Where does In-context Learning Happen in Large Language Models?

Suzanna Sia [*]
Johns Hopkins University
ssia1@jh.edu

David Mueller
Johns Hopkins University
dam@cs.jhu.edu

Kevin Duh
Johns Hopkins University
kevinduh@cs.jhu.edu

Abstract

Self-supervised large language models have demonstrated the ability to perform various tasks via in-context learning, but little is known about where the model locates the task with respect to prompt instructions and demonstration examples. In this work, we attempt to characterize the region where large language models transition from recognizing the task to performing the task. Through a series of layer-wise context-masking experiments on GPTNEO2.7B, BLOOM3B, LLAMA2-7B, LLAMA2-7B-CHAT, STARCODER2-3B and STARCODER2-7B on Machine Translation and Code generation, we demonstrate evidence of a "task recognition" point where the task is encoded into the input representations and attention to context is no longer necessary. We further observe correspondence between the low performance when masking out entire layers, and the task recognition layers. Taking advantage of this redundancy results in 45% computational savings when prompting with 5 examples, and task recognition achieved at layer 14 / 32 using an example with Machine Translation.

1 Introduction

In-context learning (ICL) refers to the phenomenon in which large generative pretrained transformers (GPTs) perform tasks with no gradient updates when shown task examples or descriptions in their context [13, 12]. There are many works on in-context learning which focused on *prompt-engineering*, treating GPT models as black boxes by focusing on which examples to provide in-context [47, 14, 19, 20, 51]. However, many of these works apply surface level interventions leaving the internal mechanism of task recognition in GPT models largely not understood.

In this work, we ask **where does in-context Learning occur** in GPT models? Our view of In-context Learning is that of “task recognition" not “task learning" [64, 41]. While in-context learning in GPT models appears to be generally applicable to any natural language task, to study task location, we focus on two tasks, Machine Translation (MT) and Code generation, as there is little to no ambiguity in evaluating whether the model has recognised the task. For MT, the model must generate tokens in a different language. For Code generation, the model must produce a working program which can be executed correctly. These two tasks are unlikely to be “learnt" from following patterns, and are more complex than a lookup in associative memory for simple Question-Answer tasks.

We focus on multi-head attention layers as a unit of study, as the self-attention mechanism is designed to allow the model to attend to it’s context during generation of the target sentence [54]. Using causal masking over different parts of the context we demonstrate that there exists a "task-recognition" point after which attention to the context is no longer necessary (Section 4). The potential implications are large computational savings when the context is several times longer than the test source sentence

[*] Corresponding Author, suzyahyah@gmail.com and Code Repository https://github.com/suzyahyah/where_does_in-context-learning_happen_in_LLMs

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

(Section 5). Having identified the layers in which "task recognition" occurs, we also conduct exploratory studies into the extent to which subsequent layers are either *redundant* or corresponding to the "task recognition" layers. Simple layer-wise masking shows that for smaller models, removing attention around the "task-recognition" layers can cause the model to fail to perform translation all-together, whereas layers towards the end of the model are much more redundant (Section 6.2).

Next, we observe that very lightweight fine-tuning of LoRA parameters [29] are most effective at earlier layers of the model compared to the later ones (Section 7.2). While there is not a strict 1-1 correlation in layers, this provides loose supports for the conjecture that earlier layers are more important for the task.

We further investigate the extent of MT *task redundancy* using differentiable L_0 regularisation to train discrete attention head gates (Section A.7). We find that around 10% of the attention heads can be masked, which indicates that the attention-heads themselves are not redundant, it is attention over all of the context that can be redundant. This fundamentally differs from the literature in supervised NMT where more than half of the attention heads can be pruned, and Transformers are highly specialised for MT [56, 38, 7].

2 Background

In-Context Learning was first demonstrated by [13] who showed that GPT-3 could be used to perform a huge variety of tasks without any task-specific parameters or training, by conditioning the model's generation on a *prompt* which included a few labeled examples of the task of interest. Since then, interest in using GPT models for ICL has grown significantly [39, 3, 61], with several recent works introducing methods such as instruction-tuning [49, 58] or chain-of-thought prompting [59] to improve downstream ICL accuracy. One key characteristic of In-context Learning is its reliance on prompt examples demonstrating the task that the model should carry out [46].

In-context Learning as Task Recognition. Ostensibly, ICL can work for nearly any task that can be defined or described in natural language, and therefore has potential for incredibly broad impact. However, ICL can often still underperform supervised fine-tuning [9], prompting research in analyzing the mechanisms underlying ICL. One line of work studies in-context learning with *linear* functions, typically linear regression, characterizing the learnability of these functions with ICL [33, 24] and even the learning algorithm a transformer uses [2, 17, 57]. A second body of work suggests that in-context learning locates *existing* latent concepts (tasks) which have been *already learnt* during pretraining [64, 60]. Notably, [52] describe function vectors which are robust to changes in context. [26] try to characterise the extent of task recognition from the pre-training data. Although there have been many studies on task recognition, our work presents a complementary perspective for task recognition, by demonstrating that there exists a point in the model's *layers* where the task has been located and causal self-attention onto the context is no longer needed for the model to perform the task.[2]

Transformer Layers and Self-attention as the Unit of Study. Many works study layers of the model as a natural unit of analysis for interpretability [28, 18, 42, 21, 8, 48]. We highlight some of the work which is more closely related to task performance. [63] study the layer-wise adaptability by a hidden-state variability ratio while [55] study evolution of representations in MT-supervised transformer models. [43] studies when model layers can be skipped by feeding intermediate representations into the final output layer of a pre-trained supervised model. Our work adds to this body of work by considering the perspective of when and where layers are responsible for task location in in-context learning models.

The self-attention mechanism specifically has been highlighted as a source of redundancy by many previous and concurrent works [10, 40, 27]. This is due to it's causal structure over the input symbols under the specific context of the input sequence within it's context window [45]. In this paper, we study a major source of causal redundancy in the input, the "prompt examples" that are provided as input-output demonstrations to the model for "in-context learning".

[2] In our experiments investigating where "task recognition" happens, we consider the actual "task performance" score as it could be possible to recognise the task as Machine Translation or Code generation, yet perform less well on it.

3 Data and Settings

Models We use GPTNEO2.7B [11], BLOOM3B [50], LLAMA2-7B and LLAMA2-7B-chat [53] in all of our experiments with Machine Translation. For code generation, we used LLAMA2-7B and LLAMA2-7B-chat, STARCODER2-3B and STARCODER2-7B[35].

GPTNEO2.7B has 32 layers and 20 heads, BLOOM3B has 30 layers and 32 heads, LLAMA2-7B has 32 layers and 32 heads and STARCODER2 has 30 layers and 24 heads. The checkpoints we use are from Meta AI (for LLAMA) and the transformers library [62]. Starcoder2 utilises grouped-query attention [1], while the rest of the models use "regular" multi-head self-attention.

GPTNEO was trained on The PILE [23], an 825GB text dataset which consists of roughly 98% English data. Despite being mostly monolingual, The PILE contains Europarl which GPTNEO was trained on at a document level (rather than a sentence level). Conversely, BLOOM was trained on the ROOTS corpus [32], a composite collection of 498 datasets that were explicitly selected to be multilingual, representing 46 natural languages and 13 programming languages. LLAMA training data consists primarily of common crawl, C4, wikipedia, stackexchange as major sources. STARCODER2 was trained on Github as well as Arxiv and Wikipedia. To our knowledge, there has not been any reports of sentence level parallel corpora in the training datasets of these models.

Data We test our models using two datasets, FLORES [25] for Translation and HUMANEVAL for Code generation. For FLORES, we experiment with `en`↔`fr` (main paper) and `en`↔`pt` (appendix). Prompt examples are drawn from the development set. We evaluate the generations using BLEU scores, following the implementation from [44]. For HUMANEVAL[15], we evaluate on the execution accuracy of the generated code using the Pass@1 metric. As HUMANEVAL does not have an explicit train set, the prompt set is drawn from the Mostly Basic Python Program (MBPP) dataset [4]. To account for example selection and ordering effects,[3] all inference runs were repeated with 5 randomly sampled prompt example sets.

Prompt Format Our prompts may consist of instructions, examples, both, or none. Importantly, we adopt *neutral* delimiters, "Q:" and "A:" to separate the prompt and the start of machine generated text. This ensures that the models do not have any information from the delimiters on what the task is and must recognise the task from examples. [4]

For the translation task, when no natural language instructions are used the model input will be `Q: {source_sentence} A:` Instructions are given in natural language and take the form: `Translate from {L1} to {L2}: Q: {source_sentence} A:`, where `L1 = English` and `L2 = French` if the source and target languages are English and French respectively. Examples are given after instructions, and similarly delimited by Q: and A:. See Appendix: Table 1 for an example.

For the code generation task, when no natural language instructions are used, the model input will be `Q: {program_description}`, where the `program_description` is Instructions are given in natural language and take the form: "Write a program for the following task:".

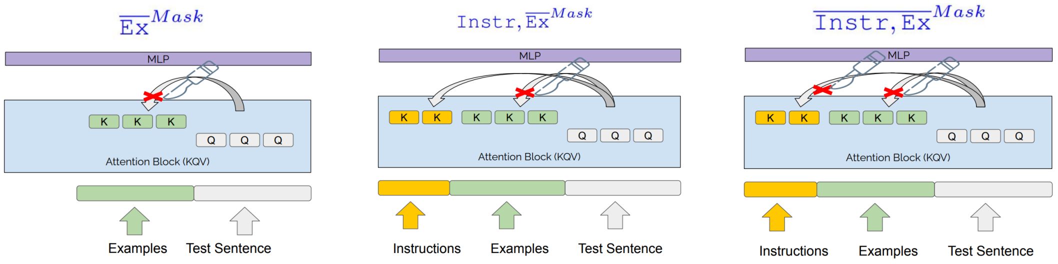
4 Where does In-context MT happen?

4.1 Analysis Methodology: Layer-from Context Masking

In-context learning differs from task-specific supervised learning in that, during test time, the desired task must be identified from the context first, then executed. At what stage in the feed-forward computation does a GPT-style model transition from an in-context learner to a translation model? To explore this question, we introduce *layer-from context-masking* which masks out all attention weights to the context (instructions or prompts) *from* a certain layer onwards (see Figure 1 for a graphical description).

[3] In-context learning models have been found to be sensitive to these order effects [36].

[4] In an earlier exploration, we found that supplying the model with language indicators only, e.g., "English:", "French:" or "English:", "Python:", was sufficient for strong models (llama7b, llama7b-chat, starcoder2-7B) to perform the task without seeing any instructions or examples in the context.



Name	$\texttt{Instr}$	$\texttt{Ex}$	
$\overline{\texttt{Ex}}^{Mask}\mathrm{Q}$	N	Y	Q: $\cdots$ A: $\cdots$ Q: $\cdots$ A:
$\texttt{Instr}, \overline{\texttt{Ex}}^{Mask}$	Y	Y	Translate French to English: Q: $\cdots$ A:$\cdots$ Q: $\cdots$ A:
$\overline{\texttt{Instr}, \texttt{Ex}}^{Mask}$	Y	Y	Translate French to English: Q: $\cdots$ A:$\cdots$ Q: $\cdots$ A:

Figure 1: Graphical explanation of Masking the Attention over Instructions and Examples. The leftmost image has instructions and masks examples ($\texttt{Instr}, \overline{\texttt{Ex}}^{Mask}$), while the right image has both instructions and examples masked ($\overline{\texttt{Instr}, \texttt{Ex}}^{Mask}$). In the table, the overline corresponds to the yellow highlights. N and Y refer to absence and presence of either Instruction of Examples. $\texttt{Instr}$: Instructions and $\texttt{Ex}$: Examples.

For Causal Decoder-only Transformer Language Models, given each position i, the Attention weight α_{ij} over context positions $j, j < i$ can be computed by a $\alpha_{ij} = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})_{ij}$. Each element in (QK^T) is the dot product between a query vector and key vector $q_i \cdot k_j$, where $q_i = W_q x_i, k_j = W_k x_j$ for trained weight matrices W_k and W_q.[5] We apply the attention mask over the context so that the attention score is $(q_i \cdot k_j) + m(j, \mathbf{u})$. Here $\mathbf{u}$ is the context that we wish to mask, and

$$m(j, \mathbf{u}) = \begin{cases} 0 & \text{if } x_j \notin \mathbf{u} \\ -\infty & \text{if } x_j \in \mathbf{u} \end{cases}$$

All masks operate from the j-th layer (ℓ_j) *onwards*, i.e. masking from ℓ_{20} means causally masking out attention to all context positions from $\ell_{20:n_\ell}$, where n_ℓ is the total number of layers. To construct Fig 2, we increment ℓ_j from 1 to n_ℓ and apply the set of masks $\{m(j, \mathbf{u})\}^{\ell_j:n_\ell}$ in each experiment and observe the performance of the model.

Under this causal masking treatment masking from layer ℓ, the model must rely on the hidden state representations of the target input sentence from layer $\ell + 1$ *only* and having self-attention on only the positions of the task test input *only* to complete the task; if the test input representations do not already encode the target task (e.g., Machine translation or Code generation) then the model will fail to generate the correct output.

In other words, the goal is to characterise where the model has "located" the task. In all experiments we mask the examples provided in the context, but to control for the effect of semantic instructions, we ablate over different treatments of the instructions by removing instructions entirely ($\overline{\texttt{Ex}}^{Mask}$), leaving them unmasked ($\texttt{Instr}\overline{\texttt{Ex}}^{Mask}$), or masking them together with the examples ($\overline{\texttt{InstrEx}}^{Mask}$). The overline notation indicates the context which are masking over (Figure 1).

We run this experiment for all layers in the model, $j = 1, \cdots, n_\ell$ and observe how task performance is affected, using 5 examples per prompt.

4.2 Results

We discuss the central findings of the paper: **Models do not need to maintain attention over all of the context across every layer to perform the task.**

[5] Readers should note that there is a W_k and W_q weight matrix for each layer and each attention head, but we omit the notation on this for readability.

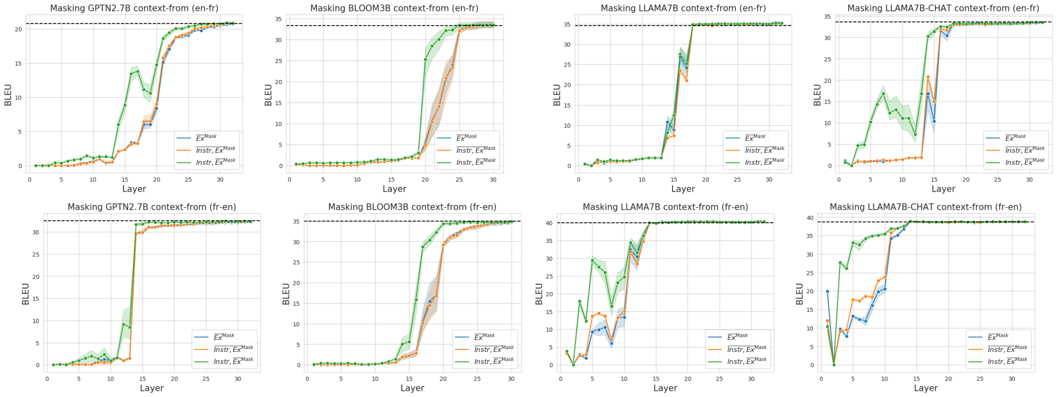


Figure 2: *Layer-from context-masking experiments* for GPTNeo2.7B, BLOOM3B, Llama7b, Llama7b-chat on `en` ↔ `fr`. The graphs show translation performance when masking contexts from the j^{th} layer onwards. Different lines indicate different treatments of the instruction, as described in Figure 1. The dashed black line is the performance when shown both examples and instructions without masking.

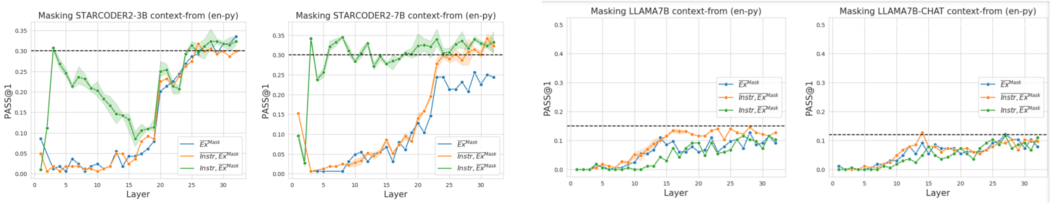


Figure 3: *Layer-from context-masking experiments* for Starcoder2-3B, Starcoder2-7B, Llama7b, Llama7b-chat on a text to code generation task. The graphs show translation performance when masking contexts from the j^{th} layer onwards. Different lines indicate different treatments of the instruction, as described in Figure 1. The dashed black line is the performance when shown both examples and instructions without masking.

In all models we observe that when applying masking from $\{m(j, \mathbf{u})\}^{\ell:n_\ell}$ over the context, performance plateaus before the final layer, i.e., when $\ell = n_\ell$. The results of our experiment for `en` → `fr` and `fr` → `en` are shown in Figure 3, and additional experiments for GPTNeo and Bloom on `en` → `pt` and `pt` → `en` are shown in Section A.2.

Different models reach this plateau point at different layers. In GPTNeo this point occurs around layer 25, in BLOOM this point occurs around layer 15-20, and in Llama models this occurs around layer 13-15. As English is the dominant language, as expected models can successfully perform translation into English upon earlier layers of masking, than translation out of English.

At this point, the models benefits only marginally, if at all, from attending to the context, suggesting most of the task "location" has already occurred.

There exists critical layers for task location. Prior to the task recognition point, around the middle layers of the models, moving the context mask up a layer results in a significant increase to performance. We consider these critical layers, as instead of a gradual increase in performance, we observe very steep jumps, accounting for more than 80% of the model's ceiling performance for that task. We conjecture that the model is locating the correct task during processing in these middle layers, after which the context is no longer necessary to perform the task.

Overall, our findings suggest a 3-phase process to in-context learning: in the first phase, moving the mask up makes little difference in performance, which is close to 0. This suggests that the context has not influenced task location at all. In the second phase, shifting the mask upwards makes a large difference in performance, suggesting that the model has started to locate the task but can improve significantly with more processing of the context. Finally, in the third phase, shifting the mask upwards again has little-to-no effect on the performance, suggesting that the model has fully recognized the task as translation and no longer requires the context to interpret the task.

4.3 Instruction-tuned vs Non-instruction Tuned Models

When comparing non-instruction tuned Llama2-7b vs instruction-tuned models Llama2-7b-chat, we do not observe any noticeable difference in where performance plateaus, i.e., where the model no longer requires attention over the context. This occurs around layers 18 for both Llama models in $\texttt{en} \rightarrow \texttt{fr}$ and around layer 14 for $\texttt{fr} \rightarrow \texttt{en}$. The main difference is that instruction-tuned model is able to achieve better performance in the earlier layers for the setting where instructions are present and examples are masked $(\texttt{Instr}, \overline{\texttt{Ex}}^{Mask})$. This is to be expected as these models are tuned towards following instructions.

Overall we find that the observation of task recognition layers and a task recognition point is present across both non-instruction tuned and instruction tuned models, and that this presents itself similarly in both types of models.

4.4 Do models have a distinct task recognition region regardless of the type of task? (Experiments on Code Generation)

For tasks that the model does not perform fluently, we do not observe a sharp increase at any particular layer. For instance, for code generation (HumanEval) where the llama2 model performs poorly, we can observe only a very gradual effect of masking the self-attention layers, and not a distinct increase as compared to the Llama2's performance on Translation.

However when we consider StarCoder2 while masking instructions or no instructions, i.e., the $\overline{\texttt{Instr, Ex}}^{Mask}$ and $\overline{\texttt{Ex}}^{Mask}$, we again see the same pattern demonstrating the task recognition phenomena on layer 19 of the 3B model, and layer 20-23 of the 7B model.

To understand Starcoder2's strong performance on the $(\texttt{Instr}, \overline{\texttt{Ex}}^{Mask})$ condition, investigations found that the instructions and the test prompt alone contain sufficient information for the model to recognise that the task is to generate a Python program, even though the model is not instruction tuned. This happens as the model is very specialised towards code generation and has a strong prior to generate python code given its prevalence in it's training data.

4.5 The Role of Instructions vs Examples

In separate experiments, we found that when shown only instructions and no examples, GPTNeo and Bloom models are unable to translate, and their performance is nearly at 0 BLEU Score. For GPTNeo and Bloom we see that the behavior of the model is similar when no instructions are present $(\overline{\texttt{Ex}}^{Mask})$ and when instructions are masked $(\overline{\texttt{Instr, Ex}}^{Mask})$. However, if the model is given complete access to instructions $(\texttt{Instr}\overline{\texttt{Ex}}^{Mask})$, it can use the intermediate processing of examples to reach baseline performance earlier.

5 Inference Efficiency

Speeding up transformer inference is of great interest to the community [22]. We highlight the potential of speeding up inference time as a direct consequence of identifying where task recognition occurs in the model and redundancy of self-attention processing. Our results indicate that we can achieve significant speedups in inference by removing the processing of context-tokens all-together after a certain point in the model, with little to no impact on downstream performance. Let ℓ_r be the r^{th} layer where we can mask out the attention of the context across subsequent layers and match the "ceiling" performance. Let k be the number of prompt examples, where each example consists of a pair of parallel sentences. Then, for a model with n_ℓ layers, the amount of processing in terms of speed and memory saved is approximately $(n_\ell - r)/n_\ell \times (k/k+1)$.

Using the example of Llama7b (32 layers) on $\texttt{en}{\rightarrow}\texttt{fr}$, we see from Figure 3 that the model is very close to it's ceiling score after processing the examples at layer 14 ($\ell = 14$). If we no longer need to process examples after $\ell = 14$, **under a prompt size of 5 the savings are approximately 45%.**

For instruction-tuned models which are typically deployed in production, even if we assume that no examples are provided, savings can be non-trivial as very long-form instructions are typically provided to the model in an attempt to control it's behavior (prompt engineering).

Although we had demonstrated the redundancy of self-attention over the input context, the significance of this computational savings extends to *all* components of the transformer during forward inference. Since all subsequent layers of forward inference no longer rely on computations on previous token positions, all processing related to those redundant token positions (from the task recognition layer onwards) can be effectively removed.

6 Characterising Redundancy in Layers

Recently, [48] found that many layers in pre-trained transformers can be dropped with little harm to downstream tasks; moreover, it is well known neural MT transformer models are known have several redundant heads which are not necessary during test time [56, 38, 7]. However, it is not clear if the same trends hold for *in-context MT* models, and how that redundancy is related to task location versus task execution. We focus on the task of `en` → `fr` in this set of experiments.

6.1 Attention to the Context vs Attention to the Input

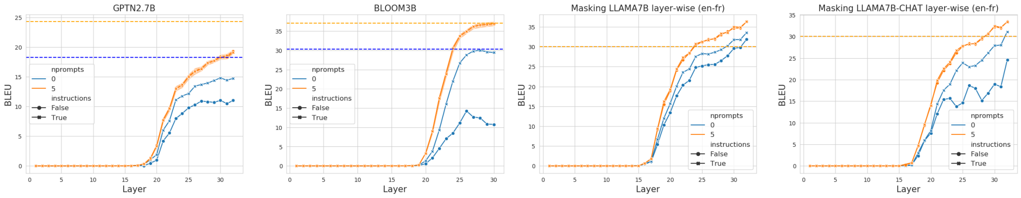


Figure 4: *Layer-from* experiments for GPTNEO2.7B, BLOOM3B, LLAMA and LLAMA7B-CHAT on `en` → `fr` when masking out from layer j onwards. Orange and blue dashed lines refer to the baselines (without masking) of 0 and 5 prompts with instructions. True/False refers to whether there are instructions provided (True) vs not provided (False). In view of the smaller models failure to translate at all under the format Q: A: with no examples, we adopt "English:", "French:" as delimiters instead of QA in generating this figure.

One possible explanation for the results in Figure 3 is that, rather than identifying the point at which the task is recognized (no longer requires attending to instructions and examples), we have identified the point at which the model no longer requires attending to *any* other input tokens (instructions, examples and source sentence). To explore this, we run experiments in the `en` → `fr` direction where we mask attention to *all inputs* from a certain layer onwards. This does not include masking over the text the model has generated.

We plot the results in Figure 4; we find that for all models, the layer at which attention can be fully removed is much higher than the layer at which we can remove attention to the context. For GPTNEO and LLAMA, translation performance is never comparable to the baseline with no masking. Conversely, when masking only the context, translation performance improves as early as layer 10 and plateaus at the no-mask baseline much earlier. This supports the interpretation that the curves we observe in Figure 3 are due to the model still requiring attention to the source sentence input.

We study the contributions of individual attention-layers by performing a simple *layer-wise* masking of all self-attention heads for a single layer. When we mask layer j, we are masking the *attention mechanism* of layer j, that is the MLP of layer j acts directly on the output of layer $j - 1$, rather than the output of the attention-head of layer j. Doing so allows us to study how *critical* each layer is, where *critical layers* is loosely defined as those that have a large negative impact when masked.

We plot results for each layer all models, using the three combinations of {0 examples, no instructions}, {5 examples, instructions}, {5 examples, no instructions} in Figure 5.[6]

[6] The combination of {0 examples, no instructions} is not meaningful as the model only receives "Q: <source sentence> A:" as the input and is not expected to do the translation task.

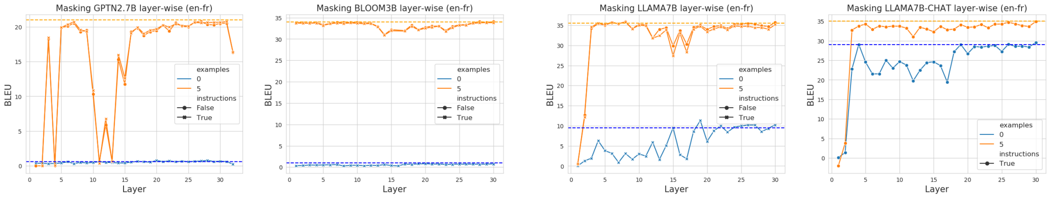


Figure 5: *Layer-wise masking* of self-attention heads for GPTNEO2.7B, BLOOM3B, LLAMA and LLAMA-CHAT on en $\rightarrow fr$. The orange and blue dotted lines refer to the baselines (without masking) of 0 and 5 prompts with instructions. True/False refers to whether there are instructions provided (True) vs not provided (False). We observe critical layers near the middle and redundant layers towards the end of the model.

6.2 Are “Critical" Layers Task Locating Layers?

In Section 4, we observed that there are layers for task location. In this section, we observe evidence that there are critical layers which correspond to the task locating layers, providing support for our earlier observations.

For instance for LLAMA7B en $\rightarrow$ fr, even in the scenarios when examples are provided, we can see a dip in performance around layer 15 to 18. Refering back to Figure 3, we see that this is where most of the task location with large jumps in performance had occurred.

For GPTNeo, we obseve a large set of contiguous layers which significantly decrease performance at around layer 10 to 15. This also corresponds to where most of the task location (large jumps in performance) had occurred for this model in Figure 3.

We note that the critical layers in different models have varying degrees of severity. It is not immediately clear why GPTNEO has such critical layers and suffers compared to the other models, although we note that this is unlikely to be due to size or model architecture as BLOOM is also around the same size as GPTNEO and performs more similarly to LLAMA. We suspect that it could be due to training data or some other factor related to the training dynamics but leave this for future work.

With regard to redundancy, we find that layers can be more safely removed towards the end without a noticeable loss in performance. We observe that for the less stable models, the model achieves close to baseline performance by layer-wise masking from ℓ_{15} for GPTNEO, ℓ_{26} for BLOOM and ℓ_{20} for LLAMA. This suggests that these later layers contain redundancy for translation.

Overall, observing redundancy in layers is not suprising. To explain why models can have redundant layers, we refer to [16] who identify a phenomena where attention heads attend almost exclusively to delimiter and separator tokens such as `[SEP]`, periods and commas. This is thought to act as a “no-op" as the value of such tokens in changing the current hidden representation is very small. Note that it is then possible to mask entire Transformer layers and still achieve a sensible output due to residual connections in the Transformer architecture at every layer.

7 Further Analysis

In the following sections, we focus on GPTNEO and BLOOM to conduct deeper analysis on the main phenomena presented in the paper.

7.1 Does the Number of Prompts Affect Task Recognition?

In Section 4 we study context-masking with a fixed number of prompts. However, it is not clear if the number of prompts affects how fast, layer-wise, the model is able to recognize the task. We plot these results for en$\rightarrow$fr in Figure 6, for both GPTNEO and BLOOM. In general, we find that the number of prompt examples has little effect on which layer the task is recognized at. While there is some variation in performance when the context is masked around the middle layers of the model, the final performance plateau occurs at the same layer regardless of the number of prompts.

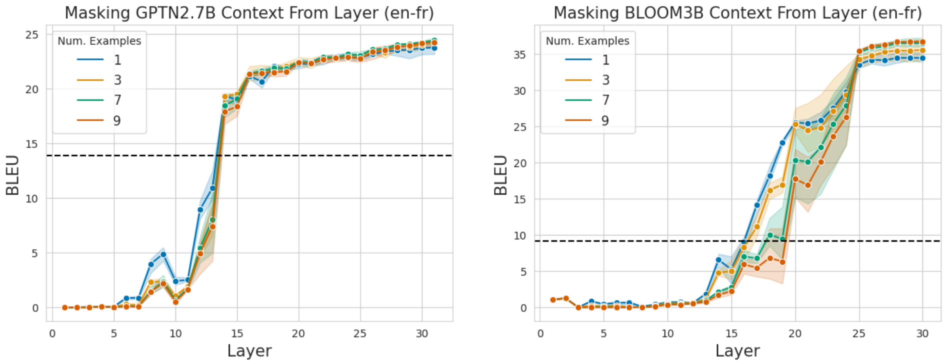


Figure 6: *Layer-from context-masking experiments* for GPTNeo and BLOOM on en → fr investigating number of examples in the $\overline{\texttt{Ex}}^{Mask}$ mask setting. The dashed black line refers to no instructions and no examples.

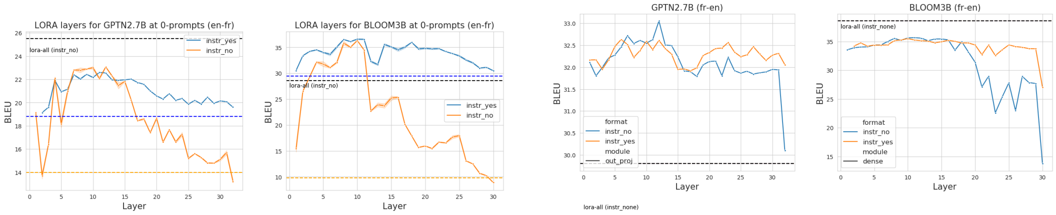


Figure 7: Performance of no-instructions trained Lora layers for GPTNeo and BLOOM on en↔fr. The dashed black line refers to training of all layers together, while the orange (test without instructions) and blue (test with instructions) dashed lines refers to no training. The layers which are most amenable to lightweight fine-tuning occur in the earlier layers before the "task recognition" point.

7.2 The Adaptability of Task Layers

Intuitively, the layers prior to "task recognition" should contain information about locating the MT task. To test this intuition, we further explore the adaptability of these layers by lightweight fine-tuning experiments. We trained a single Low-rank Adaptation matrix (LoRA; [29]) for each layer of the output projection while keeping the rest of the network frozen.[7] The model was shown parallel sentences as input, and layers were trained with no explicit translation instructions. We split the dev set of FLORES into 800 training examples and 200 dev examples. Note that this setup is designed to tune the layers for task location. It is highly unlikely that the model can learn translation knowledge from this small amount of supervision. The LoRA layers were trained for up to 50 epochs with early stopping patience$= 5$ and threshold$= 0.001$, with $\alpha = 32, r = 32$ and dropout$= 0.1$. The cross-entropy loss was computed only on the target sentence (see Section A.4 for details) and we used the best checkpoint on the 200 held out dev examples for evaluation.

While each layer can be trained to perform better than no fine-tuning at all, tuning different layers have different impacts on performance Figure 7. In particular, we find that high performing layers occur at the earlier to middle parts of the network, with the peak often occurring near the start of the "task-locating" layers from Section 4. In contrast to common fine-tuning wisdom, additional tuning on the later layers has a much smaller impact on final performance for en → fr.

7.3 Are There Specialised Attention Heads?

In Section 4, we found that the earlier part of the model is critical for *task location* from the prompt context, and in Section 6.2 we found both critical and redundant layers to the MT task. In this section, we increase the level of granularity to that of attention heads instead of layers.

A well established finding for supervised encoder-decoder MT models, is that up to 90% of the attention heads can be pruned while minimising fall in translation performance [56, 6, 38]. We note

[7] We also experimented with the training separate Key, Query and Value LoRA Layers but found this to be less effective.

that asking about the extent of pruning is a slightly ill-formed research question, as it depends on the type of pruning technique used. However broad trends of highly prunable models have been observed in the supervised MT paradigm. For instance, [5] studied attention-head importance for a broader set of ICL tasks, finding that the most important heads for ICL occur in the middle layers of the model. We train discrete attention head gates with L_0 regularisation for GPTNEO and BLOOM on `en` $\rightarrow$ `fr` (see Section A.5.1). Overall, we report that there are no "few" specialised heads, which directly contrasts with the literature on compression in supervised MT models [56, 38]. Potential reasons for this difference might be due to cross-entropy loss associated with task tuning for MT vs non-specific training on large corpora. We leave this as an open question for future work.

8 Conclusion

We demonstrate evidence that In-context Causal Decoder models locate their task at a specific layers during forward inference. To study this, we introduced causal masking of self-attention over the context from layer ℓ onwards (Section 4). The findings generalise across 4 models of different sizes and in both non instruction-tuned and instruction-tuned models. We further identify certain layers as task critical, and show that this corresponds to the task recognition point of the model (Section 6.2) and is not influenced by increasing number of examples (Section 7.1) shown to the models.

Our central finding that models do not need to maintain attention over all of the context across every layer has direct implications for inference efficiency of transformers, with estimated up to 45% cost-savings for llama model with 5 examples (Section 5).

Contrary to common fine-tuning wisdom, we show that it is sometimes beneficial to target middle layers for fine-tuning the model which could be associated with task recognition (Section 7.2). Finally, we trained attention head gates using differentiable L_0 regularisation (Section 7.3), and found that around 10% of attention heads can be masked. These are mostly distributed across the later layers of the model, providing some support for the idea that later layers are redundant. Although we have characterised this phenomena using Machine Translation and Code Generation, we believe that the broad findings are likely to generalise to other tasks.

8.1 Reproducibility

The Machine Translation dataset that we use, FLORES [25] and the code generation datasets HumanEval[15] are fully open-source and well-known in the community. All models are open-source and freely available on Huggingface [62]. We used models of "reasonable" size (3B and 7B parameters) that can be run with consumer or academic grade GPUs, making our work reproducible to most academic institutions.

8.2 Impact Statement (Ethics and Societal Consequences)

There are no known ethical concerns as these are exploratory studies on open-source LLMs.

8.3 Limitations (and Future Work)

- There is limited exploration of why different models exhibit varying behaviors in terms of their "task recognition point" and critical layers. Unfortunately, the differences are not due to easily observable hyperparameters like model size or architecture. To put in another way, why do large models exhibit different characteristics?
- This paper focuses on empirical analysis, and lacks a theoretical framework that could explain why this phenomena is being observed.

Acknowledgments

We would like to thank all the anonymous reviewers for their invaluable comments and suggestions, as well as Daniel Kashabi and Marc Marone for feedback on earlier drafts.

References

[1] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore, Dec. 2023. Association for Computational Linguistics.

[2] E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

[3] E. Akyürek, B. Wang, Y. Kim, and J. Andreas. In-context language learning: Arhitectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024.

[4] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[5] H. Bansal, K. Gopalakrishnan, S. Dingliwal, S. Bodapati, K. Kirchhoff, and D. Roth. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11833–11856, Toronto, Canada, July 2023. Association for Computational Linguistics.

[6] M. Behnke and K. Heafield. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2664–2674, 2020.

[7] M. Behnke and K. Heafield. Pruning neural machine translation for speed using group lasso. In *Proceedings of the sixth conference on machine translation*, pages 1074–1086, 2021.

[8] I. Ben-Shaul and S. Dekel. Nearest class-center simplification through intermediate layers. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pages 37–47. PMLR, 2022.

[9] K. Bhatia, A. Narayan, C. D. Sa, and C. Ré. Tart: A plug-and-play transformer module for task-agnostic reasoning, 2023.

[10] Y. Bian, J. Huang, X. Cai, J. Yuan, and K. Church. On attention redundancy: A comprehensive study. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 930–945, 2021.

[11] S. Black, G. Leo, P. Wang, C. Leahy, and S. Biderman. GPT-Neo: Large scale autoregressive language modeling with Mesh-Tensorflow, Mar. 2021.

[12] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[14] B. Chen, Z. Zhang, N. Langrené, and S. Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

[15] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[16] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[17] D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[18] N. De Cao, M. S. Schlichtkrull, W. Aziz, and I. Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, 2020.

[19] P. Denny, V. Kumar, and N. Giacaman. Conversing with copilot: Exploring prompt engineering for solving cs1 problems using natural language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 1136–1142, 2023.

[20] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[21] N. Durrani, H. Sajjad, F. Dalvi, and F. Alam. On the transformation of latent space in fine-tuned nlp models. *arXiv preprint arXiv:2210.12696*, 2022.

[22] Q. Fournier, G. M. Caron, and D. Aloise. A practical survey on faster and lighter transformers. *ACM Computing Surveys*, 55(14s):1–40, 2023.

[23] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[24] S. Garg, D. Tsipras, P. S. Liang, and G. Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[25] N. Goyal, C. Gao, V. Chaudhary, P.-J. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, and A. Fan. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. 2021.

[26] X. Han, D. Simig, T. Mihaylov, Y. Tsvetkov, A. Celikyilmaz, and T. Wang. Understanding in-context learning via supportive pretraining data. *arXiv preprint arXiv:2306.15091*, 2023.

[27] S. He, G. Sun, Z. Shen, and A. Li. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024.

[28] J. Hewitt and P. Liang. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*, 2019.

[29] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[30] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[31] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86, 2005.

[32] H. Laurençon, L. Saulnier, T. Wang, C. Akiki, A. Villanova del Moral, T. Le Scao, L. Von Werra, C. Mou, E. González Ponferrada, H. Nguyen, et al. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826, 2022.

[33] S. Li, Z. Song, Y. Xia, T. Yu, and T. Zhou. The closeness of in-context learning and weight shifting for softmax regression. *arXiv preprint arXiv:2304.13276*, 2023.

[34] C. Louizos, M. Welling, and D. P. Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

[35] A. Lozhkov, R. Li, L. B. Allal, F. Cassano, J. Lamy-Poirier, N. Tazi, A. Tang, D. Pykhtar, J. Liu, Y. Wei, et al. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*, 2024.

[36] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. 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, May 2022. Association for Computational Linguistics.

[37] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[38] P. Michel, O. Levy, and G. Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[39] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[40] B. Pan, R. Panda, Y. Jiang, Z. Wang, R. Feris, and A. Oliva. Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. *Advances in Neural Information Processing Systems*, 34:24898–24911, 2021.

[41] J. Pan. *What in-context learning "learns" in-context: Disentangling task recognition and task learning*. PhD thesis, Princeton University, 2023.

[42] A. Pasad, J.-C. Chou, and K. Livescu. Layer-wise analysis of a self-supervised speech representation model. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 914–921. IEEE, 2021.

[43] J. Phang, H. Liu, and S. R. Bowman. Fine-tuned transformers show clusters of similar representations across layers. *arXiv preprint arXiv:2109.08406*, 2021.

[44] M. Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

[45] R. Y. Rohekar, Y. Gurwicz, and S. Nisimov. Causal interpretation of self-attention in pre-trained transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] O. Rubin, J. Herzig, and J. Berant. Learning to retrieve prompts for in-context learning. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics.

[47] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[48] H. Sajjad, F. Dalvi, N. Durrani, and P. Nakov. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429, 2023.

[49] V. Sanh, A. Webson, C. Raffel, S. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. L. Scao, S. Biderman, L. Gao, T. Wolf, and A. M. Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

[50] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[51] S. Sia and K. Duh. In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. *arXiv preprint arXiv:2305.03573*, 2023.

[52] E. Todd, M. L. Li, A. S. Sharma, A. Mueller, B. C. Wallace, and D. Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

[53] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[55] E. Voita, R. Sennrich, and I. Titov. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. *arXiv preprint arXiv:1909.01380*, 2019.

[56] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics.

[57] J. von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent, 2023.

[58] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, E. Pathak, G. Karamanolakis, H. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, K. K. Pal, M. Patel, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. Doshi, S. K. Sampat, S. Mishra, S. Reddy A, S. Patro, T. Dixit, and X. Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[59] J. Wei, X. Wang, D. Schuurmans, M. Bosma, brian ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[60] N. Wies, Y. Levine, and A. Shashua. The learnability of in-context learning. *arXiv preprint arXiv:2303.07895*, 2023.

[61] N. Wies, Y. Levine, and A. Shashua. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[62] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[63] S. Xie, J. Qiu, A. Pasad, L. Du, Q. Qu, and H. Mei. Hidden state variability of pretrained language models can guide computation reduction for transfer learning. *arXiv preprint arXiv:2210.10041*, 2022.

[64] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

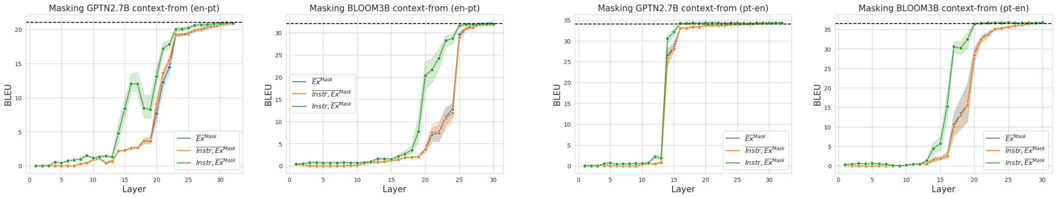


Figure 8: Context-masking and Layer-masking results on the **English** ↔ **Portugese** translation task. Critically, we see nearly identical trends to what we see in Figure 3 and Figure 5 on the English to French translation task, suggesting our results generalize across language pairs.

A Appendix

A.1 Prompt Format

Translate English to French.		
Q: A discomfort which lasts ..	A:	Un malaise qui dure
Q: HTML is a language for formatting	A:	HTML est un langage de formatage
...		...
Q: After you become comfortable with formatting ..	A:	

Table 1: A single continuous input sequence presented to the model for decoding a single test source sentence “After you become comfortable with formatting..”. Given the entire sequence as input, the model proceeds to generate the target sequence.

A.2 Additional Results on English & Portugese

In addition to the language pairs `en` → `fr` and `fr` → `en`, we also run experiments on English and Portugese language pairs, both `en` → `pt` and `pt` → `en`. Due to space limitations, we plot the results of those experiments here. Overall, we see largely identical trends on both directions of English and Portugese to what we observe on English and French translation tasks, leading us to conclude that our conclusions generalize across different translation tasks.

A.3 Autoregressive Decoder only Transformer

The transformer consists of stacked blocks of self-attention, which itself consists of smaller units of self-attention heads that are concatenated before being fed through a fully connected layer. In autoregressive decoder-only transformers, training and inference adopts a causal mask, where current positions are only able to attend to previous timesteps, instead of being able to attend to the entire input sequence. Unlike encoder-decoder NMT models where source and target sentence have separate processing transformer blocks, decoder-only means that the same model weights are both used to “encode" the source sentence and “decode" the target sentence in a single continuous sequence.

A.4 Training with Autoregressive Translation

The original language modeling objective in GPT training involves predicting the entire input token sequence which consists of both the source and target sentence (shifted by 1 position). We found this to produce slightly worse results than only minimising the negative log likelihood of predicting the target sentence to be translated, and not the entire sequence. We consider this autoregressive translation training.

A.5 L_0 Attention Gate Training

Training Details For Section A.7, We train using Adam Optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) with a batch size of 32, and learning rate of 0.001, early stopping patience of 10 and threshold of 0.01. We initialise attention head gates to be 1 instead of random or 0.5 as this leads to faster convergence. We experiment with two different training settings, the `0-prompts Train` setting and the `5-prompts Train` setting. As described in Section A.4, we train the model by predicting only the target sentence, conditioned on the context. In the 0-prompt setting, the context consists of the instructions and the source sentence to be translated. In the 5-prompt setting, the context consists of the instructions, 5 prompt examples, and the source sentence to be translated.

In the `0-prompt` setting, the conditional prefix consists of the instructions and the source sentence to be translated. In the `5-prompt setting`, the conditional prefix consists of the instruction, 5 source target sentence pairs, and the source sentence to be translated.

Data We used the first 10,000 lines of `en`→`fr` from WMT06 Europarl [31] for training.[8] To test the generalisability of trained attention head gates, we use a different test domain, FLORES [25] to reflect the scarcity of in-domain data. We also test an additional language direction `en`→`pt` in FLORES to see if training can generalise across languages.

Training Details We train using Adam Optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) with a batch size of 32, and learning rate of 0.001. We use a large early stopping patience of 10 and threshold of 0.01, and train for up to 100 epochs. This is due to the nature of L_0 training; we do not expect performance to improve over many iterations and would like the attention gates to keep training as long as there is no large loss in performance. We initialise attention head gates to be 1 instead of random or 0.5 as this leads to much faster convergence and better performance. For the regularisation weight λ, we search over a hyperparameter set of $\{0.1, 0.01, 0.001, 0.0001\}$ and found 0.01 performs best on the validation set.

A.5.1 L_0 head masking experiments.

Additional experiments on L0 head masking in the en→ fr direction.

A.5.2 Training Attention Head Gates with L_0 regularisation

For a scalable approach to pruning, we opt to train self-attention head gates following [56] using the technique of differentiable L_0 regularization [34]. Let the attention head gates $g \in \mathbb{R}^{n_h \times n_\ell}$ be a set of trainable parameters, where n_h is the number of attention heads per layer, and n_ℓ is the number of layers. Let the original output of each attention head be v_j, gated outputs $\tilde{v}_j$ are obtained by elementwise multiplication of the gate value g_j, i.e., $\tilde{v}_j = g_j \odot v_j$. For $\{(x, y)\}^n$ source sentence (x) and target sentence (y) training pairs, a model f and loss function $\mathcal{L}$, L_p regularisation adds a λ weighted penalty associated with the complexity of the parameters. [9] The L_0 loss is non-differentiable as it involves raw counts of parameters. As a work around, g can be approximated with random variables drawn from a Binary concrete distribution [37, 30].[10] We refer the reader to [34] for the relevant technical exposition. Details of training are provided in Section A.5.

A.5.3 Generalisability of L_0 gate training

We experiment with `0-prompts` and `5-prompts` in training and using $\lambda{=}0$ (no regularisation) and $\lambda{=}0.01$. L_0 training for the `0-prompts` shows some gains for the 0-prompts test case, and with no loss on the 5-prompts test case (Table 2). Notably, this persists in `en` → `pt`, a different language direction from training.

The robustness of translation performance under multiple testing conditions (number of prompts, datasets, language directions) gives some confidence that the trained discrete attention head gates from L_0 support a general ability to translate (Table 2). In contrast, the soft attention head gates without regularisation ($\lambda = 0$) appear to overfit as they perform well on some conditions but deteriorate in others.

We observe that `0-prompt` training for $L_0(\lambda = 0.01)$ also outperforms `5-prompts` which is slightly suprising since `5-prompts` has more information in the prefix to locate the translation task.

[8] Data available from `https://www.statmt.org/europarl/`

[9] L_2 regularisation has the effect of reducing the magnitude of all g, L_1 regularisation has the effect of reducing the magnitude of several attention heads to a very small value (but not exactly 0), while L_0 regularisation has the effect of driving g values to exactly 0.

[10] The class of Concrete distributions was invented to work around the problem of automatic differentiation of stochastic computation graphs.

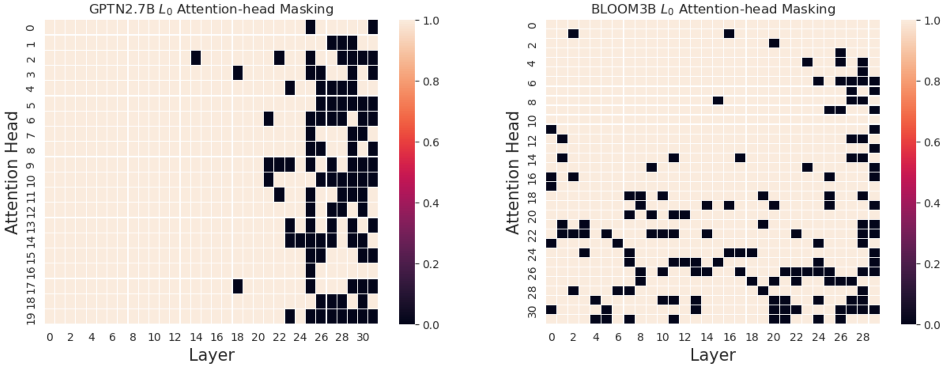


Figure 9: Visualisation of attention head masks for GPTNeo and BLOOM, learned with $L_0(\lambda = 0.01)$ regularisation under a `0-prompt train` scheme in `en` → `fr`. A value of 0 (in black) indicates that the attention head is effectively masked out by the trained attention gate. Around 10% of attention heads are masked out i.e., redundant, with a majority of them occuring at the later layers for GPTNeo and distributed across layers for BLOOM. `fr` → `en` is availble in Section A.5.1

	Base	`0-prompts`		`5-prompts`		Base	`0-prompts`		`5-prompts`	
		$\lambda=0$	$\lambda=.01$	$\lambda=0$	$\lambda=.01$		$\lambda=0$	$\lambda=.01$	$\lambda=0$	$\lambda=.01$
0-prompts	18.3	21.4	20.1	18.9	19.3	6.7	15.7	8.6	13.2	6.4
5-prompts	24.3	24.5	24.1	23.6	24.2	25.9	19.6	25.8	24.3	26.0
	Train: `en`→`fr`, Test: `en`→`fr`					Train: `en`→`fr`, Test: `en`→`pt`				

Table 2: Performance when using trained attention head gates for L_0 with regularisation $\lambda = .01$. $\lambda = 0$ refers to training without regularisation. `0` and `5` prompts were used in the context for training. We highlight values which are greater or worse than 0.5 BLEU points from baseline. Note that as these are compression experiments, we do not expect L_0 to perform better than baseline.

One possibility is that the model overfit to the Europarl domain where the training prompts were drawn from.

A.6 Qualitative Analysis of Layer-wise Masking

GPTNeo Masking $\ell_{4:8}$ results in a drop in performance for the 0-prompt setting but not the 5-prompt setting (Figure 5), which suggests that $\ell_{4:8}$ are **not** related to the processing of prompt examples. We emphasise that this interpretation mostly holds at an aggregate level and is not strictly for each instance. For Test Instance ID 575, the model still generates a copy of the English source sentence up to the masking of ℓ_{25} for the 0-prompts without instructions setting (Table 4). This suggests that uncertainty over the task is maintained across layers even though the contributions towards *task location* may be greater from specific layers.

Bloom is observed to be more robust to masking of layers; suggesting that task location is more distributed. For the 5-prompt setting, the performance only decreases very slightly. For the 0-prompt setting, we observe that similar to GPTNeo, performance drops when masking out the middle layers. At the aggregate level, Bloom appears to still be translating (> 0 BLEU) even when layers are masked. However we observe that the drop in performance is due to around 40 to 50% of the test sentences scoring < 5 BLEU points. There is a clear failure to translate, not simply producing poorer translations.

layer	id	lang	BLEU	text
1	600	cy	0.00	uffose
1	575	ca	0.00	B marriages{
2	600	et	0.00	sses room (I
2	575	no	0.00	NaN
3	600	fr	1.90	C'est la même chose que l'on a fait avec les virus.
3	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
4	600	no	0.00	NaN
4	575	no	0.00	NaN
5	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
5	575	fr	72.98	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent toujours à un dinosaur.
6	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
6	575	fr	60.29	Mais il y a beaucoup de choses à propos de oiseaux qui ressemblent encore à un dinosaur.
7	600	fr	13.94	Hershey et Chase ont implanté leur propre gène dans un bactérie.
7	575	fr	72.98	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent toujours à un dinosaur.
8	600	no	0.00	NaN
8	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosau.
9	600	no	0.00	NaN
9	575	fr	82.82	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent toujours à un dinosaure.
10	600	no	0.00	NaN
10	575	en	2.73	But there are a lot of things about birds that still look like a dinosaur.
11	600	en	4.78	Hershey and Chase used phages, or viruses, to implant their own DNA into a bacterium.
11	575	en	2.73	But there are a lot of things about birds that still look like a dinosaur.
12	600	no	0.00	NaN
12	575	no	0.00	NaN
13	600	no	0.00	NaN
13	575	fr	35.75	Mais il y a beaucoup de choses que je ne comprends pas.
14	600	en	4.78	Hershey and Chase used phages, or viruses, to implant their own DNA into a bacterium.
14	575	en	2.73	But there are a lot of things about birds that still look like a dinosaur.
15	600	fr	76.48	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bacillus.
15	575	en	2.73	But there are a lot of things about birds that still look like a dinosaur.
16	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
16	575	fr	70.18	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent toujours comme un dinosaurof.
17	600	fr	82.32	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans une bactérie.
17	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
18	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre génome dans un bactérie.
18	575	fr	66.38	Mais il y a beaucoup de choses sur les oiseaux qui aussi ressemble à un dinosaures.
19	600	fr	59.33	Les héritiers de Hershey et de Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
19	575	fr	47.91	Mais il y a beaucoup de choses à propos de les oiseaux qui ressemblent toujours à un dinosaur.
20	600	fr	48.82	Hershey et Chase ont utilisé les phages, ou les virus, pour implanter leur propre gène dans un bactérie.
20	575	en	2.73	But there are a lot of things about birds that still look like a dinosaur.
21	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
21	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
22	600	fr	48.82	Hershey et Chase ont utilisé les phages, ou les virus, pour implanter leur propre gène dans un bactérie.
22	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaurof.
23	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
23	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
24	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre génome dans un bactérie.
24	575	fr	62.72	Mais il y a beaucoup de choses à propos de les oiseaux qui ressemblent encore à un dinosaur.
25	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
25	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
26	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
26	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
27	600	fr	66.28	Hershey et Château ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
27	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
28	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
28	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
29	600	fr	59.33	Les héritiers de Hershey et de Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
29	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaurof.
30	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
30	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
31	600	fr	78.78	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
31	575	fr	88.44	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
32	600	fr	6.44	Les héritiers de Hershey et de Chase ont été capables de l'implanter dans un bactérie.
32	575	fr	51.52	Mais il y a beaucoup de choses sur les oiseaux que sont encore aussi vus comme un dinosaures.

Table 3: 0-prompts with instructions, masking layer by layer of GPTNEO2.7B

layer	id	lang	BLEU	text
1	575	no	0.0	NaN
1	600	no	0.0	NaN
2	575	NaN	0.0	,
2	600	no	0.0	NaN
3	575	fr	2.3	C’est pas un oiseau, c’est un dinosaur.
3	600	fr	0.8	[phare] Phare, phare, phare, phare, phare, phare, phare, phare, phare, phare, phare, ...,
4	575	en	2.7	"I think it’s a dinosaur, I think it’s a dinosaur."
4	600	fr	2.6	Les virus, c’est-ce qu’on dit? C’un mot? C’est pas un mot? C’un mot? C’un’un? ...
5	575	fr	42.9	Mais il y a beaucoup de choses à propos de oiseaux qui ressemblent toujours comme un dinosaur.
5	600	fr	73.6	L’Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
6	575	fr	53.8	Mais il y a beaucoup de choses à propos de oiseaux qui ressemblent toujours à un dinosaure.
6	600	fr	74.9	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
7	575	fr	76.2	Et il y a beaucoup de choses sur les oiseaux qui ressemblent toujours à un dinosaure.
7	600	fr	83.8	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bactérie.
8	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosau.
8	600	fr	83.8	L’usine Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bactérie.
9	575	en	1.5	The bird is a dinosaur.
9	600	fr	33.9	Les hémoglobine et Chase utilisent les phages, ou les virus, pour implanter leur propre gène dans un bactérie.
10	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
10	600	fr	11.4	Les phages, ou virus, ont implanté leur propre gène dans un bactérie.
11	575	en	2.6	I think it’s a good idea to have a little bit of a bird in your pocket.
11	600	en	0.0	The French have a saying: "The French have a saying: "The French have a saying: "The French have a saying:...
12	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
12	600	en	1.7	The bacterium was then able to use the phage to infect other bacteria.
13	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
13	600	fr	18.7	L’entreprise Hershey a utilisé des phages pour implanter leur propre DNA dans leur bactérie.
14	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
14	600	en	4.8	Hershey and Chase used phages, or viruses, to implant their own DNA into a bacterium.
15	575	fr	3.0	C’est pas un truc de poulet, c’est un truc de poulet.
15	600	fr	35.7	L’université de Paris-Sud a utilisé des phages, ou viraux, pour implanter leur propre gène dans un bacillus.
16	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
16	600	fr	74.9	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre génome dans un bactérie.
17	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
17	600	fr	82.3	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans une bactérie.
18	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
18	600	fr	74.9	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre génome dans un bactérie.
19	575	fr	44.2	Mais il y a beaucoup de choses à propos de oiseaux qui ressemblent toujours à un dinosaur.
19	600	fr	59.3	Les héritiers de Hershey et de Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
20	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
20	600	fr	46.4	Les Hershey et Chase ont utilisé les phages, ou les virus, pour implanter leur propre gène dans un bactérie.
21	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
21	600	fr	74.9	L’usine Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
22	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
22	600	fr	56.3	Les Hershey et Chase ont utilisé les phages, ou les virus, pour implanter leur propre ADN dans un bactérie.
23	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
23	600	fr	82.9	L’Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bactérie.
24	575	fr	60.3	Mais il y a beaucoup de choses à propos de oiseaux qui ressemblent encore à un dinosaur.
24	600	fr	37.0	L’usine Hershey et Chase utilisaient les phages, ou les virus, pour implanter leur propre génome dans un bactérie.
25	575	en	2.7	But there are a lot of things about birds that still look like a dinosaur.
25	600	fr	74.9	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
26	575	fr	71.6	Mais il y a beaucoup de choses à propos de oiseaux qui ressemblent encore à un dinosaure.
26	600	fr	73.6	L’Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
27	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaurof.
27	600	fr	63.0	Les Hershey et Château ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
28	575	fr	44.2	Mais il y a beaucoup de choses à propos de oiseaux qui ressemblent toujours à un dinosaur.
28	600	fr	74.9	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
29	575	fr	87.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaurofrench:
29	600	fr	53.4	L’entreprise de la filière Hershey a utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
30	575	fr	82.8	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent toujours à un dinosaure.
30	600	fr	74.9	Les Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
31	575	fr	82.8	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent toujours à un dinosaure.
31	600	fr	59.3	Les hémoglobins de Hershey et de Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
32	575	fr	67.5	Mais il y a beaucoup de choses sur les oiseaux qui aussi ressemblent à un dinosaurof.
32	600	fr	6.7	L’Hershey et le Chase ont été capables de l’implanter leur propre gène dans un bactérie.

Table 4: 0-prompts without instructions, masking layer by layer of GPTNEO2.7B

layer	id	lang	BLEU	text
1	902	en	0.0	: of
1	575	en	0.0	of
2	902	en	0.0	of(n, very very- ofS First
2	575	da	0.0	f(
3	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
3	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
4	902	en	0.0	the, the French, the French, the English, the, the, the, the, the, the, the, the, the, the,
4	575	no	0.0	NaN
5	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
5	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
6	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
6	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
7	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
7	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
8	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
8	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
9	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
9	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
10	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
10	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
11	902	fr	1.4	Fondamentalement, vous afficherez des annonces pour proposer votre aide, arpenterez les quais, ...
11	575	fr	1.3	Fondamentalement, vous afficherez des annonces pour proposer votre aide, arpenterez les quais, ...
12	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
12	575	fr	42.5	Mais il y a beaucoup de choses qui ressemblent à un dinosaur.
13	902	fr	34.5	Les scènes sont déclarées sur les pyramides et les pyramides sont déclarées sur les pyramides.
13	575	fr	5.5	Les oiseaux sont des animaux, mais ils sont aussi des êtres humains.
14	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
14	575	fr	73.3	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent à un dinosauresque.
15	902	fr	64.0	Les scènes sont affichées sur les pyramides et les pyramides différents sont éclairés.
15	575	fr	26.7	Mais il y a beaucoup de choses à propos de la façon dont les oiseaux se ressemblent, même si c'est un peu plus tard.
16	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
16	575	fr	76.7	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore comme un dinosaures.
17	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
17	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
18	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
18	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosauresque.
19	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont illuminés.
19	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
20	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
20	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
21	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
21	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
22	902	fr	64.0	Les scènes sont affichées sur les pyramides et les pyramides différents sont éclairés.
22	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosauroide.
23	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
23	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
24	902	fr	78.3	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairees.
24	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
25	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
25	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
26	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
26	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
27	902	fr	100.0	Les scènes sont affichées sur les pyramides et les différentes pyramides sont éclairées.
27	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
28	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
28	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
29	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
29	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
30	902	fr	78.3	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairees.
30	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
31	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont éclairés.
31	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
32	902	fr	65.9	Les scènes sont affichées sur les pyramides et les différents pyramides sont illuminées.
32	575	fr	67.5	Mais il y a beaucoup de choses sur les oiseaux qui aujourd'hui ressemblent à un dinosauro.

Table 5: 5-prompts with instructions, masking layer by layer of GPTNeo2.7B

layer	id	lang	BLEU	text
1	600	tl	0.0	*- ing
1	575	en	0.6	fl.of,
2	600	en	1.6	in " " - (–, -, - (es," " " so " whats " whats" " between whats –what e,
2	575	en	2.3	" ",what awaited ico " " " "_, .
3	600	fr	86.6	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans une bactérie.
3	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
4	600	en	0.3	the, etc.
4	575	ro	0.0	are: are: are: are: are: are: are: are: are: are: are: are: are: are: are: are: are: ...
5	600	fr	76.5	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bacille.
5	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
6	600	fr	88.1	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bactérie.
6	575	fr	62.7	Mais il y a beaucoup de choses à propos de les oiseaux qui ressemblent encore à un dinosaur.
7	600	fr	88.1	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bactérie.
7	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
8	600	fr	85.7	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bacillus.
8	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
9	600	fr	78.8	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
9	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
10	600	fr	30.1	En gros, vous mettre en place des phages, ou viraux, pour implanter leur propre gène dans un bactérie.
10	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
11	600	fr	1.3	Fondamentalement, vous afficherez des annonces pour proposer votre aide, arpenterez les quais, aborderez les personnes nettoyant leurs yachts, ...
11	575	fr	1.4	Fondamentalement, vous afficherez des annonces pour proposer votre aide, arpenterez les quais, aborderez les personnes nettoyant leurs yachts, .
12	600	fr	12.4	Les phages sont utilisés pour implanter leur propre gène dans un virus.
12	575	fr	36.8	Mais il y a beaucoup de choses qui semblent être des dinosaures.
13	600	fr	4.3	Les phages sont des virus qui sont implantés dans la cellule d'un organisme.
13	575	fr	5.5	Les oiseaux sont des animaux, mais ils sont aussi des êtres humains.
14	600	fr	74.4	Hershey et Chase ont utilisé des phages, ou virus, pour implanter leur propre ADN dans un bactérie.
14	575	fr	73.3	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent à un dinosauresque.
15	600	fr	62.6	Hershey et Chase ont utilisé des phages, ou virus, pour implanter leur propre gène dans un bacille.
15	575	fr	26.7	Mais il y a beaucoup de choses à propos de la façon dont les oiseaux se ressemblent, même si c'est un peu plus tard.
16	600	fr	85.7	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bacillus.
16	575	fr	76.7	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore comme un dinosaures.
17	600	fr	85.7	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bacille.
17	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosauresque.
18	600	fr	85.7	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bacille.
18	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosauresque.
19	600	fr	76.5	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bacille.
19	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
20	600	fr	76.5	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bacillus.
20	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
21	600	fr	88.1	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bactérie.
21	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
22	600	fr	85.7	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bacille.
22	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosauro.
23	600	fr	88.1	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bactérie.
23	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
24	600	fr	85.7	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre ADN dans un bacille.
24	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
25	600	fr	76.5	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bacillus.
25	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
26	600	fr	76.5	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bacille.
26	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
27	600	fr	55.9	Hershey et Château utilisaient des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
27	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
28	600	fr	76.5	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bacillus.
28	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
29	600	fr	78.8	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
29	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
30	600	fr	78.8	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
30	575	fr	88.4	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaur.
31	600	fr	78.8	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
31	575	fr	100.0	Mais il y a beaucoup de choses sur les oiseaux qui ressemblent encore à un dinosaure.
32	600	fr	78.8	Hershey et Chase ont utilisé des phages, ou des virus, pour implanter leur propre gène dans un bactérie.
32	575	fr	57.2	Mais il y a beaucoup de choses sur les oiseaux que pourraient encore ressembler à un dinosaures.

Table 6: 5-prompts without instructions, masking layer by layer of GPTNEO2.7B

A.7 Studying Redundancy via Compression

We noted that GPTNEO has some critical differences from BLOOM and LLAMA in terms of having critical layers (see Section 6.2). To what extent are there specialised attention heads for MT in the GPT-style models? If there were specialised heads, we would expect the model to be highly compressable/prunable to a select few heads. We plot a grid map of learned attention gate values for `en` → `fr`, where 0 indicates that the head is masked out (Figure 9). We find that most of the masked heads are distributed at the later layers for GPTNeo and are distributed across layers for BLOOM. This appears consistent with Section 6.2's observations that redundancy is more focused at certain layers in GPTNeo, and more spread out across the layers for Bloom.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We outline our contributions precisely in our abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our limitations in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not present theoretical results in this work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe hyperparameters, models, and methods clearly. Our models are downloaded from Huggingface and are all opensource.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will provide access to all of the code used to run experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We use publicly available dataset splits and our code will detail precisely what samples are used during in-context learning.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our core results present error bars over different random seeds and sets of in-context exemplars when applicable.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information to reproduce all of our experiments (all of our experiments can be run on consume-grade GPUs such as RTX 6000).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We adhere to the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work contains an Impact Statement in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All of our data and models are publicly available, and we reference their sources in the main paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: N/A

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.