Attention Consistency for LLMs Explanation

Tian Lan[1,*], Jinyuan Xu[2,*], Xue He[3,4], Jenq-Neng Hwang[5], Lei Li[5,6,†]

[1]Milkuya Studio [2]ERTIM, INALCO [3]Sorbonne University
[4]IRD [5]University of Washington [6]VitaSight

Abstract

Understanding the decision-making processes of large language models (LLMs) is essential for their trustworthy development and deployment. However, current interpretability methods often face challenges such as low resolution and high computational cost. To address these limitations, we propose the **Multi-Layer Attention Consistency Score (MACS)**, a novel, lightweight, and easily deployable heuristic for estimating the importance of input tokens in decoder-based models. MACS measures contributions of input tokens based on the consistency of maximal attention. Empirical evaluations demonstrate that MACS achieves a favorable trade-off between interpretability quality and computational efficiency, showing faithfulness comparable to complex techniques with a 22% decrease in VRAM usage and 30% reduction in latency.

1 Introduction

The rapid advancements in Large Language Models (LLMs) have revolutionized natural language processing, powering diverse applications (Cai et al., 2025a,b; He et al., 2025; Jin et al., 2025; Li et al., 2025a,b; Yao et al., 2025). As these models become increasingly integrated into critical systems, understanding their decision-making processes is crucial for ensuring trustworthiness, reliability, and enabling targeted improvements. Explainable AI (XAI), particularly methods attributing outputs to input tokens, is thus essential. However, existing approaches for interpreting deep decoder-only Transformers face notable limitations: full attention aggregation methods like Attention Rollout can produce noisy or diffuse attributions, potentially due to phenomena like Over-squashing, Over-mixing, and softmax dispersion which can obscure important signals in deep models and long sequences. Concurrently, many other XAI techniques, including gradient and perturbation-based approaches, are often too computationally intensive, limiting their practicality for real-time diagnostics and require model modifications or specialized computation paths. Table 1 summarizes the comparison of MACS and these state-of-the-art XAI methods, outlining their respective advantages and limitations across key interpretability dimensions. [1] [2]

Metric	**MACS**	Rollout	AttnLRP	AtMan
Faithfulness	✓	✗	✓	✗
Usability	✓	✓	✗	✗
Comp. Eff.	✓	✗	✓	✗
VRAM Eff.	✓	✗	✓	✓

Table 1: Comparative summary of MACS (ours) and other XAI methods, highlighting their strengths and weaknesses across key interpretability criteria. (✓) denotes strong performance, (✗) indicates poor performance. MACS demonstrates competitive results across most metrics.

To address the need for efficient and insightful interpretability, we propose the **Multi-Layer Attention Consistency Score (MACS)**, a novel, lightweight heuristic for estimating the importance of input tokens in Transformer models. Unlike conventional attribution techniques, MACS quantifies token contribution by evaluating the consistency of the strongest attention connection from an output query to each input token across all layers. This consistency is derived via layer-wise max pooling followed by element-wise (Hadamard) multiplication. MACS can be applied directly during inference without requiring any model modifications.

Our main contributions are as follows:

- **MACS Methodology**: We propose the Multi-Layer Attention Consistency Score (MACS), a lightweight and computationally

[1]* Both authors contributed equally to this research.
[2]† The corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1736–1750
November 4-9, 2025

efficient heuristic for input token attribution in Transformer-based models, leveraging cross-layer attention consistency.

- **Theoretical Motivation**: We provide a theoretical grounding for MACS as a response to known information propagation challenges (Over-squashing, Over-mixing, Softmax Dispersion), which often limit the effectiveness of existing attribution techniques.
- **Empirical Evaluation**: Through experiments on Question Answering tasks, we demonstrate that MACS identifies salient tokens more effectively than Attention Rollout and achieves faithfulness comparable to more complex attribution methods.
- **Practical Utility**: We demonstrate the real-time capability of MACS by highlighting its efficiency and showcase its potential applicability beyond text-only models to other Transformer architectures, as preliminarily explored in the context of Visual Question Answering (see Appendix B).

We show that MACS achieves a favorable trade-off between computational efficiency and attribution quality, making it a practical component for interpreting LLMs.

2 Background

Understanding the decision-making process of Transformer-based models, especially Large Language Models (LLMs), is a critical research objective in the field of Explainable AI (XAI) (Liu et al., 2025; Shi et al., 2025). Existing attribution methods that link model predictions to input tokens can be broadly categorized by their primary mechanisms, including attention aggregation, perturbation, and backpropagation.

2.1 Explainability via Attention Aggregation and its Challenges

The attention mechanism has been a natural candidate for interpretability due to its explicit weighting of token interactions(Zhang et al., 2023). Early studies demonstrated its usefulness in tasks such as machine translation (Zenkel et al., 2019) and document classification (Yang et al., 2016). Subsequent analyses of models like BERT revealed that individual attention heads can capture linguistically meaningful patterns (Clark et al., 2019)

However, interpreting raw attention from single layers or heads in deep, multi-head Transformers is often insufficient. To obtain a more holistic view, some methods focusing on attention aggregation across layers, like Attention Rollout (Abnar and Zuidema, 2020), have been proposed. Rollout typically propagates attention by multiplying attention matrices across layers, thereby tracing and accumulating influence through all paths. Although more comprehensive than raw attention, such full-aggregation techniques may still yield "noisy" or diffuse attribution maps, making it challenging to identify critical input tokens.

Recent theoretical studies have highlighted challenges associated with aggregating attention signals in deep decoder-based Transformer models(Jia and Li, 2024). Barbero et al. (2024, 2025) formalized phenomena like Over-squashing, where influence from distant inputs is severely attenuated, and Over-mixing/Representational Collapse, where token representations lose distinctiveness. Furthermore, Veličković et al. (2024) demonstrate that standard softmax attention inherently disperses and loses its ability to focus sharply as sequence length increases out-of-distribution. These theoretical limitations suggest that methods like Rollout, by design, may struggle to produce clear attributions in deep, long-context models. This motivates the need for alternative heuristics like the one proposed in this work.

2.2 Perturbation and Backpropagation-Based Explainability

Beyond direct attention aggregation, prominent XAI paradigms include perturbation-based and gradient-based methods, each offering distinct approaches to attribution.

Perturbation-based methods assess feature importance by modifying input features or, in Transformer-specific variants like AtMan (Deiseroth et al., 2025), by altering pre-softmax attention weights and measuring the impact on outputs. This family also includes local surrogate models such as LIME (Ribeiro et al., 2016) and LORE (Guidotti et al., 2018), which approximate the model locally using interpretable proxies, and SHAP (Lundberg and Lee, 2017), which leverages game-theoretic principles. While generally model-agnostic, these methods require numerous forward passes, making them computationally expensive. Moreover, surrogate-based explanations may fail to capture the internal representations of the original model.

Gradient-based methods propagate signals from the output back to the input using gradient information. Techniques such as SmoothGrad (Smilkov et al., 2017) and Integrated Gradients (Sundararajan et al., 2017) belong to this family, however their effectiveness can degrade in large Transformers (Achtibat et al., 2024). Other approaches, like gradient-weighted attention (Chefer et al., 2021) (e.g., Grad-CAM (Selvaraju et al., 2019) variants), and rule-based techniques like Layer-wise Relevance Propagation (LRP) (Binder et al., 2016) and AttnLRP (Achtibat et al., 2024), provide more structured attributions but face challenges in handling Transformer-specific non-linearities while maintaining theoretical guarantees like relevance conservation.

Both families are typically post hoc and computationally demanding, often requiring backward passes or repeated evaluations. This limits their scalability for real-time or interactive scenarios, motivating the development of efficient inference-time alternatives like MACS.

3 Methodology: From Full Aggregation to Attention Consistency

This section outlines the motivation and formulation of our proposed method, MACS. We begin by discussing the limitations inherent in full attention aggregation approaches, which motivates our alternative strategy based on measuring attention consistency.

3.1 More is Less: Limitations of Full Aggregation Methods

Aggregation-based methods like Attention Rollout aim to approximate how influence propagates through a Transformer model. These methods operate on the principle that each layer uses self-attention to redistribute influence among token representations. From a graph-theoretic perspective, the model can be viewed as a directed graph, where nodes correspond to token representations and edges represent attention-based interactions. Attention Rollout seeks to estimate the total influence flowing from an initial input token j to a token representation i at a later layer L by aggregating attention patterns across layers.

A common formulation involves recursively multiplying attention matrices $A^{(l)}$ (often adjusted to account for residual connections):

$$\tilde{A}^{(L)} = A^{(L)} A^{(L-1)} \cdots A^{(0)} \quad (1)$$

The value $(\tilde{A}^{(L)})_{ij}$ provides an estimate of the total influence accumulated along all paths from input j to position i at layer L.

However, a criticism of such full aggregation methods is that the resulting attribution map can be "noisy" (Achtibat et al., 2024). These methods may highlight a large number of tokens, making it difficult to identify which ones were truly influential. Recent studies on the dynamics of deep decoder-only Transformers have identified several contributing factors:

Over-squashing (Barbero et al., 2024, 2025): In deep causal models, information propagating over long distances (many layers or across many tokens) can become progressively weaker or "squashed". Aggregating contributions from all paths, as Rollout does, means accumulating potentially numerous weak, squashed signals.

Over-mixing / Representational Collapse (Barbero et al., 2024, 2025): Through successive layers of transformation, token representations can become less distinct and more similar to each other. This may make attention patterns less discriminative, and aggregating them might lead to a blurred or averaged-out view of influence.

Softmax Dispersion (Veličković et al., 2024): The softmax attention mechanism inherently struggles to maintain sharp focus as the number of attended items grows, with attention weights dispersing towards uniform. This affects methods relying on aggregating these weights.

Because Attention Rollout structurally sums influence over all possible paths via matrix multiplication, it is susceptible to accumulating noise from these weak (squashed) or potentially indistinct (mixed) signals. This can result in dense or diffuse attribution maps where identifying the most critical input tokens for a specific output is challenging. This motivates our exploration of a more focused approach.

3.2 Less is More: MACS - Measuring Attention Consistency Across Layers

To provide an alternative perspective that might yield clearer attribution while remaining computationally efficient for real-time diagnostic purposes, we propose **Multi-Layer Attention Consistency Score (MACS)**.

Instead of summing contributions over all forward paths like Rollout, MACS adopts a different heuristic that focuses on measuring the **consistency of attention links** directed backward from

the generated token's query to each input token's key, across all layers.

The core idea is that input tokens maintaining a sustained, strong attention connection from the output query across the network's depth are likely key contributors. This provides a distinct measure of contribution based on the consistency of focused attention rather than total aggregated flow. We hypothesize that emphasizing consistent, strong links can yield clearer and potentially sparser attribution maps by filtering out weak or inconsistent signals.

3.2.1 Setup and Generation Process

Given an initial input sequence $X = \{x_1, x_2, \ldots, x_N\}$, where x_i represents the embedding of the i-th input token, a decoder-only Transformer generates subsequent tokens autoregressively. The token t_k (at sequence position $n = N + k$) is generated based on the conditional probability:

$$t_k \sim P\big(t_k \mid x_1, \ldots, x_N,\ t_1, \ldots, t_{k-2}, t_{k-1}\big) \quad (2)$$

Modern decoders utilize a caching mechanism for efficiency. At generation step k (producing token t_k at position $n = N + k$), only the representation of the most recently generated token t_{k-1} (at position $n - 1 = N + k - 1$)[3] typically issues new queries into the attention mechanism. These queries attend to the keys and values associated with all preceding tokens in the sequence (both original inputs and previously generated tokens).

For MACS, we focus on the attention patterns generated when predicting token t_k (at position n). Let L be the number of layers in the Transformer (indexed $l = 0$ to L). For each layer l and head h, let $a_{uv}^{(l,h)}$ denote the attention weight from the query at position u to the key at position v. The attention vector $\mathbf{a}^{(n-1,l,h)} \in \mathbb{R}^{n-1}$ computed by the query from position $n - 1$ (associated with predicting t_k) is:

$$\mathbf{a}^{(n-1,l,h)} = \mathrm{Attn}^{(l,h)}\Bigg(\underbrace{x_1, \ldots, x_N}_{\text{keys: initial input}},\ \underbrace{t_1, \ldots, t_{k-2}, t_{k-1}}_{\text{keys: generated so far}} \Bigg|\ \text{query: } n-1 \Bigg) \quad (3)$$

where $\mathrm{Attn}^{(l,h)}(\cdot)$ denotes the standard scaled dot-product attention computation for head h at layer l under causal masking.

3.2.2 MACS Calculation Steps

MACS processes these attention vectors layer-by-layer for each generation step k (predicting token t_k at position n) to compute a consistency score $C_i^{(k)}$ [4] for each input token $i \in \{1, \ldots, N\}$.

Step 1: Attention Extraction and Redistribution For each layer l and head h:

- Extract attention to inputs: $\mathbf{a}_I^{(n-1,l,h)} \in \mathbb{R}^N$, where $(\mathbf{a}_I^{(n-1,l,h)})_i = a_{n-1,i}^{(l,h)}$ for $1 \le i \le N$.
- Extract attention to previous outputs: $\mathbf{a}_O^{(n-1,l,h)} \in \mathbb{R}^{k-1}$, where $(\mathbf{a}_O^{(n-1,l,h)})_p = a_{n-1,(N+p)}^{(l,h)}$ for $1 \le p \le k-1$.
- Calculate redistributed attention $\mathbf{a}_R^{(n-1,l,h)} \in \mathbb{R}^N$:

$$(\mathbf{a}_R^{(n-1,l,h)})_i = a_{I,i}^{(n-1,l,h)} + \underbrace{\frac{1}{N}\sum_{p=1}^{k-1} a_{O,p}^{(n-1,l,h)}}_{\text{Avg. Attention to Outputs}} \quad (4)$$

Justification: This step incorporates indirect influence ($t_{k-1} \to t_p \to i$) by uniformly distributing attention from previous outputs back to inputs, aiming to balance attention in long generations where direct attention to distant inputs might decay due to effects like over-squashing.

Step 2: Max-Pooling Across Heads For each layer l, compute the element-wise maximum across heads:

$$(\mathbf{m}_l')_i = \max_{h \in \{1,\ldots,H\}} \Big((\mathbf{a}_R^{(n,l,h)})_i \Big) \quad (5)$$

This yields $\mathbf{m}_l' \in \mathbb{R}^N$.

Justification: This step isolates the strongest attention signal directed toward input token i across all heads at layer l. By focusing on peak signals from potentially specialized heads, it filters out weaker, diffuse attention often encountered in over-mixing scenarios, thereby highlighting the most decisive attention link per layer and mitigating noise from less informative heads.

[3] At the first generation step ($k = 1$, where $n = N + 1$), MACS is bootstrapped using the attention vector from the input's last token x_N, representing the model's attention pattern just before generation begins.

[4] For simplicity, we omit the step index (k) in the layer-wise calculations and denote the token i consistency score as C_i. The index (k) will be reintroduced when presenting scores across multiple steps.

Step 3: Incorporate floor vector Introduce a minimum score offset using hyperparameter α (e.g., 0.8):

$$\mathbf{m}_l = \alpha \mathbf{m}'_l + (1 - \alpha)\mathbf{1}^N \tag{6}$$

Justification: This step ensures that even if $\mathbf{m}'_l$ has near-zero entries (i.e., no head strongly attended to token i at layer l), the score going forward ($\mathbf{m}_l$ is at least $1 - \alpha$. This prevents the subsequent Hadamard product from prematurely zeroing out the contribution score for tokens whose relevance might only emerge in deeper layers.

Step 4: Multi-layer Attention Consistency Score (MACS)

Measure sustained layer-wise attention strength through layer-wise multiplication:

Initialize $\mathbf{c}_0 = \mathbf{m}_0$.

For $l = 1, \ldots, L$:

$$\mathbf{c}_l = \mathbf{m}_l \odot \mathbf{c}_{l-1} \tag{7}$$

The final consistency vector is $\mathbf{c}_L \in \mathbb{R}^N$ where $C_i = (\mathbf{c}_L)_i$

Justification: The Hadamard product directly measures the **consistency** of the processed maximal attention link across layers. For $(\mathbf{c}_L)_i$ to be large, the corresponding $(\mathbf{m}_l)_i$ must be consistently large across most/all layers l. This contrasts with additive aggregation and aims to yield clearer attribution by emphasizing sustained relevance, filtering transient signals that could contribute to noise exacerbated by over-squashing or softmax dispersion.

3.2.3 Final Attribution Scores via Z-Scoring

The raw MACS score vector $\mathbf{c}_L^{(k)} \in \mathbb{R}^N$ at generation step k captures each token's layer-consistent maximal attention strength. To render these values comparable and highlight the most salient tokens at each step, we normalize them directly into Z-scores:

$$z_i^{(k)} = \frac{\left(\mathbf{c}_L^{(k)}\right)_i - \mu^{(k)}}{\sigma^{(k)}}, \tag{8}$$

where $\mu^{(k)}$ and $\sigma^{(k)}$ are the mean and standard deviation across the N input token scores within the vector $\mathbf{c}_L^{(k)}$. Each $z_i^{(k)}$ directly measures how many standard deviations token i's attention consistency deviates from the average, immediately flagging tokens with statistically significant focus during generation.

3.3 Algorithm Overview

The complete MACS algorithm is summarized as follows:

Algorithm 1 MACS (streaming)

```
for each generation step k do
    Extract and redistribute attention
    for each transformer layer l do
        Max-pooling across attention heads
        Add weighted floor vector
        Calculate attention consistency
    end for
    Compute step Z-score z^(k)
    yield z^(k)
end for
```

4 Experiments

In the Experiments section, we address two key research questions. **First**, how does our proposed method compare to alternative approaches in terms of accuracy and reliability? **Second**, what is the impact of our method on the model's inference performance?

We perform all experiments on an NVIDIA A800 (80 GB) GPU, using **Llama 3.1-8B** (Grattafiori et al., 2024) as our primary model. This choice was made because Llama 3.1-8B is a powerful, widely-used, and publicly available LLM whose architecture is characteristic of many current state-of-the-art decoder-only Transformers, ensuring our findings are broadly relevant to a significant class of contemporary LLMs.

We benchmark MACS against four baseline methods representing distinct explainability paradigms:

- **Random**: Assigns random importance scores, serving as a basic sanity check.
- **Attention Rollout**: An attention aggregation method that sums influence over all paths. Chosen as a widely recognized aggregation-based baseline.
- **AttnLRP**: A state-of-the-art gradient-based method adapted from Layer-wise Relevance Propagation for Transformers. Chosen as a strong, more complex gradient-based baseline.
- **AtMan**: A perturbation-based method that manipulates pre-softmax attention. Chosen to represent post hoc perturbation techniques.

The key hyperparameter α for MACS is set to 0.8. For AttnLRP, we use their official *lxt* [5] library, and for AtMan, we adapt their publicly available code[6] to Llama 3.1-8B.

4.1 Question Answering (QA) Task Attribution

4.1.1 Dataset, Metrics and Implementation Details

We evaluate attribution performance on a Question Answering task using a subset of 350 question-context-answer triples from the SQuAD 2.0 dataset (Rajpurkar et al., 2018), where answers are guaranteed to be spans within the context. we use the following prompt:

```
Answer the question based on the following text. Keep your response short and simple. Do not quote the original text.
Question: {question}
Context: {context}
```

This subset has an average context length of 169 tokens (max 512), and we set the maximum generation length to 256 tokens. If multiple answers exist, metrics are averaged.

Our primary evaluation metric for ranking relevant input tokens is the Area Under the Precision-Recall Curve (AUC-PR), which evaluates how well each method ranks the ground-truth answer tokens (answer spans) within the input context. The final AUC-PR reported for a given sample is the **maximum AUC-PR achieved across all generation steps** for that sample. This "best step" approach acknowledges that an attribution method might highlight the answer most clearly at a particular point during the generation of the response.

To assess faithfulness, we adopt the Symmetric Relevance Gain (SRG) (Blücher et al., 2024), which quantifies the difference between the model's performance (e.g., output text similarity to the original or model confidence) when progressively removing the least influential tokens versus removing the most influential tokens. We report SRG based on model perplexity (SRG-PP) and ROUGE-L F1 score (SRG-RL). For SRG-PP, more negative values indicate better faithfulness (as perplexity should ideally increase more with important token removal); for other metrics, higher is better. The SRG for the Random baseline is computed using two independent random perturbation orderings and is expected to be near zero. Confidence intervals (95%) are reported as mean ± half-width. Full details and additional results are in Appendix A and Table 5.

[5]https://github.com/rachtibat/LRP-eXplains-Transformer

[6]https://github.com/Aleph-Alpha-Research/AtMan

4.1.2 Results and Discussions

Method	mAUC-PR ↑	mSRG-PP ↓	mSRG-RL ↑
Random	0.113 ± 0.01	0.003 ± 0.01	-0.006 ± 0.01
Rollout	0.147 ± 0.02	-0.039 ± 0.02	0.082 ± 0.02
AttnLRP	0.565 ± 0.03	**-0.126** ± 0.01	**0.323** ± 0.02
AtMan	0.315 ± 0.03	-0.021 ± 0.01	0.055 ± 0.02
MACS(Ours)	**0.601** ± 0.03	-0.118 ± 0.01	0.315 ± 0.02

Table 2: Performance on the QA task. SRG = Symmetric Relevance Gain; PP = Perplexity; RL = ROUGE-L; higher/negative is better; The SRG for the Random baseline is computed using two independent random perturbation orderings and is expected to be near zero. ± value represents the half-width of the 95% CI. m represents the averaged value across all samples.

Table 2 presents the attribution performance among the tested XAI methods. MACS leads in identifying relevant answer tokens, achieving an mAUC-PR of 0.601.This significantly outperforms Attention Rollout by over 300% (0.147 ± 0.02) and AtMan by over 90% (0.315 ± 0.03).

In terms of faithfulness, MACS demonstrates performance statistically comparable to the more complex AttnLRP method across both mSRG-RL and mSRG-PP. Both substantially outperform Rollout and AtMan.

The poor performance of Attention Rollout across both mAUC-PR and faithfulness metrics suggests its tendency to produce diffuse or misleading attributions, a behavior consistent with the theoretical challenges of signal degradation from over-squashing and softmax dispersion in deep models.

4.1.3 Analysis of "Best Step" Timing

To further understand the nature of the attention consistency captured by MACS, we analyzed when its "best step" (the generation step with highest AUC-PR for identifying the answer span in the context) occurs relative to the actual generation of the answer. Across our 350 SQuAD samples, we found that in **97.68% of cases, MACS achieved its peak AUC-PR towards the answer span before the model began to generate the answer tokens themselves**. An illustrative example is shown in Figure 3, where MACS highlights the answer in the context based on the query from an early generated

token, prior to the answer appearing in the output. This finding suggests that MACS captures anticipatory attention related to information retrieval and comprehension crucial for answer formation, rather than merely reflecting a surface-level similarity during the token-by-token generation of the answer.

4.2 Performance Comparison

Beyond attribution quality, the practical utility of XAI methods hinges on their computational efficiency, especially for real-time applications or analysis of large models and long contexts. In this section, we evaluate the efficiency of MACS against other attribution methods.

4.2.1 Dataset, Metrics, and Implementation Details

We focus on two key aspects: peak VRAM usage and throughput (tokens generated per second). It's important to distinguish how these metrics apply across XAI methods. For real-time capable methods such as MACS, AttnLRP, and Rollout, VRAM and throughput reflect overhead added during the generation process. In contrast, post-hoc methods like AtMan report "Peak VRAM" as the sum of baseline inference memory and the additional memory used for attribution, such as perturbations. Similarly, AtMan's "Throughput" is calculated using the total time, including both inference and perturbation, divided by the number of tokens generated in the original inference, capturing the full explanation cost.

To evaluate computational efficiency under demanding conditions, we sampled 70 CNN/DailyMail (Hermann et al., 2015) examples with input lengths ranging from 173 to 3,936 tokens, paired with a summarization task requiring up to 512 output tokens. This setup imposes substantial and diverse workloads, allowing for a clear assessment of VRAM usage and throughput scaling across different attribution methods as context length increases. The prompt used is as follows:

```
Summarize the following text.
Text: {context}
```

4.2.2 Results and Discussions

As shown in Figure 1a and Table 3, A clear advantage of MACS is its low VRAM footprint, with only a 11% mean increase over baseline inference; this is much more efficient than AttnLRP, which shows a 33% increase. Attention Rollout's VRAM consumption is substantially higher, increasing by 62% over baseline for the samples it could complete, its need to manipulate entire attention matrices leads to rapidly escalating VRAM costs and Out-of-Memory errors, especially with longer inputs. AtMan imposes negligible impact on VRAM usage (only a 1% increase); however, this comes at the cost of greatly reduced throughput.

Regarding inference speed (Figure 1b, Table 3), MACS has the least impact on inference time, showing only a 23% mean decrease in throughput. Its performance degrades gently with increasing context length and is nearly identical to the baseline for inputs under 500 tokens. This is noticeably better than AttnLRP, which sees a 53% decrease in throughput. Attention Rollout slows inference considerably (a 59% throughput decrease) due to its full matrix aggregations. AtMan, being post hoc and requiring $1 + length_{input}$ perturbations, becomes prohibitively slow (a 78% decrease) for long contexts, rendering it unsuitable for real-time applications.

Method	mPeak VRAM (MB) ↓	mThroughput (Tokens/Sec) ↑
Pure Inference	16088	7.46
Rollout	42649 (+62%)	3.04 (-59%)
AtMan	**16253 (+1%)**	1.66 (-78%)
AttnLRP	24112 (+33%)	3.49 (- 53%)
MACS (Ours)	17998 (+11%)	**5.69 (-23%)**

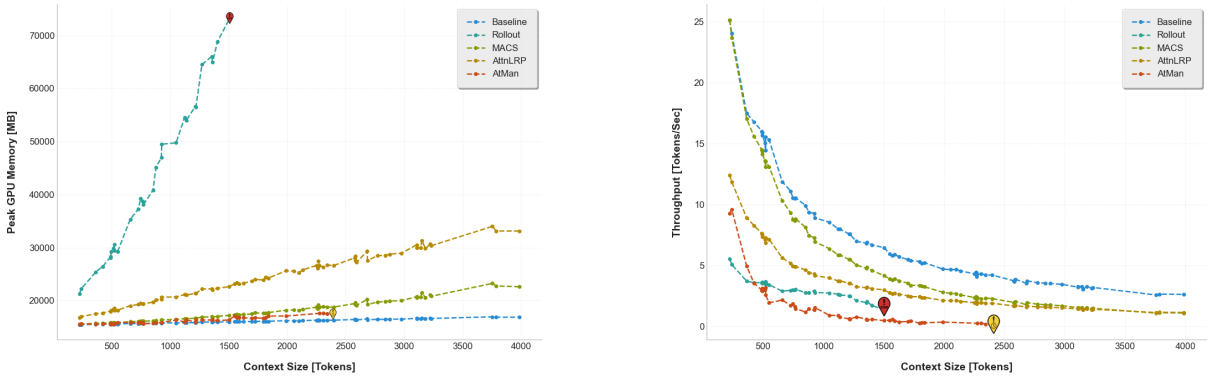
Table 3: Efficiency comparison. For AtMan and Rollout, reported mean values are based on the subset of samples completed before encountering errors (e.g., Out-of-Memory or excessive processing time); means for other methods are over all 70 samples.↑ represents increase compared to the pure inference baseline

5 Ablation Study

To validate the design of MACS and understand the contribution of its distinct components, we conduct a series of ablation studies. We systematically remove or modify key elements of the MACS algorithm and evaluate the impact on performance. All ablations are performed on the SQuAD 2.0 subset using AUC-PR as the primary metric, as described in Section 4.1.1. The full MACS method (detailed in Section 3.2.2) serves as the baseline for comparison.

5.1 Ablated Variants

We evaluate the following variations of MACS:



(a) Peak VRAM vs Context Size

(b) Throughput vs Context Size

Figure 1: Peak VRAM usage (a) and Throughput (b) against context size for different XAI methods. denotes Out-of-Memory (OOM) errors. indicates prohibitive inference times (>10 mins or near-zero throughput). Baseline is the inference without any XAI method. MACS maintains high efficiency across context lengths.

1. **w/o Redistribution**: The redistribution step is removed. The calculation uses only the direct attention to inputs $\mathbf{a}_I^{(n-1,l,h)}$ instead of $\mathbf{a}_R^{(n-1,l,h)}$. (Section 3.2.2 Step 1)

2. **Pooling**: Max-pooling across heads is replaced with mean-pooling/min-pooling across heads. (Section 3.2.2 Step 2)

3. **The Floor vector** : Evaluate performance using different values for the floor hyperparameter in the $\alpha(1-\alpha)\mathbf{1}^N$. (Section 3.2.2 Step 3)

5.1.1 Discussion

The results in Table 4 validate the core design choices of MACS.

Max-pooling is critical, replacing it with mean-pooling or min-pooling significantly degrades mAUC-PR (from 0.601 to 0.570 and 0.443, respectively). This strongly supports our hypothesis that identifying and propagating the strongest consistent attention link via max-pooling is key to MACS's effectiveness, aligning with the idea that salient contributions are often marked by peak attention signals rather than averaged ones.

The weighted floor vector is also essential, its removal drops mAUC-PR to 0.546, confirming its key role in preventing the Hadamard product from prematurely nullifying token scores that only gain relevance in deeper layers. Interestingly, while the baseline is essential, the exact value of α has a limited impact, implying the main benefit lies in setting a non-zero floor rather than fine-tuning the weight.

Removing the attention redistribution step had a negligible impact on mAUC-PR in this experimental setup, possibly due to the moderate generation lengths (max 256 tokens) where its benefits for capturing distant indirect influences may be less pronounced.

Method Variant	mAUC-PR
Full MACS (Baseline)	**0.601**
w/o Redistribution	0.600
Mean-Pooling (vs Max)	0.570
Min-Pooling (vs Max)	0.443
w/o floor vector ($\alpha = 1$)	0.546
Floor Vector ($\alpha = 0.2$)	0.599
Floor Vector ($\alpha = 0.5$)	0.599

Table 4: Ablation study results for MACS

6 Conclusion

We have introduced Multi-Layer Attention Consistency Score (MACS), a novel heuristic for quantifying input token importance in decoder-only models. Designed to be **lightweight, computationally efficient and usable without model modifications**. Our empirical evaluations reveal that MACS frequently matches more sophisticated attribution techniques. These findings also support our hypothesis that measuring the consistent strength of maximal attention links across layers offers a clear and effective way to identify salient input contributions. Consequently, MACS serves as a practical tool for obtaining rapid insights, thus contributing an efficient way toward enhancing interpretability in LLMs.

7 Limitations

While MACS demonstrates compelling advantages in efficiency and offers strong empirical performance, its unique approach warrants careful consideration. MACS quantifies input contribution by measuring the consistency of maximal attention links across layers. This provides a distinct perspective on explainability compared to methods focused on input sensitivity (e.g., gradient-based) or marginal impact (e.g., perturbation-based).

The precise interpretation of this "attention consistency" score, and how it relates to or complements the insights from other established XAI paradigms, is an important area for consideration. While our results show it effectively identifies salient inputs, understanding the full implications of this consistency measure versus, for example, total information flow, requires careful interpretation based on its specific mechanism.

8 Future Work

This novel perspective itself opens avenues for future research into different aspects of model reasoning and what various forms of "contribution" signify (a more detailed discussion on interpreting attention consistency is provided in Section 4.1.3 and Appendix B.0.3). Other factors, such as the reliance on the underlying model's attention quality and the information selection via max-pooling, also define the scope of the current method.

Future work should therefore involve deeper theoretical analysis of this attention consistency measure, further exploring its connections to model behavior and cognitive processes across diverse tasks and Transformer architectures. Investigating its role as both a standalone diagnostic and as a component in hybrid XAI approaches also remain a promising direction.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. *Preprint*, arXiv:2005.00928.

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. 2024. Attnlrp: Attention-aware layer-wise relevance propagation for transformers. *Preprint*, arXiv:2402.05602.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João G. M. Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. 2024. Transformers need glasses! information over-squashing in language tasks. *Preprint*, arXiv:2406.04267.

Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. 2025. Why do llms attend to the first token? *Preprint*, arXiv:2504.02732.

Alexander Binder, Grégoire Montavon, Sebastian Bach, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. *Preprint*, arXiv:1604.00825.

Stefan Blücher, Johanna Vielhaben, and Nils Strodthoff. 2024. Decoupling pixel flipping and occlusion strategy for consistent xai benchmarks. *Preprint*, arXiv:2401.06654.

Chengkun Cai, Haoliang Liu, Xu Zhao, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, John Lee, Jenq-Neng Hwang, and Lei Li. 2025a. Bayesian Optimization for Controlled Image Editing via LLMs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chengkun Cai, Xu Zhao, Haoliang Liu, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, Jenq-Neng Hwang, and Lei Li. 2025b. The Role of Deductive and Inductive Reasoning in Large Language Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hila Chefer, Shir Gur, and Lior Wolf. 2021. Transformer interpretability beyond attention visualization. *Preprint*, arXiv:2012.09838.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. 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. Association for Computational Linguistics.

Björn Deiseroth, Mayukh Deb, Samuel Weinbach, Manuel Brack, Patrick Schramowski, and Kristian Kersting. 2025. Atman: Understanding transformer predictions through memory efficient attention manipulation. *Preprint*, arXiv:2301.08110.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh

Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems. *Preprint*, arXiv:1805.10820.

Yangfan He, Sida Li, Jianhui Wang, Kun Li, Xinyuan Song, Xinhang Yuan, Keqin Li, Kuan Lu, Menghao Huo, Jingqun Tang, and 1 others. 2025. Enhancing low-cost video editing with lightweight adaptors and temporal-aware inversion. *arXiv preprint arXiv:2501.04606*.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NIPS)*.

Sen Jia and Lei Li. 2024. Adaptive masking enhances visual grounding. *arXiv preprint arXiv:2410.03161*.

Can Jin, Tianjin Huang, Yihua Zhang, Mykola Pechenizkiy, Sijia Liu, Shiwei Liu, and Tianlong Chen. 2025. Visual prompting upgrades neural network sparsification: A data-model perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4111–4119.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981.

Lei Li, Sen Jia, Jianhao Wang, Zhaochong An, Jiaang Li, Jenq-Neng Hwang, and Serge Belongie. 2025a. Chatmotion: A multimodal multi-agent for human motion analysis. *arXiv preprint arXiv:2502.18180*.

Lei Li, Sen Jia, Jianhao Wang, Zhongyu Jiang, Feng Zhou, Ju Dai, Tianfang Zhang, Zongkai Wu, and Jenq-Neng Hwang. 2025b. Human motion instruction tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17582–17591.

Pei Liu, Haipeng Liu, Xingyu Liu, Yiqun Li, Junlan Chen, Yangfan He, and Jun Ma. 2025. Scene-aware explainable multimodal trajectory prediction. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10786–10792. IEEE.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Preprint*, arXiv:1705.07874.

Perhols. n.d. Copyright Summary Diagram. Illustration licensed under GNU FDL 1.2 or later; CC BY-SA 3.0; CC BY 2.5. Accessed via Wikimedia Commons.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Preprint*, arXiv:1602.04938.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359.

Jingzhe Shi, Qinwei Ma, Hongyi Liu, Hang Zhao, Jeng-Neng Hwang, and Lei Li. 2025. Explaining context length scaling and bounds for language models. *arXiv preprint arXiv:2502.01481*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *Preprint*, arXiv:1706.03825.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *Preprint*, arXiv:1703.01365.

Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. 2024. softmax is not enough (for sharp out-of-distribution). *Preprint*, arXiv:2410.01104.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Ziyu Yao, Xuxin Cheng, Zhiqi Huang, and Lei Li. 2025. CountLLM: Towards Generalizable Repetitive Action Counting via Large Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *Preprint*, arXiv:1901.11359.

Tianfang Zhang, Lei Li, Siying Cao, Tian Pu, and Zhenming Peng. 2023. Attention-guided pyramid context networks for detecting infrared small target under complex background. *IEEE Transactions on Aerospace and Electronic Systems*, 59(4):4250–4261.

A Experiments details

This appendix provides further details on the datasets, metrics, and implementation specifics for the experiments presented in the main section.

The QA experiments utilized a subset of 350 question-context-answer triples from the SQuAD 2.0 dataset, selected for instances where the answer span is present in the context. For evaluation, the *original input sequence* refers to the tokenized context part of the prompt, excluding any special tokens or instructional prompt text. The ground truth ("golden answer") consists of a list of all possible tokenized answers. If multiple correct answer spans were provided in SQuAD 2.0 for a given question, our metrics were computed for each and then averaged.

A.1 AUC-PR Calculation

The Area Under the Precision-Recall Curve (AUC-PR) is our primary metric for evaluating how well an attribution method ranks the tokens from the ground-truth answer span(s) higher than non-answer tokens within the input context. For each generation step k when predicting token t_k, an attribution method produces a score $s_i^{(k)}$ for every token x_i in the input context (length N_{ctx}). Given:

- A tokenized ground-truth answer span $A = \{a_1, a_2, \ldots, a_M\}$.
- A list of $(token\_pos\_id_i, s_i^{(k)})$ pairs for the input context tokens.

We construct two vectors for the average precision score (APS) function:

- y_{true}: A binary vector of length N_{ctx}. For each input context token x_i at position j:

$$(y_{true})_j = \begin{cases} 1 & \text{if } x_i \in A \\ 0 & \text{otherwise} \end{cases}$$

- y_{score}: A vector of the attribution scores for the corresponding input context tokens: $(y_{score})_j = s_j^{(k)}$ (score for token x_i at position j).

The AUC-PR for a single answer span A at a specific generation step k is then:

$$\text{AUC-PR}(A, k) = \mathsf{APS}(\mathbf{y}_{\text{true}}, \mathbf{y}_{\text{score}})$$

This score reflects the ability of an attribution score to correctly rank the answer tokens highly. If $any(y_{true})$ is false, the AUC-PR is treated as 0.0 for that specific case.

As a XAI method produces attribution scores at each generation step, and SQuAD 2.0 can have multiple reference answers, the final reported AUC-PR for a given sample is calculated as follows:

- For each generation step during the model's response, the AUC-PR is calculated against each tokenized ground-truth answer. The mean of these AUC-PRs is taken as the step's score.

$$\text{MeanStepAUC}^{(k)} = \frac{1}{|A|} \sum_{A_j \in A} \text{AUC-PR}(A_j, k)$$

- The reported AUC-PR for the entire sample is the **maximum** MeanStepAUC observed across all generation steps for that sample:

$$\text{SampleAUC-PR} = \max_k \left(\text{MeanStepAUC}^{(k)} \right)$$

A.2 Faithfulness Score Calculation

To assess the faithfulness of the attribution scores generated by MACS and baseline methods, we adapt the "pixel flipping" (or feature removal) paradigm, commonly used in XAI, to the text domain using attention masking. This section details the calculation of the **Most Influential First (MIF)**, **Least Influential First (LIF)**, and **Symmetric Relevance Gain (SRG)** metrics based on this approach.

A.2.1 Perturbation Strategy: Attention Masking

For each input sample and a given attribution method, we first obtain the attribution scores (Z-scores for MACS) for all N_{ctx} tokens in the input context. These scores are then used to rank the input tokens from most to least influential.

Perturbations are performed by masking tokens in the attention mechanism. Instead of removing tokens from the input sequence (which would change token positions and require re-tokenization), we modify the attention mask during the model's generation process. If a token x_j is selected for "removal" based on its attribution score[7], the attention mask is altered such that no query position can attend to key x_j. This effectively makes the token invisible to the attention mechanism for subsequent generation steps.

[7]For methods like MACS, AttnLRP, and Rollout that produce scores at each generation step, the attribution score used for ranking tokens for perturbation is the overall score, typically an average across all generation steps. For post hoc methods like AtMan that produce a single attribution map for the entire generated sequence, that single map is used directly.

We perform these perturbations at predefined fractions $F = \{f_0, f_1, \ldots, f_M\}$ of the total number of input context tokens, where $f_0 = 0\%$ (baseline, no tokens masked) and $f_M = 20\%$ in our experiments. The specific fractions used are $F = \{0.00, 0.01, 0.05, 0.10, 0.15, 0.20\}$.

A.2.2 Performance Curves

For each attribution method ϕ and each sample:

1. **MIF Ordering (π^ϕ)**: Input tokens are sorted in descending order of their attribution scores (most influential first).
2. **LIF Ordering ($(\pi^\phi)^r$**: Input tokens are sorted in ascending order of their attribution scores (least influential first).
3. **Random Ordering (π^{RND})**: Input tokens are sorted in a random order.

For each ordering (π^ϕ, $(\pi^\phi)^r$, π^{RND}) and each perturbation fraction $s \in F$, we generate a new output sequence by masking the corresponding fraction s of tokens. We then measure a model performance metric $v(s, \pi)$ (e.g., logit of the original next token, perplexity of the generated sequence, ROUGE-L F1, BLEU, or Semantic Similarity compared to the original unperturbed generation).

These performance values are then normalized relative to the baseline performance at $s = f_0 = 0\%$ perturbation (i.e., $v(f_0, \pi)$):

$$c(s, \pi) = \frac{v(s, \pi)}{v(f_0, \pi)}$$

This results in three normalized performance curves for each sample and base metric:

- $c_{MIF}(s)$: Normalized performance curve when removing tokens by π^ϕ.
- $c_{LIF}(s)$: Normalized performance curve when removing tokens by $(\pi^\phi)^r$.
- $c_{RND}(s)$: Normalized performance curve when removing tokens by π^{RND}.

A.2.3 AUC Calculation

The Area Under the normalized Perturbation Curve (AUC) is calculated using the trapezoidal rule, normalized by the maximum perturbation fraction $f_M = 0.20$:

$$\begin{aligned}\mathrm{AUC}[\pi] &= \frac{1}{f_M} \int_0^{f_M} c(s, \pi)\, ds \\ &\approx \frac{1}{F[-1]} \sum_{j=1}^{M} \left[\frac{c(F[j], \pi) + c(F[j-1], \pi)}{2} \cdot \Delta F_j \right]\end{aligned}$$

where $\Delta F_j := F[j] - F[j-1]$ and $F[j]$ is the j-th fraction in our fraction array F. This yields $AUC_{MIF}[\phi]$, $AUC_{LIF}[\phi]$, and $AUC_{RND}[\phi]$ for each sample.

A.2.4 Faithfulness Metrics

The final faithfulness metrics reported are averages for all samples.

- **mMIF Score**: The average $AUC_{MIF}[\phi]$
- **mLIF Score**: The average $AUC_{LIF}[\phi]$
- **mSRG Score**: $SRG[\phi] = AUC_{LIF}[\phi] - AUC_{MIF}[\phi]$

The SRG for a random baseline (SRG_{RND}) is computed using two independent random perturbation orderings and is expected to be close to zero.

We show the comprehensive comparison across all metrics and attribution methods in Table 5

B Exploring MACS's Applicability to Multi-Modal Models (Visual Question Answering)

The core mechanism of MACS, which measures the consistency of maximal attention links across layers, is not inherently limited to text-only decoder models. It primarily requires access to layer-wise attention weights. This suggests potential applicability to other Transformer-based architectures, including multi-modal models.

B.0.1 Motivation and Approach

To explore this potential, we conducted a preliminary study applying MACS to a Visual Question Answering (VQA) task. A key advantage of MACS in this context is its out-of-the-box integration capability. Unlike perturbation or gradient-based XAI techniques that often require significant adaptation or architectural modifications to handle multi-modal inputs (e.g., separate handling of text and image feature perturbations, or complex gradient paths through vision and language encoders), MACS can be applied by analyzing the attention patterns within the language processing or cross-modal attention layers of the multi-modal model without changes to the underlying network.

Given the difficulty of adapting other XAI methods for a direct, rigorous comparison in this multi-modal setup without extensive engineering, our primary goal here is to demonstrate MACS's straightforward applicability and to qualitatively observe its behavior.

Figure 2: During generation, MACS dynamically highlights in the image the regions corresponding to the text as it's being produced. The generated text is: "The image shows a young child and a white dog, sitting together in a grassy outdoor setting. The child is wearing a red cap, a red and gray jacket, and has a backpack on." Source: Perhols (n.d.)

B.0.2 Experimental Setup

Similar to the QA task, we selected 200 annotated images from the Open Images Dataset V4 (Kuznetsova et al., 2020), spanning 10 everyday object categories[8], with 20 samples per category. To simplify the setup and reduce model load, we only included images containing exactly one annotated object. Each image was paired with the following prompt:

```
What is in the image?
```

The goal is to evaluate if MACS effectively highlights image regions relevant to the model's predicted answer, specifically the object in the image. We report AUR PC scores calculated over the labeled masked regions. The QVA task uses the **Qwen 2.5-VL-7B** model (Bai et al., 2025).

B.0.3 Preliminary Observations

Our application of MACS to the Qwen-VL model for this simplified VQA task yielded a mean mAUC-PR of 0.602 (averaged over 200 samples, taking the best step per sample). This score indicates that MACS is generally able to rank pixels belonging to the target object significantly higher than background pixels, demonstrating a promising level of attribution accuracy for identifying relevant visual regions.

[8]Orange, Apple, Dog, Cat, Book, Laptop, Guitar, Piano, Bus, Airplane

Qualitatively, we observe a distinct pattern in MACS's attributions on these VQA samples. As illustrated in Figure 2.

MACS often produces sparse or "peaky" heatmaps. Instead of highlighting the entire extent of the ground-truth object, it tends to concentrate high attribution scores on a small, often well-defined, sub-region within the target object. We hypothesize that MACS produces concentrated attribution patterns due to its core mechanism. By measuring the consistency of maximal attention using layer-wise max-pooling and a Hadamard product, MACS highlights input features that receive consistently strong attention across layers. This results in sparse attributions focused on the most discriminative parts, rather than spreading importance evenly. While this "peaky" attribution leads to good ranking performance,not all ground truth object pixels receive high scores, a characteristic important for interpreting MACS outputs visually.

Q: In what country is Normandy located?
G: Normandy

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Q: In what country is Normandy located?
G: Normandy is

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy , a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Q: In what country is Normandy located?
G: Normandy is located

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy , a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Q: In what country is Normandy located?
G: Normandy is located in

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Figure 3: MACS demonstrating anticipatory attention on a QA example. The heatmap shows MACS attribution scores on the input context prior to the model generating the answer "France". High consistency scores (darker red) on "France" in the context indicate MACS identifies the answer span before its generation by the model (Q: Question, G: Generated tokens).

Output Type	Metric	MACS	AttnLRP	Rollout	AtMan	Random
Main Task	AUC-PR	0.601 ± 0.033	0.565 ± 0.030	0.147 ± 0.021	0.315 ± 0.032	0.113 ± 0.012
Mean Logits	mMIF (AUC)	0.919 ± 0.006	0.915 ± 0.006	0.967 ± 0.005	0.945 ± 0.006	–
	mLIF (AUC)	1.003 ± 0.002	1.005 ± 0.002	0.995 ± 0.003	0.964 ± 0.005	–
	mSRG	0.084 ± 0.006	0.090 ± 0.006	0.028 ± 0.005	0.019 ± 0.006	-0.003 ± 0.003
Perplexity	mMIF (AUC)	1.114 ± 0.013	1.120 ± 0.014	1.040 ± 0.010	1.046 ± 0.010	–
	mLIF (AUC)	0.996 ± 0.004	0.994 ± 0.004	1.002 ± 0.005	1.025 ± 0.008	–
	mSRG	-0.118 ± 0.013	-0.126 ± 0.013	-0.039 ± 0.009	-0.021 ± 0.010	0.003 ± 0.005
ROUGE-L F1	mMIF (AUC)	0.619 ± 0.021	0.609 ± 0.021	0.823 ± 0.019	0.783 ± 0.019	–
	mLIF (AUC)	0.933 ± 0.013	0.932 ± 0.013	0.906 ± 0.014	0.838 ± 0.018	–
	mSRG	0.315 ± 0.021	0.323 ± 0.022	0.082 ± 0.019	0.055 ± 0.020	-0.006 ± 0.011
BLEU	mMIF (AUC)	0.571 ± 0.020	0.558 ± 0.019	0.780 ± 0.022	0.718 ± 0.021	–
	mLIF (AUC)	0.903 ± 0.017	0.902 ± 0.016	0.869 ± 0.019	0.784 ± 0.021	–
	mSRG	0.332 ± 0.021	0.344 ± 0.021	0.089 ± 0.021	0.066 ± 0.022	-0.011 ± 0.012
Semantic Sim.	mMIF (AUC)	0.758 ± 0.020	0.752 ± 0.020	0.892 ± 0.014	0.870 ± 0.015	–
	mLIF (AUC)	0.962 ± 0.009	0.962 ± 0.009	0.944 ± 0.011	0.906 ± 0.013	–
	mSRG	0.205 ± 0.019	0.210 ± 0.019	0.051 ± 0.014	0.036 ± 0.015	-0.004 ± 0.009

Table 5: Comprehensive Comparison of Attribution Methods. All values are averaged across all samples.