: 2403.08295*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proc. of ACL*.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. In *Proc. of ICLR*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Proc. of NeurIPS*.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022. Transformers learn in-context by gradient descent. In *Proc. of ICML*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of BlackboxNLP*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. ArXiv:2303.03846.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL*.

James F. Woodward. 2003. *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, New York.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *Proc. of ICLR*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proc. of NeurPS*.

A Probing Study

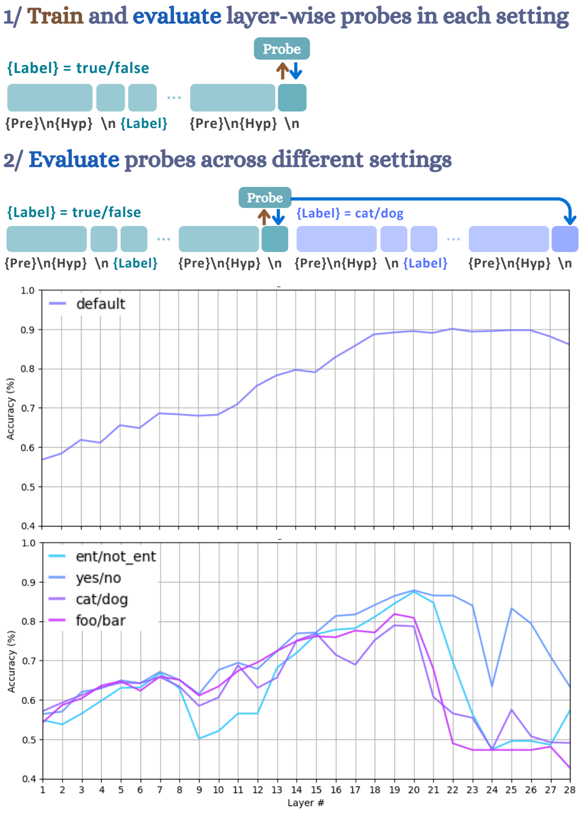


Figure 6: **Probing setting and results. Top:** an illustration of the probing experiment, where we train a probe on the last token presentation to predict the NLI label. **Bottom:** results of the probing study. The first graph shows the probing accuracy when the probe is trained and evaluated on the default setting; while the second graph displays the probing accuracy when the probe is trained on the default setting and evaluated on other settings with different label words. By comparing the out-of-distribution curves (in the second graph) with the in-distribution curve (in the first graph), we can see an obvious bifurcation starting around the 18th-20th layer. This indicates (1) that the representation of the answer has been fully developed around these layers, and (2) the first process (before the bifurcation point) is roughly invariant to the changes in label space, while the second process (after the bifurcation point) is heavily dependent on the label space.

To identify the low-level implementations that align with our hypothesized high-level abstractions, we start with identifying processes *shared* across different model runs and then verify if they satisfy all properties we hypothesize. As a proxy, we conduct a pilot study where we use probing to examine how the intermediate representations, i.e., the neural activations, correlate with the high-level concept of the answer to the ICL task and the label words specified in the prompt.

Probe. A probe (Alain and Bengio, 2016; Peters et al., 2018; Tenney et al., 2019; Clark et al., 2019; Hupkes et al., 2018) is usually a linear classifier or shallow MLP that takes the intermediate representation as the input and output labels for some property, aimed at testing how easily the representations can be linearly separated. A high probing accuracy on a hold-out test set indicates that the information about the property is encoded in the intermediate representation.

Experimental details. We implement probes as logistic regressions with Scikit-learn (Pedregosa et al., 2011) and the L-BFGS optimization algorithm (Liu and Nocedal, 1989). We train one probe for each layer with representations generated on the RTE training set by GEMMA-7B. Probes are then applied to the RTE validation set with different label words. Results are averaged over three trials with different sets of demonstrations. We repeat this process for each layer.

Experiments. We train probes to predict the output labels on the last token representations of one run with default label words ("true"/"false") in the NLI task. We first apply to probe to the representations produced by runs with default label words to observe the development of the representation of the output throughout the forward pass. Then, we test if this probe can generalize to the representations produced by runs with remapped label words (e.g., when the default label space "true"/"false" is remapped to "cat"/"dog"). This aims to identify if there are shared processes in runs with different label words. Specifically, the process on certain layers is interpreted as being *shared* if the probing accuracy on these layers are similar across different runs; on the contrary, if the probing accuracy on certain layers of one run diverges from the probing accuracy on another run, we conclude that the model implements different processes on these layers in each run.

Findings. For probes trained and tested on the runs with default label words (Figure 6, Middle), the probing accuracy is monotonically increasing, suggesting the development of the representation of the output label. For probes trained on the runs with default label words and tested on runs with different label words, the probes achieve high accuracy on other label words in layers 15-20 and then drop to a random baseline in the last 8 layers. In other words, the probing accuracy on different runs first converges and then diverges.

These results indicate that the information about the NLI labels is encoded at least in two different patterns across the model's forward pass. In the middle layer, where the probe can generalize, the representations share a similar structure in relation to the NLI label, enabling the probe trained on the default label words to still correctly classify in the other label words cases. In the late layers, the representation structures start to vary based on the label words, causing a decline in generalization performance. From these results, we can categorize the model's intermediate representations into two groups: one that is sensitive to label words and another that is not. Given these findings, it is natural to assume that the processes that generate the representations share the same characteristics. That is, the first process implemented by the middle layers is `inference` function related since it produces unified representations of the task target. The second process implemented by the late layers is `verbalization` function related, as its intermediate representation structure varies across different label words.

Caveat. We note that probing results only tell us whether the representations encode certain information without guaranteeing that these representations indeed play *causal roles* in the generation of certain model's behaviors, i.e., *used* by the model (Belinkov, 2022), or *genuinely represented* in the layers where they are detected (Harding, 2023). Claiming about the causal effects we want to study will require further hypotheses and experiments introduced later in the paper. Nonetheless, we take the probing results as an initial point, and the three key messages delivered: (1) the two functions may be located *separately*, (2) if located separately, the `inference` function comes first then the `verbalization` function, and (3) the location of the two functions may be *consistent* across settings.

Clarification of "localization". We do not commit to the notion of "verbalization layer", i.e., layers that are active exclusively in response to label words in ICL, resembling the classic example "grandmother cell" in neuroscience. We do not follow the traditional view of located representation positing that a specific vehicle of the representation (neurons or small sets of neurons) could be responsible for representing complex concepts. What we find and we will claim is that the model *perform or implement* the high-level `verbalization` function consistently in certain layers under the background conditions we test (prompting strategies, etc).

B Illustrations of the Reconstructed Task Intervention

Figure 7 illustrates and summarizes our reconstructed task intervention outlined in Section 3.2.

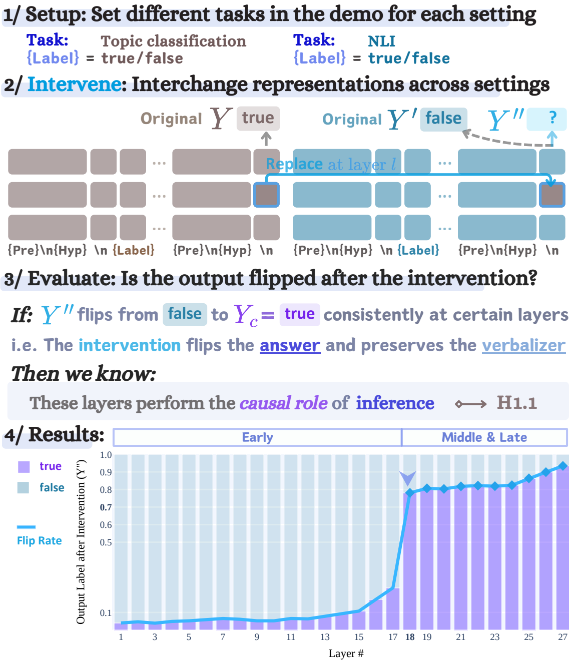

1/ Setup: Set different tasks in the demo for each setting
Task: Topic classification
{Label} = true/false
Task: NLI
{Label} = true/false
2/ Intervene: Interchange representations across settings
Original Y true
Original Y' false
Y'' ?
Replace at layer l
{Pre}\n{Hyp} \n {Label}
{Pre}\n{Hyp} \n
{Pre}\n{Hyp} \n {Label}
{Pre}\n{Hyp} \n
3/ Evaluate: Is the output flipped after the intervention?
If: Y'' flips from false to Y_c = true consistently at certain layers
i.e. The intervention flips the answer and preserves the verbalizer
Then we know:
These layers perform the causal role of inference ⟷ H1.1
4/ Results:
Early
Middle & Late
true
false
Flip Rate
Output Label after Intervention (Y'')
1.0
0.9
0.8
0.7
0.6
0.5
0.1
1
3
5
7
9
11
13
15
17
18
19
21
23
25
27
Layer #

Figure 7: **Experiment with reconstructed tasks on MultiNLI.** (1) We induce changes in the `inference` function by prompting the model to perform a different task on the same input space. (2) We perform intervention by taking the representation from the **sourced model** (prompted by the alternative task) and replacing the representation of the **intervened model** (prompted by the NLI task) at layer l. (3) Intervention effects are evaluated by calculating the matching rate between the intervened output Y'' and the hypothetical `counterfactual` output Y_c.

C Implications of Poor ICL Performance

We do not expect to observe effective intervention results on cases where the ICL performance is inadequate, and such cases will not weaken our argument. This is because the poor ICL performance is very likely to come from the model's failure to develop the necessary abstractions for solving the task, which would include recognizing the task, inferring the correct answer, recognizing the label space, and verbalizing the answer correctly. If the model achieves high ICL performance with some

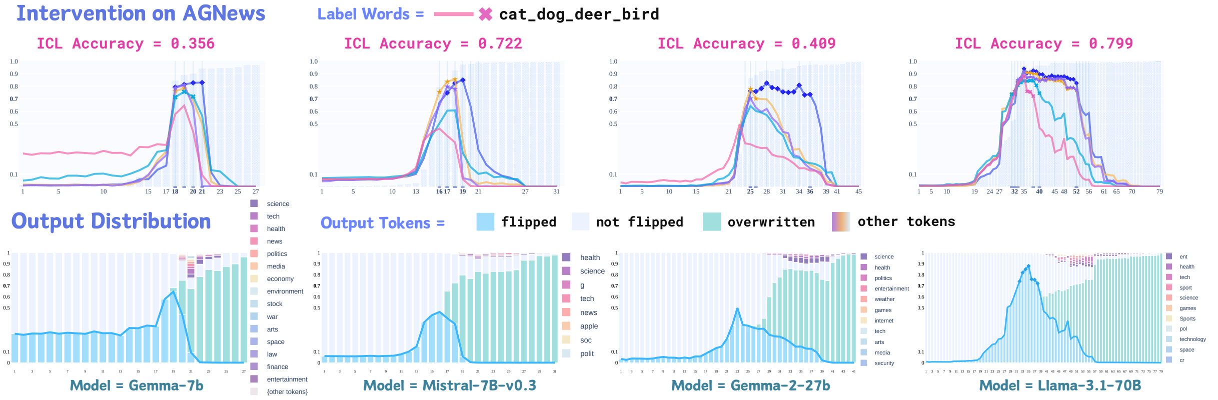


Figure 8: **Exploration of negative cases where the ICL performance is low. Top:** flip rates of intervention with remapped label spaces. **Bottom:** Distribution of predicted token from the intervened model with the under-performing label words.

label spaces while not with others, it may employ different sets of abstractions to solve these two ICL tasks.

Nonetheless, we intentionally include some cases where the model fails to solve the ICL task, as complementary results (ICL performance is summarized in Table 5 with the low accuracy colored in orange or red). In our experiments, we find that (1) if the ICL accuracy is good enough, e.g., ≥ 0.7, most settings will show a strong pattern, as summarized before; while some inconsistencies may exist, though rarely. One example is the case where MISTRAL-7B-V0.3 is used on AGNews with the label words with the label words "cat"/"dog", as shown in Figure 8, second column); (2) if the ICL is not good enough (around 0.6 for binary classification and around 0.4 for four-class classification), we find the flip rates still generally follow the same trend, though the patterns are *weaker* than settings with high ICL accuracy due to the lack of necessary abstractions developed. Examples include: cases with the label space "foo"/"bar" and "black"/"white" for GEMMA-7B and MISTRAL-7B-V0.3 on RTE (Figure 3, second row) and "cat"/"dog"/"deer"/"bird" for MISTRAL-7B-V0.3 on AGNews (Figure 8, first and third column).

D Prompting Strategy to Improve ICL Performance

We slightly adjust the selection of irrelevant label words and prompting strategies for each setting, to facilitate the model in achieving adequate ICL performance.

We observe that reproducing the phenomena of interest on *base models* can be challenging with some settings. For instance, 7-8 models on ANLI are difficult to handle with irrelevant label words despite performing adequately with default label words. And AGNews is generally hard for all models because it is a four-class classification and naturally requires more shots.

We adjust our template selection based on the corresponding ICL performance. We always start with a 16-shot demonstration with the "sentence" template, aiming for ICL performance above 0.7. If this benchmark is not met, we increase the demonstration to 32 shots. Should challenges persist, we modify the template to "sent_label" and consider adding specific keywords in front of the `answer` to implicitly hint for the intended tasks, such as "topic:" for AGNews and "sentiment:" for IMDb.

Note on the notion of generalizability. By the generalizability of our findings, we do not mean that a single set of irrelevant labels should work universally across all models in all settings. Again, our study of causal mechanisms focuses on cases where the model achieves high ICL performance, which means it develops the necessary causal abstractions required to solve the task we want to study. Since LMs are sensitive to contexts and to different prompting strategies, the required settings unsurprisingly vary model by model.

Address concerns about cherry-picking. Concerns may arise that our selection of successful cases constitutes cherry-picking. However, we select cases based on their ICL performance, not their intervention results, and we test a variety of irrelevant label words that are representative of broader cases. We also discuss cases where ICL performance is not good and find that they still generally follow the patterns, though they do so to a lesser extent. (See Appendix C).

E Predicted Tokens from the experiment with remapped label spaces

We present the full results of the intervened model's prediction distributions on all five datasets used in our intervention with remapped label spaces: MultiNLI (Figure 9), RTE (Figure 10), ANLI (Figure 11), IMDb (Figure 12), AGNews (Figure 13 & 14).

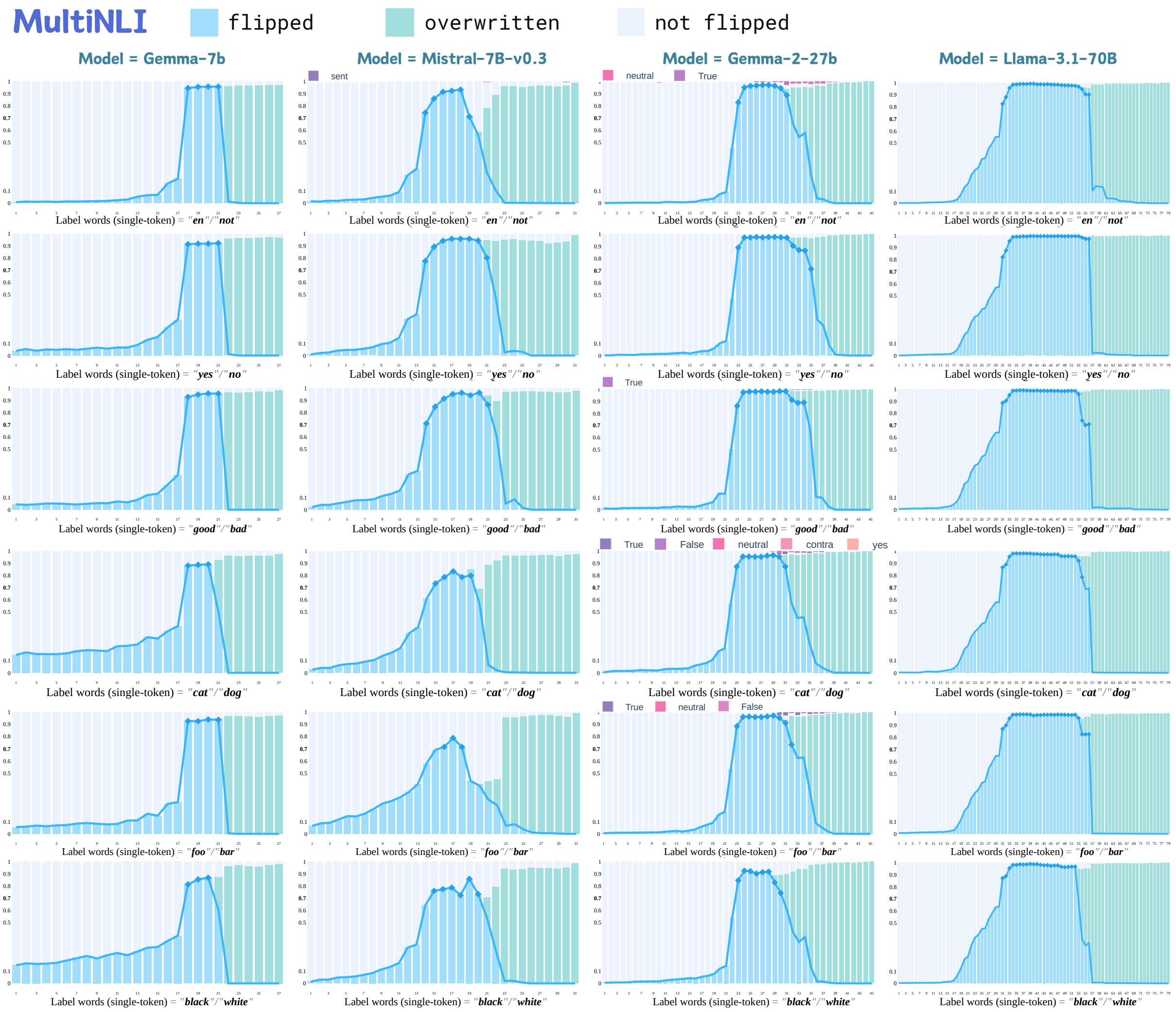


Figure 9: Full results of intervened model's prediction distributions on the MultiNLI dataset.

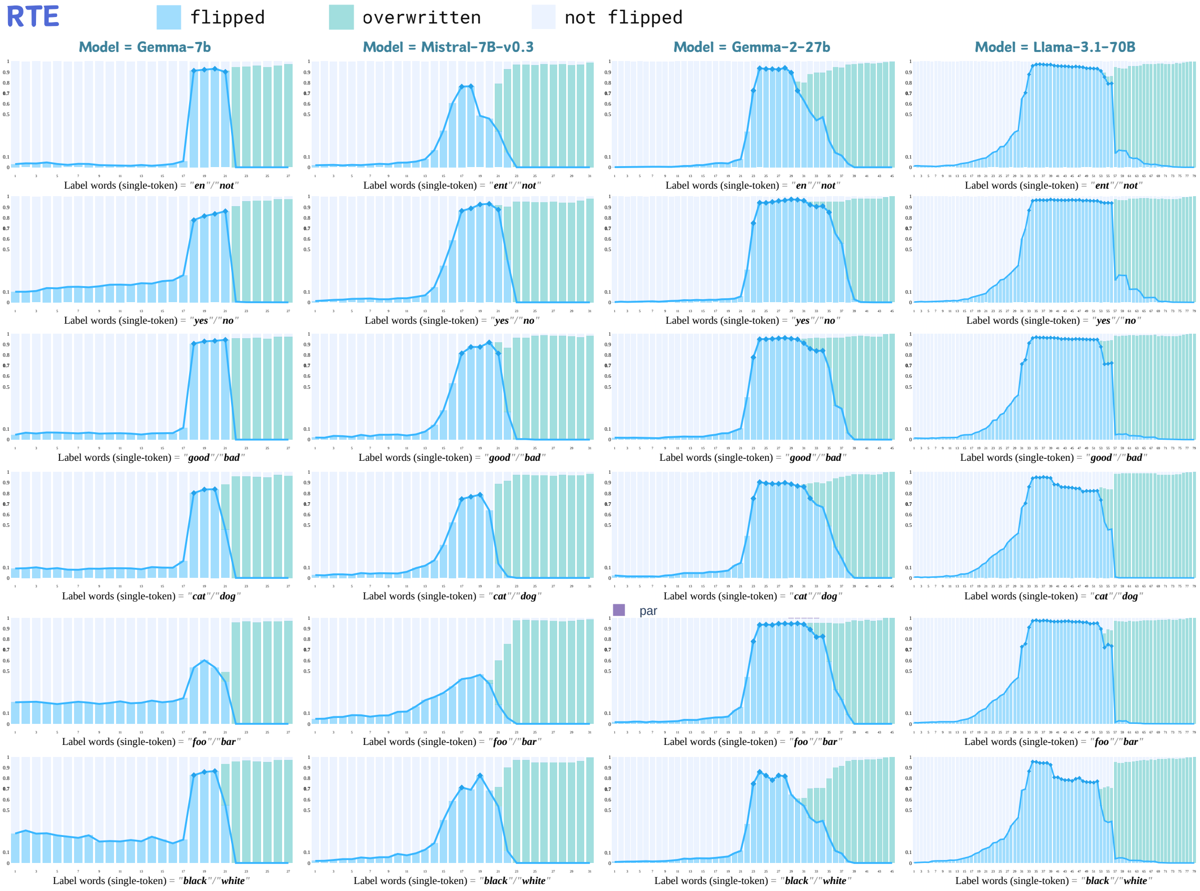


Figure 10: Full results of intervened model's prediction distributions on the RTE dataset.

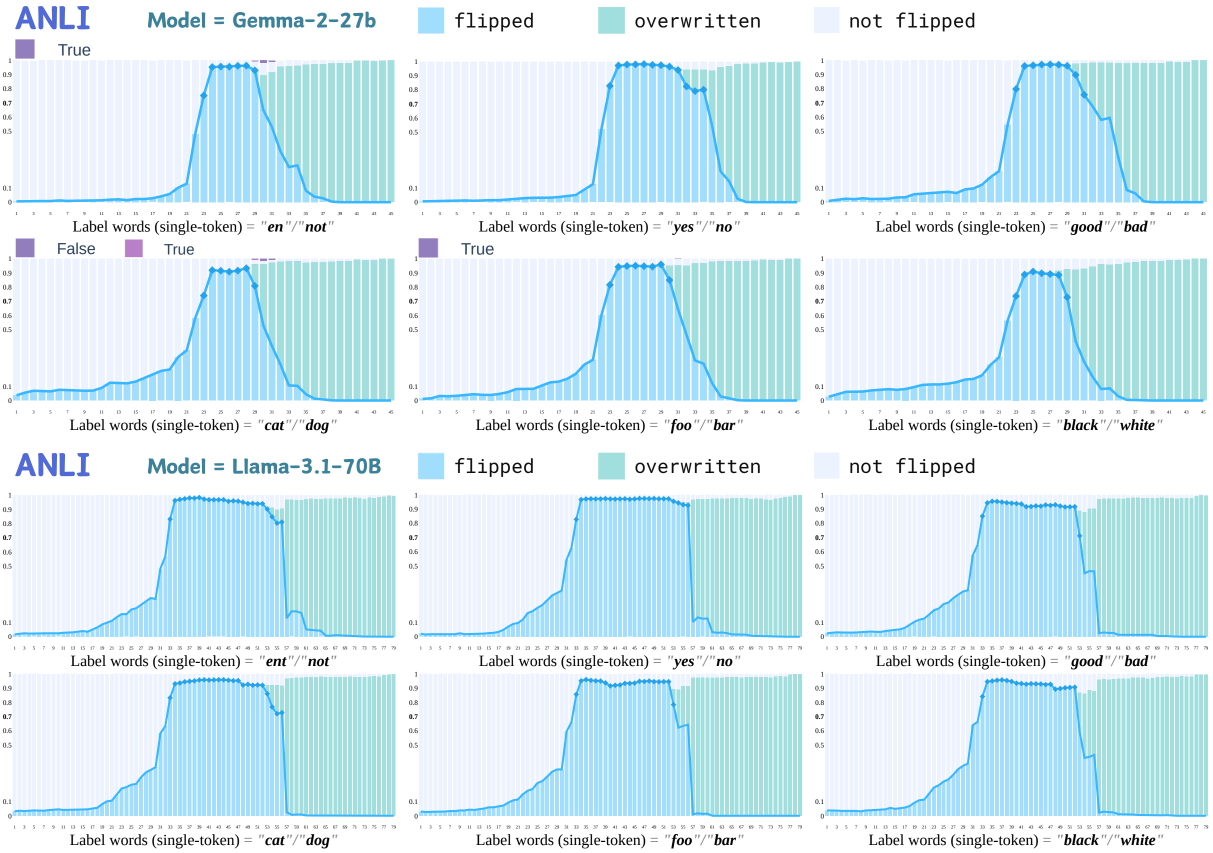


Figure 11: Full results of intervened model's prediction distributions on the ANLI dataset.

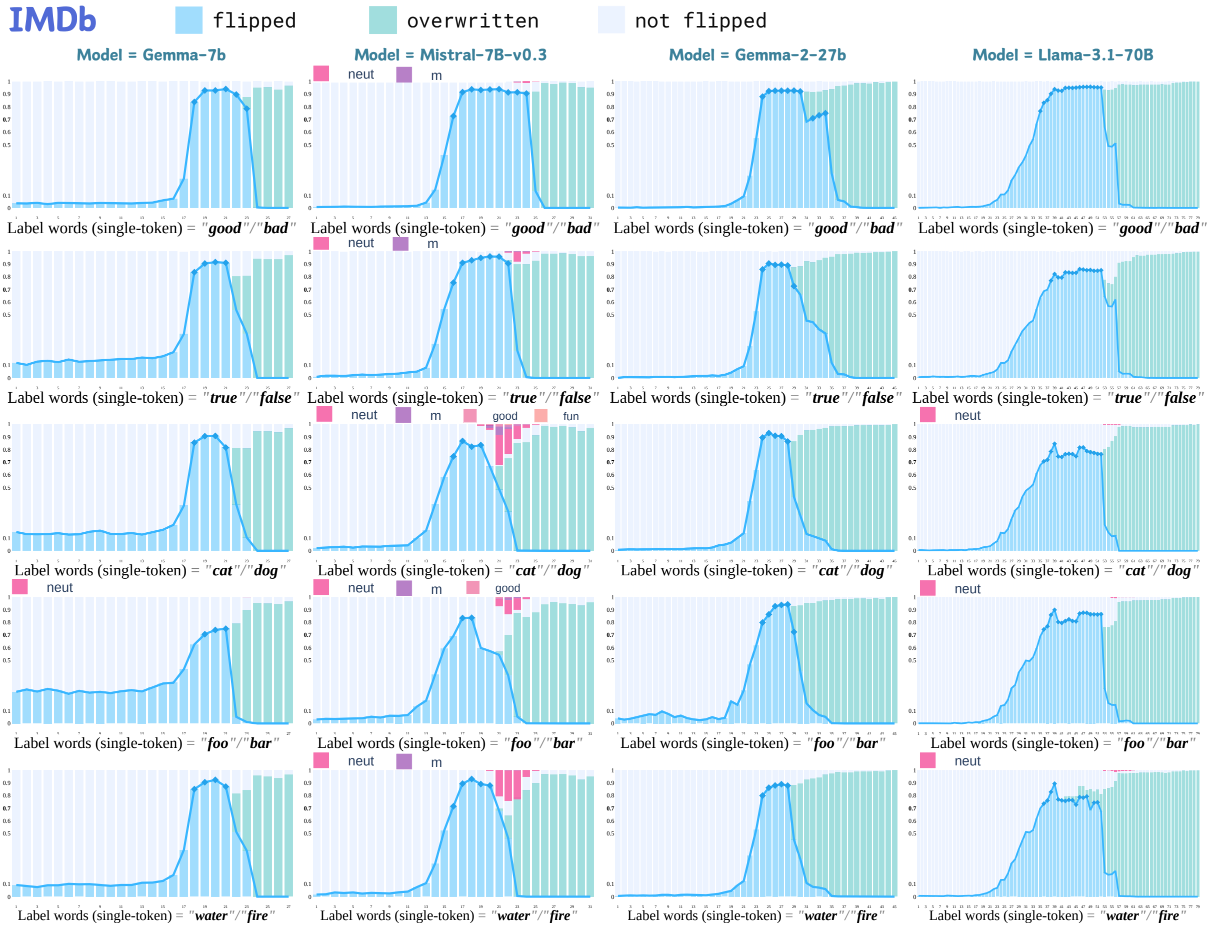


Figure 12: Full results of intervened model's prediction distributions on the IMDb dataset.

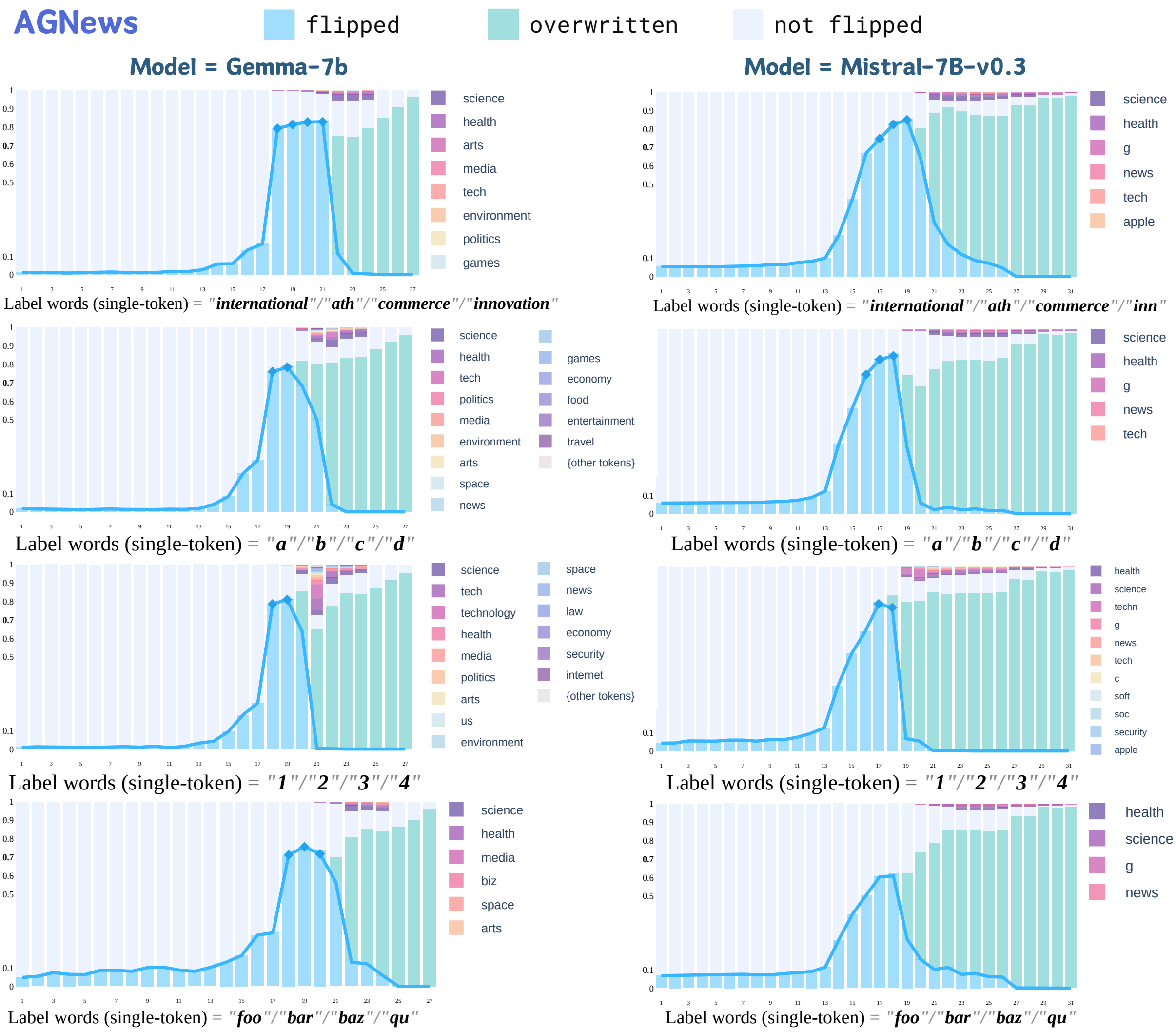


Figure 13: Full results of intervened model's prediction distributions on the AGNews dataset. Part 1.

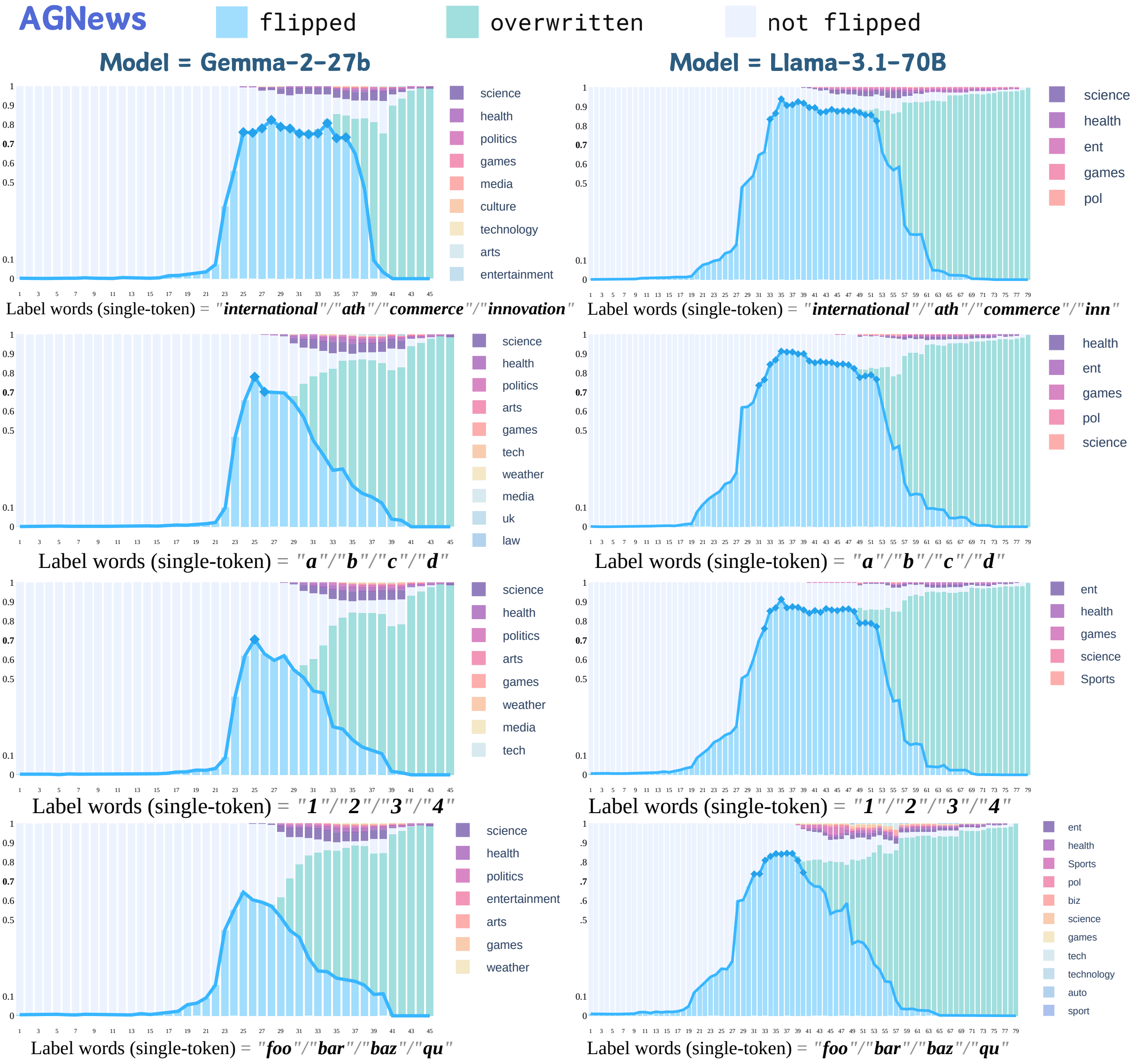


Figure 14: Full results of intervened model's prediction distributions on the AGNews dataset. Part 2.

Dataset	Verbalizers	Predicted Single Tokens			
		GEMMA-7B	MISTRAL-7B-V0.3	GEMMA-2-27B	LLAMA-3.1-70B
MultiNLI/	*"true"/"false"*	*"true"/"false"*	*"true"/"false"*	*"true"/"false"*	*"true"/"false"*
RTE/	"en"/"not"	"en"/"not"	"ent"/"not"	"en"/"not"	"ent"/"not"
ANLI	"yes"/"no"	"yes"/"no"	"yes"/"no"	"yes"/"no"	"yes"/"no"
	"good"/"bad"	"good"/"bad"	"good"/"bad"	"good"/"bad"	"good"/"bad"
	"cat"/"dog"	"cat"/"dog"	"cat"/"dog"	"cat"/"dog"	"cat"/"dog"
	"foo"/"bar	"foo"/"bar	"foo"/"bar	"foo"/"bar	"foo"/"bar
	"black"/"white"	"black"/"white"	"black"/"white"	"black"/"white"	"black"/"white"
IMDb	*"positive"/"negative"*	*"positive"/"negative"*	*"pos"/"negative"*	*"positive"/"negative"*	*"positive"/"negative"*
	"good"/"bad"	"good"/"bad"	"good"/"bad"	"good"/"bad"	"good"/"bad"
	"true"/"false"	"true"/"false"	"true"/"false"	"true"/"false"	"true"/"false"
	"cat"/"dog"	"cat"/"dog"	"cat"/"dog"	"cat"/"dog"	"cat"/"dog"
	"foo"/"bar"	"foo"/"bar"	"foo"/"bar"	"foo"/"bar"	"foo"/"bar"
	"water"/"fire"	"water"/"fire"	"water"/"fire"	"water"/"fire"	"water"/"fire"
AGNews	*"world"/"sports"/* *"business"/"sci/tech"*	*"world"/"sports"/* *"business"/"sci"*	*"world"/"s"/* *"business"/"sc"*	*"world"/"sports"/* *"business"/"sci"*	*"world"/"sports"/* *"business"/"sci"*
	"international"/"athletics"/ "commerce"/"innovation"	"international"/"ath"/ "commerce"/"innovation"	"intern"/"ath"/ "commerce"/"inn"	"international"/"ath"/ "commerce"/"innovation"	"international"/"ath"/ "commerce"/"inn"
	"a"/"b"/"c"/"d"	"a"/"b"/"c"/"d"	"a"/"b"/"c"/"d"	"a"/"b"/"c"/"d"	"a"/"b"/"c"/"d"
	"1"/"2"/"3"/"4"	"1"/"2"/"3"/"4"	"1"/"2"/"3"/"4"	"1"/"2"/"3"/"4"	"1"/"2"/"3"/"4"
	"foo"/"bar"/"baz"/"qux"	"foo"/"bar"/"baz"/"qu"	"foo"/"bar"/"b"/"qu"	"foo"/"bar"/"baz"/"qu"	"foo"/"bar"/"baz"/"qu"

Table 1: **Tokenization of verbalizers**: LM tokenizers may break a word into sub-tokens. Our intervention method requires verbalizers to be differentiable by the first sub-token since it only evaluates the first generated token. Here we present how each model's tokenizer break-down the verbalizers.

Intervention	Task	Dataset	# of Test Examples		
			Original	Experiments (7-8b models)	Experiments (27-70b models)
Change Verbalizer	Natural	RTE	277	277	277
	Language	MultiNLI	9820	300	300
	Inference	ANLI	1000	300	300
	Sentiment Analysis	IMDb	25000	300	300
	Topic Classification	AGNews	7600	300	300
Change Task	Multiple Tasks	MultiNLI	9820	300	300

Table 2: **Statistic of datasets:** Due to the computational budget, we reduced the amounts of test examples for all datasets other than the RTE to around 300 examples which we believe is sufficient to test the generalizability of our hypothesized functions. Subsets are sampled randomly using a fix random seed 42 from each corresponding task dataset.

Intervention	Task	Dataset	GEMMA-7B		MISTRAL-7B-V0.3		GEMMA-2-27B		LLAMA-3.1-70B	
			Shot	Template	Shot	Template	Shot	Template	Shot	Template
Change Verbalizer	Natural	MultiNLI	16	sentence	16	sentence	16	sentence	16	sentence
	Language	RTE	16	sent_label	16	sent_label	16	sentence	16	sentence
	Inference	ANLI	/	/	/	/	16	sentence	16	sentence
	Sentiment Analysis	IMDb	16	passage_label	16	passage_label	8	passage_label	16	passage_sentiment
	Topic Classification	AGNews	32	text_topic	32	text_topic	32	text_topic	32	text_linebreak
Change Task	Multiple Tasks	MultiNLI	32	ambi_instruct	32	ambi_instruct	16	ambi_instruct	16	ambi_instruct

Table 3: **ICL settings details**: we select the combinations of the prompt template and number of demonstrations based on the **ICL performance**. We search the optimal number of shots in $\{8, 16, 24, 32\}$. See Table 6 for details of prompt templates and Table 8 for instructions used in the Change Task setting.

Dataset	Intervention on Label Words	Label Words	GEMMA-7B			MISTRAL-7B-V0.3		
			sentence 16shots	sentence 32shots	sent_label 16shots	sentence 16shots	sentence 32shots	sent_label 16shots
RTE	***default → default***	***"true"/"false"***	***0.807***	***0.786***	***0.819***	***0.827***	***0.838***	***0.813***
	default → relevant	"en"/"not"	0.798	0.787	0.788	0.755	0.795	0.773
	default → irrelevant	"yes"/"no"	0.838	0.795	0.745	0.742	0.826	0.771
		"good"/"bad"	0.780	0.767	0.810	0.721	0.767	0.724
		"cat"/"dog"	0.574	0.621	0.759	0.558	0.573	0.637
		"foo"/"bar	0.777	0.761	0.633	0.536	0.539	0.567
		"black"/"white"	0.611	0.537	0.620	0.607	0.639	0.609

Dataset	Intervention on Label Words		GEMMA-2-27B		LLAMA-3.1-70B	
		sentence 8shots	sentence 16shots	sentence 8shots	sentence 16shots	
ANLI	***default → default***	***"true"/"false"***	***0.824***	***0.851***		***0.854***
	default → relevant	"en"/"not"	0.801	0.816		0.841
	default → irrelevant	"yes"/"no"	0.811	0.843		0.821
		"good"/"bad"	0.758	0.790		0.802
		"cat"/"dog"	0.628	0.643		0.766
		"foo"/"bar	0.633	0.728		0.808
		"black"/"white"	0.667	0.710		0.794

Table 4: An (representative) example (RTE for GEMMA-7B and MISTRAL-7B-V0.3; ANLI for GEMMA-2-27B and LLAMA-3.1-70B). Additional information we want to provide; for informing the justification of we choose the setting we are using instead of others we start from "sentence" template and 16 shot (for small models) and 32 for large models chosen setting are bold/colored in blue LLAMA-3.1-70B in general does not work well with "label" and so we create some

Dataset	Intervention on Label Words	Label Words	GEMMA-7B			MISTRAL-7B-V0.3			GEMMA-2-27B			LLAMA-3.1-70B		
			ICL	Start	End	ICL	Start	End	ICL	Start	End	ICL	Start	End
MultiNLI	***default → default***	***"true"/"false"***	***0.889***	***18***	***27***	***0.842***	***15***	***31***	***0.939***	***24***	***45***	***0.956***	***34***	***79***
	default → relevant	"en"/"not"	0.921	18	21	0.870	15	18	0.943	24	29	0.958	34	52
		"yes"/"no"	0.856	18	21	0.820	15	20	0.944	24	31	0.959	34	53
		"good"/"bad"	0.834	18	21	0.747	15	20	0.913	24	31	0.950	34	52
	default → irrelevant	"cat"/"dog"	0.702	18	20	0.691	15	19	0.912	24	29	0.949	34	52
		"foo"/"bar	0.770	18	21	0.616	15	17	0.908	24	29	0.950	34	52
		"black"/"white"	0.701	18	20	0.772	15	19	0.906	24	28	0.944	34	52
RTE	***default → default***	***"true"/"false"***	***0.819***	***18***	***27***	***0.813***	***17***	***31***	***0.857***	***24***	***45***	***0.854***	***34***	***79***
	default → relevant	"en"/"not"	0.788	18	21	0.773	17	18	0.868	24	28	0.839	34	52
		"yes"/"no"	0.745	18	21	0.771	17	20	0.863	24	31	0.854	34	52
		"good"/"bad"	0.810	18	21	0.724	17	20	0.846	24	30	0.829	34	52
	default → irrelevant	"cat"/"dog"	0.759	18	20	0.637	17	19	0.805	24	31	0.835	34	39
		"foo"/"bar	0.633	/	/	0.567	/	/	0.841	24	31	0.830	34	52
		"black"/"white"	0.620	18	20	0.609	17	19	0.833	24	28	0.838	34	39
ANLI	***default → default***	***"true"/"false"***	***0.723***	-	-	***0.679***	-	-	***0.851***	***24***	***45***	***0.854***	***34***	***79***
	default → relevant	"en"/"not"	0.693	-	-	0.589	-	-	0.816	24	28	0.841	34	52
		"yes"/"no"	0.721	-	-	0.659	-	-	0.843	24	30	0.821	34	52
		"good"/"bad"	*0.676*	-	-	0.606	-	-	*0.790*	24	*29*	*0.802*	34	*52*
	default → irrelevant	"cat"/"dog"	0.522	-	-	0.519	-	-	0.643	24	28	0.766	34	52
		"foo"/"bar	0.601	-	-	0.521	-	-	0.728	24	29	0.808	34	52
		"black"/"white"	0.511	-	-	0.542	-	-	0.710	24	28	0.794	34	52
IMDb	***default → default***	***"positive"/"negative"***	***0.844***	***18***	***27***	***0.873***	***17***	***31***	***0.889***	***24***	***45***	***0.913***	***39***	***79***
	default → relevant	"good"/"bad"	0.887	18	21	0.904	17	24	0.919	24	30	0.923	39	52
		"true"/"false"	0.733	18	21	0.772	17	21	0.912	24	28	0.918	39	52
	default → irrelevant	"cat"/"dog"	0.709	18	20	0.778	17	17	0.880	24	27	0.908	39	52
		"foo"/"bar"	0.617	18	21	0.821	17	18	0.740	24	28	0.903	39	52
		"water"/"fire"	0.760	18	20	0.816	17	18	0.870	24	28	0.888	39	51
AGNews	***default → default***	***"world"/"sports"/ "business"/"sci/tech"***	***0.822***	***18***	***27***	***0.852***	***16***	***31***	***0.850***	***25***	***45***	***0.874***	***35***	***79***
	default → irrelevant	"international"/"athletics"/"commerce"/ "innovation"	0.822	18	21	0.832	17	19	0.841	25	34	0.869	35	48
		"a"/"b"/"c"/"d"	0.810	18	19	0.752	16	18	0.834	25	25	0.856	35	47
		"1"/"2"/"3"/"4"	0.807	18	19	0.723	17	18	0.830	25	25	0.826	35	47
		"foo"/"bar"/ "baz"/"qux"	0.572	18	19	0.683	/	/	0.769	/	/	0.811	35	38
MultiNLI	***NLI → NLI classification (entail vs. not entail)***	***"true"/"false"***	***0.800***	***18***	***27***	***0.794***	**17**	***31***	***0.876***	***24***	***45***	***0.860***	***34***	***79***
	NLI → Domain classification (government vs. fiction)	"true"/"false"	0.606	18	20	0.864	16	18	0.744	24	28	0.841	40	52
	NLI → Domain classification (government vs. telephone)	"true"/"false"	0.713	18	20	0.951	16	18	0.936	24	28	0.987	41	52
	NLI → Detect negation (negation vs. no negation)	"true"/"false"	0.590	18	20	0.591	16	18	0.752	24	26	0.782	34	52
	NLI → Detect Overlap (overlap vs. non-overlap)	"true"/"false"	0.584	18	20	0.776	16	18	0.686	24	28	0.811	34	52

Table 5: **ICL Performance Across Settings and Identification of the “Platitude” Stage.** The *default ← default baseline* are italic and bold. For **remap label space** experiment, we delineate the approximate middle stage (start and end of layers) where we achieve the counterfactual scenario. This stage should see peak flip rates, positioned between two monotonic trends—one converging towards the baseline, the other diverging. For the **change task** experiment, we summarize the start and end of the first sub-phase (the first platitude we see), aligning with the middle stage of the **remap label space** setting. We do not conduct intervention on settings where the ICL performance is inadequate and note in the corresponding cell with “-” (for ANLI) or “/” (for label words such as “foo”/“bar" on RTE). Flip rates below 0.6 are highlighted in orange, and those below 0.7 in red.

Name	Type	Prompt Templates
sentence	Blank	Sentence 1: {premise}\nSentence 2: {hypothesis}\n{answer}
sent_label	Blank	Sentence 1: {premise}\nSentence 2: {hypothesis}\nLabel:{answer}
passage_label	Blank	Passage: {premise}\nLabel:{answer}
passage_sentiment	Imply	Passage: {premise}\nSentiment\n{answer}
text_linebreak	Blank	Text: {premise}\n{answer}
text_topic	Imply	Text: {premise}\nTopic:{answer}

Table 6: **Prompt templates:** We intentionally avoid using prompts with instructions and explicit hints for the verbalizers. We find that Llama-3.1-70B is very sensitive to the ending token of the prompt and often yields degenerated results when the prompt does not end with a newline token. Therefore we use prompts different from other models in IMDb and AGNews.

Task Name	Classification Task
default (nli)	Determine whether the first sentence entails the second sentence,
government files vs. fiction	Determine whether the two sentences are from government files or fiction.
government files vs. telephone recordings	Determine whether the two sentences are from government files or telephone recordings.
detect overlap	Determine whether all words from the second sentence also appear in the first sentence.
detect negation	Determine whether the second sentence includes any negation words ("not", "no", "n’t").

Table 7: **Alternative tasks for the change task setting**: We use the dataset from Si et al. (2023) from Si et al. (2023), which provides a variant of MultiNLI with labels for the alternative tasks.

Name	Task	Prompt Templates
ambi_inst	default (nli)	In this task, you will be presented with a premise sentence (the first sentence) and a hypothesis sentence (the second sentence). Determine whether the premise sentence entails (implies) or does not entail the hypothesis sentence. Please answer with \"{pos_label}\" for entailment and \"{neg_label}\" for non-entailment.
	government files vs. fiction	In this task, you will be presented with a premise sentence (the first sentence) and a hypothesis sentence (the second sentence). Determine whether they come from government files or fiction. Please answer with \"{pos_label}\" for government and \"{neg_label}\" for fiction.
	government files vs. telephone recordings	In this task, you will be presented with a premise sentence (the first sentence) and a hypothesis sentence (the second sentence). Determine whether they come from government files or telephone. Please answer with \"{pos_label}\" for government and \"{neg_label}\" for telephone.
	detect overlap	In this task, you will be presented with a premise sentence (the first sentence) and a hypothesis sentence (the second sentence). Determine whether all words in the second sentence also appear in the first sentence. If so, answer \"{pos_label}\"; if not, answer \"{neg_label}\".
	detect negation	In this task, you will be presented with a premise sentence (the first sentence) and a hypothesis sentence (the second sentence). Determine whether there are any negation words in the second sentence (\"not\", \"no\", \"n’t\"). Please answer with \"{pos_label}\" for not having negations and \"{neg_label}\" for having negations.

Table 8: **Instruction prompts for Change Task experiments:** We use the instructions from Si et al. (2023). {pos_label} and {neg_label} are replaced with their corresponding verbalizers.