

Beyond Token Probes: Hallucination Detection via Activation Tensors with ACT-ViT

Technical Appendices

A Extended Experimental Section

Our experiments were conducted using the PyTorch [48] framework (License: BSD), using a single NVIDIA L-40 GPU for all experiments. We use a fixed batch size of 128 for all experiments, other than the ones with ACT-ViT(s), ACT-MLP(s), where we used a batch size of 64. We used 4 heads in the transformer part of ViT for all experiments. Hyperparameter tuning was performed utilizing the Weight and Biases framework [8] – see Appendix A.1

Optimizer and Schedulers. For all datasets, we use the AdamW optimizer [37] in combination with a cosine learning rate scheduler, incorporating a warm-up phase over the first 10% of training epochs.

LLMs. We consider the following LLMs for our experiments:

1. **Mistral-7b-instruct-v0.2** [25] (License: Apache-2.0). Referred to as Mis-7B in the main text and accessed through the Hugging Face interface at <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>.
2. **Llama-3-8b-Instruct** [61] (License: Llama-3⁴). Referred to as LLaMa-8B in the main text and accessed through the Hugging Face interface at <https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>.
3. **Qwen-2.5-7b-Instruct** (License: Apache-2.0): Referred to as Qwen-7B in the main text and accessed through the Hugging Face interface at <https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>.

Generating Activation Tensors. Given a large language model (LLM) M and an output sequence of N tokens, we define the *activation tensor* $\mathbf{A} \in \mathbb{R}^{L_M \times N \times D_M}$, where L_M is the number of layers in M , and D_M is the hidden dimension. Let $\mathbf{A}^{(l)} := \mathbf{A}[l, :, :]$ denote the hidden representations at the l -th layer across the entire sequence. In a standard transformer architecture, these representations are computed as:

$$\mathbf{A}^{(l)} = \mathbf{A}^{(l-1)} + \left[\text{ReLU} \left(\text{Attn}(\mathbf{A}^{(l-1)}) \mathbf{W}_l^1 + \mathbf{b}_l^1 \right) \right] \mathbf{W}_l^2 + \mathbf{b}_l^2, \quad (1)$$

where Attn denotes the output of the multi-head attention mechanism, followed by batch normalization. The matrices $\mathbf{W}_l^1, \mathbf{W}_l^2 \in \mathbb{R}^{D_M \times D_M}$ and biases $\mathbf{b}_l^1, \mathbf{b}_l^2 \in \mathbb{R}^{D_M}$ correspond to the feed-forward subnetwork at layer l . Throughout this paper, we use the term *activation tensor* to refer to \mathbf{A} as defined in Equation (1).

A.1 HyperParameters

This section outlines the hyperparameter search performed for our experiments. We employ the same hyperparameter grid for both our primary model, ACT-ViT, and our proposed baseline, ACT-MLP⁵. Hyperparameter grid search is performed to optimize the AUC on the validation set. The best hyperparameters are selected based on the run that achieves the highest validation AUC, for each of our two proposed methods, namely ACT-ViT, and ACT-MLP. The hyperparameter grids used for all experiments are outlined below, corresponding to each experimental setup:

1. Training on a single LLM-specific dataset – see Table 3
2. Joint training on all 15 datasets simultaneously – see Table 4
3. Leave-one-out training (excluding 1 out of 15 datasets) – upper part of Table 5
4. Low-data regime adaptation on the held-out dataset – lower part of Table 5. “–” refers to slot which are inapplicable, or taken from the pre-trained model.
5. Leaving out one LLM (excluding 5 out of 15 LLM-dataset combinations) – see Table 6

⁴<https://huggingface.co/meta-llama/Meta-Llama-3-8B/blob/main/LICENSE>

⁵The patch size parameter does not apply to these baselines, as it is not part of their configuration.

For the probing baselines—specifically Token[n] for $n \in \mathcal{N}$ (see Section 4) and Probe[*]—we performed a grid search over inverse regularization strengths $C \in \{10000, 100, 1.0, 0.01, 0.0001\}$. For each token probe, we selected the best-performing model based on validation set performance, corresponding to the optimal value of C . In the case of Probe[*], we additionally selected the best token position using the validation set.

Table 3: Hyperparameter search grid for ACT-ViT(s) and ACT-MLP(s), for each of the 15 LLM-dataset combinations.

Hyperparameter Search Grid for each of the 15 LLM-dataset combinations	
Number of layers	{1, 3}
Learning rate	0.001
Embedding size	{128, 1024}
Epochs	15
Dropout	0.3
Weight Decay	{1, 0.001}
Patch Size	{(1, 1), (8, 1), (4, 2)}

Table 4: Hyperparameter search grid for ACT-ViT and ACT-MLP, in the setting where training is performed jointly across all 15 LLM-dataset combinations.

Hyperparameter Search Grid for joint-training of the 15 LLM-dataset combinations	
Number of layers	3
Learning rate	{0.001, 0.0005}
Embedding size	128
Epochs	5
Dropout	0.3
Weight Decay	{10, 0.001}
Patch Size	{(1, 1), (1, 2), (1, 4), (2, 1), (4, 1)}

Table 5: Hyperparameter search grid used for ACT-ViT and ACT-MLP when trained on the combined set of all 14 LLM datasets, in the “leaving one dataset out” setup. “–” denotes slots that are inapplicable, as their values are inherited from the pre-trained model.

Num. layers	Learning rate	Embedding size	Epochs	Dropout	Weight Decay	Patch size
Pre training (and Zero-shot)						
3	0.001	128	15	0.3	0.001	(1, 1)
Low-data regime LA adaptation (over {5%, 10%, 20%, 50%, 100%} of 10,000 test samples)						
–	0.001	–	5	–	0.001	–

A.2 Dataset Description

In this section, we provide an overview of the five datasets used in our analysis; we mostly follow the framework given in [46] in constructing the datasets. We aimed to cover diverse tasks, reasoning skills, and datasets, highlighting each one’s unique value and how it complements the rest.

For all datasets, we used a consistent split of 10,000 training samples and 10,000 test samples, unless otherwise specified. From the 10,000 training samples, 20% (i.e., 2,000) were selected in a stratified manner for validation, using a fixed random seed of 42.

- HotpotQA with and without context** (License: CC-BY-SA-4.0) [65]: HotpotQA is a multi-hop question answering dataset featuring diverse questions that require reasoning across multiple sources. Each instance includes supporting Wikipedia documents. We use two settings in our analysis: (1) *Without context*, where questions are presented alone, testing

Table 6: Hyperparameter search grid used for ACT-ViT and ACT-MLP when trained on the combined set of 10 LLM datasets, in the “leaving one LLM out” setup. “–” denotes slots that are inapplicable, as their values are inherited from the pre-trained model.

Num. layers	Learning rate	Embedding size	Epochs	Dropout	Weight Decay	Patch size
Pre training						
{3, 5}	0.001	128	5	0.3	0.001	{(1, 1), (8, 1), (4, 2)}
LA Adaptation						
–	0.001	–	15	–	{0.1, 0.01, 10, 20}	–

factual recall and reasoning; and (2) *With context*, where supporting documents are provided, emphasizing the model’s ability to leverage contextual information effectively.

2. **Movies** [46] (License: MIT): We use this dataset to evaluate generalization in scenarios regarding movies involving factual inaccuracies (i.e., hallucinations). This dataset contains 7857 test samples.
3. **IMDB** (originally released with no known license by Maas et al. [38]): This dataset consists of movie reviews for sentiment classification. Following the method in [46], we employed a one-shot prompt to help the large language model (LLM) apply the predefined sentiment labels accurately.
4. **TriviaQA** (originally released with no known license by [26]): A dataset of trivia question-answer pairs presented to the LLM without any supporting context, relying only on its internal parametric knowledge. Multiple acceptable answer variants are provided to facilitate automatic evaluation of response accuracy.

A.3 Additional Results

Below in Table 7 we provide a comparison of ACT-ViT with LOS-Net.

Table 7: Comparison of ACT-ViT with LOS-Net. For LOS-Net, we report the 1-sigma error-bars. Best is in **Bold**.

Method	HotpotQA	IMDB	Movies	HotpotQA	IMDB	Movies	HotpotQA	IMDB	Movies
	Mis-7B			LlaMa-8B			Qwen-7B		
LOS-Net	72.92 ± 0.45	94.73 ± 0.58	72.20 ± 0.66	72.60 ± 0.34	90.57 ± 0.28	77.43 ± 0.66	73.71 ± 1.21	88.19 ± 0.88	88.01 ± 0.39
ACT-ViT	84.33	97.03	79.63	82.73	94.12	84.81	87.62	96.22	95.08

A.4 Run-Time

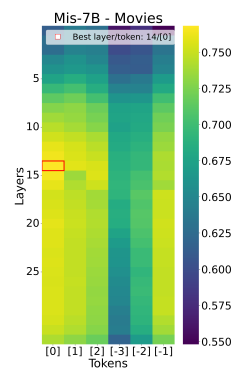
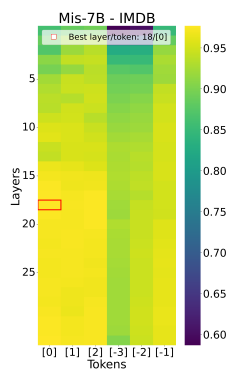
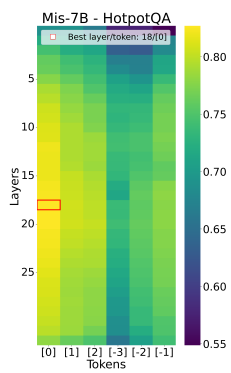
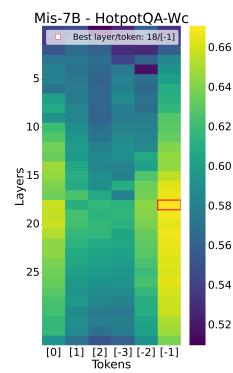
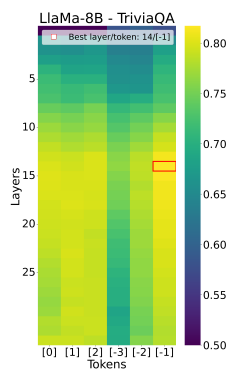
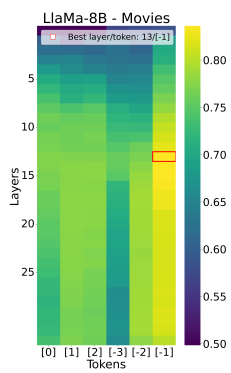
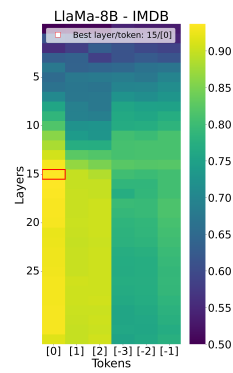
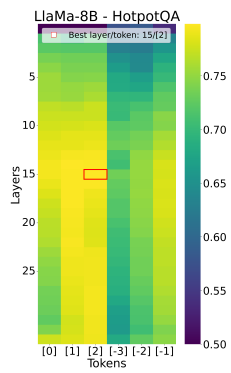
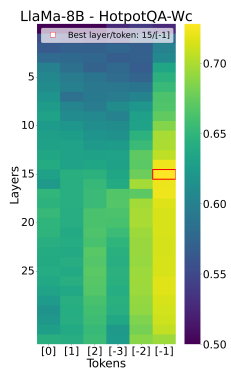
We present the run-time of ACT-ViT(s), on a single NVIDIA L-40 GPU, compared with probing classifiers, on each of the 15 LLM-dataset combinations considered in the paper; see Table 8.

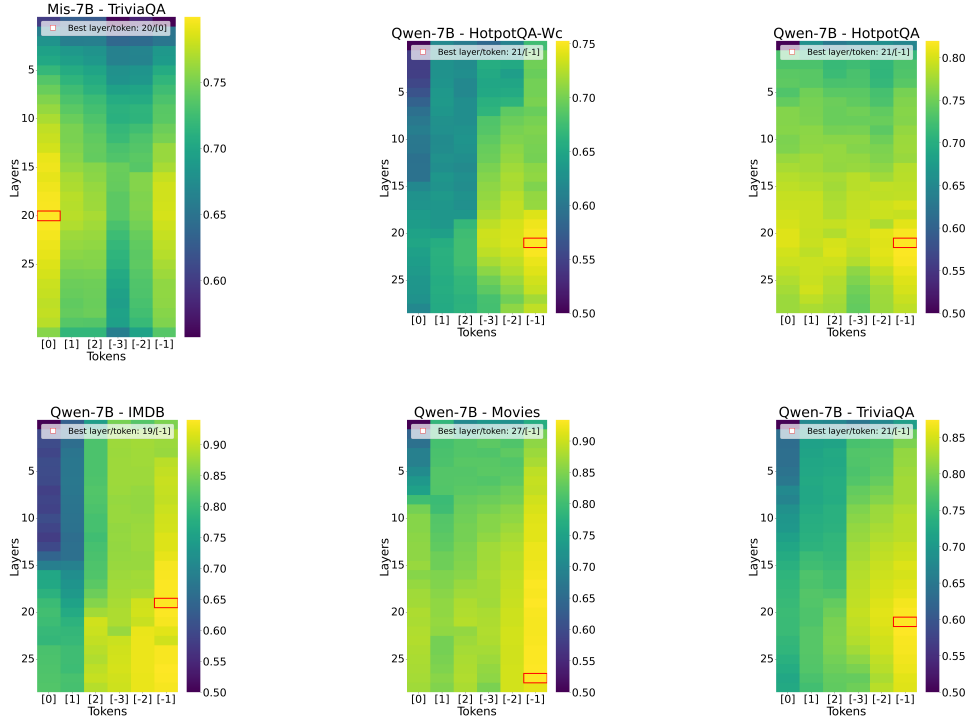
Table 8: Training time (in minutes [m] and seconds [s]) for Probe[*] and ACT-ViT across all 15 LLM–dataset combinations. Each ACT-ViT training run was performed on a single NVIDIA L-40 GPU.

LLM	Method	HotpotQA	Movies	HQA-Wc	IMDB	TriviaQA
Mis-7B	Probe[*]	2[m] 45[s]	6[m] 23[s]	1[m] 41[s]	1[m] 38[s]	1[m] 50[s]
	ACT-ViT	13[m] 2[s]	11[m] 28[s]	12[m] 49[s]	14[m] 17[s]	12[m] 42[s]
LlaMa-8B	Probe[*]	2[m] 50[s]	2[m] 5[s]	1[m] 54[s]	2[m] 22[s]	2[m] 36[s]
	ACT-ViT	13[m] 48[s]	27[m] 5[s]	16[m] 34[s]	13[m] 1[s]	27[m] 21[s]
Qwen-7B	Probe[*]	2[m] 0[s]	1[m] 43[s]	1[m] 51[s]	1[m] 39[s]	1[m] 53[s]
	ACT-ViT	10[m] 29[s]	13[m] 12[s]	11[m] 49[s]	11[m] 30[s]	10[m] 46[s]

1026 **A.5 Layer/Token Visualizations**

1027 Below we present the layer-token heatmaps, corresponding to each of the 15 LLM-dataset combination
1028 considered in our paper.





1029 A.6 Low-Data Regime Adaptation

1030 Below, we present additional results in the low-data training regime for all 15 LLM-dataset combina-
 1031 tions considered in our paper.

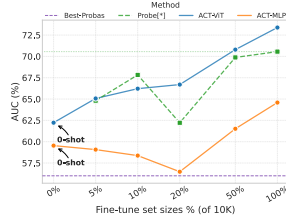


Figure 6: LLaMa-8B – HotpotQA-Wc

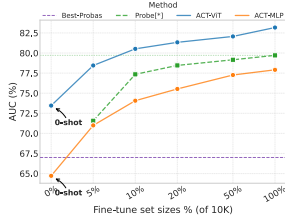


Figure 7: LLaMa-8B – HotpotQA

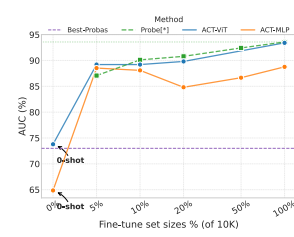


Figure 8: LLaMa-8B – IMDB

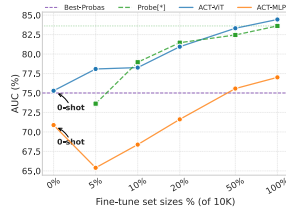


Figure 9: LLaMa-8B – Movies

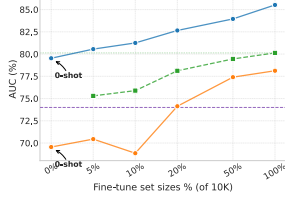


Figure 10: LLaMa-8B – TriviaQA

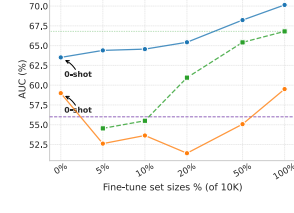


Figure 11: Mis-7B – HotpotQA-Wc

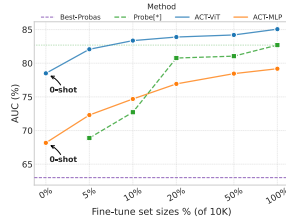


Figure 12: Mis-7B – HotpotQA

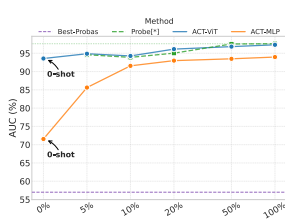


Figure 13: Mis-7B – IMDB

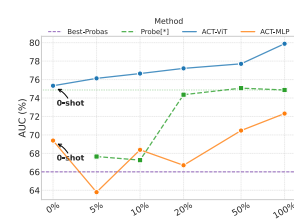


Figure 14: Mis-7B – Movies

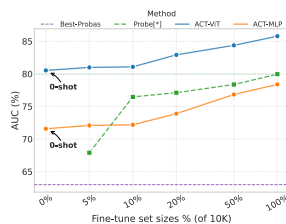


Figure 15: Mis-7B – TriviaQA

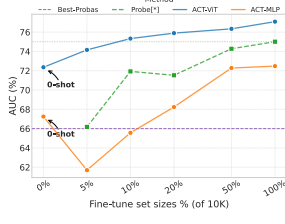


Figure 16: Qwen-7B – HotpotQA-Wc

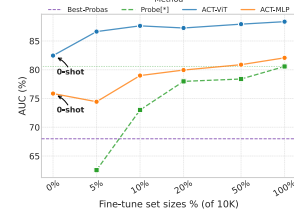


Figure 17: Qwen-7B – HotpotQA

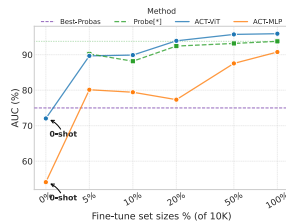


Figure 18: Qwen-7B – IMDB

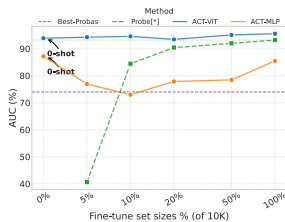


Figure 19: Qwen-7B – Movies

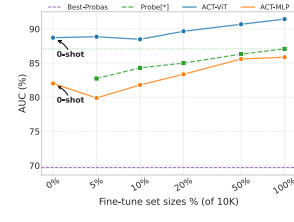


Figure 20: Qwen-7B – TriviaQA

B Pooling Algorithm

Our pooling algorithm is presented in Algorithm 1.

Algorithm 1 Pooling

Require: Tensor $\mathbf{A} \in \mathbb{R}^{L_M \times N \times D_M}$, target sizes L_p, N_p
Ensure: Tensor $\mathbf{A} \in \mathbb{R}^{L_p \times N_p \times D_M}$
 1: $\mathbf{A} \leftarrow \text{permute}(\mathbf{A}, [D_M, L_M, N])$
 2: Pad \mathbf{A} so that L_M and N are divisible by L_p and N_p
 3: Compute patch sizes: $f_L = \frac{L_{\text{pad}}}{L_p}, f_N = \frac{N_{\text{pad}}}{N_p}$
 4: Apply pooling:
 $\mathbf{A} \leftarrow \text{F.max_pool2d}(\mathbf{A}, \text{kernel_size} = (f_L, f_N))$
 5: $\mathbf{A} \leftarrow \text{permute}(\mathbf{A}, [L_p, N_p, D_M])$
 6: **return** \mathbf{A}

C Permutation Symmetries

Given the same number of layers and hidden dimensions, two LLMs can compute the exact same function and yet produce entirely different activation tensors. This discrepancy stems from internal symmetries in the model’s weight space. To illustrate this, we consider a simplified transformer architecture with L layers, omitting residual connections and batch normalization for clarity (though the analysis extends to those cases as well).

Recall that \mathbf{A}^l is the activation tensor at layer $l \in [L]$. Let $(\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l)$ denote the query, key, and value matrices at layer l , computed as:

$$(\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l) = (\mathbf{A}^{l-1} \mathbf{W}_l^{Q_l}, \mathbf{A}^{l-1} \mathbf{W}_l^{K_l}, \mathbf{A}^{l-1} \mathbf{W}_l^{V_l}). \quad (2)$$

Assume the feed-forward network (FFN) at layer l produces output:

$$\mathbf{A}^l = \text{ReLU}(\text{Attn}(\mathbf{A}^{l-1}) \mathbf{W}_l^1 + \mathbf{b}_l^1) \mathbf{W}_l^2 + \mathbf{b}_l^2. \quad (3)$$

Now assume the model uses d -dimensional hidden features, and let $P \in \mathbb{R}^{d \times d}$ be a permutation matrix, and define the following weight transformation:

$$(\mathbf{W}_l^2, \mathbf{b}_l^2) \rightarrow (\mathbf{W}_l^2 P^\top, \mathbf{b}_l^2 P^\top), \quad (4)$$

$$(\mathbf{W}_{l+1}^{Q_{l+1}}, \mathbf{W}_{l+1}^{K_{l+1}}, \mathbf{W}_{l+1}^{V_{l+1}}) \rightarrow (P \mathbf{W}_{l+1}^{Q_{l+1}}, P \mathbf{W}_{l+1}^{K_{l+1}}, P \mathbf{W}_{l+1}^{V_{l+1}}). \quad (5)$$

Although the new weights (the right hand side) are very different, it is straightforward to verify that the permutation matrices cancel out, leaving the underlying function unchanged. However, recalling Equation (3), the activation tensors differ significantly, as the transformation permutes the feature dimension as follows:

$$\mathbf{A}[l, n, d] \rightarrow \mathbf{A}[l, n, \sigma(d)], \quad (6)$$

where σ is the permutation induced by P .

This illustrative example remarks the relevance of resorting to LLM-specific LA modules, as opposed to using a single shared linear adapter. Although padding can side-step the dimensionality problem, such a single adapter would be required to implicitly learn (to be invariant to) all feature permutations that may potentially arise ($D!$). While we note that principled approaches for handling such symmetries exist – such as designing layers that are invariant to feature permutations by design – we consider these less relevant to us given the limited number of considered LLMs. Exploring symmetry-aware architectures still remains a promising direction we envision to explore in future endeavors.

Extended Analysis to Standard Transformers Activations. The implications extend naturally to a standard transformer that includes residual connections and batch normalization. Residual connections do not affect the analysis, since in our analysis the permutation symmetry is applied uniformly across layers—the same permutation is used for all weight matrices, so the residual

1062 addition remains consistent with our analysis. Batch normalization also preserves this symmetry, as
1063 it is a permutation-equivariant operation: applying a permutation to the input features results in a
1064 correspondingly permuted output.

1065 **D Broader Impact**

1066 By enhancing the detection of hallucinations in Large Language Models, our work contributes to the
1067 responsible development and deployment of generative AI by promoting greater transparency and
1068 trust. However, we acknowledge that our findings also reveal aspects of the predictive information
1069 embedded within LLM internals. This insight could be misused by malicious actors, potentially
1070 motivating restrictions on LLM internals access and thereby slowing progress in open research.