Attention Sink-driven Hallucination in Multimodal LLMs: A Purple Teaming Study

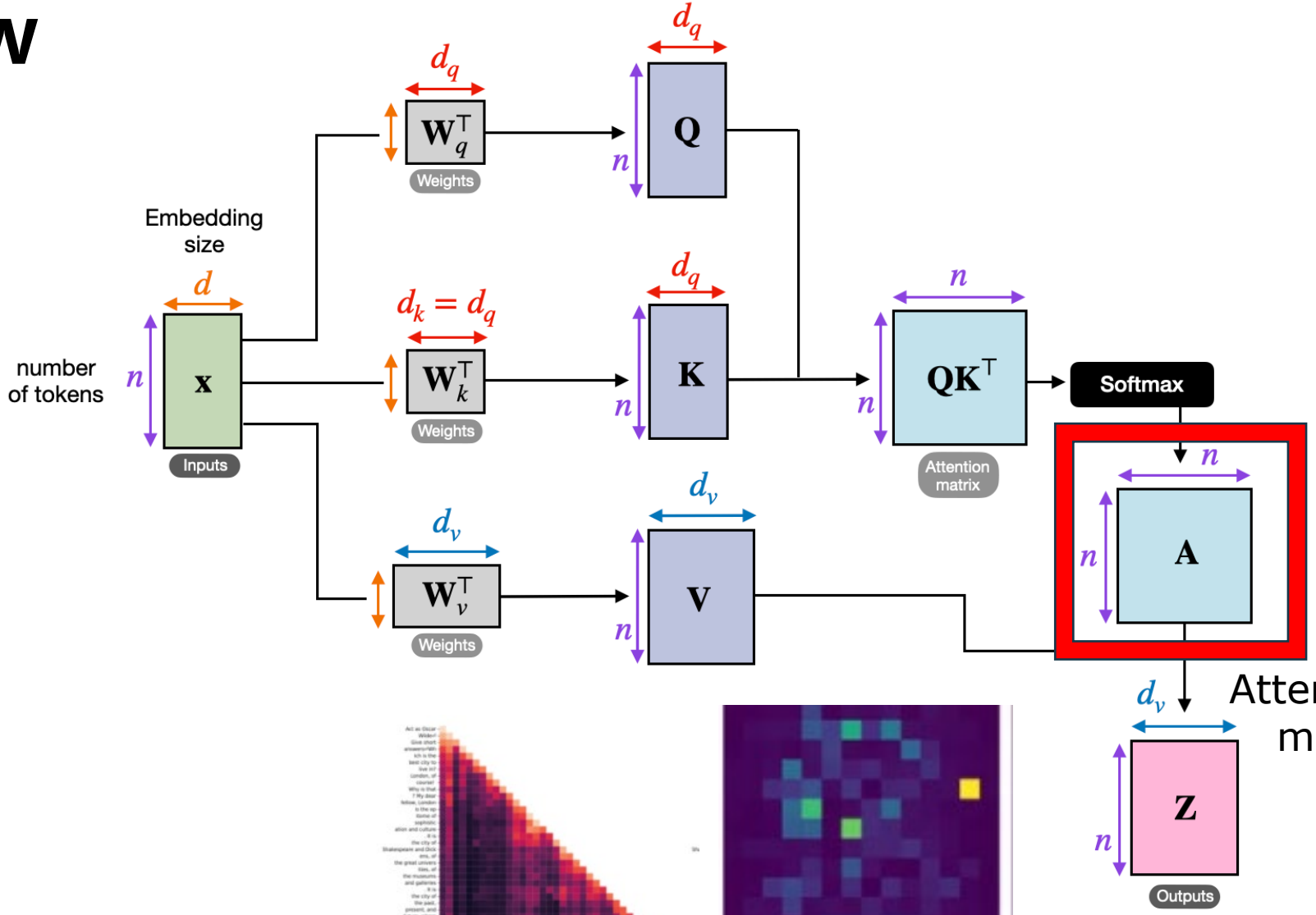
Byeongseo Min

Electrical Engineering (**EE**)

POSTECH

Quick Review

- Attention scores:
  $$QK^T$$
- Attention map:
  $$\mathrm{A} = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})$$
- Attention map represents the pairwise dependency or relevance between tokens



Examples of attention map

Quick Review

- There was a phenomenon called the *attention sink* in the Transformer
- Attention sinks receive excessively large attention scores
  - It is not a major issue when truly important tokens become sinks
  - Non-semantic tokens becoming sinks: can lead to unintended model behavior

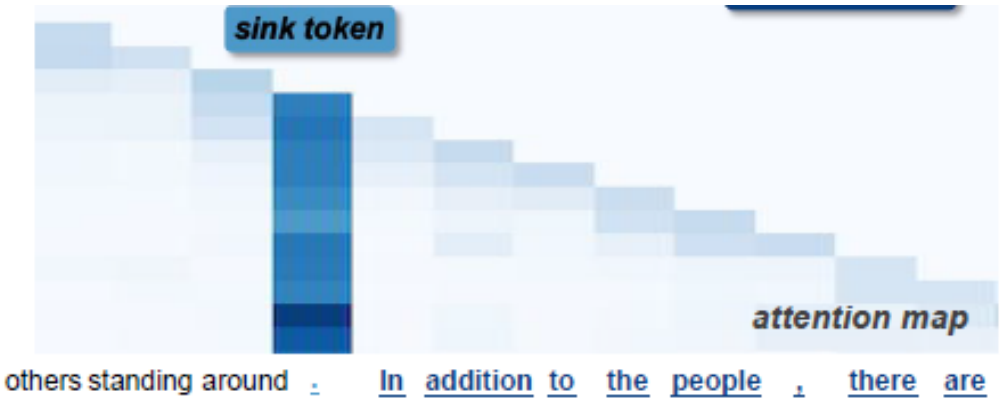


Figure 1: An illustration of the attention sink phenomenon in MLLM responses. The sink token receives high attention scores in a columnar pattern. The hallucinated responses are marked bold with **indigo tokens**.

Quick Review

- In HW1, we conducted research on attention sink-aware adversarial attacks for ViT
- The motivation was to design a transferable attack by exploiting attention sinks, a common vulnerability of Transformer models

Quick Review

- However, there exists a hallucination attack on MLLMs that is attention-sink-aware!

***Mirage in the Eyes*: Hallucination Attack on Multi-modal Large Language Models with *Only* Attention Sink**

Yining Wang[1], Mi Zhang[1,*], Junjie Sun[1], Chenyue Wang[1], Min Yang[1,*],
Hui Xue[2], Jialing Tao[2], Ranjie Duan[2], Jiexi Liu[2]
[1] *Fudan University*, [2] *Alibaba Group*
{ynwang22@m., mi_zhang@, jjsun22@m., wangcy23@m., m_yang@}fudan.edu.cn
{hui.xueh, jialing.tjl, ranjie.drj, liujiexi.ljx}@alibaba-inc.com

[1] Wang, Yining, et al. "Mirage in the eyes: Hallucination attack on multi-modal large language models with only attention sink." *USENIX Security 25*. 2025.

Contents

1. Implementation of the Mirage in the Eyes

POSTECH

POSTECH GSAI
GRADUATE SCHOOL OF ARTIFICIAL INTELLIGENCE

POSTECH COMPUTER SCIENCE & ENGINEERING

Mirage in the Eyes

- Hallucination in MLLMs

Human: Please describe this image in detail.

Assistant: The image depicts a lively scene in a park, where two men are competing to catch a frisbee in mid-air...There are several other people in the park, some sitting on the grass and others standing around. **In addition to the people, there are several balls scattered throughout the scene, possibly belonging to the park-goers. A car can also be seen parked in the background, adding to the overall atmosphere of the park.**

hallucination!

Mirage in the Eyes

- Motivation
  - Decoding-stage attention-sinks cause hallucination
  - Induce hallucination attacks by stealthily creating attention sinks during decoding
- Add adversarial perturbation to the image

visual input

Hallucination Attack

Human: Please describe this image in detail.

Step:1, *Sink token: items*

Assistant: The image features a man sitting at a dining table with a bowl of cereal in front of him. The table is fill ed with various food ***items***, including **a cup** of cereal...

Step:3, *Sink token: is*

Assistant: The image features a man sitting at a dining table with a bowl of cereal in front of him. He ***is*** **holding a spoon**, ready to enjoy his breakfast. The table is...

Step:10, *Sink token: hand*

Assistant: The image features a man sitting at a dining table, **holding a bowl in his** ***hand***. The table is set with various items, including a cereal box, a carton of milk...

Step:18, *Sink token: .*

Assistant: The image features a **woman** sitting at a din -ing table with a bowl of cereal in front of her. **She is holding a box of cereal, possibly a box of Cheer...**

Optimizing Adversarial Perturbations

- Determine adversarial sink token **idx**

$$s_i^{(l)} = \mathrm{Sim}(h_i^{(l)}, \overline{h}_g^{(l)}),\ 1 \le l < L,\ i > M+N-1 \quad , \overline{h}_g^{(l)} = \mathrm{Mean}(\{h_i^{(l)}\}_{i=0}^{M+N-1}).$$

$$\mathrm{idx} = \arg\max_i s_i^{(l)}$$

  - $h_i^{(\ell)}$:hidden state of the *i-th generated token*
  - $h_g^{(\ell)}$:global input representation
    - mean of hidden states from *image + prompt tokens*
  - Select the generated token with the highest cosine similarity to the global input embedding

Optimizing Adversarial Perturbations

- Attention loss

$$\mathcal{L}_{\text{attn}}(\mathbf{x}^v, \mathbf{x}^t) = CE(A'^{(l)}, \text{idx})$$

  - $A'^{(\ell)}$:attention of tokens after the sink
  - idx: selected sink token
  - Encourages subsequent tokens to attend to the sink token
  - Induces a **columnar attention pattern**
  - Goal
    - Force future tokens to rely on the sink token

Optimizing Adversarial Perturbations

- Embedding loss

$$\mathcal{L}_{\text{emb}}(\mathbf{x}^v, \mathbf{x}^t) = \max(0, \sigma - \text{Sim}(h_{\text{idx}}^{(l)}, \overline{h}_g^{(l)}))$$

  - $h_{\text{idx}}^{(\ell)}$:sink token hidden state
  - $h_g^{(\ell)}$:global input representation
  - σ: similarity threshold (hyperparameter)
  - Increases similarity between sink and input
  - Makes the sink token **input-like**
  - Goal
    - Make the sink a meaningful (but imperfect) summary

Optimizing Adversarial Perturbations

- Adversarial Objective

$$\begin{aligned} \min\ & \mathcal{L}_{\text{attn}}(\tilde{\mathbf{x}}^v, \mathbf{x}^t) + \alpha \mathcal{L}_{\text{emb}}(\tilde{\mathbf{x}}^v, \mathbf{x}^t) \\ \text{s.t.},\ & \tilde{\mathbf{x}}^v = \mathbf{x}^v + \delta,\ ||\delta||_p < \varepsilon \end{aligned} \tag{10}$$

  - x^v :original image
  - $\tilde{x}^v$:adversarial image
  - δ: perturbation
  - ϵ: attack budget
  - α: weighting factor
  - PGD-style
  - **Optimize the image to create and amplify a misleading attention sink in decoding stage**

White-box Attack

Table 2: Results of GPT-4 assisted hallucination evaluation for the image captioning task on white-box models. All of the MLLM responses are generated with *beam search* decoding. We report six aspects of evaluation, including the number of sentences per image (**SPI**), the number of words per image (**WPI**), the number of hallucinated sentences per image (**HSPI**), the number of hallucinated words per image (**HWPI**), the ratio of hallucinated sentences (**HSR**), and the ratio of hallucinated words (**HWR**). A larger HSPI, HWPI, HSR, and HWR indicate a higher level of hallucination in MLLM responses. The best results are marked in bold, and the number in brackets indicates the hallucination improvement compared to the clean image.

Target Model	Input	SPI	WPI	HSPI	HWPI	HSR(%)	HWR(%)
InstructBLIP	clean image	4.54	75.64	2.83	48.05	62.91%	64.93%
	ε=2/255	4.60	80.19	2.97 (+0.14)	55.14 (+7.09)	64.92% (+2.01%)	68.23% (+3.30%)
	ε=5/255	4.47	80.48	3.04 (+0.21)	54.90 (+6.85)	**68.41%** (+5.50%)	**70.84%** (+5.91%)
	ε=8/255	4.41	79.71	2.89 (+0.06)	52.91 (+4.86)	66.79% (+3.88%)	69.45% (+4.52%)
LLaVA-1.5	clean image	4.60	116.24	2.68	79.08	59.62%	71.68%
	ε=2/255	4.64	96.60	2.76 (+0.08)	62.97 (-16.11)	60.26% (+0.64%)	68.17% (-3.51%)
	ε=5/255	4.49	108.03	2.67 (-0.01)	74.85 (-4.23)	62.36% (+2.74%)	**75.74%** (+4.06%)
	ε=8/255	4.53	103.58	2.92 (+0.24)	75.45 (-3.63)	**65.07%** (+5.45%)	75.08% (+3.40%)
MiniGPT-4	clean image	3.98	60.56	2.31	37.34	58.13%	62.77%
	ε=2/255	4.10	59.20	2.49 (+0.18)	37.97 (+0.63)	61.42% (+3.29%)	65.01% (+2.24%)
	ε=5/255	3.97	66.27	2.41 (+0.10)	43.48 (+6.14)	61.02% (+2.89%)	67.09% (+4.32%)
	ε=8/255	4.00	64.51	2.55 (+0.24)	40.83 (+3.49)	**64.59%** (+6.46%)	**67.97%** (+5.20%)
Shikra	clean image	3.11	46.13	1.56	23.39	52.95%	53.16%
	ε=2/255	3.13	45.99	1.69 (+0.13)	25.65 (+2.26)	56.04% (+3.09%)	57.93% (+4.77%)
	ε=5/255	3.26	46.82	1.83 (+0.27)	26.51 (+3.12)	**57.88%** (+4.93%)	58.25% (+5.09%)
	ε=8/255	3.12	45.19	1.69 (+0.13)	25.69 (+2.30)	56.31% (+3.36%)	**59.11%** (+5.95%)

Black-box Attack

Table 3: Results of GPT-4 assisted hallucination evaluation for the image captioning task on black-box models. All of the MLLM responses are generated with *beam search* decoding. The six aspects of evaluation are the same as in Tab. 2. A larger HSPI, HWPI, HSR, and HWR indicate a higher level of hallucination in MLLM responses. The best results are marked in bold, and the number in brackets indicates the hallucination improvement compared to the clean image for each target model.

Surrogate Model	Target Model	SPI	WPI	HSPI	HWPI	HSR(%)	HWR(%)
InstructBLIP	InstructBLIP	4.47	80.48	3.04 (+0.21)	54.90 (+6.85)	68.41% (+5.50%)	70.84% (+5.91%)
	LLaVA-1.5	4.46	99.77	2.64 (-0.04)	70.51 (-8.57)	59.42% (-0.20%)	71.48% (-0.20%)
	MiniGPT-4	3.84	63.00	2.31	40.54 (+3.20)	**61.81%** (+3.68%)	**68.21%** (+5.44%)
	Shikra	3.20	48.95	1.79 (+0.23)	27.77 (+4.38)	56.14% (+3.19%)	57.09% (+3.93%)
LLaVA-1.5	LLaVA-1.5	4.49	108.03	2.67 (-0.01)	74.85 (-4.23)	62.36% (+2.74%)	75.74% (+4.06%)
	InstructBLIP	4.47	78.31	2.81 (-0.02)	51.85 (+3.80)	65.37% (+2.46%)	68.75% (+3.82%)
	MiniGPT-4	3.95	63.60	2.32 (+0.01)	42.25 (+4.91)	60.79% (+2.66%)	68.14% (+5.37%)
	Shikra	3.08	45.94	1.94 (+0.38)	29.96 (+6.57)	**63.85%** (+10.90%)	**65.90%** (+12.74%)
MiniGPT-4	MiniGPT-4	4.00	64.51	2.55 (+0.24)	40.83 (+3.49)	64.59% (+6.46%)	67.97% (+5.20%)
	InstructBLIP	4.36	79.62	2.96 (+0.13)	54.82 (+6.77)	**68.96%** (+6.05%)	**71.94%** (+7.01%)
	LLaVA-1.5	4.27	116.50	2.51 (-0.17)	75.86 (-3.22)	60.84% (+1.22%)	73.67% (+1.99%)
	Shikra	3.33	49.67	1.92 (+0.36)	28.99 (+5.60)	58.86% (+5.91%)	59.61% (+6.45%)
Shikra	Shikra	3.12	45.19	1.69 (+0.13)	25.69 (+2.30)	56.31% (+3.36%)	59.11% (+5.95%)
	InstructBLIP	4.48	80.36	2.90 (+0.07)	54.39 (+6.34)	67.48% (+4.57%)	70.28% (+5.35%)
	LLaVA-1.5	4.43	110.61	2.71 (+0.03)	75.34 (-3.74)	**64.77%** (+5.15%)	77.35% (+5.67%)
	MiniGPT-4	3.97	72.35	2.35 (+0.04)	46.90 (+9.56)	60.86% (+2.73%)	**70.07%** (+7.30%)

Defense: Attacking Mitigation Mechanisms

Table 5: Results of GPT-4 assisted hallucination evaluation against mitigation mechanisms on LLaVA-1.5. (*), (°), and (◇) denote methods through *decoding*, *model retraining*, and *post-processing* respectively. Best results are marked in bold.

Mitigation	Input	HSR(%)	HWR(%)
OPERA* [28]	clean	50.27%	51.93%
	ε=2/255	53.50% (+3.23%)	56.13% (+4.20%)
	ε=5/255	52.33% (+2.06%)	54.37% (+2.44%)
	ε=8/255	**55.86%** (+5.59%)	**58.18%** (+6.25%)
VCD* [35]	clean	51.38%	53.58%
	ε=2/255	54.46% (+3.08%)	57.02% (+3.44%)
	ε=5/255	57.69% (+6.31%)	60.12% (+6.54%)
	ε=8/255	**62.42%** (+11.04%)	**64.95%** (+11.37%)
Less is More° [81]	clean	43.74%	45.78%
	ε=2/255	46.22% (+2.48%)	48.23% (+2.45%)
	ε=5/255	47.68% (+3.94%)	49.91% (+4.13%)
	ε=8/255	**52.77%** (+9.03%)	**54.07%** (+8.29%)

Table 6: Results of GPT-4 assisted hallucination evaluation against mitigation mechanisms on MiniGPT-4. (*), (°), and (◇) denote methods through *decoding*, *model retraining*, and *post-processing* respectively. Best results are marked in bold.

Mitigation	Input	HSR(%)	HWR(%)
OPERA* [28]	clean	43.71%	45.79%
	ε=2/255	57.09% (+13.38%)	59.34% (+13.55%)
	ε=5/255	**60.78%** (+17.07%)	**63.75%** (+17.96%)
	ε=8/255	59.03% (+15.32%)	61.82% (+16.03%)
LRV-Instruction° [44]	clean	67.19%	70.82%
	ε=2/255	69.73% (+2.54%)	73.81% (+2.99%)
	ε=5/255	70.75% (+3.56%)	75.03% (+4.21%)
	ε=8/255	**71.43%** (+4.24%)	**75.54%** (+4.72%)
LURE◇ [85]	clean	48.57%	53.54%
	ε=2/255	58.21% (+9.64%)	64.44% (+10.90%)
	ε=5/255	59.42% (+10.85%)	67.41% (+13.87%)
	ε=8/255	**59.97%** (+11.40%)	**67.85%** (+14.31%)

Defense: Attacking Mitigation Mechanisms

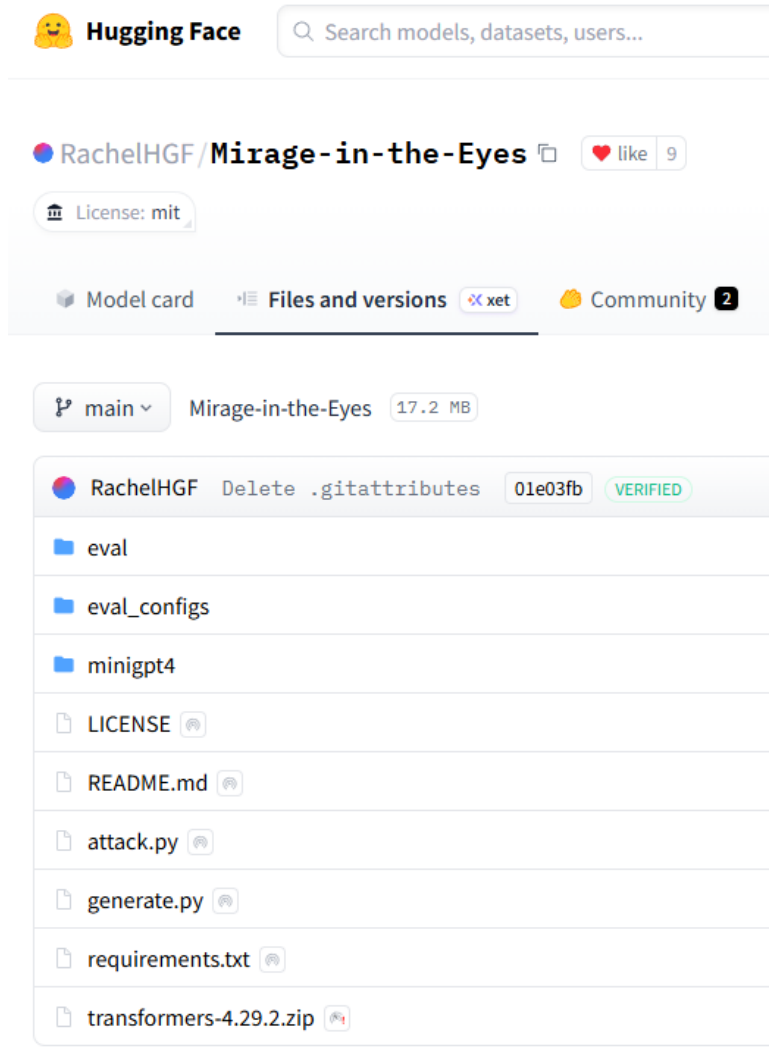
- Why existing defenses fail
  - OPERA (decoding-based)

$$\text{score}(y_t) = \log p\,(y_t, \mid xy_{<t}) - \lambda \cdot \text{SinkPenalty}$$

    - Only penalizes sink patterns at output selection stage, but cannot fix earlier representation-level manipulation
  - Other approaches also do not intervene in how attention is formed during generation
    - E.g., VCD, LRV-Instruction, LURE, EOS-based
- Existing defenses fail because they operate after or outside attention formation, while Mirage manipulates the attention mechanism itself

Implementation

- Implementation based on their official Hugging Face code
- Setting
  - Judge: gpt-4.1-mini (openrouter API)
  - Models: LLaVA-5.1, MiniGPT-4
  - Dataset: HalluBench (50 images)
  - Epsilon: 8/255



Implementaion Results

- White-box attack

Target Model	Method / Condition	HSR(%)	HWR(%)
LLaVA-1.5	Clean baseline	33.9%	36.0%
	LLaVA mirage	39.0%	40.3%
	LLaVA mirage + OPERA	42.7%	46.4%
MiniGPT-4	Clean baseline	33.6%	39.2%
	MiniGPT mirage	42.6%	44.9%
	MiniGPT mirage + OPERA	43.9%	48.6%

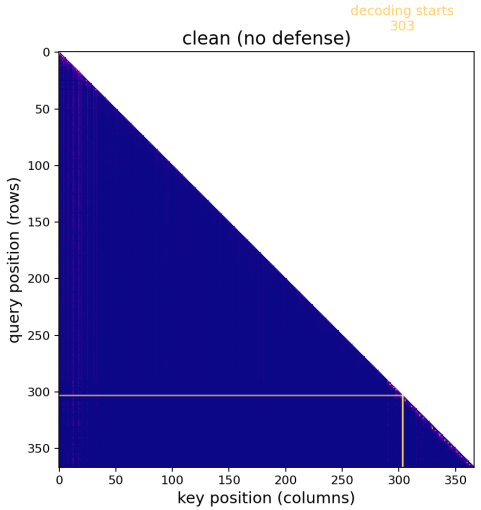
- Black-box attack

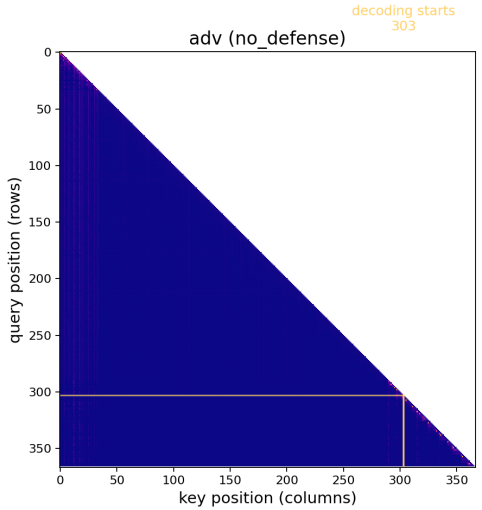
Target Model	Method / Condition	HSR(%)	HWR(%)
LLaVA-1.5	Clean baseline	33.9%	36.0%
	MiniGPT mirage	31.4%	35.7%
MiniGPT-4	Clean baseline	33.6%	39.2%
	LLaVA mirage	43.1%	44.9%

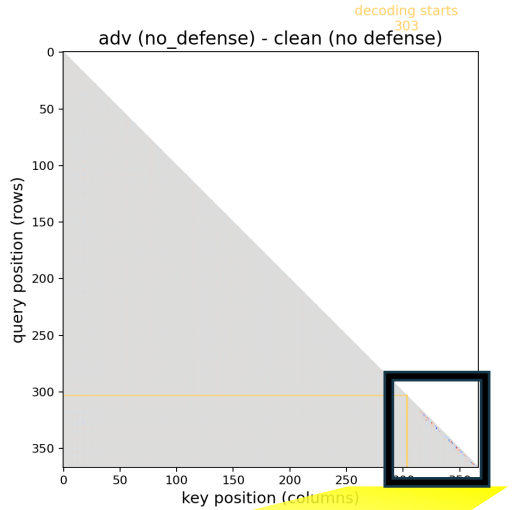
Implementaion Results (Attention Map)

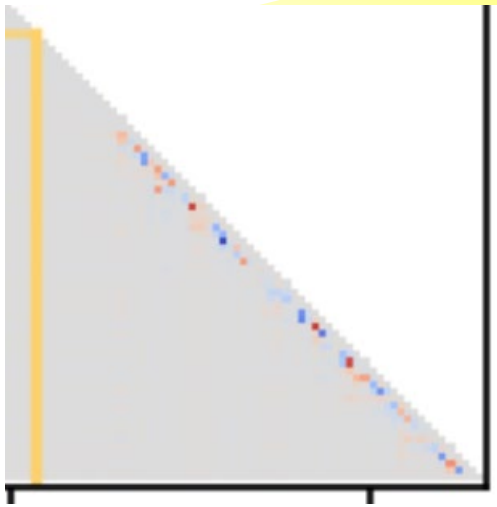
llava-1.5 image 2414349 no_defense L1 H19

- LLaVA-1.5 attention map (clean/adv/adv-clean)





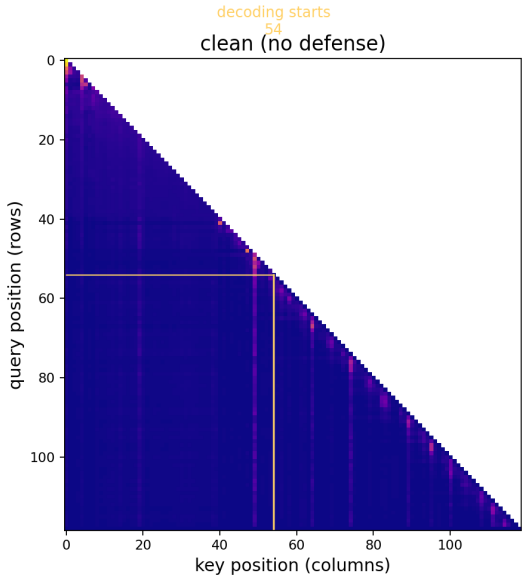


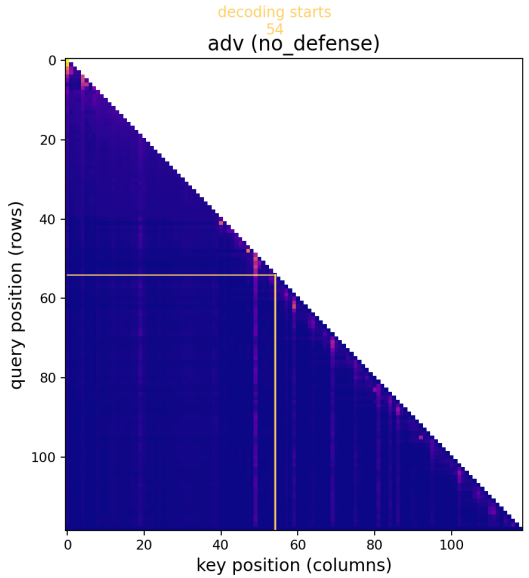


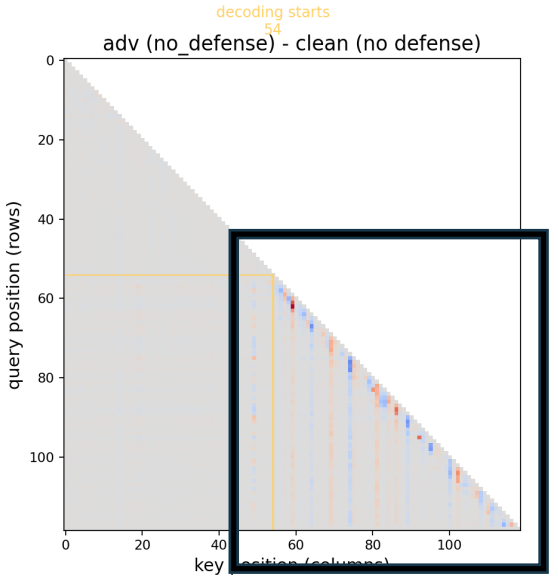
Implementaion Results (Attention Map)

minigpt4 image 2414349 no_defense L1 H4

- MiniGPT-4 attention map (clean/adv/adv-clean)







- Adversarial sink in decoding stage is matter!

2. Defense of Mirage in the eyes - Prefix Mitigation

Prefix Mitigation

- A paper in the LLM quantization field
- Mitigates statistical activation outliers by adding prefix tokens before attention

Prefixing Attention Sinks can Mitigate Activation Outliers for Large Language Model Quantization

Seungwoo Son[1,2*], Wonpyo Park[2], Woohyun Han[2], Kyuyeun Kim[2], Jaeho Lee[1,2†]
[1]POSTECH [2]Google
{swson, jaeho.lee}@postech.ac.kr
{wppark, woohyun, kyuyeunk}@google.com

Prefix Mitigation

- Motivation: maybe it could remove adversarial attention sink in decoding stage

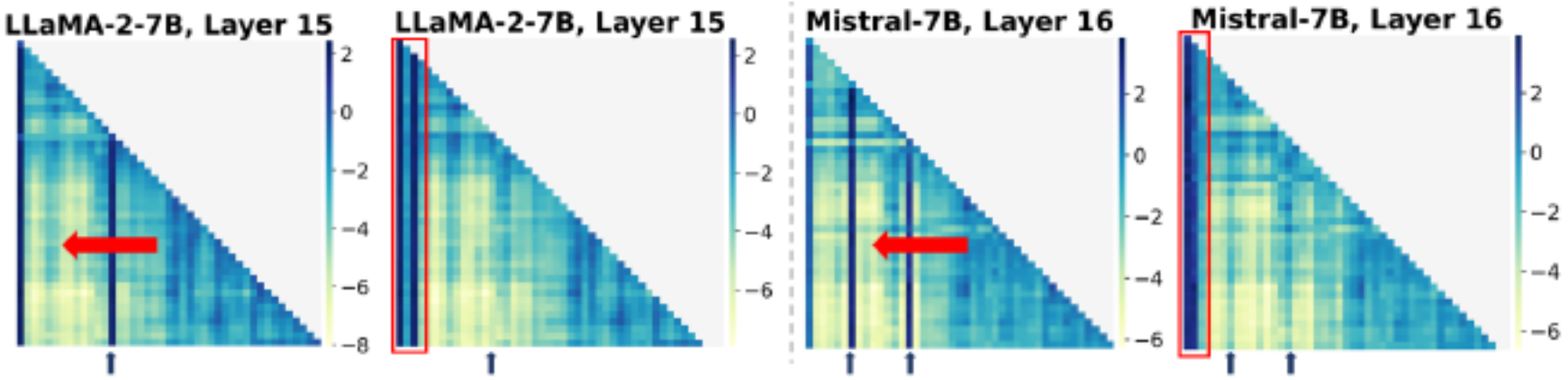


Figure 3: **Attention patterns before and after applying CushionCache in LLaMA3-8B and Mistral-7B.** The first and third panels show the attention patterns in models without CushionCache, where the attention sinks are quite prevalent in the generated token sequence. The second and fourth panels illustrate the attention patterns after inserting CushionCache. By adding the CushionCache, the attention is redirected toward the CushionCache tokens, preventing the attention sink from arising in the subsequent tokens.

Experiments

- Prefix

Target Model	Prefix Location	Hard Prefix
LLaVA-1.5	Before <ImageHere>	<s><s><s><s>
MiniGPT-4	Before <ImageHere>	</s></s></s>

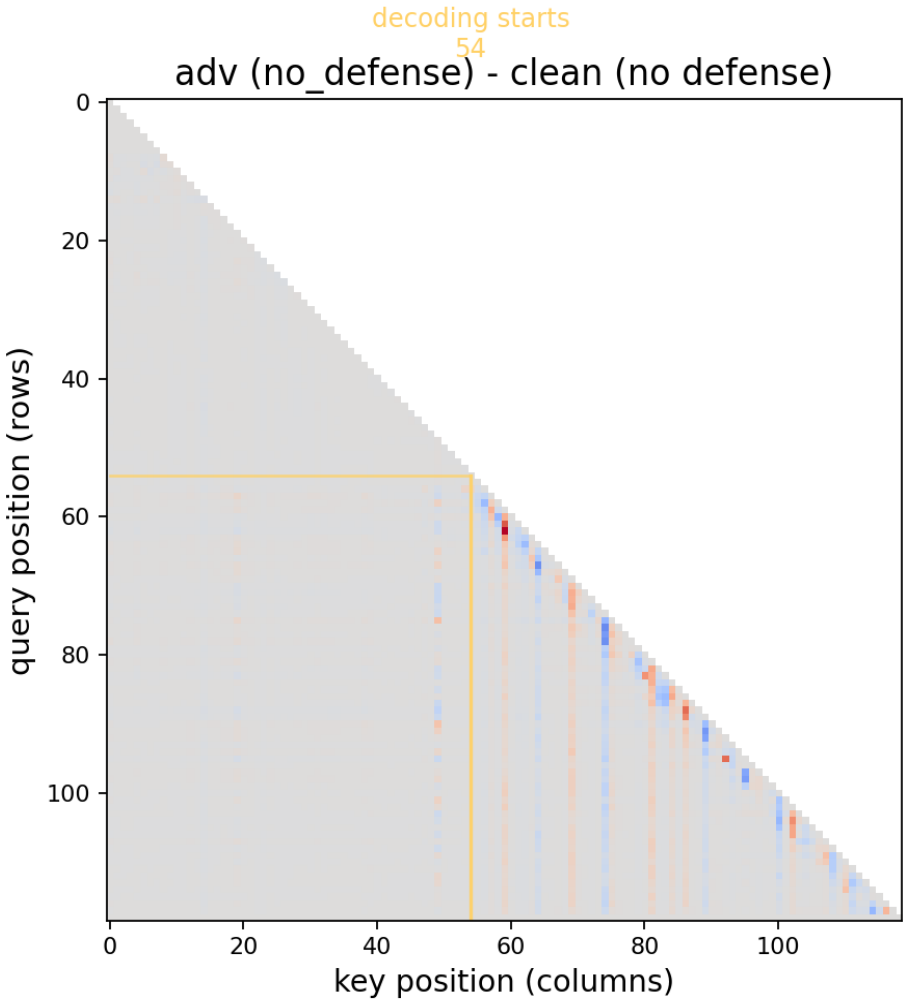
  - LLaVA: USER: <s><s><s><s><ImageHere> Please describe this image in detail. ASSISTANT:
  - MiniGPT-4: ###Human: <Img></s></s></s><ImageHere></Img> Please describe this image in detail. ###Assistant:

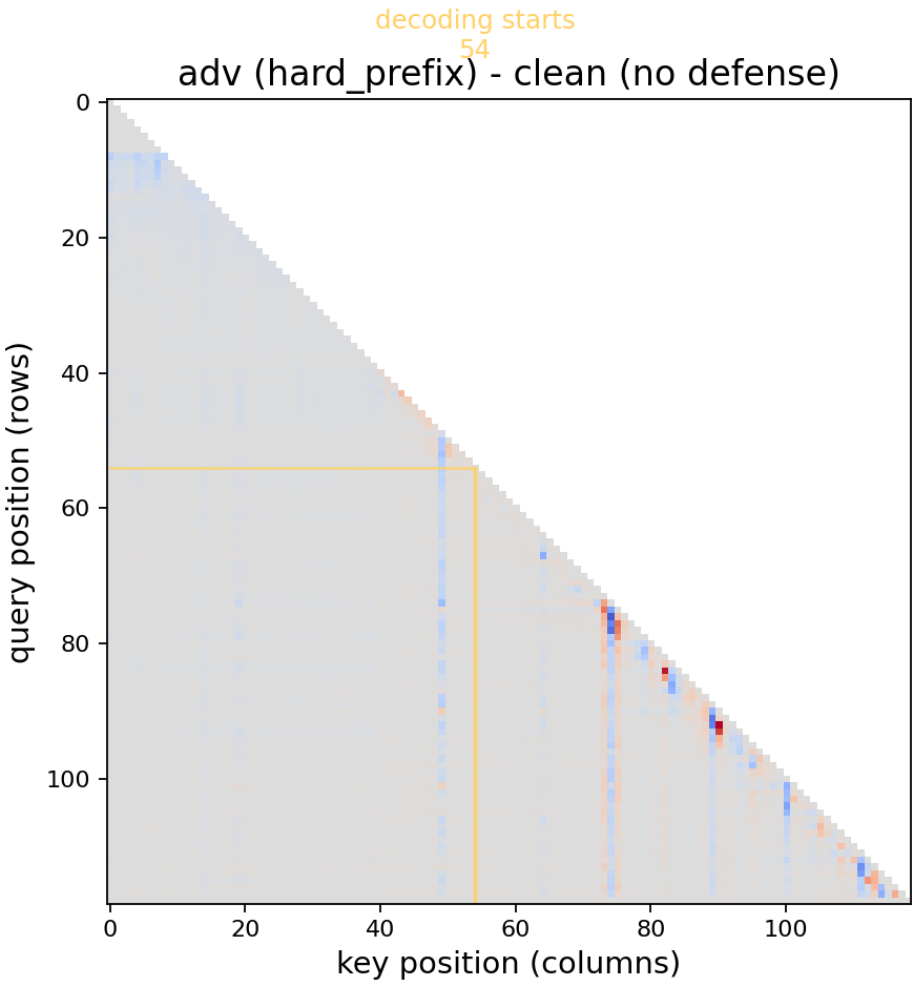
Experiments Results

- Although it mitigates attention sinks, it is insufficient as a defense

Target Model	Method / Condition	HSR(%)	HWR(%)
LLaVA-1.5	Clean baseline	33.9%	36.0%
	LLaVA mirage	39.0%	40.3%
	LLaVA mirage + OPERA	42.7%	46.4%
	LLaVA mirage + Prefix	40.2%	42.0%
MiniGPT-4	Clean baseline	33.6%	39.2%
	MiniGPT mirage	42.6%	44.9%
	MiniGPT mirage + OPERA	43.9%	48.6%
	MiniGPT mirage + Prefix	48.5%	48.9%

Experiments Results (Attention map)





3. Defense of Mirage in the eyes - DASH

POSTECH

POSTECH GSAI GRADUATE SCHOOL OF ARTIFICIAL INTELLIGENCE

POSTECH COMPUTER SCIENCE & ENGINEERING

ACT (Attention Calibration Technique)

- A **training-free method** to optimize attention distributions during inference
- Identifies and reduces **harmful attention sinks** in selected heads
  - Redistributes attention to other tokens
- Goal: Improve model accuracy by correcting attention allocation

---

Unveiling and Harnessing Hidden Attention Sinks: Enhancing Large Language Models without Training through Attention Calibration

---

Zhongzhi Yu [*1] Zheng Wang [*1] Yonggan Fu [1] Huihong Shi [1] Khalid Shaikh [1] Yingyan (Celine) Lin [1]

ACT (Attention Calibration Technique)

- There is a consistent location pattern (layer/head) in which attention sinks should be recalibrated to improve model performance

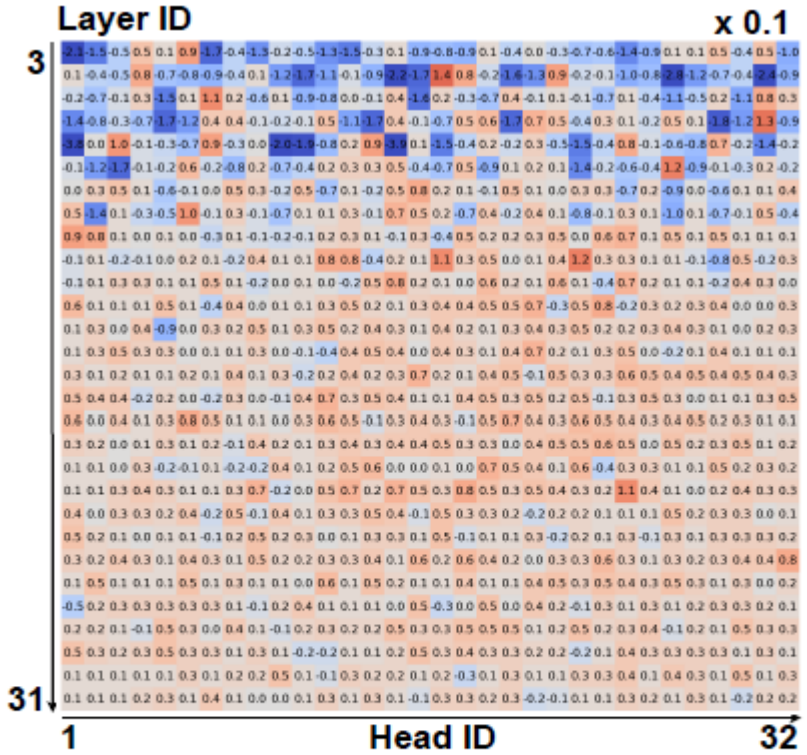


Figure 3. Visualization of accuracy improvement in the MMLU dataset (Hendrycks et al., 2020) achieved by reducing the attention score of attention sinks in the middle of input sequences for each individual head separately.

ACT (Attention Calibration Technique)

- Head selection
  - Step 1: Evaluate each head individually
    $\text{Acc}_{\ell,h}$ = Performance after calibrating head (ℓ, h)
    - Apply calibration to **one head at a time**
  - Step 2: Select beneficial heads
    $$H = \{(\ell, h) \mid Acc_{\ell,h} > Acc_{baseline}\}$$
    - Keep only heads where calibration improves accuracy
    - Exclude heads where calibration hurts performance
  - Step 3: Apply ACT only to selected heads
    $(\ell, h) \in H \Rightarrow$ apply attention calibration
    - Calibration is not applied globally
    - Only applied to **selected layer-head pairs**

ACT (Attention Calibration Technique)

- Attention Redistribution
  - Identify attention sink tokens in each layer/head:

$$S_h^\ell = \left\{ t \mid a_h^\ell[t] > \alpha \cdot \frac{1}{N} \right\}$$

  - Reduce attention to sink tokens:

$$\hat{A}_h^\ell[k, s] = \beta A_h^\ell[k, s], s \in S_h^\ell$$

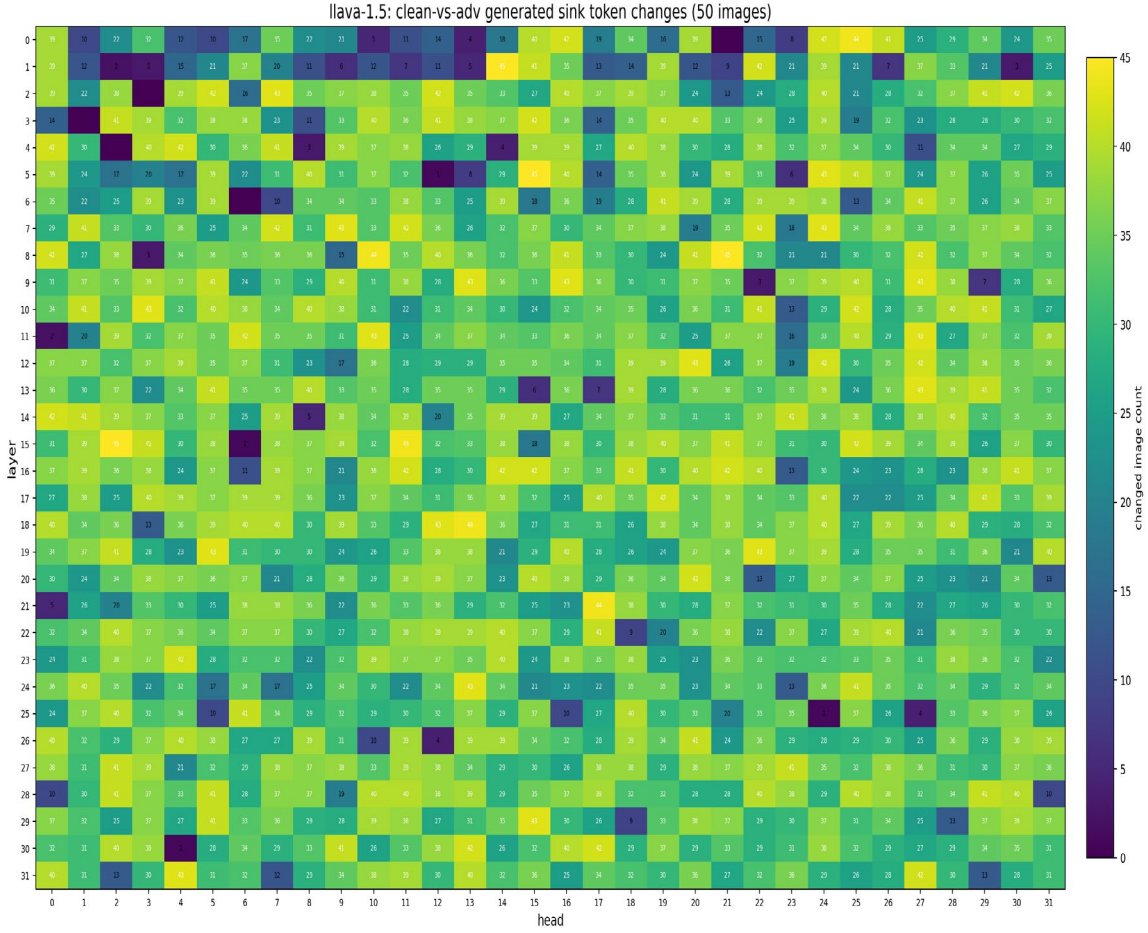
  - Redistribute the reduced attention mass to non-sink tokens:

$$\sum_j \hat{A}_h^\ell[k, j] = 1$$

  - Apply this only to selected heads where calibration improves performance

ACT (Attention Calibration Technique)

- Motivation
  - Does Mirage, which targets hallucination attacks caused by non-semantic attention sinks, also exhibit patterns in the locations of adversarial sinks?



DASH: Decoding-stage Attention Sink Head Mitigation

- 1. Measure Decoding-stage Sink Strength

$$C_{\ell,h}(j) = \frac{1}{T-j} \sum_{q=j+1}^{T} A_{\ell,h}(q, j)$$

  - y_j :generated token
  - $A_{\ell,h}(q, j)$:attention from later query $q$to token j
  - High $C_{\ell,h}(j)$ means token y_j acts as a decoding-stage sink

- 2. Find Mirage-induced Sink Heads

$$E_{\ell,h}(j) = \max\left(0, C_{\ell,h}^{\text{adv}}(j) - C_{\ell,h}^{\text{clean}}(j) - m\right)$$

  - Compare clean vs. adversarial images
  - Large $E_{\ell,h}(j)$ indicates an adversarially amplified sink

$$\boldsymbol{H}_{\mathbf{adv}} = \textit{heads with large } E_{\ell,h}$$

DASH: Decoding-stage Attention Sink Head Mitigation

- 3. Find Clean-safe Heads

$$\boldsymbol{H_{\text{safe}}} = \{(\ell, h): \text{clean behavior is preserved}\}$$

  - Clean-safety criteria:
    - Token F1 is preserved
    - Response length ratio is stable
    - No-degrade rate remains high
  - Only modify heads that **do not harm** clean generation.

- 4. Select DASH Heads

$$H_{\text{DASH}} = H_{\text{adv}} \cap H_{\text{safe}}$$

  - **Adv-sensitive:** actually affected by Mirage
  - **Clean-safe:** safe to recalibrate
  - DASH modifies only the intersection

DASH: Decoding-stage Attention Sink Head Mitigation

- 5. Conservative Head Pruning
  - Start from all intersection heads:

$$H_{\mathrm{DASH}}^{\mathrm{all}} = H_{\mathrm{safe}} \cap H_{\mathrm{adv}}$$

  - First apply redistribution to all candidate heads
  - If clean generation becomes unstable, remove heads conservatively:

late heads → middle heads → early heads

  - Motivation:
    - Later layers are more directly tied to language generation
    - Modifying them may more easily distort clean responses

DASH: Decoding-stage Attention Sink Head Mitigation

- Clean-safety Proxy
  - For each candidate set, compare clean outputs with and without DASH.

$$\text{Token F1} \geq 0.9$$

$$0.8 \leq \frac{|\text{DASH output}|}{|\text{baseline output}|} \leq 1.25$$

$$\text{ImageAttn}_{DASH} \geq 0.8 \cdot \text{ImageAttn}_{baseline}$$

  - No malformed outputs
  - No-degrade rate ≥ 0.95
    - roughly ≥ 48 / 50 clean samples
- Final Clean-safety Check

$$\text{HSR}_{\text{DASH,clean}} \leq \text{HSR}_{\text{clean}} + 0.02$$

$$\text{HWR}_{\text{DASH,clean}} \leq \text{HWR}_{\text{clean}} + 0.02$$

  - DASH is accepted only if clean hallucination does not noticeably increase

DASH: Decoding-stage Attention Sink Head Mitigation

- 6. Apply Attention Redistribution

$$A' = \text{Redistribute}(A; \beta)$$

  - Reduce excessive attention to adversarial sink tokens
  - Redistribute attention to valid non-sink tokens
  - Use mitigation strength:

$$\beta = 0.2$$

  - Protect BOS, special tokens, and punctuation to avoid unnecessary disruption

DASH vs ACT

Aspect	ACT	DASH
Goal	General accuracy improvement	Mirage hallucination defense
Target	LLM attention sinks	MLLM decoding-stage sinks
Head selection	Heads that improve accuracy	Adv-sensitive ∩ clean-safe heads
Attack awareness	Not considered	Uses clean vs. adversarial sink shift
Sink target	General attention sinks	Adversarial decoding sinks
Clean criterion	Improves performance	Does not degrade clean behavior
Token handling	Default sink logic	Protects BOS/special/punctuation tokens

Experiments

- Setup
  - Finding DASH layer/head from 50 images in HalluBench
  - Evaluated the selected heads on a separate set of 50 images
  - Ensured no overlap between head selection and evaluation data
- Head selecting
  - LLaVA example

Procedure	Heads	No-degrade rate	Clean proxy pass
Initial = H_{DASH}^{all}	L1H19, L18H5, L22H11, L26H4	0.90	False
Remove L26H4	L1H19, L18H5, L22H11	0.92	False
Remove L22H11	L1H19, L18H5	0.92	False
Remove L18H5	L1H19	0.96	True

  - Final layer/head: LLaVA-1.5 (L1H19) MiniGPT-4 (L1H4)

Experiment Results (White-box Attack)

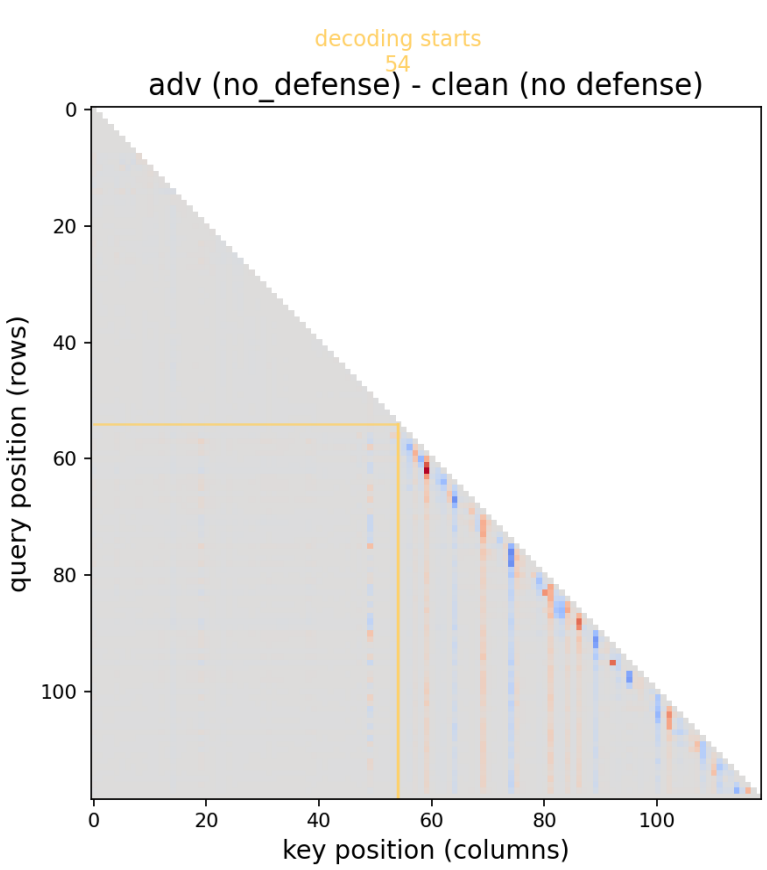
Target Model	Method / Condition	HSR(%)	HWR(%)
LLaVA-1.5	Clean baseline	33.9%	36.0%
	LLaVA mirage	39.0%	40.3%
	LLaVA mirage + OPERA	42.7%	46.4%
	LLaVA mirage + DASH	**36.1%**	**39.0%**
MiniGPT-4	Clean baseline	33.6%	39.2%
	MiniGPT mirage	42.6%	44.9%
	MiniGPT mirage + OPERA	43.9%	48.6%
	MiniGPT mirage + DASH	**42.2%**	**46.3%**

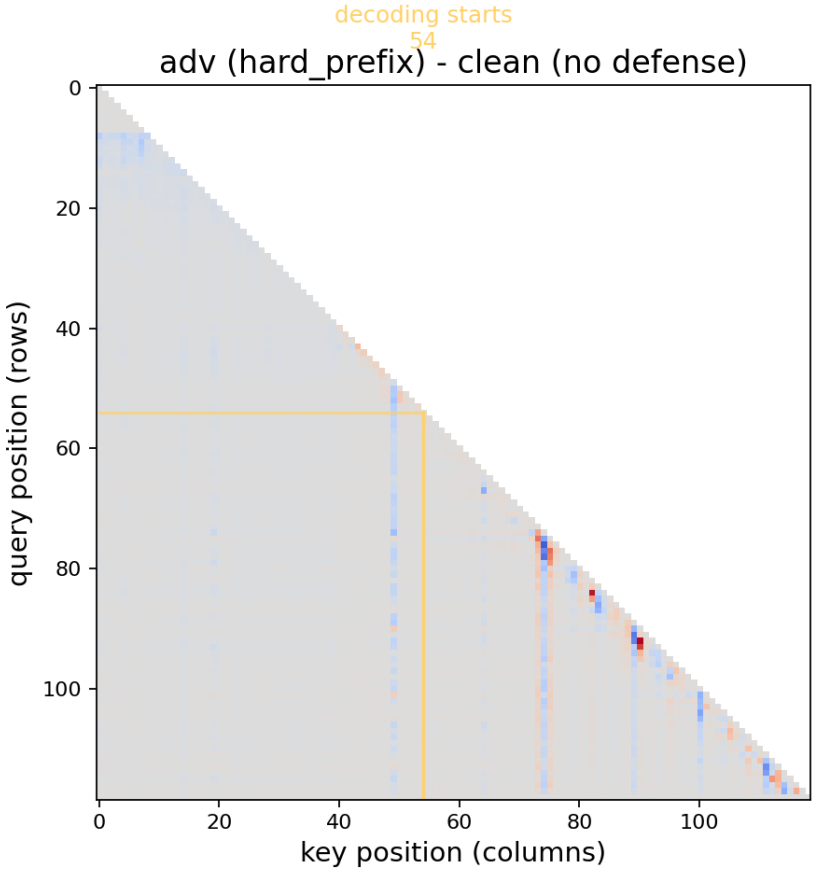
Experiment Results (Black-box Attack)

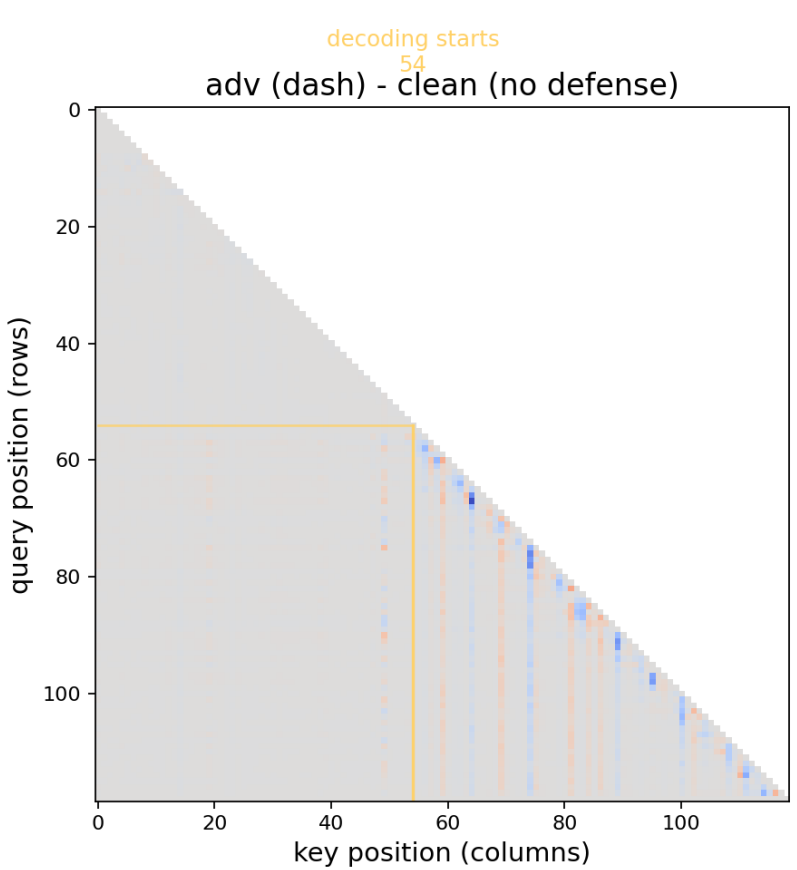
Target Model	Method / Condition	HSR(%)	HWR(%)
LLaVA-1.5	Clean baseline	33.9%	36.0%
	MiniGPT mirage	31.4%	35.7%
	MiniGPT mirage + DASH	**29.7%**	**31.3%**
MiniGPT-4	Clean baseline	33.6%	39.2%
	LLaVA mirage	43.1%	44.9%
	LLaVA mirage + DASH	**43.3%**	**46.7%**

Experiments Results (Attention map)

- MiniGPT-4

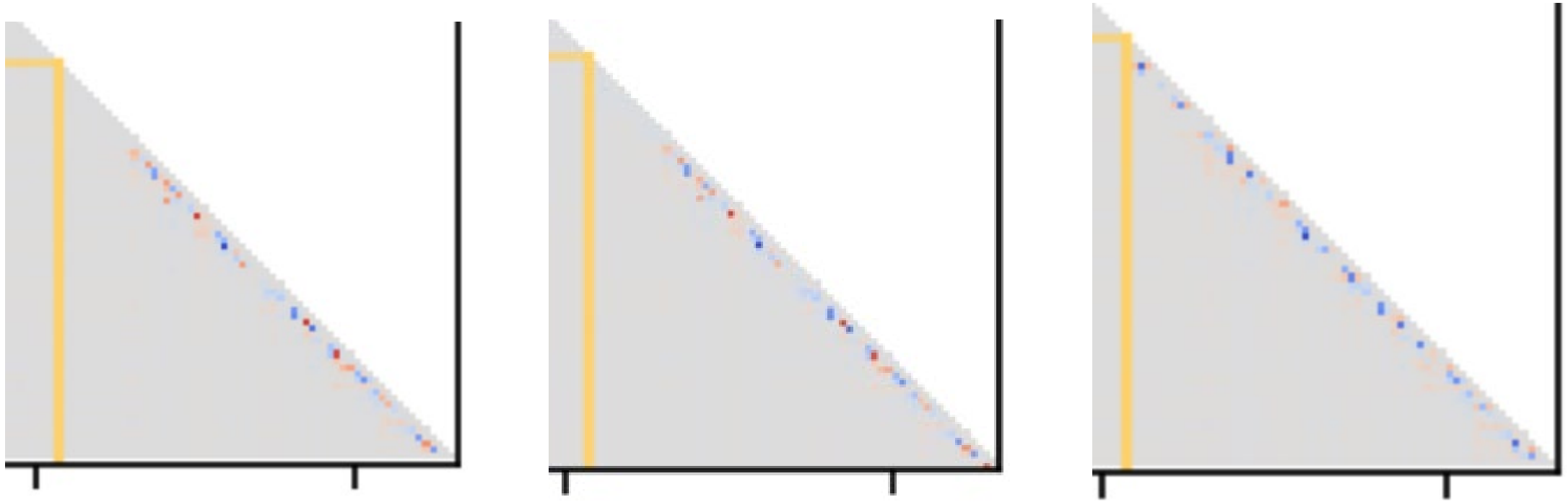






Experiments Results (Attention map)

- LLaVA-1.5



Future Works

- Scaling up
  - Use more images
  - Evaluate on larger datasets
- Stronger Judge
  - Use more reliable models (e.g., GPT-4.1)
- More Diverse Settings
  - Vary thresholds and layer/head combinations
  - Test on more models
  - Explore different ϵ values
- Adaptive Mirage Attack
  - Introduce randomness across layers/heads
  - Design objectives that avoid **fixed sink patterns**

Take-home Messages

- Attention sink–aware attacks are feasible
  - Attention sink pattern during the token generation stage is the critical factor
- However, they often exhibit identifiable patterns
- In simple (naïve) settings, ACT-based methods show partial effectiveness in mitigating these attacks.

References

- Wang, Yining, et al. "Mirage in the eyes: Hallucination attack on multi-modal large language models with only attention sink." *34th USENIX Security Symposium (USENIX Security 25)*. 2025.
- Son, Seungwoo, et al. "Prefixing attention sinks can mitigate activation outliers for large language model quantization." *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024.
- Yu, Zhongzhi, et al. "Unveiling and harnessing hidden attention sinks: enhancing large language models without training through attention calibration." *Proceedings of the 41st International Conference on Machine Learning*. 2024.

Q&A