Membership Inference Attacks against Large Vision-Language Models

Li Z, Wu Y, Chen Y, et al. at NeurIPS 2024

https://proceedings.neurips.cc/paper_files/paper/2024/hash/b2c892312af07f8a77afbeed188391f4-Abstract-Conference.html

Presented by Tong, Ziyi
Jan 8th 2026

Background

- **Large vision-language model** (VLLM) exhibit promising capabilities for various application scenarios, the use of VLLMs become increasingly prevalent .
- **Membership Inference Attacks** (MIAs) aim to determine whether a given data record was included in a model's training dataset.
- The study of MIAs plays an important role in **preventing test data contamination** and **protecting data security**.

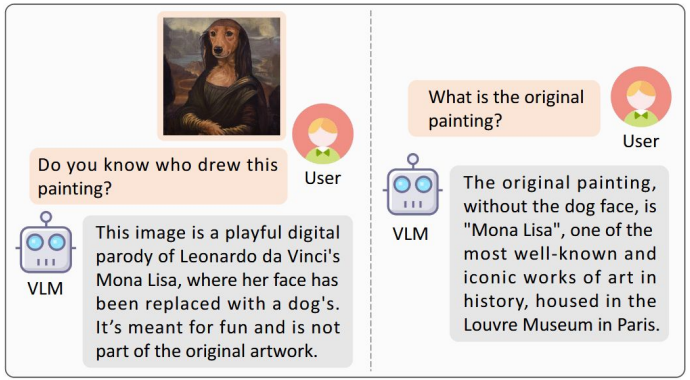
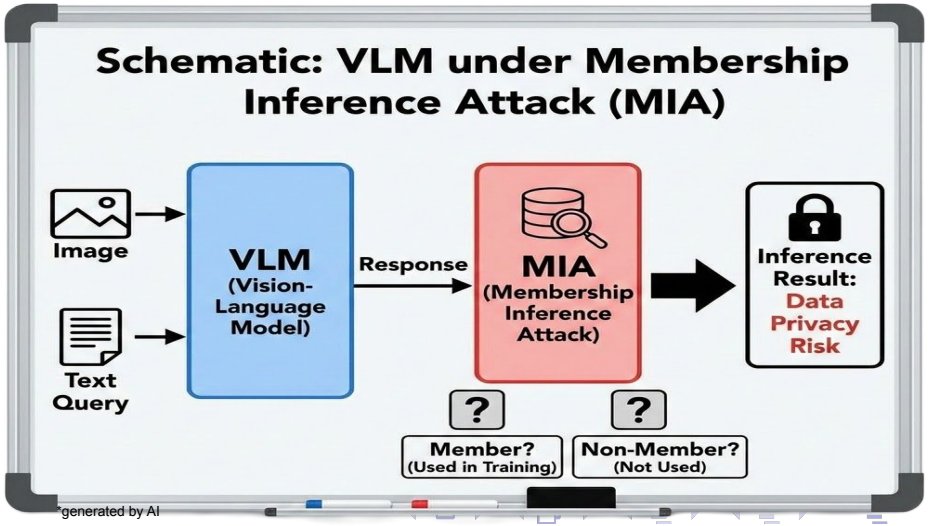


Figure 1: An example of the interaction with a VLM



Problem

- Absence of **dataset** for develop and evaluate MIAs in VLLMs
- Lack of efficient **techniques** to detect a single modality in VLLMs. (Image or text, single modality is more common in real world scenario)

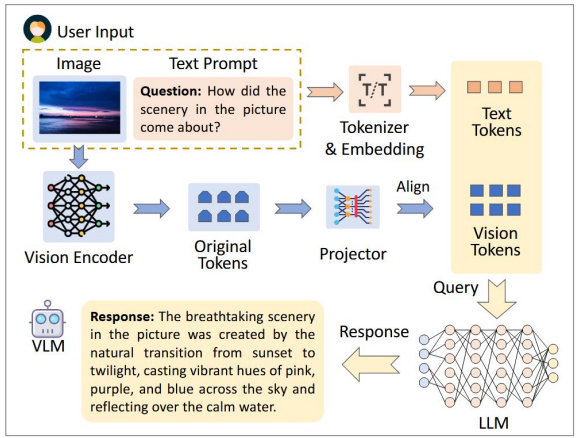


Figure 2: General Structure of VLMs

- Images are **challenging** to detect because image embedding are **projected** through an MLP layer and mapped into the language model's embedding space.
- **No discrete image tokens** to probe. For text, attacker probes token probabilities, but for image, attackers never sees image tokens.

Related work

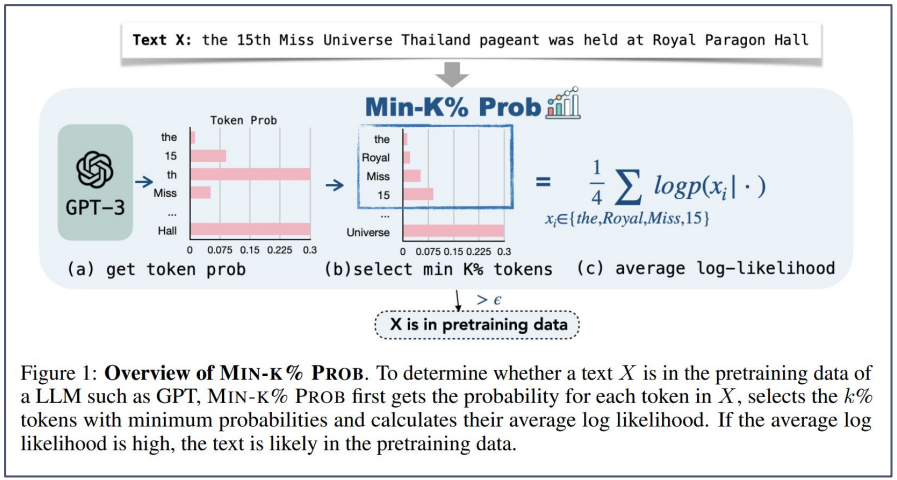


Figure 1: **Overview of MIN-K% PROB**. To determine whether a text X is in the pretraining data of a LLM such as GPT, MIN-K% PROB first gets the probability for each token in X, selects the k% tokens with minimum probabilities and calculates their average log likelihood. If the average log likelihood is high, the text is likely in the pretraining data.

Current sota method, **Min-K% Prob is designed for text LLM.** This idea come from the simple principle that : an unseen example is likely to contain a few outlier words with low probabilities under the LLM, while a seen example is less likely to have words with such low probabilities. **The process is :** get the log probabilities of the target text, and select the k% tokens that have lowest probability and then calculate their average

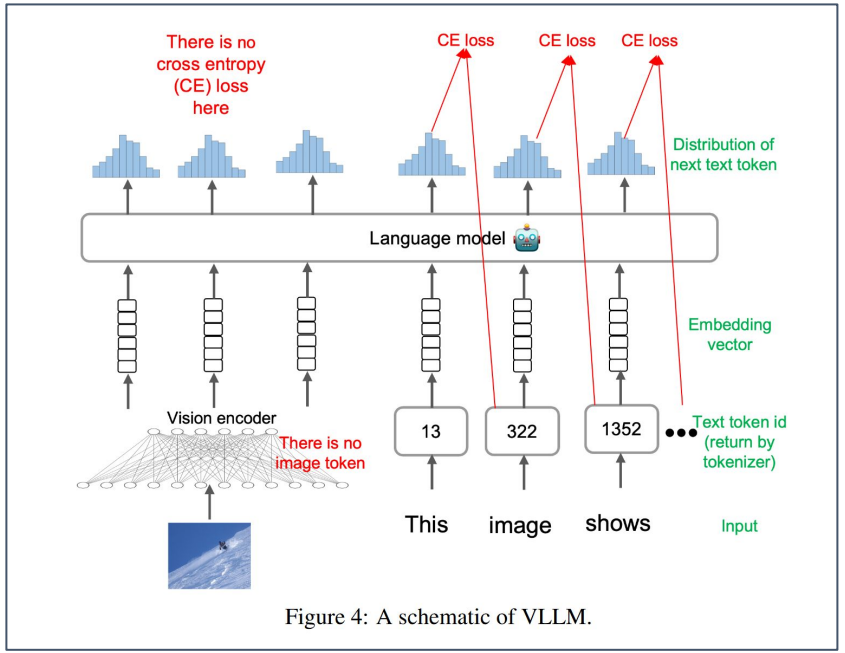


Figure 4: A schematic of VLLM.

Methods : dataset construction

VL-MIA dataset

Table 1: **Overview of VL-MIA dataset**: VL-MIA covers image and text modalities and can be applied for dominant open-sourced VLLMs.

Dataset	Modality	Member data	Non-member data	Application
VL-MIA/DALL-E	image	LAION_CCS	DALL-E-generated images	LLaVA 1.5 MiniGPT-4 LLaMA_adapter v2
VL-MIA/Flickr	image	MS COCO (from Flickr)	Latest images on Flickr	LLaVA 1.5 MiniGPT-4 LLaMA_adapter v2
VL-MIA/Text	text	LLaVA v1.5 instruction-tuning text	GPT-generated answers	LLaVA 1.5 LLaMA_adapter v2
		MiniGPT-4 instruction-tuning text	GPT-generated answers	MiniGPT-4

Existed training data

Generated/
Latest data

Target Model

Data examples and prompts

Table 16: Examples in VL-MIA/image non-member data are generated by DALL-E or collected from recent Flickr websites; text non-member data are generated by GPT-4.

Dataset	Member data	Non-member data
VL-MIA/DALL-E		
		
VL-MIA/Flickr		
		

VL-MIA/Text for MiniGPT-4	The image shows a bedroom with a wooden headboard and nightstands on either side of the bed. The bed is made with a white comforter and pillows, and there are two lamps	The image shows a bathroom with cream-colored walls. On the left, there is a vanity with a granite countertop and wooden cabinets below. A soap dispenser is placed on the countertop, and
	This image shows a blue pickup truck, which appears to be a Volkswagen Beetle, parked in a driveway in front of a house. The hood of the truck is open, exposing the	The image depicts a well-used kitchen with various cooking utensils and food items scattered throughout. On the left, there is a gas stove with a white oven beneath it. Above the stove
VL-MIA/Text for LLaVA 1.5 and LLaMA Adapter v2	To enjoy the last two pieces of cake equally and fairly, I suggest using a knife which, according to the image, is already present on the table. Carefully cut each of the	To enjoy the remaining pieces of this delectable cake fairly, I would recommend dividing the slices equally among those present, ensuring that each person gets an identical portion, or alternatively, one could
	The young boy is demonstrating the important habit of maintaining good oral hygiene by brushing his teeth. In the image, he is standing in front of a mirror holding a toothbrush, which	In the image, a young boy is engaging in the imperative habit of brushing his teeth, which is fundamental for maintaining oral hygiene, preventing dental issues like cavities and gum disease, and

Table 6: Different prompts we use for dataset construction.

Dataset	Prompt	Model
VL-MIA/DALL-E	{original caption in member data}	dall-e-2
VL-MIA/Text for MiniGPT-4	Provide a caption for this image, beginning with "The image".	gpt-4-vision-preview
VL-MIA/Text for LLaVA 1.5	{original question in member data}	gpt-4-vision-preview

Methods : MaxRenyi-K%

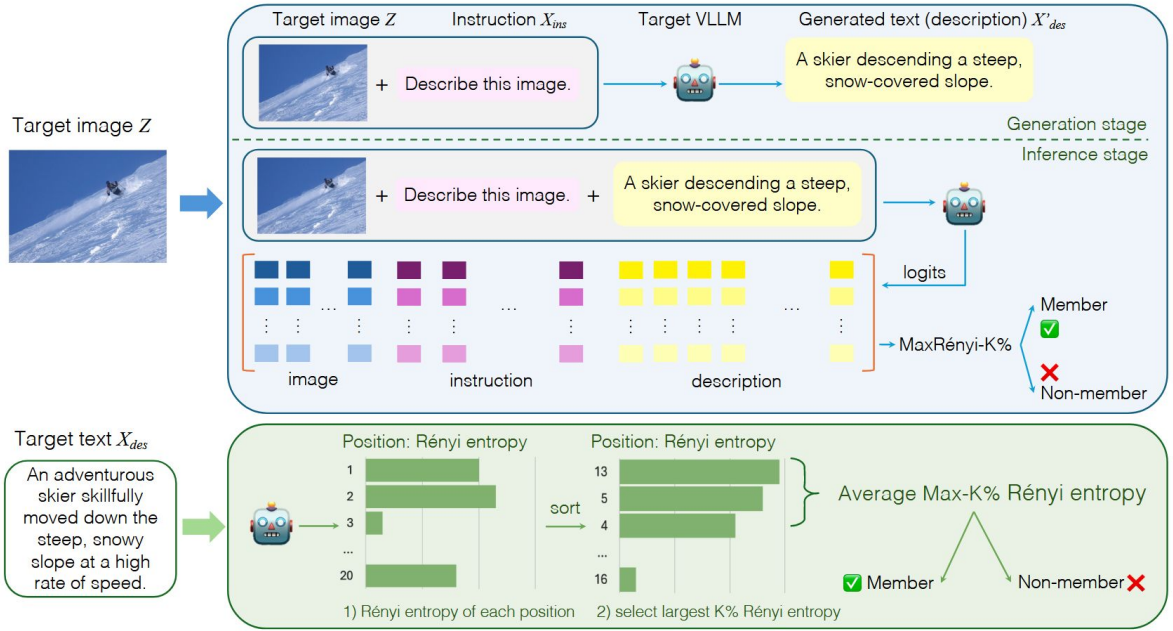


Figure 1: **MIAs against VLLMs**. **Top:** Our image detection pipeline: In the generation stage, we feed the image and instruction to the target model to obtain a description; then during the inference stage, we input the image, instruction, and generated description to the model, and extract the logits slices to calculate metrics. **Bottom:** `MaxRényi-K%` metric: we first get the Rényi entropy of each token position, then select the largest $k\%$ tokens and calculate the average Rényi entropy.

Pipeline of Two stages:

1. **In generation stage**, we provide the model with an image followed by an instruction to generate a textual sequence(description).
2. **In inference stage**, we feed the model with the concatenation of the same image, instruction, and generated description text.

MaxReny-K%:
If the model has seen this data before, the model will be more confident in the next token and thus have smaller Reny entropy.

Methods : MaxRenyi-K%

Given a probability distribution p, the Rényi entropy [47] of order α, is defined as $H_\alpha(p) = \frac{1}{1-\alpha}\log\left(\sum_j (p_j)^\alpha\right), 0 < \alpha < \infty, \alpha \neq 1$. $H_\alpha(p)$ is further defined at $\alpha = 1, \infty$, as $H_\alpha(p) = \lim_{\gamma\to\alpha} H_\gamma(p)$ by,

$$\bullet\ H_1(p) = -\sum_j p_j \log p_j, \quad \bullet\ H_\infty(p) = -\log\max p_j.$$

To be more specific, given a token sequence $X := (x_1, x_2, \ldots, x_L)$, let $p^{(i)}(\cdot) = \mathbb{P}(\cdot|x_1, \cdots, x_i)$ be the probability of next-token distribution at the i-th token. Let Max-K%(X) be the top K% from the sequence X with the largest Rényi entropies, the `MaxRényi-K%` score of X equals

$$\texttt{MaxRényi-K\%}(X) = \frac{1}{|\text{Max-K\%}(X)|}\sum_{i\in\text{Max-K\%}(X)} H_\alpha(p^{(i)}).$$

When $K = 0$, we define the `MaxRényi-K%` score to be $\max_{i\in[L-1]} H_\alpha(p^{(i)})$. When $K = 100$, the `MaxRényi-K%` score is the averaged Rényi entropy of the sequence X.

Methods : ModRenyi-K%

We also extend our `MaxRényi-K%` to the target-based scenarios, denoted by `ModRényi`. We first consider linearized Rényi entropy, $\overline{H}_\alpha(p) = \frac{1}{1-\alpha}\left(\sum_j (p_j)^\alpha - 1\right), 0 < \alpha < \infty, \alpha \neq 1$. $\overline{H}_\alpha(p)$ is also further defined at $\alpha = 1$, as $\overline{H}_1(p) = \lim_{\alpha \to 1} \overline{H}_\alpha(p) = H_1(p)$. Assuming the next token ID is y, recall that a small entropy value or a large p_y value indicates membership, we want our modified entropy to be monotonically decreasing on p_y and monotonically increasing on $p_j, j \neq y$. Therefore, we propose the modified Rényi entropy on a given next token ID y, denoted by $\overline{H}_\alpha(p, y)$:

$$\overline{H}_\alpha(p, y) = -\frac{1}{|\alpha - 1|}\left((1 - p_y)p_y^{|\alpha-1|} - (1 - p_y) + \sum_{j \neq y} p_j(1 - p_j)^{|\alpha-1|} - p_j\right).$$

Let $\alpha \to 1$, we have $\overline{H}_1(p, y) = \lim_{\alpha \to 1} \overline{H}_\alpha(p, y) = -\sum_{j \neq y} p_j \log(1 - p_j) - (1 - p_y) \log p_y$, which is equivalent to the `Modified Entropy` [58]. In addition, our more general method does not

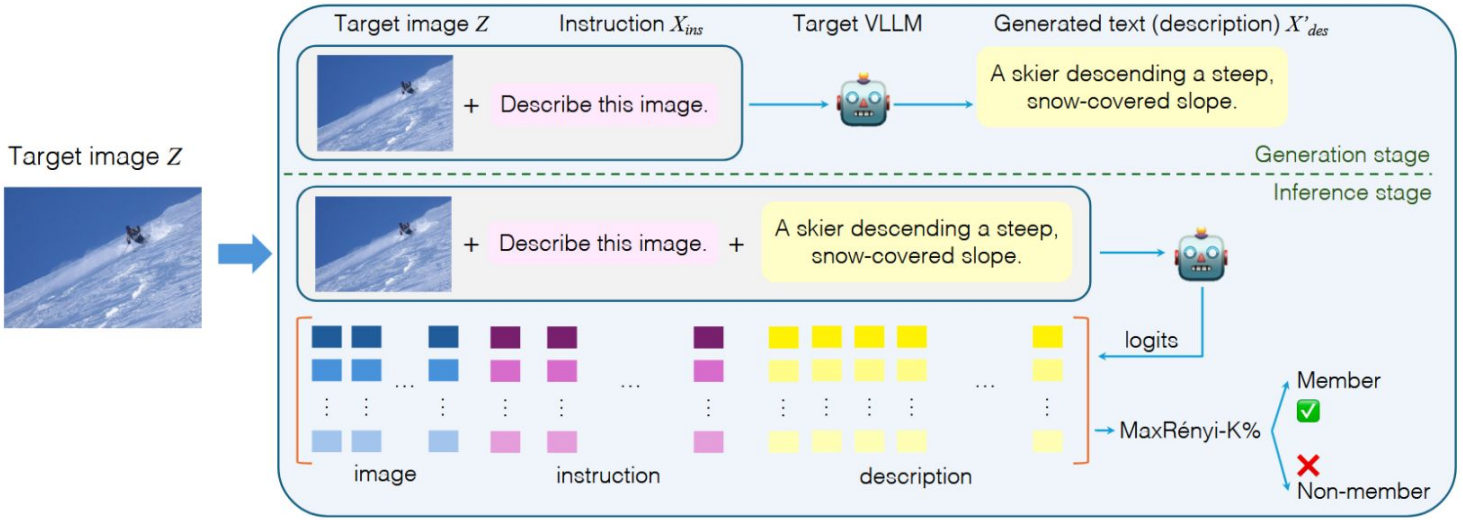
Experiments

Metric: AUROC (higher the better)

Baseline: take existing MIA methods as baselines. (perplexity, zlib,Min-K%)

Dataset : VL-MIA

Pipeline:



Experiments - Image MIA

Table 2: **Image MIA**. AUC results on VL-MIA under our pipeline. "img" indicates the logits slice corresponding to image embedding, "inst" indicates the instruction slice, "desp" the generated description slice, and "inst+desp" is the concatenation of the instruction slice and description slice. We use an asterisk * in superscript to indicate the target-based metric. **Bold** indicates the best AUC within each column and underline indicates the runner-up.

VL-MIA/Flickr												
Metric		**LLaVA**				**MiniGPT-4**				**LLaMA Adapter**		
		img	inst	desp	inst+desp	img	inst	desp	inst+desp	inst	desp	inst+desp
Perplexity*		N/A	0.378	0.667	0.559	N/A	0.414	0.649	0.497	0.380	0.661	0.425
Min_0% Prob*		N/A	0.357	0.651	0.357	N/A	0.272	0.569	0.274	0.462	0.566	0.463
Min_10% Prob*		N/A	0.357	0.669	0.390	N/A	0.272	0.603	0.265	0.437	0.591	0.438
Min_20% Prob*		N/A	0.374	0.670	0.370	N/A	0.293	0.628	0.303	0.437	0.611	0.424
Aug_KL		0.596	0.539	0.492	0.508	0.462	0.458	0.438	0.435	0.428	0.422	0.427
Max_Prob_Gap		0.577	0.601	0.650	0.650	0.664	0.695	0.609	0.626	0.475	0.671	0.661
ModRényi*	$\alpha = 0.5$	N/A	0.368	0.651	0.614	N/A	0.483	0.636	0.592	0.430	0.662	0.555
	$\alpha = 1$	N/A	0.359	0.659	0.502	N/A	0.371	0.635	0.417	0.394	0.646	0.423
	$\alpha = 2$	N/A	0.370	0.645	0.611	N/A	0.492	0.636	0.605	0.434	0.665	0.579
Rényi ($\alpha = 0.5$)	Max_0%	0.515	0.689	0.687	0.689	0.437	0.624	0.542	0.626	0.497	0.570	0.499
	Max_10%	0.557	0.689	0.691	0.719	0.493	0.624	0.592	0.707	0.432	0.573	0.622
	Max_100%	**0.702**	**0.726**	**0.713**	0.728	**0.671**	**0.795**	**0.664**	0.724	**0.633**	0.674	**0.697**
Rényi ($\alpha = 1$)	Max_0%	0.503	0.708	0.685	0.725	0.429	0.645	0.579	0.652	0.517	0.602	0.517
	Max_10%	0.623	0.708	0.698	**0.743**	0.489	0.645	0.627	0.710	0.456	0.610	0.565
	Max_100%	0.702	0.720	0.702	0.721	0.626	0.776	0.662	**0.741**	0.597	**0.678**	0.687
Rényi ($\alpha = 2$)	Max_0%	0.583	0.682	0.673	0.705	0.444	0.697	0.587	0.665	0.580	0.617	0.581
	Max_10%	0.621	0.682	0.685	0.725	0.482	0.697	0.621	0.734	0.499	0.615	0.584
	Max_100%	0.682	0.694	0.683	0.703	0.627	0.785	0.656	0.735	0.572	0.670	0.668
Rényi ($\alpha = \infty$)	Max_0%	0.588	0.646	0.651	0.674	0.462	0.693	0.569	0.657	0.604	0.566	0.603
	Max_10%	0.593	0.646	0.669	0.699	0.488	0.693	0.603	0.704	0.506	0.591	0.584
	Max_100%	0.669	0.673	0.667	0.687	0.632	0.769	0.649	0.725	0.564	0.661	0.659

AUC results on VL-MIA under designed pipeline. On subset Flickr.

1. **Methods performances differences:**
   - MaxReny-K% **surpasses** other methods in most scenario.
   - **α=0.5** yeilds the best performance.
   - Overall, ModReny and MaxReny **outperform** baseline.

Experiments - Image MIA

VL-MIA/DALL-E

Metric		LLaVA				MiniGPT-4				LLaMA Adapter		
		img	inst	desp	inst+desp	img	inst	desp	inst+desp	inst	desp	inst+desp
Perplexity*		N/A	0.338	0.564	0.448	N/A	0.356	0.517	0.421	0.491	0.577	0.506
Min_0% Prob*		N/A	0.482	0.559	0.482	N/A	0.422	0.494	0.421	0.448	0.554	0.448
Min_10% Prob*		N/A	0.482	0.563	0.425	N/A	0.422	0.495	0.462	0.447	0.556	0.455
Min_20% Prob*		N/A	0.434	0.559	0.353	N/A	0.462	0.501	0.401	0.560	0.460	0.456
Aug_KL		0.408	0.463	0.505	0.489	0.396	0.421	0.460	0.446	0.474	0.489	0.476
Max_Prob_Gap		0.529	0.575	**0.597**	0.602	0.527	0.407	0.505	0.490	0.518	0.553	0.555
ModRényi*	$\alpha = 0.5$	N/A	0.360	0.560	0.523	N/A	0.399	**0.518**	0.465	0.479	0.580	0.546
	$\alpha = 1$	N/A	0.342	0.560	0.425	N/A	0.372	0.516	0.450	0.478	0.576	0.489
	$\alpha = 2$	N/A	0.384	0.561	0.536	N/A	0.416	**0.518**	0.477	0.486	**0.581**	0.559
Rényi ($\alpha = 0.5$)	Max_0%	0.537	0.597	0.563	0.598	0.518	**0.496**	0.493	**0.497**	**0.625**	0.503	**0.624**
	Max_10%	0.622	0.597	0.563	**0.648**	**0.563**	**0.496**	0.504	0.482	0.573	0.516	0.573
	Max_100%	0.421	0.604	0.575	0.582	0.528	0.448	0.504	0.481	0.511	0.552	0.531
Rényi ($\alpha = 1$)	Max_0%	0.549	0.569	0.551	0.576	0.523	0.477	0.497	0.486	0.598	0.522	0.597
	Max_10%	0.667	0.569	0.558	0.586	0.555	0.477	0.512	0.472	0.532	0.530	0.553
	Max_100%	0.469	**0.637**	0.564	0.584	0.548	0.428	0.517	0.477	0.519	0.555	0.532
Rényi ($\alpha = 2$)	Max_0%	0.591	0.549	0.545	0.558	0.524	0.401	0.489	0.445	0.504	0.529	0.503
	Max_10%	**0.707**	0.549	0.553	0.575	0.548	0.401	0.503	0.428	0.526	0.534	0.528
	Max_100%	0.526	0.606	0.560	0.576	0.548	0.406	**0.518**	0.476	0.509	0.556	0.530
Rényi ($\alpha = \infty$)	Max_0%	0.623	0.559	0.559	0.567	0.534	0.386	0.494	0.439	0.461	0.554	0.460
	Max_10%	0.699	0.559	0.563	0.580	0.555	0.386	0.495	0.416	0.510	0.556	0.515
	Max_100%	0.545	0.587	0.564	0.577	0.550	0.394	0.517	0.473	0.506	0.577	0.530

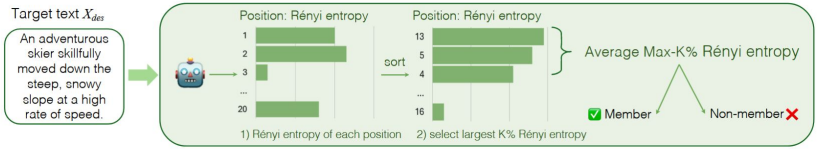
AUC results on VL-MIA under designed pipeline. On subset DALL-E.

- **Model performance differences** : LLaVA outperforme MiniGPT-4 and LLaVA Adapter, because LLaVA update more parameters while training. MiniGPT4 only updates the parameters of the image projection layer and LLaVA Adapter applies parameter-efficient fine-tuning approaches. The more parameter's updates, the easier the model memorize training data.
- **Dataset performance differences:** DALL-E is more challenging than Flickr. For DALLE-E, non- member data is genrated on the description of a member data, which makes the non member data and member data has a one-to-one correspondence and very similar, therefore, harder to distinguish.

Experiments - Text MIA

Table 3: **Text MIA.** AUC results on LLaVA.

Metric		VLLM Tuning 32	VLLM Tuning 64	LLM Pre-Training 32	LLM Pre-Training 64	LLM Pre-Training 128	LLM Pre-Training 256
Perplexity*		0.779	0.988	0.542	0.505	0.553	0.582
Perplexity/zlib*		0.609	0.986	0.56	0.537	0.581	0.603
Perplexity/lowercase*		**0.962**	0.977	0.493	0.518	0.503	0.583
Min_0% Prob*		0.522	0.522	0.455	0.451	0.425	0.448
Min_10% Prob*		0.461	0.883	0.468	0.487	0.526	0.534
Min_20% Prob*		0.603	0.980	0.505	0.498	0.549	0.562
Max_Prob_Gap		0.461	0.545	0.574	0.544	0.565	0.629
ModRényi*	$\alpha = 0.5$	0.809	0.979	0.557	0.500	0.536	0.567
	$\alpha = 1$	0.808	**0.993**	0.544	0.503	0.546	0.567
	$\alpha = 2$	0.779	0.963	0.559	0.497	0.529	0.560
Rényi ($\alpha = 0.5$)	Max_0%	0.506	0.514	0.541	0.515	0.489	0.571
	Max_10%	0.458	0.776	0.518	0.525	0.606	0.65
	Max_100%	0.564	0.835	0.555	0.531	0.6	0.631
Rényi ($\alpha = 1$)	Max_0%	0.552	0.579	0.566	0.571	0.603	0.668
	Max_10%	0.566	0.809	0.553	0.541	0.623	0.65
	Max_100%	0.554	0.750	0.544	0.523	0.588	0.621
Rényi ($\alpha = 2$)	Max_0%	0.589	0.625	0.594	0.606	0.659	0.657
	Max_10%	0.607	0.787	0.583	0.556	0.629	0.663
	Max_100%	0.553	0.709	0.592	0.576	0.568	0.649
Rényi ($\alpha = \infty$)	Max_0%	0.600	0.638	**0.607**	**0.615**	**0.688**	**0.669**
	Max_10%	0.618	0.763	0.586	0.548	0.627	0.667
	Max_100%	0.557	0.694	0.546	0.527	0.584	0.634



Two types of text MIA

- **Instruction-tuning text**
  - VL-MIA dataset (VLLM tuning)
  - ModRenyi outperform others
  - Member data are seen in recent tuning, the next token will convey more causal relations in the sequence remembered by the model, thus target-based method are better
- **Pretraining text**
  - WikiMIA dataset (LLM pre-training)
  - MaxReny outperform others.
  - During pretraining, model's parameters change a lot, target-free MIA methods, which use the whole distribution to compute statistics are more robust

Experiments-Image MIA on GPT4

Table 4: **Image MIA on GPT-4.**

Metric		VL-MIA/ DALL-E	VL-MIA/ Flickr
Perplexity/zlib*		0.807	0.520
Max_Prob_Gap		0.516	0.486
Rényi ($\alpha = 0.5$)	Max_0%	0.697	0.571
	Max_10%	0.749	0.604
	Max_100%	**0.815**	0.605
Rényi ($\alpha = 1$)	Max_0%	0.688	0.572
	Max_10%	0.747	0.591
	Max_100%	0.790	**0.630**
Rényi ($\alpha = 2$)	Max_0%	0.678	0.572
	Max_10%	0.723	0.574
	Max_100%	0.786	0.595
Rényi ($\alpha = \infty$)	Max_0%	0.685	0.561
	Max_10%	0.708	0.549
	Max_100%	0.781	0.583

Image MIA on closed- source GPT-4

API: GPT-4-vision-preview

Pipeline: random select 200 images, prompt GPT-4 to describe in 64 words, then apply MIA based on the generated descriptions.

Problem: GPT-4 only provide the top5 probability distribution.

Solution: Assume the size of entire token set is 32000 and the probability of the remain tokens are uniformly distributed.

Result: MaxRényi-K% (α = 0.5) can achieve an AUC of 0.815

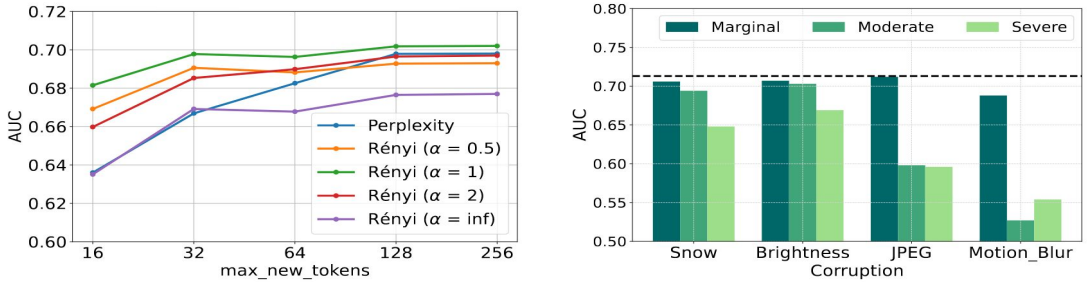
Ablation Study

Q1 : Does the length of description affect the image MIA performance?

yes, see figure2(a).

When the length of the description increases, the AUC of the MIA becomes higher and enters a plateau when max_new_tokens reaches 128.

Assumption: shorter text contains insufficient information.



(a) Ablation study on `max_new_tokens`. (b) `MaxRényi-K%` on corrupted images.

Figure 2: **Ablation study** (a) on `max_new_tokens` with `MaxRényi-10%`. Allowing VLLMs to generate longer descriptions can increase the AUC of "desp" slices, but we encounter a plateau when `max_new_tokens` equals 128. (b) on image MIAs against corrupted versions of VL-MIA/Flickr with `MaxRényi-K%` ($\alpha = 0.5$). Three levels of corruption are applied to the images: Marginal, Moderate, and Severe. The dotted line indicates the AUC on raw images without corruption.

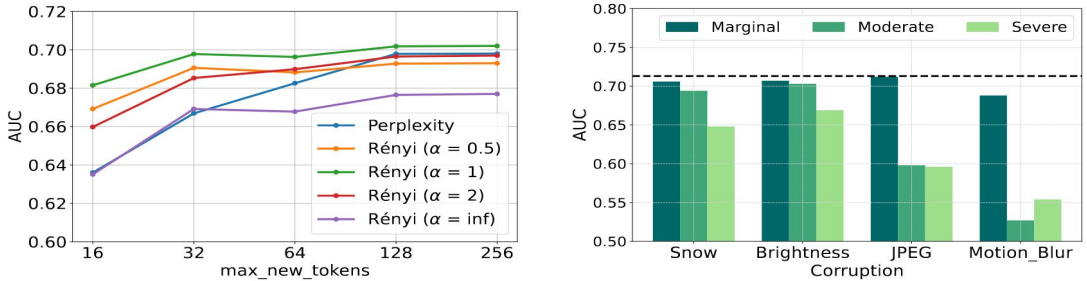
Ablation Study

Q2 : Can we still detect corrupted member images?

yes, see figure2(b).

Corrupted member images make MIAs more difficult, but can still be detected successfully.

Observation: reducing model quality (JPEG) or adding blur (Motion_Blur) degrade MIA performance more than changing the base parameter (Brightness) or overlaying texture (Snow)



(a) Ablation study on `max_new_tokens`. (b) `MaxRényi-K%` on corrupted images.

Figure 2: **Ablation study** (a) on `max_new_tokens` with `MaxRényi-10%`. Allowing VLLMs to generate longer descriptions can increase the AUC of "desp" slices, but we encounter a plateau when `max_new_tokens` equals 128. (b) on image MIAs against corrupted versions of VL-MIA/Flickr with `MaxRényi-K%` ($\alpha = 0.5$). Three levels of corruption are applied to the images: Marginal, Moderate, and Severe. The dotted line indicates the AUC on raw images without corruption.

Ablation Study

Q3 : Can we use different instructions?

Setting: three different instruction texts: “Describe this image concisely.”, “Please introduce this painting.”, and “Tell me about this image.”

Result: Yes, successfully detects member images on every instruction, which indicates robustness across different instruction texts.

Table 14: **Image MIA**. AUC results on VL-MIA/Flickr with LLaVA 1.5 when we change the instruction text. “Describe” indicates “Describe this image concisely.”, ”Please” indicates “Please introduce this painting.”, and “Tell” indicates “Tell me about this image.”.

Metric		Describe				Please				Tell			
		img	inst	desp	inst+desp	img	inst	desp	inst+desp	img	inst	desp	inst+desp
Perplexity*		N/A	0.378	0.667	0.559	N/A	0.379	0.671	0.549	N/A	0.362	0.662	0.530
Min_0% Prob*		N/A	0.357	0.651	0.357	N/A	0.421	0.629	0.421	N/A	0.341	0.622	0.341
Min_10% Prob*		N/A	0.357	0.669	0.390	N/A	0.421	0.656	0.414	N/A	0.341	0.646	0.367
Min_20% Prob*		N/A	0.374	0.670	0.370	N/A	0.411	0.661	0.381	N/A	0.356	0.650	0.360
Aug_KL		0.596	0.539	0.492	0.508	0.602	0.497	0.506	0.498	0.599	0.493	0.480	0.482
Max_Prob_Gap		0.577	0.601	0.650	0.650	0.577	0.462	0.652	0.649	0.577	0.543	0.661	0.663
ModRényi*	$\alpha = 0.5$	N/A	0.368	0.651	0.614	N/A	0.431	0.661	0.636	N/A	0.359	0.649	0.605
	$\alpha = 1$	N/A	0.359	0.659	0.502	N/A	0.396	0.664	0.503	N/A	0.355	0.653	0.474
	$\alpha = 2$	N/A	0.370	0.645	0.611	N/A	0.444	0.655	0.641	N/A	0.371	0.644	0.610
Rényi ($\alpha = 0.5$)	Max_0%	0.515	0.689	0.687	0.689	0.515	**0.683**	0.651	0.683	0.515	0.700	0.663	0.701
	Max_10%	0.557	0.689	0.691	0.719	0.557	**0.683**	0.663	0.666	0.557	0.700	0.672	0.723
	Max_100%	**0.702**	**0.726**	**0.713**	0.728	**0.702**	0.609	**0.700**	0.704	**0.702**	**0.709**	**0.708**	**0.726**
Rényi ($\alpha = 1$)	Max_0%	0.503	0.708	0.685	0.725	0.503	0.619	0.649	0.643	0.503	0.614	0.649	0.635
	Max_10%	0.623	0.708	0.698	**0.743**	0.623	0.619	0.670	0.707	0.623	0.614	0.671	0.699
	Max_100%	0.702	0.720	0.702	0.721	0.702	0.613	0.693	0.702	0.702	0.663	0.696	0.714
Rényi ($\alpha = 2$)	Max_0%	0.583	0.682	0.673	0.705	0.583	0.584	0.637	0.669	0.583	0.585	0.630	0.619
	Max_10%	0.621	0.682	0.685	0.725	0.621	0.584	0.660	0.670	0.621	0.585	0.655	0.672
	Max_100%	0.682	0.694	0.683	0.703	0.682	0.571	0.678	0.681	0.682	0.624	0.676	0.690
Rényi ($\alpha = \infty$)	Max_0%	0.588	0.646	0.651	0.674	0.588	0.636	0.629	0.666	0.588	0.578	0.622	0.603
	Max_10%	0.593	0.646	0.669	0.699	0.593	0.636	0.656	0.673	0.593	0.578	0.646	0.657
	Max_100%	0.669	0.673	0.667	0.687	0.669	0.539	0.671	0.667	0.669	0.608	0.662	0.675

Conclusion

- In this work, we take an initial step towards detecting training data in **VLLMs**.
- Specifically, we construct a **comprehensive dataset** to perform MIAs on both image and text modalities.
- Additionally, we uncover **a new pipeline** for conducting MIA on VLLMs cross-modally and propose a novel method based on Rényi entropy.

Contributions

- The first **benchmark** for detection of training data in VLLMs: VL-MIA.
- **Perform the first individual image or description MIAs on VLLMs in a cross-modal manner**. We demonstrate that we can perform image MIAs by computing statistics from the image or text slices of the VLLM's output logits.
- **Propose MaxReny-K% and its modified version ModReny-K%.** We demonstrate their effectiveness on open-source VLLMs and closed-source GPT-4. We achieve an AUC of 0.815 on GPT-4 in image MIAs.

Appendix

Table 5: Main notations.

Notation	Meaning
$\mathcal{V}$	The token set of a VLLM
$\mathbf{A}_{\text{image}}$	Membership inference attacker for image
$\mathbf{A}_{\text{des}}$	Membership inference attacker for text
θ	Model parameters
X_{des}, Z	Text description and image to be detected
X_{ins}	Text instruction, e.g., "Describe this image"
X'_{des}	Generated text description
X_{ept}	Empty instruction
Z_{ept}	All-black image
p	Some probability distribution
$p^{(i)}$	The next-token probability distribution at position i
p_j	The probability corresponding the j-th token in $\mathcal{V}$
$H_\alpha(p)$	Rényi entropy of order α
$\overline{H}_\alpha(p)$	Linearized Rényi entropy of order α
$\overline{H}_\alpha(p, y)$	Modified Rényi entropy of order α with target y