Cross-Layer Attention Probing for Fine-Grained Hallucination Detection

Malavika Suresh[1,*], Rahaf Aljundi[2], Ikechukwu Nkisi-Orji[1] and Nirmalie Wiratunga[1]

[1]Robert Gordon University, Aberdeen, United Kingdom
[2]Toyota Motor Europe, Brussels, Belgium

Abstract

With the large-scale adoption of Large Language Models (LLMs) in various applications, there is a growing reliability concern due to their tendency to generate inaccurate text, i.e. *hallucinations*. In this work, we propose Cross-Layer Attention Probing (CLAP), a novel activation probing technique for hallucination detection, which processes the LLM activations across the entire residual stream as a joint sequence. Our empirical evaluations using five LLMs and three tasks show that CLAP improves hallucination detection compared to baselines on both greedy decoded responses as well as responses sampled at higher temperatures, thus enabling fine-grained detection, i.e. the ability to disambiguate hallucinations and non-hallucinations among different sampled responses to a given prompt. This allows us to propose a detect-then-mitigate strategy using CLAP to reduce hallucinations and improve LLM reliability compared to direct mitigation approaches. Finally, we show that CLAP maintains high reliability even when applied out-of-distribution.

Keywords

hallucination detection, activation probing, large language models

1. Introduction

Large Language Models (LLMs) have become increasingly accessible and scalable for commercial use, largely due to the API-based access offered by several LLM platform providers. From AI-generated summaries in search engines to chatbots in various health and business sector applications, LLM generated text is being widely consumed by a large proportion of the population. Such widespread adoption increases the risk of spreading misinformation and causing harm to users through factually incorrect LLM-generated text, i.e. *hallucinations*. Improving the trustworthiness by detecting and mitigating hallucinations is therefore an important research objective.

Current approaches for tackling LLM hallucinations fall under three broad categories: black-box, grey-box and open-box methods. Among open-box methods, some recent works [1, 2, 3] have established the potential of building binary hallucination detectors using raw LLM activations as input, termed *activation probing*. Other works have shown that hallucinations can be mitigated by directly editing activations [4, 5] or output probabilities [6] at generation time. However, the effect of activation-editing on other model capabilities is not well understood. While prior work on activation probing has focused on individual layers, in this work we introduce Cross-Layer Attention Probing (CLAP), a fundamentally different approach that utilises the full residual stream (i.e. activations from all LLM layers) to probe model behaviour more comprehensively. We hypothesise that the contribution of activations at different layers to hallucination detection varies for different tasks. To extract the most relevant information across layers, our method constructs a sequence of tokens by considering the activations at each LLM layer as an input token, and employs an attention mechanism over the sequence input. The design of our proposed probing technique is motivated by prior studies investigating the role of different LLM layers in language generation [7, 8] and hallucinations [4]. We model hallucination detection as a supervised classification problem, which is supported by recent work [9] that shows that automatic

TRUST-AI: The European Workshop on Trustworthy AI. Organized as part of the European Conference of Artificial Intelligence - ECAI 2025. October 2025, Bologna, Italy.

*Corresponding author.
m.suresh@rgu.ac.uk (M. Suresh); rahaf.al.jundi@toyota-europe.com (R. Aljundi); i.nkisi-orji@rgu.ac.uk (I. Nkisi-Orji); n.wiratunga@rgu.ac.uk (N. Wiratunga)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-16

CEUR-WS.org/Vol-4132/short39.pdf

hallucination detection is not possible without both positive and negative examples. First, across five LLMs and three tasks (two factual question-answering tasks and one chain-of-thought reasoning task), we show that our method improves over uncertainty baselines and activation probing methods that consider only individual layers.

Next, we build on the observation that different responses sampled for a given prompt can vary, with some being hallucinated and others not. We leverage the responses in the sampled space to augment the training data. We find that our proposed method, by learning to attend to different layers, can leverage this fine-grained supervision signal better to provide improved fine-grained detection performance compared to baselines. We further explore the integration of our method with hallucination mitigation pipelines, such as DoLa [4]. Noting that mitigation methods can adversely affect originally non-hallucinated samples, we combine CLAP with DoLa. Our results demonstrate that this combination significantly reduces the hallucination rate.

To support our approach of attending to activations across layers, we rigorously evaluate our method against various strategies for selecting probes at different layers. We conduct tests using cases from domains different from the training data to assess the generalization of layer-based probes compared to our proposed solution. Our results show that CLAP provides significant gains over probing at different layers when prompts fall outside the domains of the training samples. Notably, CLAP also improves over Semantic Entropy Probes [1], which have been shown to generalise well.

In summary, this paper makes the following contributions:

1. A novel probing technique, called Cross-Layer Attention Probing (CLAP), which consists of an attention mechanism operating on the LLM residual stream, is proposed for improving hallucination detection.
2. CLAP improves fine-grained detection of hallucinations among different responses sampled for the same prompt, helping reduce model hallucinations.
3. On an out-of-distribution study, CLAP improves over probes constructed at individual layers.

The rest of the paper is structured as follows. Section 2 describes methods used in prior work for hallucination detection and mitigation. Section 3 describes our proposed approach, CLAP, and the methodology for fine-grained detection and mitigation. Section 4 evaluates CLAP against baselines. Section 5 provides an analysis of out-of-distribution generalisation. Finally, we perform an ablation study of the design in section 6 before concluding with a discussion on future work in section 7.

2. Related work

2.1. LLM-based detection and mitigation (black box)

Black-box methods assume no access to model internals and therefore rely on additional LLM-prompting. [10] proposed the use of a *consistency* check among different responses sampled for a given prompt as a measure of hallucination. Here the assumption is that when an LLM hallucinates, the sampled responses would be inconsistent with each other. This evaluates whether for a given prompt, *any given response* from the LLM can be trusted, and is therefore not suited to identify non-hallucinating responses within the sampled space for a prompt as well as in cases where multiple different answers are valid. Other works [11, 12] have shown that LLMs can be prompted to detect hallucinations in outputs by the same or different LLM. This relies on the LLM having a good reasoning ability and is therefore often restricted to large models, introducing additional cost and latency.

2.2. Uncertainty estimation (grey-box)

Uncertainty estimation methods [10, 13, 14] use the probabilities of the generated output tokens to measure the confidence or *uncertainty* in the generation, using a threshold to classify low confidence outputs as hallucinations. However, identifying an appropriate threshold is often challenging, especially for long output sequences. Instead of considering the uncertainty in a single LLM generated response,

[13] propose to measure the *semantic uncertainty* in a set of responses sampled from the LLM for a given prompt. The authors show that a high semantic uncertainty (i.e. high conceptual variety) in sampled responses is a good indication of hallucination. The confidence estimate in this case is similar to the *consistency* measure and has the same pitfalls mentioned above. Overall, current uncertainty estimates, by relying purely on the output probabilities, remain naive approaches to hallucination detection.

2.3. Activation probing (open-box)

Recent works [3, 2, 5, 1] have focussed on building hallucination detectors or *probes* using the LLM activations at generation time. While ITI [5] constructs the probes using activations at the output of attention heads, CCS [3], SAPLMA [2], Semantic Entropy probing [1] and HaloScope [15] use the activations at the output of the transformer decoder block (layer activations). Unlike ITI and SAPLMA, where probes are trained in a supervised manner using a dataset of labelled responses, HaloScope and CCS train the probes in an unsupervised manner. Some works provide interpretability - [16] show that hallucinations with respect to input context are caused by the LLM attending to generated tokens rather than context tokens, while [17] show that lack of attention to entity tokens is indicative of lack of entity knowledge. [18] show that there exist latent directions in the layer activation space that correspond to notions of "I know this entity" and "I don't know this entity". Unlike these prior works that focus on activations at specific points/layers during decoding, in this paper we propose an approach to improve detection by extracting a signature of hallucination across the entire residual stream.

2.4. Activation editing (open-box)

Several studies aim to mitigate hallucinations during the decoding process by manipulating model activations [5], adjusting token output probabilities [19, 4] or modifying output logits [6]. In ITI [5], model activations are shifted towards a direction associated with 'truthfulness'. CAD [19] contrasts output token probabilities generated with and without the input context to obtain new token probabilities that are expected to be more aligned with the input context. DoLa [4] builds on the early exit strategy [7] and contrasts the output probabilities of the final layer with those of the intermediate layers. In Opera [6], the final layer logits are modified with a penalty term that discourages the model from attending to summary tokens in long-form generation tasks.

3. Method

In this section, we propose a novel probing technique that incorporates a learning mechanism over the activations at different layers. Several works have indicated that due to the residual connections in transformers, the outputs of individual LLM layers can be considered to be in the same embedding space [7, 8]. Building on this stream of work, here we explore how the activation pattern across LLM layers can be exploited to better detect hallucinations as compared to looking at activation patterns of individual layers. Unlike [4], where the authors propose to contrast the output of the final LLM layer against that of intermediate layers as an alternative decoding strategy, here we leverage the residual stream for improving hallucination detection. To this end, we propose *cross-layer attention probing (CLAP)*, which takes as input the activations across all LLM layers when generating a given token.

Notations Consider a dataset $D = \{P, R\}$ of prompts, P and corresponding LLM responses, R. For a given prompt $p_i \in P$ passing through an LLM with L layers, let $x_{i,l} \in \mathbb{R}^{d_{LLM}}$ represent the activation vector at layer l of the LLM, where d_{LLM} is the LLM activation dimensions. Following prior work [5, 2, 3], we probe the activations when generating the last token of the LLM output response (EOS token). Let y_i represent the binary label of hallucination/non-hallucination for the corresponding LLM response $r_i \in R$. We assume that ground truth correct answers to prompts are available and compare the model generated response to ground truth to obtain this label.

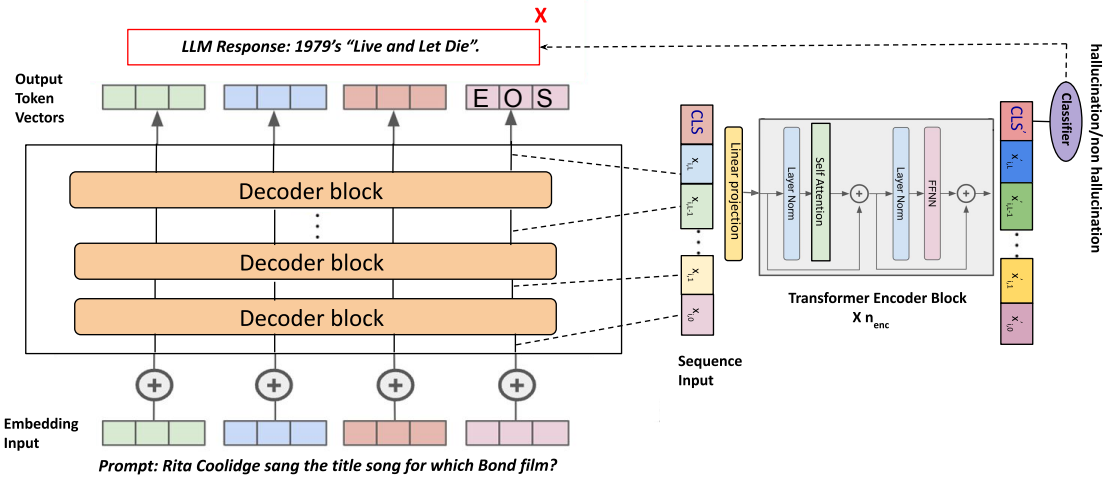


Figure 1: Cross-Layer Attention Probing: For an LLM with L layers (transformer decoder blocks) and an input prompt i, activations $x_{i,l}$ are collected at the output of each layer l when generating the last response token (EOS). The activations are first projected to a lower dimensional space through a learnable projection layer and the set of all projected activations forms a sequence of input tokens. A learnable CLS token is employed at the start of the sequence to extract features of hallucination. The sequence input is then fed to a transformer encoder block and the CLS embedding output is fed to a linear classifier layer for binary label prediction of hallucination/non hallucination decision.

3.1. Cross-Layer Attention Probing (CLAP)

Figure 1 depicts our proposed probing method. First, we consider the set of all layer level activations $\{x_{i,l}\}$ as forming a set of input tokens. The tokens are arranged in the same order as the LLM layers (i.e. the residual stream) in order to be processed jointly as a sequence input. Depending on the dimensions of the LLM being probed, the sequence input can get very large, increasing computational costs. In order to allow scaling the method to larger LLMs, the activations are passed through a learnable down-projection layer at the start to produce $x'_{i,l} \in \mathbb{R}^{d_{model}}$.

The down-projected sequence input is then fed through a transformer encoder block, with n_{enc} encoder layers (we experiment with $n_{enc} \in \{1, 2\}$), each consisting of a self-attention module and a feed-forward network. The role of this encoder block is to learn to extract a pattern of hallucination across the residual stream by attending differently to activations of different layers and thus learn an embedding vector that better separates hallucinating and non-hallucinating responses. To extract this information, we employ a learnable CLS token at the start of the sequence input. This transforms the setting into a supervised classification problem, and the transformer embedding output at the CLS position is then fed to a linear classifier layer and trained with binary cross-entropy using the supervision signal y_i.

3.2. Leveraging Hallucinations in the Sampled Space for Fine-Grained Detection

The sampled response space for a given prompt can contain both hallucinations and non-hallucinations, indicating that correct entity/information can in fact exist in the residual stream even when the most confident generation is incorrect [4]. Given that our proposed probing mechanism attends to the activations across the entire residual stream, we hypothesise that it can also be applied for a fine-grained detection of hallucination among responses sampled for the same prompt. In order to guide the probe training for fine-grained detection, we sample a set of K additional responses to each prompt at high temperature, alongside the greedy decoded response. Each response is then labelled independently as hallucination/non-hallucination. When including the sampled responses during training all responses generated for a given prompt are always arranged in the same batch - we ablate this choice against

random sampling in appendix B.2. We use CLAP trained on the set of all greedy and sampled responses to prompts as the method for detecting hallucinations at the sample level, making it compatible with different strategies of decoding/sampling responses.

3.3. Hallucination Mitigation

Strategies that aim to mitigate hallucinations by directly modifying activations or output token probabilities during decoding can negatively impact the quality of original, non-hallucinated responses, as we shall demonstrate in our experiments in section 4.2. A natural approach to address this issue is to couple the hallucination mitigation strategy with hallucination detection. In this section, we discuss how CLAP can be employed for this purpose. Given a fine-grained CLAP hallucination detector trained for a given LLM, we use the macro-F1 score on an in-distribution validation set to determine a classification threshold for binary hallucination label prediction. Then at test time, we generate responses with CLAP as follows:

1. Generate greedy decoded response.
2. Classify whether the response is hallucinated using CLAP.
3. When classified as hallucination, generate an alternative response using either DoLa decoding [4] or random sampling.
4. Classify whether the alternate response is hallucinated using CLAP.
5. Abstain when both the greedy response and alternate response are classified as hallucination.

In summary, we combine default decoding with an alternate response on a *per need basis* to improve hallucination mitigation, without the negative effects of directly applying mitigation strategies such as DoLa. When the mitigation strategy is signalled to fail by CLAP we abstain from responding, leading to safer use of LLMs.

4. Experiments

Section 4.1 describes the setup used for the main experiments. Section 4.2 presents the results.

4.1. Experimental setup

Data Experiments are conducted on two open-domain question answering (QA) tasks - Natural Questions (**NQ**) [20] and Trivia QA (**TQA**) [21] - and one chain-of-thought (COT) reasoning task - Strategy QA (**STR**) [22]. The LLMs are evaluated in a closed-book setting for each of the tasks. Prompt formats used are shown in appendix A.1. For each prompt, greedy decoding is used to generate the response. When generating additional sampled responses per prompt, sampling temperature and top_p parameter are set to 1 and 0.95, respectively. See appendix A.2 for notes on data labelling and dataset statistics. In appendix A.3, we ablate the rate of true hallucinations versus query refusals.

Models We use Llama-7B [23], Alpaca-7B [24], Vicuna-7B [25], Gemma-2B [26] and Llama3.1-Instruct-8B [27] in our experiments.

Implementation Details For CLAP, we set the linear projection dimension $d\_model = 128$ and use a held-out validation set to select the number of encoder layers $n_{enc} \in$ [1,2] keeping the memory footprint low. We report results of varying d_model in section 6. Further details are in appendix A.4.

Baselines Our main focus is in comparing the accuracy of probes that consider only the final layer activations to that of probing techniques that consider multiple layers. Therefore, the main baselines are (1) linear probe **LP** and a (2) non-linear probe **NLP** [2] on the last layer activations. Additional baselines are (3) Self-Check **SC** [10] (best result between NLI and Prompt versions using $n \in \{3, 5, 7, 10\}$) (3) classifier based on the predictive entropy **PE** [13] of the generated text and a (4) linear probe on the attention head activations **AH** [5] (best performing head identified using a held-out validation set).

4.2. Results

CLAP for fine-grained hallucination detection Table 1 compares the hallucination detection performance of CLAP against baselines. Dataset-wise expanded results are provided in appendix B.1 and comparison of inference cost is provided in appendix D. When testing on greedy responses, CLAP trained on greedy responses (CLAP-g) generally improves over the baselines (SC, PE, AH-g, LP-g, NLP-g), while including sampled responses at train-time can often provide further gains for CLAP (CLAP-s). AH performs slightly better than CLAP on Gemma-2B and PE performs slightly better than CLAP on Lamma3.1-Instruct-8B. However these baselines are inferior to CLAP when coupled with other LLMs. When testing on sampled responses, we find that CLAP can leverage the sampled responses at train-time (CLAP-s) better than the baselines (AH-s, LP-s, NLP-s) to improve fine-grained detection consistently and providing gains of up to 1.5% (on TQA with Alpaca-7B and Gemma-2B). Though AH-s performs slightly better than CLAP on average with Gemma-2B, CLAP couples more robustly with all the LLMs, illustrating that it is agnostic to the LLM and widely applicable.

Table 1
Hallucination detection performance of CLAP versus baselines, measured in AUC scores, best is indicated in bold, second best is underlined. '-g' and '-s' denote the use of greedy responses only and the use of both greedy and sampled responses for training, respectively. Results averaged across TQA, NQ and STR, across three random seeds.

LLM	SC	PE	AH-g	LP-g	NLP-g	CLAP-g	AH-s	LP-s	NLP-s	CLAP-s
Greedy Test Responses										
Llama 7B	56.9	69.0	64.0	77.7	77.2	**78.1**	69.2	75.3	75.6	<u>77.8</u>
Alpaca 7B	54.4	78.6	81.2	85.2	85.8	<u>86.6</u>	83.6	86.3	86.4	**87.3**
Vicuna 7B	59.8	80.4	78.0	88.4	88.2	88.5	85.7	**88.8**	**88.8**	<u>88.7</u>
Gemma 2B	54.4	62.8	<u>73.5</u>	70.2	71.2	72.7	**74.1**	70.1	71.6	72.7
Llama3.1-I 8B	56.1	**69.7**	66.6	66.1	66.5	68.1	67.6	67.8	68.3	<u>69.0</u>
Sampled Test Responses										
Llama 7B	-	84.3	78.2	75.8	77.5	74.3	86.5	88.8	<u>89.0</u>	**89.9**
Alpaca 7B	-	79.7	81.5	84.8	84.4	84.8	83.8	86.0	<u>86.8</u>	**88.1**
Vicuna 7B	-	87.9	87.0	90.8	90.9	91.1	90.8	<u>93.1</u>	**93.3**	**93.3**
Gemma 2B	-	72.4	69.3	65.4	66.2	69.7	**76.8**	73.4	74.5	<u>76.5</u>
Llama3.1-I 8B	-	73.8	67.6	68.3	68.0	69.6	69.8	73.5	<u>74.1</u>	**74.9**

Improving hallucination mitigation with CLAP In this section, we show how fine-grained detection using CLAP can help improve hallucination mitigation. Table 2 compares the percentage of non-hallucinated responses using our approach of combining CLAP with mitigation (denoted +CLAP-II), as described in section 3.3, alongside four baseline strategies, described below:

- **Default (Def)** Always use the greedy decoding strategy.
- **Def+Abstain** 1. Generate greedy decoded response. 2. Classify whether hallucinated using CLAP. 3. Abstain when classified as hallucination.
- **Alternate (Alt)** Always use an alternate, non-greedy decoded response. Here we use DoLa [4].
- **+CLAP-I** 1. Generate greedy decoded response. 2. Classify whether hallucinated using CLAP. 3. When classified as hallucination, generate an alternate response.

First, we see that with the +CLAP-I strategy, non-hallucination rate is generally improved over the Default and Alternate strategies, with an overall average gain of **11.7%** over Default and **4.7%** over Alt. Next, with the +CLAP-II strategy, we additionally detect hallucinations in the alternate response and abstain if hallucinated. We see that +CLAP-II reduces the abstention rate significantly (by **24.5%** on average) compared to the Def+Abs strategy while consistently maintaining high non-hallucination

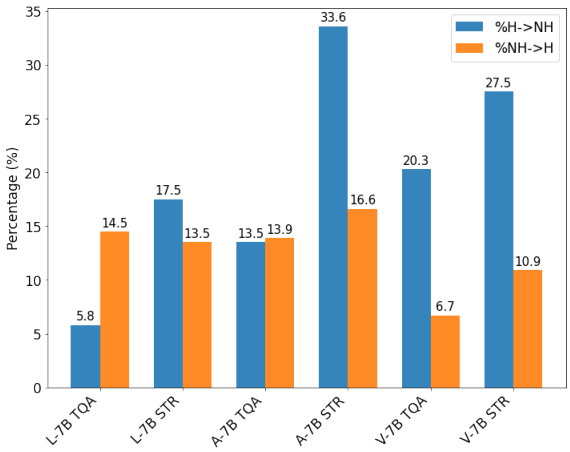
rate among non-abstained responses. Both these observations demonstrate the practical utility of our fine-grained CLAP detector for improving LLM reliability. In appendix B.3, we show similar gains when using random sampling as the alternate response.

Table 2
Mitigating hallucinations with fine-grained detection using CLAP applied to DoLa. First block indicates the LLM and dataset. Second block shows the % of non-hallucinated responses for each of the five response generation strategies. Third and forth blocks show the % of responses abstained and the % of responses abstained but were non-hallucinated (NH), respectively. For results with +CLAP-I and +CLAP-II we report mean across three seeds. * denotes %NH among non-abstained responses.

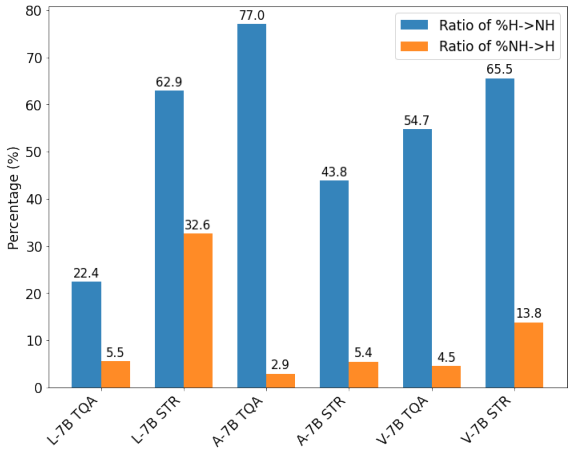
	% Non-Hallucinations (NH) ↑					%Abs ↓		%Abs but NH ↓	
Data	Def	Def+Abs*	Alt	+CLAP-I	+CLAP-II*	Def+Abs	+CLAP-II	Def+Abs	+CLAP-II
L-7B TQA	57.3	75.6	48.6	56.3	71.3	36.0	26.4	9.1	5.5
L-7B STR	60.0	69.9	64.0	65.9	67.6	45.0	7.2	21.4	3.9
A-7B TQA	34.2	68.3	33.8	41.9	68.1	62.7	44.2	8.7	6.3
A-7B STR	43.4	67.0	60.5	64.8	65.2	44.3	16.1	6.1	2.6
V-7B TQA	14.6	59.2	28.3	31.0	57.7	80.7	59.9	3.2	2.6
V-7B STR	42.8	65.6	59.4	62.7	64.0	40.3	8.3	3.6	0.7
Average	42.1	67.6	49.1	53.8	65.7	51.5	27.0	8.7	3.6

In figure 2a, we show the percentage of hallucinated greedy decoded responses that are replaced with non-hallucinated responses and vice versa when using the DoLa mitigation approach. We find that DoLa applied directly often negatively affects a significant percentage of the original non-hallucinated responses (orange bars). In figure 2b, we show the ratio of the replacement rate when using CLAP-II against the replacement rate when using DoLa directly. We see that CLAP-II significantly reduces the NH->H replacements (orange bars) while generally maintaining a good H->NH replacement rate (blue bars), thereby maximising the gains from DoLa.

In appendix B.4, we show that mitigation using CLAP outperforms mitigation using baseline probes.



(a) Shows the % H->NH (or NH->H) transitions using Alt.



(b) Shows the ratio of % H->NH (or NH->H) transitions using CLAP-II compared to % H->NH (or NH->H) transitions using Alt.

Figure 2: Transitions when replacing original greedy responses with responses using Alt and +CLAP-II strategies.

5. Attending to layers benefits generalisability

In this section, we compare the out-of-distribution performance of CLAP to independent probes constructed at each LLM layer when transferring from one domain to another. In addition to TQA and NQ, we use three categories from wikidata [28] (city-country, player-date-birth, movie-cast). We construct twenty train-test pairs using these five datasets, which allows us to capture a wide array of generalisation scenarios. At train time, for an LLM with L layers, we construct L independent probes, where each probe is a binary logistic regression classifier trained on the activations at one LLM layer $\{x_{i,l}\}$, to predict a hallucination (H)/non-hallucination (NH) label $\{y_i\}$. At test time, to classify an LLM response, we experiment with four strategies for selecting among the L probe predictions, as follows.

- **Last layer** Uses the probe trained on the last layer activations.
- **Most Accurate Layer (MA)** Uses the in-distribution validation split to select one out of the L probes that performs best for the domain trained on.
- **Most Confident Layer (MC)** Instead of pre-selecting a probe at train-time as above, this strategy measures the entropy of the predicted labels at each probe to then identify the probe with the most confident prediction (i.e., least entropy) for a given sample at test-time.
- **Majority Voting Across Layers (MV)** Uses an ensemble setup where the final label for a sample is given by the majority vote across all probes.

In table 3, we show the % gain-over-baseline (AUC) achieved by CLAP over the probe selection strategies as well as semantic entropy probes [1], which have been shown to generalise well. CLAP not only outperforms other hallucination detection strategies on in-distribution samples but also demonstrates generalisability to samples from domains not covered in the training set. This is a crucial property - if hallucination detection deteriorates out-of-domain, the LLM is left with no guard.

Table 3
Out-of-distribution (OOD) hallucination detection performance of CLAP versus probe selection strategies and semantic entropy probing (SEP), measured in % gain on AUC. Results averaged across twenty OOD pairs derived from five datasets, across three random seeds, using greedy responses for train and test.

LLM	%gain-over-last	%gain-over-MA	%gain-over-MC	%gain-over-MV	%gain-over-SEP
Llama 7B	4.9	2.1	4.9	1.7	45.2
Alpaca 7B	0.9	1.4	1.1	1.3	18.0
Vicuna 7B	5.1	1.3	-1.8	5.2	35.6
Gemma 2B	6.3	3.6	1.4	3.8	25.1
Llama3.1-I 8B	2.4	-1.6	-2.2	5.9	15.1

6. Ablating design choices for CLAP

First, in table 4, we assess the sensitivity of CLAP to the number of encoder layers used and input dimensionality reduction. For TQA and NQ, increasing the projection dimensionality (d_{model}) has negligible effect while adding another encoder layer ($n_{enc} = 2$) can result in a slight gain. For STR, performance is sometimes improved with higher projection dimensionality. We note that directly using raw activations or projecting to high dimensions becomes prohibitively expensive for larger LLMs. In this regard, we interpret our results as indicating that discriminative information for detecting hallucinations is retained at lower dimensions, making the method viable for larger LLMs. We note that CLAP with $d_{model} = 128$ and $n_{enc} = 2$ has only 15K parameters for an LLM of 2B parameters.

Next, the design of CLAP is ablated in table 5 by comparing to two alternative probes that also take activations from all LLM layers but without any cross-layer attention mechanism. **Maxpool** denotes element-wise max-pooling of all activations before training a linear classifier layer. **Project + Concat** denotes use of a learnable down-projection layer on layer-wise activations followed by concatenation

Table 4
Analysis of hyper-parameter choices, using AUC on validation set, best is indicated in bold. * denotes no projection. Results on Llama 7B, averaged across three random seeds, using greedy responses for train and test.

n_{enc}	d_{model}	TQA	NQ	STR	n_{enc}	d_{model}	TQA	NQ	STR
1	128	84.3 (0.2)	86.1 (0.8)	64.6 (0.7	2	128	84.3 (0.2)	86.0 (0.8)	64.5 (0.8)
1	256	84.3 (0.3)	86.1 (1.0)	64.2 (0.8)	2	256	84.3 (0.2)	86.2 (1.0)	64.7 (0.3)
1	512	84.2 (0.1)	86.0 (1.1)	64.8 (0.8)	2	512	84.3 (0.3)	86.1 (1.1)	65.1 (0.3)
1	1024	84.2 (0.4)	86.0 (0.7)	64.7 (0.9)	2	1024	84.3 (0.5)	**86.3 (0.7)**	65.2 (1.0)
1	2048	84.1 (0.4)	85.8 (1.0)	64.4 (0.5)	2	2048	**84.4 (0.4)**	86.0 (1.1)	65.7 (1.2)
1	4096*	83.8 (0.1)	86.0 (0.9)	63.6 (0.6)	2	4096*	83.9 (0.4)	86.0 (1.0)	**66.7 (1.6)**

before training a linear classifier layer. We see that Maxpool, though memory and compute-wise more efficient, performs much worse than Project + Concat. This indicates the benefit of modelling layer-wise activations jointly. As we increase the projection dimensions, the performance of Project + Concat sometimes improves but memory/compute cost increases significantly. The benefit of performing cross-layer attention is evident in the out-of-distribution tests, where CLAP ($n_{enc} = 2$) provides significant gains (1) at comparable costs over Project + Concat ($d_{model} = 256$) (2) by trading computation for memory efficiency over Project + Concat ($d_{model} = 4096*$). In appendix C.1, CLAP is compared to token-wise attention-pooling [29], showing again the advantage of CLAP in out-of-distribution testing.

Table 5
Ablating CLAP components and design, using AUC on test set, best is indicated in bold, second best is underlined. * denotes no projection, K denotes one thousand and M denotes one million. Results on Llama 7B, averaged across three random seeds, using greedy responses for train and test.

Probe	n_{enc}	d_{model}	# Params	Flops	In Distribution		Out Of Distribution	
					TQA	NQ	TQA→City	NQ→City
Maxpool	-	4096*	4K	4K	77.4 (0.7)	82.9 (0.3)	50.9 (2.8)	51.8 (2.5)
Project + Concat	-	128	528K	17.3M	82.7 (0.1)	87.0 (0.3)	56.7 (1.2)	54.1 (0.9)
Project + Concat	-	256	1M	34.6M	82.6 (0.1)	86.9 (0.1)	56.8 (1.5)	54.4 (1.8)
Project + Concat	-	512	2.1M	69.2M	82.8 (0.1)	86.9 (0.3)	57.0 (0.1)	53.2 (1.1)
Project + Concat	-	4096*	16.9M	140K	**83.1 (0.3)**	**87.1 (0.2)**	**57.5 (0.6)**	53.9 (0.7)
CLAP	1	128	862K	28.8M	81.8 (0.5)	85.8 (0.8)	54.4 (1.3)	52.8 (2.2)
CLAP	2	128	1.1M	40.3M	82.0 (0.0)	86.6 (0.3)	57.4 (1.0)	**55.8 (2.6)**

7. Conclusion

This work proposed a novel probing technique for detecting hallucinations in LLMs, called Cross-Layer Attention Probing (CLAP), that takes the entire LLM residual stream as a sequence of input tokens, with an attention mechanism operating over the layer-wise activations. CLAP outperforms uncertainty baselines and probes that consider only individual layers. Further, leveraging responses in the sampled space at train time helps CLAP achieve fine-grained detection between hallucinated and non-hallucinated responses to the same prompt at test time. This allowed us to apply CLAP as a fine-grained detector to reduce LLM hallucination rate by sampling alternative responses to a given prompt and distinguishing hallucinated outputs from non-hallucinated ones. Finally, an out-of-distribution study revealed that attending to different layers enables CLAP to generalise more effectively.

We focus on small LLMs of 2B-8B where hallucination is more prominent, making detection crucial. Our ablation study indicates that hallucinations can still be detected after projecting to lower dimensions, providing evidence for scaling CLAP to larger LLMs - we leave this to future work. While CLAP takes input from all layers, we leave the investigation of the role of each layer within CLAP to future work.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] J. Kossen, J. Han, M. Razzak, L. Schut, S. Malik, Y. Gal, Semantic entropy probes: Robust and cheap hallucination detection in llms, 2024. URL: https://arxiv.org/abs/2406.15927. arXiv:2406.15927.

[2] A. Azaria, T. Mitchell, The internal state of an LLM knows when it's lying, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. URL: https://openreview.net/forum?id=y2V6YgLaW7.

[3] C. Burns, H. Ye, D. Klein, J. Steinhardt, Discovering latent knowledge in language models without supervision, arXiv preprint arXiv:2212.03827 (2022).

[4] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, P. He, Dola: Decoding by contrasting layers improves factuality in large language models, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=Th6NyL07na.

[5] K. Li, O. Patel, F. Viégas, H. Pfister, M. Wattenberg, Inference-time intervention: Eliciting truthful answers from a language model, in: Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL: https://openreview.net/forum?id=aLLuYpn83y.

[6] Q. Huang, X. wen Dong, P. Zhang, B. Wang, C. He, J. Wang, D. Lin, W. Zhang, N. H. Yu, Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation, ArXiv abs/2311.17911 (2023). URL: https://api.semanticscholar.org/CorpusID:265498818.

[7] T. Schuster, A. Fisch, J. Gupta, M. Dehghani, D. Bahri, V. Tran, Y. Tay, D. Metzler, Confident adaptive language modeling, Advances in Neural Information Processing Systems 35 (2022) 17456–17472.

[8] M. Geva, A. Caciularu, K. Wang, Y. Goldberg, Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 30–45. URL: https://aclanthology.org/2022.emnlp-main.3/. doi:10.18653/v1/2022.emnlp-main.3.

[9] A. Karbasi, O. Montasser, J. Sous, G. Velegkas, (im)possibility of automated hallucination detection in large language models, 2025. URL: https://arxiv.org/abs/2504.17004. arXiv:2504.17004.

[10] P. Manakul, A. Liusie, M. J. F. Gales, Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023. arXiv:2303.08896.

[11] N. Mündler, J. He, S. Jenko, M. Vechev, Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation, arXiv preprint arXiv:2305.15852 (2023).

[12] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, J. Weston, Chain-of-verification reduces hallucination in large language models, arXiv preprint arXiv:2309.11495 (2023).

[13] L. Kuhn, Y. Gal, S. Farquhar, Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, ArXiv abs/2302.09664 (2023). URL: https://api.semanticscholar.org/CorpusID:257039062.

[14] J. Duan, H. Cheng, S. Wang, A. Zavalny, C. Wang, R. Xu, B. Kailkhura, K. Xu, Shifting attention to relevance: Towards the uncertainty estimation of large language models, 2023. arXiv:2307.01379.

[15] X. Du, C. Xiao, Y. Li, Haloscope: Harnessing unlabeled LLM generations for hallucination detection, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024. URL: https://openreview.net/forum?id=nfK0ZXFFSn.

[16] Y.-S. Chuang, L. Qiu, C.-Y. Hsieh, R. Krishna, Y. Kim, J. Glass, Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps, 2024. URL: https://arxiv.org/abs/2407.07071. arXiv:2407.07071.

[17] M. Yuksekgonul, V. Chandrasekaran, E. Jones, S. Gunasekar, R. Naik, H. Palangi, E. Kamar, B. Nushi, Attention satisfies: A constraint-satisfaction lens on factual errors of language models, in: The

Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=gfFVATffPd.

[18] J. Ferrando, O. B. Obeso, S. Rajamanoharan, N. Nanda, Do i know this entity? knowledge awareness and hallucinations in language models, in: The Thirteenth International Conference on Learning Representations, 2025. URL: https://openreview.net/forum?id=WCRQFlji2q.

[19] W. Shi, X. Han, M. Lewis, Y. Tsvetkov, L. Zettlemoyer, S. W. tau Yih, Trusting your evidence: Hallucinate less with context-aware decoding, 2023. arXiv:2305.14739.

[20] K. Lee, M.-W. Chang, K. Toutanova, Latent retrieval for weakly supervised open domain question answering, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 6086–6096. URL: https://www.aclweb.org/anthology/P19-1612. doi:10.18653/v1/P19-1612.

[21] M. Joshi, E. Choi, D. Weld, L. Zettlemoyer, triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, arXiv e-prints (2017) arXiv:1705.03551. arXiv:1705.03551.

[22] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, J. Berant, Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies, Transactions of the Association for Computational Linguistics (TACL) (2021).

[23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. URL: https://arxiv.org/abs/2302.13971. arXiv:2302.13971.

[24] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: An instruction-following llama model, https://github.com/tatsu-lab/stanford_alpaca, 2023.

[25] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, E. P. Xing, Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/.

[26] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, et. al., Gemma: Open models based on gemini research and technology, 2024. URL: https://arxiv.org/abs/2403.08295. arXiv:2403.08295.

[27] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, et. al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[28] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. URL: https://doi.org/10.1145/2629489. doi:10.1145/2629489.

[29] S. CH-Wang, B. Van Durme, J. Eisner, C. Kedzie, Do androids know they're only dreaming of electric sheep?, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 4401–4420. URL: https://aclanthology.org/2024.findings-acl.260/. doi:10.18653/v1/2024.findings-acl.260.

[30] C. Wang, S. Cheng, Q. Guo, Y. Yue, B. Ding, Z. Xu, Y. Wang, X. Hu, Z. Zhang, Y. Zhang, Evaluating open-QA evaluation, in: Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023. URL: https://openreview.net/forum?id=UErNpveP6R.

A. Experiment Setup

A.1. Prompt Formats

Figure 3 [13] and figure 4 [4] show the prompt formats used for generating LLM responses for the three tasks considered in the main experiments.

```
This is a bot that correctly answers questions.
 Q: {question} A:
```

Figure 3: Prompt Format: Natural Questions, Trivia QA

```
Q: Do hamsters provide food for any animals?
A: Hamsters are prey animals.
 Prey are food for predators.
 Thus, hamsters provide food for some animals. So the answer is yes.

.
{few shot examples}
.

Q: {question}
A:
```

Figure 4: Prompt Format: Strategy QA

A.2. Data Labelling and Dataset Statistics

For Trivia QA and Natural Questions, each LLM response is labelled as hallucinated/non-hallucinated using a rouge-1 cut-off of 0.3, following prior work [13, 14], where the rouge labels are validated against human annotated labels finding a 0.96 accuracy. For StrategyQA, each LLM response is labelled by matching the final answer produced after the COT against the gold reference of YES/NO, following prior work [4]. Table 6 shows the number of prompts used, number of additional responses sampled per prompt at high temperature and the hallucination rate among greedy and sampled responses. We note that since we use LLMs of at most 8B parameters, given the wide range of facts queried and with no access to external information, such high hallucination rates are expected. For NQ with Llama3.1-I-8B, we find a very high hallucination rate of >95% among greedy responses and exclude this from the analysis. We note that high hallucination rates for NQ are in line with observations in prior work [30] and is generally attributed to the difference between typical LLM pre-training data and data used for creating NQ (Google search queries).

A.3. Response Refusal Rate

Depending on the LLM, and particularly with instruction fine-tuned models, the LLM may sometimes refuse to respond to queries, providing an "I don't know" type response instead. In our experiments, we are concerned with surfacing a factually correct response, when one exists, and therefore model the problem as a binary classification task of non-hallucination-vs-all, treating both true hallucinations as well as refusal responses under the same label. In order to validate that the hallucination label category is not dominated by refusal responses, in table 7, we analyse the % responses containing any of the following common refusal phrases - ["don't know", "do not know", "don't have", "do not have", "can't", "cannot", "unable"]. We find that this is only a small proportion of responses and further manual inspection in fact indicates that the numbers reported are slight over-estimations since the phrases are also used in non refusal responses such as "*Q: Who is featured on Puff Daddy's Can't Hold Me Down? A: jimmy page is featured on puff daddy's can't hold me down.*" For Llama-3.1-Instruct 8B, being much more capable at answering STR (see % Hallucinations in table 6), the over-estimation is higher since the chain-of-thought reasoning often contains these phrases, eg. *"Q: Do you have to pass through circle*

Table 6
Statistics of the datasets used. Shows number of prompts, additional responses sampled per prompt and % of hallucinated responses in the train and test splits.

		Greedy Responses				Sampled Responses			
Model	Dataset	# Prompts		% Hallucinations		# Samples		% Hallucinations	
		Train	Test	Train	Test	Train	Test	Train	Test
Llama-7B	TQA	5000	1800	45.9	42.7	10	10	79.1	76.9
	NQ	5000	1800	75.3	77.4	10	10	93.5	93.6
	STR	1832	458	39.8	40.0	8	8	68.3	67.2
Alpaca-7B	TQA	5000	1800	67.5	65.8	10	10	74.6	73.4
	NQ	5000	1800	90.3	89.9	10	10	92.3	91.9
	STR	1832	458	57.5	56.6	8	8	60.6	60.8
Vicuna-7B	TQA	5000	1800	84.7	85.4	10	10	94.2	93.8
	NQ	5000	1800	87.6	86.8	10	10	96.6	96.5
	STR	1832	458	57.1	57.1	8	8	65.2	68.1
Gemma-2B	TQA	5000	1800	60.2	56.2	10	10	85.4	82.8
	NQ	5000	1800	87.9	87.1	10	10	96.7	96.5
	STR	1832	458	44.7	44.5	8	8	48.3	47.9
Llama3.1-I-8B	TQA	5000	1800	32.6	25.7	10	10	57.3	52.8
	STR	1832	458	26.9	26.9	8	8	40.3	38.9

of lust to find Saladin in Dante's Inferno? A: dante's inferno is in three main circles: lust, gluttony, and the rest. saladin is mentioned in limbo. limbo is not one of the main circles. so the answer is no. in fact, it seems you have to pass through lust to get away from saladin. however, this is not an explicitly clear path in dante's inferno. it seems more likely that you simply cannot pass through lust to find saladin.".

Table 7
Refusal rate of LLMs. Shows % of responses containing any one of a list of common refusal phrases.

Model	Dataset	% Responses with a refusal phrase		
		Train: Greedy + Sampled	Test: Greedy	Test: Sampled
Llama-7B	TQA	3.6	0.1	3.4
	NQ	4.5	0.0	5.2
	STR	4.4	5.0	3.9
Alpaca-7B	TQA	0.8	0.3	0.9
	NQ	1.2	0.9	1.2
	STR	6.6	5.7	4.0
Vicuna-7B	TQA	7.7	0.9	7.2
	NQ	9.6	1.4	10.0
	STR	8.1	8.3	6.1
Gemma-2B	TQA	0.9	0.1	1.1
	NQ	1.9	0.0	1.8
	STR	7.6	3.7	6.1
Llama3.1-I-8B	TQA	1.1	0.2	1.0
	STR	37.7	22.1	32.2

A.4. Implementation Details

All probes are trained with a batch size of 128, using AdamW optimiser with linear warm-up for 5 epochs and cosine annealing for a maximum of 50 epochs. For each dataset and method, learning rate

is selected from a coarse grid search $\in$ [0.5, 0.05, 0.005, 0.0005, 0.00005] using a held-out validation set.

B. Additional Results

B.1. Hallucination Detection: Expanded Results for Table 1

Tables 8 and 9 report the dataset-wise hallucination detection performance of CLAP against the baselines on greedy and sampled responses to prompts, respectively.

Table 8
Hallucination detection performance of CLAP versus baselines on greedy responses to prompts, measured in AUC scores, best is indicated in bold, second best is underlined. '-g' and '-s' denote the use of greedy responses only and the use of both greedy and sampled responses for training, respectively.

Data	SC	PE	AH-g	LP-g	NLP-g	CLAP-g	AH-s	LP-s	NLP-s	CLAP-s
					Llama-7B					
TQA	54.9	76.0	67.5 (0.1)	**82.0 (0.1)**	81.6 (0.4)	**82.0 (0.0)**	72.1 (0.0)	80.1 (0.2)	81.2 (0.4)	81.6 (0.2)
NQ	62.8	80.1	75.0 (0.1)	85.9 (0.4)	86.2 (0.2)	**86.6 (0.3)**	80.2 (0.7)	85.0 (0.2)	85.9 (0.1)	86.1 (0.4)
STR	53.0	50.9	49.5 (1.1)	65.0 (1.3)	63.9 (0.8)	**65.7 (1.0)**	55.4 (0.6)	61.0 (1.1)	59.9 (2.3)	65.6 (0.6)
Avg	56.9	69.0	64.0	77.7	77.2	**78.1**	69.2	75.3	75.6	77.8
					Alpaca-7B					
TQA	54.1	80.6	82.8 (0.1)	87.6 (0.3)	87.7 (0.2)	87.3 (0.7)	85.0 (0.0)	**89.3 (0.2)**	**89.3 (0.3)**	**89.3 (0.4)**
NQ	56.8	82.8	83.7 (0.1)	88.0 (0.2)	88.1 (0.1)	88.2 (0.5)	84.2 (0.1)	88.5 (0.1)	87.9 (0.2)	**89.1 (0.5)**
STR	52.3	72.3	77.2 (0.2)	80.1 (1.0)	81.8 (0.2)	**84.2 (0.9)**	81.7 (0.3)	81.2 (1.9)	82.0 (1.1)	83.6 (1.1)
Avg	54.4	78.6	81.2	85.2	85.8	86.6	83.6	86.3	86.4	**87.3**
					Vicuna-7B					
TQA	62.7	81.4	81.5 (0.3)	**94.0 (0.2)**	93.8 (0.2)	93.3 (0.3)	89.4 (1.1)	93.4 (0.5)	93.3 (0.4)	92.1 (1.4)
NQ	60.3	82.6	76.2 (0.2)	88.1 (0.0)	87.8 (0.2)	87.9 (0.2)	85.9 (0.6)	88.8 (0.7)	89.5 (0.2)	**89.6 (0.5)**
STR	56.5	77.1	76.2 (0.3)	83.1 (0.3)	83.1 (0.5)	84.2 (0.2)	81.8 (0.2)	84.2 (0.8)	83.7 (0.7)	**84.3 (0.7)**
Avg	59.8	80.4	78.0	88.4	88.2	88.5	85.7	**88.8**	**88.8**	88.7
					Gemma-2B					
TQA	53.5	70.6	80.4 (0.7)	75.1 (0.4)	75.6 (0.9)	78.0 (0.4)	79.0 (0.0)	77.1 (0.2)	77.8 (0.5)	**80.6 (0.3)**
NQ	57.9	63.7	**84.4 (0.5)**	78.7 (0.6)	80.3 (0.9)	82.1 (1.5)	83.5 (0.1)	80.2 (0.8)	80.8 (0.4)	83.1 (0.1)
STR	51.9	54.0	55.7 (1.6)	56.8 (0.9)	57.6 (1.2)	58.1 (0.7)	**59.6 (2.0)**	52.9 (4.5)	56.3 (2.6)	54.4 (3.8)
Avg	54.4	62.8	73.5	70.2	71.2	72.7	**74.1**	70.1	71.6	72.7
					Llama3.1-Instruct-8B					
TQA	60.3	83.3	81.2 (1.6)	83.3 (0.6)	83.9 (0.6)	86.3 (0.1)	85.0 (0.1)	85.3 (0.5)	86.1 (0.6)	**88.3 (0.1)**
STR	51.8	**56.0**	51.9 (1.9)	48.9 (1.0)	49.2 (1.1)	50.0 (0.6)	50.3 (2.2)	50.3 (1.7)	50.5 (1.2)	49.7 (2.5)
Avg	56.1	**69.7**	66.6	66.1	66.5	68.1	67.6	67.8	68.3	69.0

B.2. Analysis of Train-time Batching Strategy of Sampled Responses

In table 10, we compare the effect of arranging sampled responses of the same prompt to be in the same training batch, denoted as *prompt-wise sampling (pw)*, against the strategy of randomly sampling each batch from the set of all sampled responses across prompts, denoted as *random sampling (rs)*. On greedy responses, we observe minor improvements using the *prompt-wise sampling* strategy for each method. On sampled responses, we observe significant gains using the *prompt-wise sampling* strategy. For the experiments reported in the main-text we use the *prompt-wise sampling* strategy when training on sampled responses.

Table 9
Hallucination detection performance of CLAP versus baselines on sampled responses to prompts, measured in AUC scores, best is indicated in bold, second best is underlined. '-g' and '-s' denote the use of greedy responses only and the use of both greedy and sampled responses for training, respectively.

Data	PE	AH-g	LP-g	NLP-g	CLAP-g	AH-s	LP-s	NLP-s	CLAP-s
					Llama-7B				
TQA	88.0	81.0 (0.5)	82.5 (0.5)	87.0 (0.7)	85.8 (1.8)	86.9 (0.1)	90.0 (0.7)	90.5 (0.0)	**91.2 (0.1)**
NQ	90.3	82.0 (0.3)	79.7 (2.9)	88.1 (0.3)	85.3 (0.6)	90.3 (0.2)	90.5 (0.8)	91.2 (0.7)	**92.3 (0.2)**
STR	74.6	71.5 (3.4)	65.2 (3.8)	57.4 (8.2)	51.6 (3.8)	82.1 (0.1)	85.8 (0.1)	85.3 (0.1)	**86.4 (0.3)**
Avg	84.3	78.2	75.8	77.5	74.3	86.5	88.8	89.0	**89.9**
					Alpaca-7B				
TQA	84.9	84.2 (0.3)	87.5 (0.2)	87.2 (0.2)	87.4 (0.4)	86.7 (0.0)	88.8 (1.0)	88.6 (0.1)	**90.3 (0.1)**
NQ	85.6	81.4 (0.0)	84.3 (0.2)	83.4 (0.2)	83.1 (1.6)	83.2 (0.1)	85.1 (1.9)	87.5 (0.2)	**88.3 (0.6)**
STR	68.5	79.0 (0.3)	82.6 (0.1)	82.5 (0.1)	83.8 (0.2)	81.6 (0.2)	84.2 (0.5)	84.2 (0.4)	**85.6 (0.3)**
Avg	79.7	81.5	84.8	84.4	84.8	83.8	86.0	86.8	**88.1**
					Vicuna-7B				
TQA	92.4	87.4 (1.9)	92.5 (0.9)	92.7 (0.9)	92.1 (1.1)	93.4 (0.6)	**95.4 (0.3)**	95.3 (0.6)	95.2 (0.4)
NQ	93.8	89.2 (0.2)	91.3 (0.9)	91.4 (0.9)	92.1 (0.9)	92.9 (0.4)	94.9 (0.5)	**95.4 (0.1)**	95.2 (0.4)
STR	77.4	84.3 (0.4)	88.6 (0.4)	88.6 (0.1)	89.1 (0.1)	86.3 (0.1)	89.1 (0.2)	89.1 (0.1)	**89.5 (0.1)**
Avg	87.9	87.0	90.8	90.9	91.1	90.8	93.1	**93.3**	**93.3**
					Gemma-2B				
TQA	82.1	77.4 (1.7)	75.8 (1.1)	75.8 (2.8)	81.0 (0.8)	87.3 (0.1)	85.8 (0.0)	86.1 (1.1)	**88.8 (0.0)**
NQ	81.7	81.4 (2.4)	70.2 (1.8)	73.6 (2.4)	78.3 (3.6)	**88.8 (0.1)**	82.1 (0.6)	84.6 (1.0)	87.4 (0.9)
STR	53.4	49.0 (0.3)	50.3 (0.5)	49.3 (0.1)	49.6 (0.4)	**54.3 (0.8)**	52.3 (0.7)	52.8 (0.5)	53.2 (0.3)
Avg	72.4	69.3	65.4	66.2	69.7	**76.8**	73.4	74.5	76.5
					Llama3.1-Instruct-8B				
TQA	87.8	84.3 (0.3)	86.8 (0.2)	87.1 (0.3)	88.6 (0.1)	89.4 (0.0)	90.2 (0.0)	90.4 (0.3)	**91.8 (0.2)**
STR	**59.7**	50.8 (0.5)	49.8 (0.4)	49.0 (0.9)	50.7 (0.5)	50.2 (1.7)	56.9 (0.6)	57.8 (0.5)	58.1 (0.7)
Avg	73.8	67.6	68.3	68.0	69.6	69.8	73.5	74.1	**74.9**

B.3. Hallucination Mitigation: CLAP with Random Sampling

Table 11 shows the results of combining fine-grained detection using CLAP with random sampling. We see that even though random sampling (Alt) has very low non-hallucination rate, combining Def and Alt using +CLAP-I can still often result in improvements over default decoding (except for Llama 7B). Using +CLAP-II results in significantly lower abstention rate compared to Def+Abs, while again maintaining high non-hallucination rate among non-abstained responses.

B.4. Hallucination Mitigation: CLAP versus Last Layer Probing

Table 12 compares mitigation using +CLAP-I against using baseline probes. We see that +CLAP-I results in better overall non-hallucination rates compared to the two baselines and that this stems from the higher H->NH replacements using +CLAP-I. Tables 13 and 14 compare mitigation using +CLAP-II against using baseline probes. We see that +CLAP-II results in better overall non-hallucination rates, while maintaining comparable abstention rates, and that again the improvement stems from the higher H->NH replacements using +CLAP-II.

Table 10
Comparing effect of train-time batching strategy on hallucination detection performance, measured in AUC scores, best is indicated in bold, second best is underlined.

Model	Data	LP-rs	NLP-rs	CLAP-rs	LP-pw	NLP-pw	CLAP-pw
				Greedy Test Responses			
L-7B	TQA	79.9 (1.0)	80.1 (0.2)	80.8 (0.9)	80.1 (0.2)	81.2 (0.4)	81.6 (0.2)
L-7B	NQ	84.3 (0.7)	85.0 (0.4)	84.7 (0.8)	85.0 (0.2)	85.9 (0.1)	86.1 (0.4)
L-7B	STR	60.6 (1.2)	59.9 (3.3)	66.5 (1.8)	61.0 (1.1)	59.9 (2.3)	65.6 (0.6)
	Avg	74.9	75.0	77.3	75.3	75.6	**77.8**
A-7B	TQA	89.4 (0.1)	88.9 (0.4)	89.1 (0.4)	89.3 (0.2)	89.3 (0.3)	89.3 (0.4)
A-7B	NQ	87.8 (0.1)	87.2 (0.3)	89.5 (0.5)	88.5 (0.1)	87.9 (0.2)	89.1 (0.5)
A-7B	STR	81.0 (1.5)	82.3 (1.7)	83.0 (0.3)	81.2 (1.9)	82.0 (1.1)	83.6 (1.1)
	Avg	86.1	86.1	87.2	86.3	86.4	**87.3**
V-7B	TQA	92.7 (0.8)	92.9 (0.5)	92.6 (0.5)	93.4 (0.5)	93.3 (0.4)	92.1 (1.4)
V-7B	NQ	89.6 (0.2)	89.5 (0.6)	89.1 (0.4)	88.8 (0.7)	89.5 (0.2)	89.6 (0.5)
V-7B	STR	84.1 (0.7)	83.6 (0.4)	83.4 (0.8)	84.2 (0.8)	83.7 (0.7)	84.3 (0.7)
	Avg	**88.8**	88.6	88.4	**88.8**	**88.8**	88.7
G-2B	TQA	76.6 (0.2)	76.8 (0.9)	79.9 (0.9)	77.1 (0.2)	77.8 (0.5)	80.6 (0.3)
G-2B	NQ	79.9 (0.4)	81.0 (0.2)	82.0 (0.7)	80.2 (0.8)	80.8 (0.4)	83.1 (0.1)
G-2B	STR	55.2 (1.7)	58.5 (0.0)	56.3 (4.3)	52.9 (4.5)	56.3 (2.6)	54.4 (3.8)
	Avg	70.6	72.1	**72.7**	70.1	71.6	**72.7**
L3I-8B	TQA	84.7 (0.7)	85.1 (0.7)	86.9 (0.7)	85.3 (0.5)	86.1 (0.6)	88.3 (0.1)
L3I-8B	STR	49.9 (1.0)	49.3 (0.7)	48.9 (1.9)	50.3 (1.7)	50.5 (1.2)	49.7 (2.5)
	Avg	67.3	67.2	67.9	67.8	68.3	**69.0**
				Sampled Test Responses			
L-7B	TQA	74.0 (1.2)	77.4 (3.1)	81.3 (1.4)	90.0 (0.7)	90.5 (0.0)	91.2 (0.1)
L-7B	NQ	72.7 (0.8)	73.8 (1.4)	76.2 (2.7)	90.5 (0.8)	91.2 (0.7)	92.3 (0.2)
L-7B	STR	85.7 (0.2)	85.6 (0.1)	86.5 (0.3)	85.8 (0.1)	85.3 (0.1)	86.4 (0.3)
	Avg	77.5	78.9	81.3	88.8	89.0	**89.9**
A-7B	TQA	77.1 (0.8)	77.2 (3.4)	75.3 (3.6)	88.8 (1.0)	88.6 (0.1)	90.3 (0.1)
A-7B	NQ	71.6 (1.2)	69.3 (3.7)	68.6 (0.4)	85.1 (1.9)	87.5 (0.2)	88.3 (0.6)
A-7B	STR	84.4 (0.6)	84.6 (0.6)	85.4 (0.3)	84.2 (0.5)	84.2 (0.4)	85.6 (0.3)
	Avg	77.7	77.0	76.5	86.0	86.8	**88.1**
V-7B	TQA	77.0 (6.3)	75.4 (2.4)	73.1 (3.9)	95.4 (0.3)	95.3 (0.6)	95.2 (0.4)
V-7B	NQ	76.5 (5.4)	76.7 (7.3)	79.3 (2.8)	94.9 (0.5)	95.4 (0.1)	95.2 (0.4)
V-7B	STR	89.3 (0.2)	89.1 (0.2)	88.9 (0.4)	89.1 (0.2)	89.1 (0.1)	89.5 (0.1)
	Avg	80.9	80.4	80.4	93.1	**93.3**	**93.3**
G-2B	TQA	85.3 (0.3)	85.7 (0.5)	88.3 (0.2)	85.8 (0.0)	86.1 (1.1)	88.8 (0.0)
G-2B	NQ	83.2 (0.4)	84.9 (0.5)	87.5 (0.7)	82.1 (0.6)	84.6 (1.0)	87.4 (0.9)
G-2B	STR	52.7 (0.4)	53.4 (0.5)	53.3 (0.5)	52.3 (0.7)	52.8 (0.5)	53.2 (0.3)
	Avg	73.7	74.7	76.3	73.4	74.5	**76.5**
L3I-8B	TQA	90.4 (0.1)	90.4 (0.3)	91.4 (0.2)	90.2 (0.0)	90.4 (0.3)	91.8 (0.2)
L3I-8B	STR	57.5 (0.6)	57.3 (0.4)	57.3 (0.3)	56.9 (0.6)	57.8 (0.5)	58.1 (0.7)
	Avg	73.9	73.8	74.4	73.5	74.1	**74.9**

C. Design Ablations

C.1. Comparing CLAP with Token-wise Attention-pooling

In table 15 we compare CLAP with attention pooling [29], which implements a learnable query vector followed by softmax pooling to aggregate token-wise activations at each layer before training a logistic

Table 11
Mitigating hallucinations with fine-grained detection using CLAP applied to Random Sampling. Second block shows the % of non-hallucinated responses for each of the five response generation strategies. Third and forth blocks show the % of responses abstained and the % of responses that were abstained but were non-hallucinated (NH), respectively. For results with +CLAP-I and +CLAP-II we report mean across three seeds. * denotes %non-hallucinations among non-abstained responses.

	% Non-Hallucinations ↑					%Abs ↓		%Abs but NH ↓	
Data	Def	Def+Abs*	Alt	+CLAP-I	+CLAP-II*	Def+Abs	+CLAP-II	Def+Abs	+CLAP-II
L-7B TQA	57.3	75.6	22.8	51.8	71.8	36.0	29.7	9.1	6.9
L-7B STR	60.0	69.9	32.3	51.9	59.9	45.0	22.7	21.4	10.9
A-7B TQA	34.2	68.3	26.9	37.6	67.3	62.7	50.4	8.7	6.3
A-7B STR	43.4	67.0	39.7	48.2	61.4	44.3	31.6	6.1	3.0
V-7B TQA	14.6	59.2	6.6	15.6	53.7	80.7	72.6	3.2	2.8
V-7B STR	42.8	65.6	32.8	51.2	59.9	40.3	22.6	3.6	2.1
Average	42.1	67.6	26.9	42.7	62.3	51.5	38.3	8.7	5.3

Table 12
Hallucination Mitigation: CLAP-I versus Last Layer Probing when combined with DoLa, best is indicated in bold. Second block shows the % of non-hallucinated responses using each method. Third and fourth blocks show the ratio of % H->NH (or NH->H) replacements using LP-I/NLP-I/CLAP-I against % H->NH (or NH->H) replacements using Alt. Results averaged across TQA and STR, across three seeds.

	% Non-Hallucinations ↑			H -> NH ↑			NH -> H ↓		
LLM	+LP-I	+NLP-I	+CLAP-I	+LP-I	+NLP-I	+CLAP-I	+LP-I	+NLP-I	+CLAP-I
Llama 7B	60.2	60.0	**61.1**	55.8	45.8	**57.6**	34.4	**26.8**	32.8
Alpaca 7B	51.6	52.7	**53.4**	77.2	78.1	**80.1**	26.4	**21.6**	21.8

Table 13
Hallucination Mitigation: CLAP-II versus Last Layer Probing when combined with DoLa, best is indicated in bold. Second block shows the % of non-hallucinated responses using each method. Third and fourth blocks show the % of responses abstained and the % of responses that were abstained but were non-hallucinated (NH), respectively. * denotes %non-hallucinations among non-abstained responses. Results averaged across TQA and STR, across three seeds.

	% Non-Hallucinations ↑			%Abs ↓			%Abs but NH ↓		
LLM	+LP-II*	+NLP-II*	+CLAP-II*	+LP-II	+NLP-II	+CLAP-II	+LP-II	+NLP-II	+CLAP-II
Llama 7B	67.8	68.5	**69.5**	**16.4**	18.5	16.8	4.9	5.3	**4.7**
Alpaca 7B	66.6	64.9	**66.7**	33.1	**29.2**	30.2	5.6	**4.2**	4.5

regression probe on the pooled activation vector. Following the original work, we train 2L attention-pooling probes where L denotes the number of LLM decoder layers and probes are trained at both layer output as well as attention output (after residual connection) positions. After training the 2L probes, the individual probe weights are frozen and an ensemble logistic regression probe is trained on the output of the individual probes. **Att-Pool (MA)** denotes the best individual probe out of 2L probes (chosen using in-distribution validation data), while **Att-Pool-Ens** denotes the ensemble probe. We implement attention pooling with 20 tokens, taking either the last 20 or padding to 20 with zero vectors, as required[1]. We find that while token-wise attention pooling slightly outperforms CLAP on in-distribution testing, CLAP significantly outperforms in the out-of-distribution setting, demonstrating its superiority.

[1]We train all probes including CLAP on 2000 samples instead of the 5000 samples used for the main experiments, due to the GPU memory constraint of loading token-wise activations for all layers when training.

Table 14
Comparison of LP-II/NLP-II with DoLa versus CLAP-II with DoLa, best is indicated in bold. Shows the ratio of % H->NH (or NH->H) replacements using LP-II/NLP-II/CLAP-II against % H->NH (or NH->H) replacements using Alt. Results averaged across TQA and STR, across three seeds.

	H -> NH ↑			NH -> H ↓		
LLM	+LP-II	+NLP-II	+CLAP-II	+LP-II	+NLP-II	+CLAP-II
Llama 7B	41.0	27.5	**42.6**	19.8	**11.7**	19.0
Alpaca 7B	49.2	57.0	**60.4**	**3.8**	4.2	4.2

Table 15
Comparing CLAP with token-wise attention-pooling, using AUC scores on test data, best is indicated in bold. Results on Gemma 2B, using greedy responses for train and test.

Probe	# Params	Flops	In Distribution TQA->TQA	Out Of Distribution TQA->City
Att-Pool (MA)	2K	84K	**77.7 (1.2)**	72.3 (4.4)
Att-Pool-Ens	76K	3.1M	77.3(0.4)	66.4 (4.9)
CLAP	930K	22.2M	76.3 (0.3)	**79.0 (3.9)**

C.2. Analysis of Hyper-parameter Choices for CLAP

Table 16 reports the effect of varying the two architectural hyper-parameters n_{enc} and d_{model} on the validation data for the Alpaca 7B, Vicuna 7B, Gemma 2B and Llama3.1-Instruct 8B models.

D. Inference cost

Table 17 shows the memory and computation cost at inference time for the compared hallucination detection methods, measured in terms of the number of parameters and the number of floating point operations (flops), respectively. For the black-box methods that involve additional response sampling, flops for generating one output token is estimated using the standard formula for transformers - 2 x N, where N denotes the number of parameters of the LLM. The total cost of detection then involves the cost of generating n_s additional samples of n_t tokens each and the cost of NLI-based/prompt-based comparison of the greedy response against each of the n_s sampled responses. For Self-Check NLI, the recommended DeBERTa-v3-large-mnli model is assumed. For Self-Check Prompt, a single token YES/NO response is assumed. Unsurprisingly, the probing based methods are significantly more compute efficient than the black-box methods. Amongst the probing based methods, while CLAP increases the compute cost, this is still negligible compared to performing black-box detection.

Table 16
Analysis of hyper-parameter choices for CLAP, using AUC scores on validation data.

n_{enc}	d_{model}	TQA	NQ	STR	TQA	NQ	STR
		Alpaca 7B			Vicuna 7B		
1	128	87.3 (0.4)	89.0 (1.2)	83.8 (0.2)	93.0 (0.9)	88.1 (0.1)	85.1 (1.4)
1	256	87.2 (0.5)	**89.0 (1.0)**	83.6 (0.2)	92.8 (0.8)	88.3 (0.3)	84.7 (1.2)
1	512	**87.4 (0.5)**	88.6 (1.2)	83.8 (0.6)	92.8 (0.9)	88.1 (0.3)	85.3 (1.3)
1	1024	87.3 (0.5)	88.5 (1.0)	83.7 (0.3)	**93.0 (0.8)**	88.1 (0.2)	85.1 (1.4)
1	2048	**87.4 (0.5)**	88.3 (1.0)	83.5 (0.3)	92.8 (0.9)	**88.3 (0.2)**	85.4 (1.4)
1	4096*	87.3 (0.7)	88.6 (1.0)	83.4 (0.4)	92.9 (0.8)	88.1 (0.1)	**85.6 (1.6)**
2	128	87.2 (0.7)	88.9 (1.3)	83.4 (0.3)	92.9 (0.8)	87.9 (0.1)	84.8 (1.5)
2	256	87.3 (0.5)	**89.0 (1.0)**	83.4 (0.3)	92.7 (0.8)	88.2 (0.2)	84.8 (1.2)
2	512	87.3 (0.4)	88.7 (1.2)	83.7 (0.5)	92.8 (0.8)	88.2 (0.3)	85.0 (1.0)
2	1024	87.2 (0.5)	88.7 (1.2)	83.8 (0.4)	**93.0 (0.8)**	88.0 (0.1)	85.0 (1.6)
2	2048	87.2 (0.4)	88.6 (1.3)	**83.9 (0.4)**	93.0 (0.9)	88.2 (0.3)	85.2 (1.6)
2	4096*	87.0 (0.4)	88.3 (1.3)	83.7 (0.8)	92.6 (0.8)	88.1 (0.3)	85.4 (1.5)
		Gemma 2B			Llama3.1-Instruct 8B		
1	128	80.8 (0.6)	79.5 (0.8)	59.1 (1.6)	87.6 (0.5)	-	55.5 (0.6)
1	256	80.8 (0.5)	79.3 (0.4)	59.2 (2.3)	87.9 (0.6)	-	55.9 (0.7)
1	512	80.9 (0.3)	79.4 (0.5)	59.5 (1.3)	87.9 (0.4)	-	55.5 (0.3)
1	1024	80.6 (0.6)	**79.6 (0.1)**	**60.0 (1.9)**	87.8 (0.4)	-	55.9 (0.3)
1	2048	**81.3 (0.4)***	79.1 (1.0)*	59.6 (2.6)*	87.8 (0.4)	-	56.0 (0.3)
1	4096*	-	-	-	87.2 (0.4)	-	55.5 (0.6)
2	128	80.7 (0.8)	79.5 (0.2)	59.5 (2.0)	87.9 (0.6)	-	55.7 (0.4)
2	256	80.8 (0.7)	**79.6 (0.1)**	59.3 (2.0)	88.2 (0.6)	-	55.8 (1.0)
2	512	80.6 (0.6)	79.3 (0.4)	59.2 (1.4)	88.2 (0.3)	-	56.1 (0.6)
2	1024	80.6 (0.9)	79.1 (0.4)	58.8 (1.8)	**88.4 (0.3)**	-	55.7 (0.9)
2	2048	80.7 (0.5)*	78.9 (0.8)*	58.0 (1.9)*	88.3 (0.2)	-	55.8 (0.6)
2	4096*	-	-	-	87.8 (0.3)	-	**56.7 (0.7)**

Table 17
Comparison of memory and computation cost of hallucination detection methods, measured using number of parameters and number of floating point operations (Flops), respectively and estimated for the Llama-7B model dimensions. K denotes one thousand, M denotes one million, B denotes one billion, n_s denotes number of responses sampled and n_t denotes number of tokens generated in the response.

Method Category	Method	# Params	Flops
Single-layer probing	AH	128	128
Single-layer probing	LP/SEP/Most Accurate	4K	4K
Single-layer probing	NLP	1.1M	1.1M
Multi-layer probing	Most Confident/Majority Voting	135K	135K
Multi-layer probing	CLAP ($n_{enc} = 1, d_{model} = 128$)	826K	28.8M
Multi-layer probing	CLAP ($n_{enc} = 2, d_{model} = 128$)	1.1M	40.3M
Black-box	Self-Check NLI	7B +304M	(n_s x n_t x 2 x 7B) +(n_s x 2 x 304M)
Black-box	Self-Check Prompt	7B	(n_s x n_t x 2 x 7B) +(n_s x 2 x 7B)