Your Large Vision-Language Model Only Needs A Few Attention Heads For Visual Grounding

PNU CSE AI Department

오지현

Overview

Large Vision-Language Model (LVLM)을 활용하여

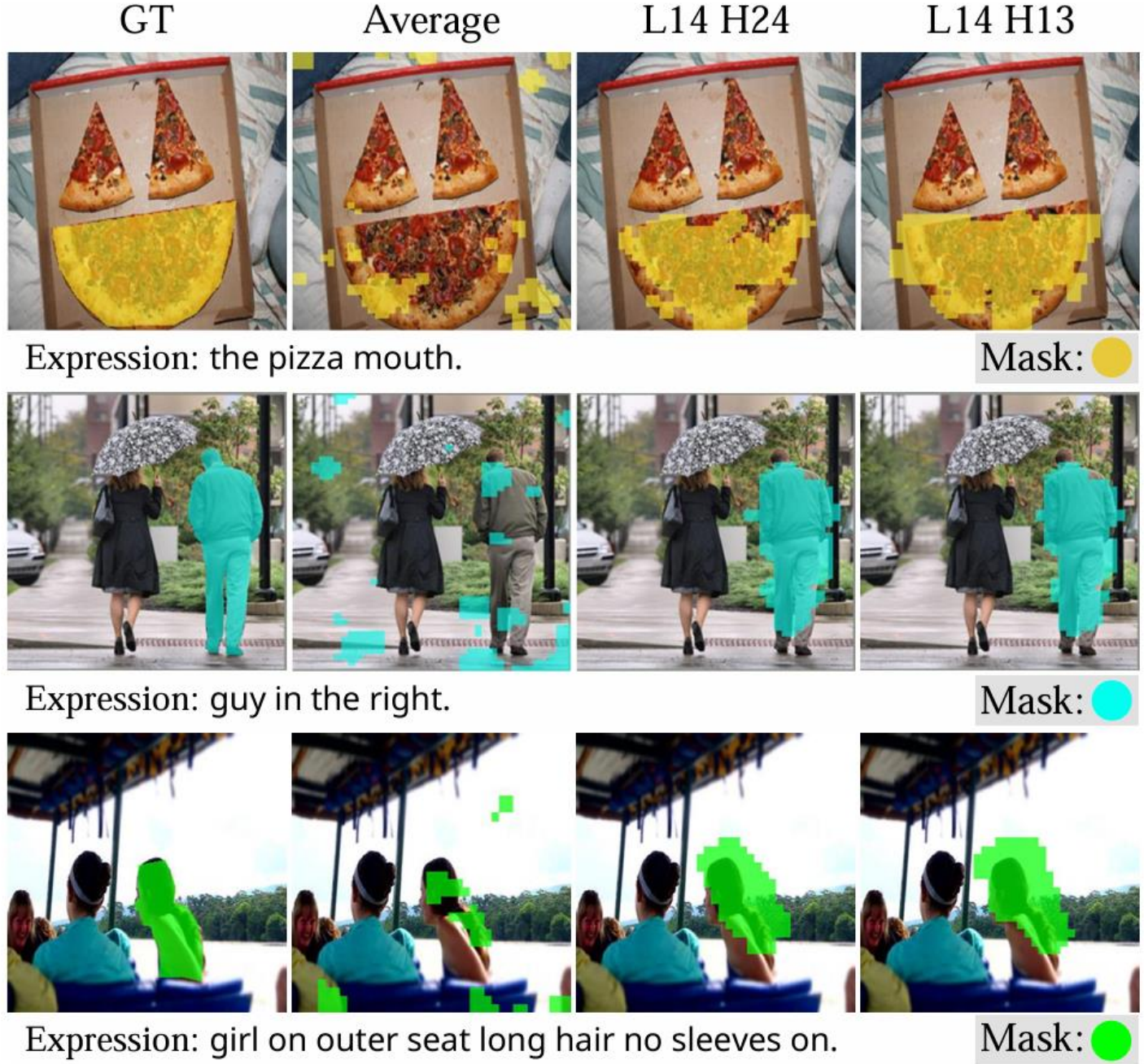
추가학습 없이 Visual Grounding을 수행할 수 있는 방법을 제안.

Overview

- Visual Grounding
  - 자연어 설명에 따라 이미지 내에서 특정 객체를 찾아내는 작업
  - 기존 대형 비전-언어 모델(LVLM)은 텍스트 생성에 최적화되어 있어, 해당 작업을 위해서는 추가적인 fine-tuning과 **모델 구조 수정이 필요함.**

LVLM이 이미 이미지의 특정 영역에 대한 질문을 잘 이해하고
그에 맞는 답을 생성하는데, 이 과정을 통해 이미지 내에서 텍스트와 관련된
지역을 정확하게 찾는 능력을 이미 특정 매커니즘에서 가지고 있다는 점을 발견.
(Localization heads)

추가 학습 없이 이러한 능력을 바로 활용할 수 있을까?


GT
Average
L14 H24
L14 H13
Expression: the pizza mouth.
Mask:
Expression: guy in the right.
Mask:
Expression: girl on outer seat long hair no sleeves on.
Mask:

Related work

Visual Grounding

- 텍스트로 설명된 객체를 이미지에서 식별하는 작업, CV와 자연어 처리 기술의 결합을 필요로 함.

- Referring Expression Comprehension (REC) : bounding box 생성
- Referring Expression Segmentation (RES) : segmentation mask 생성

"left white horse one next to black."

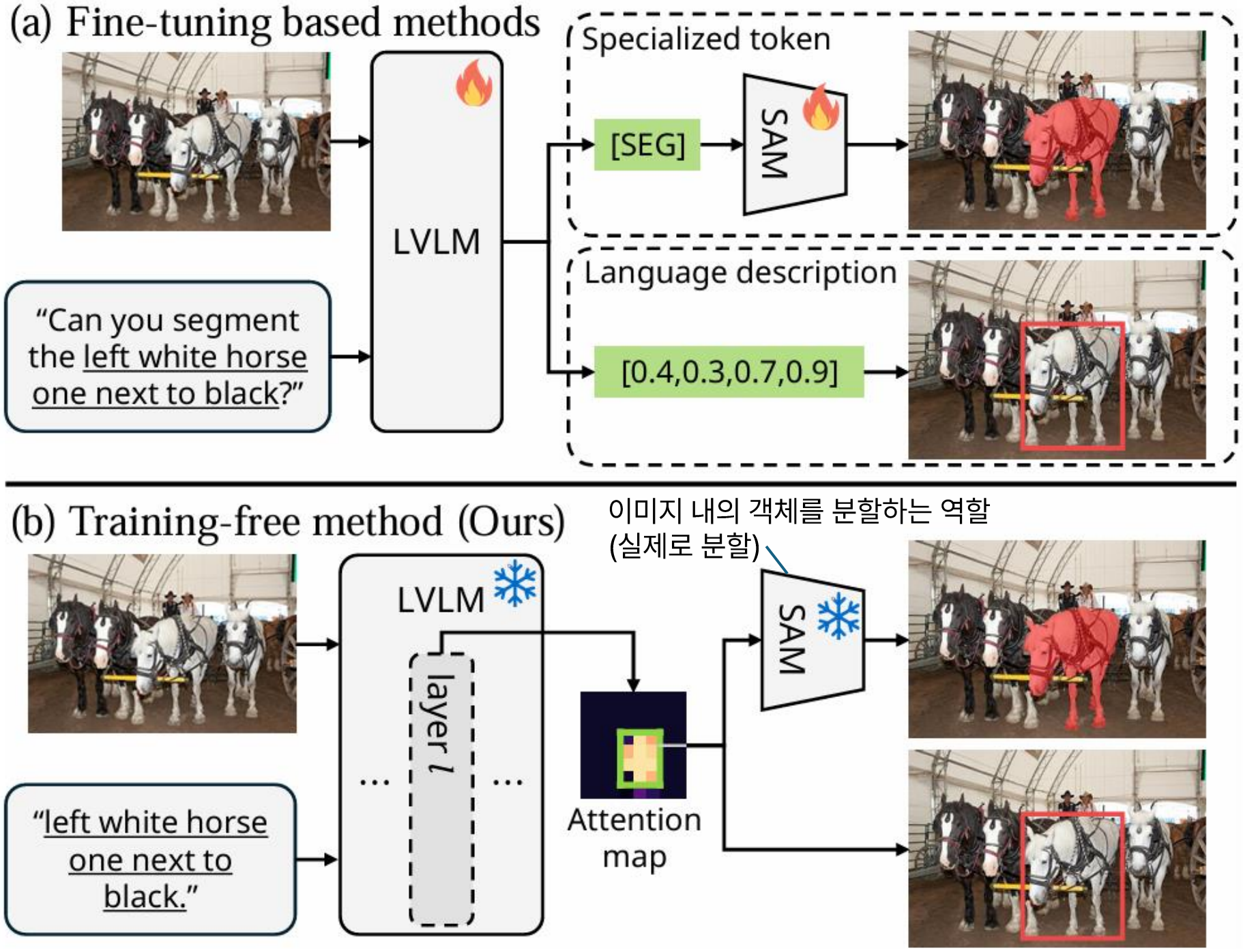
Visual Grounding Model

RES

REC

- LISA (Language-Instructed Segmentation Assistant)
  - 기존 MLLM(Multi-modal Large Language Model)에 binary segmentation mask 생성 능력을 부여

  - 모델의 응답에 **<SEG>라는 특수 토큰**이 포함되도록 유도
    - 별도의 디코더를 학습하므로 훈련 비용이 커짐.
  - 이 토큰의 hidden embedding을 기반으로 Segmentation Decoder에서 마스크 생성
  - Ground Truth와 비교 -> 파인튜닝 학습


(a) Fine-tuning based methods
Specialized token
[SEG]
SAM
LVLM
"Can you segment the left white horse one next to black?"
Language description
[0.4,0.3,0.7,0.9]
(b) Training-free method (Ours)
이미지 내의 객체를 분할하는 역할
(실제로 분할)
LVLM
layer l
Attention map
SAM
"left white horse one next to black."

Proposed method

1. Investigation of Image-Text Interaction

Analyze how attention heads in the LVLM decoder capture relations between image and text tokens

2. Criteria to Find Localization Heads

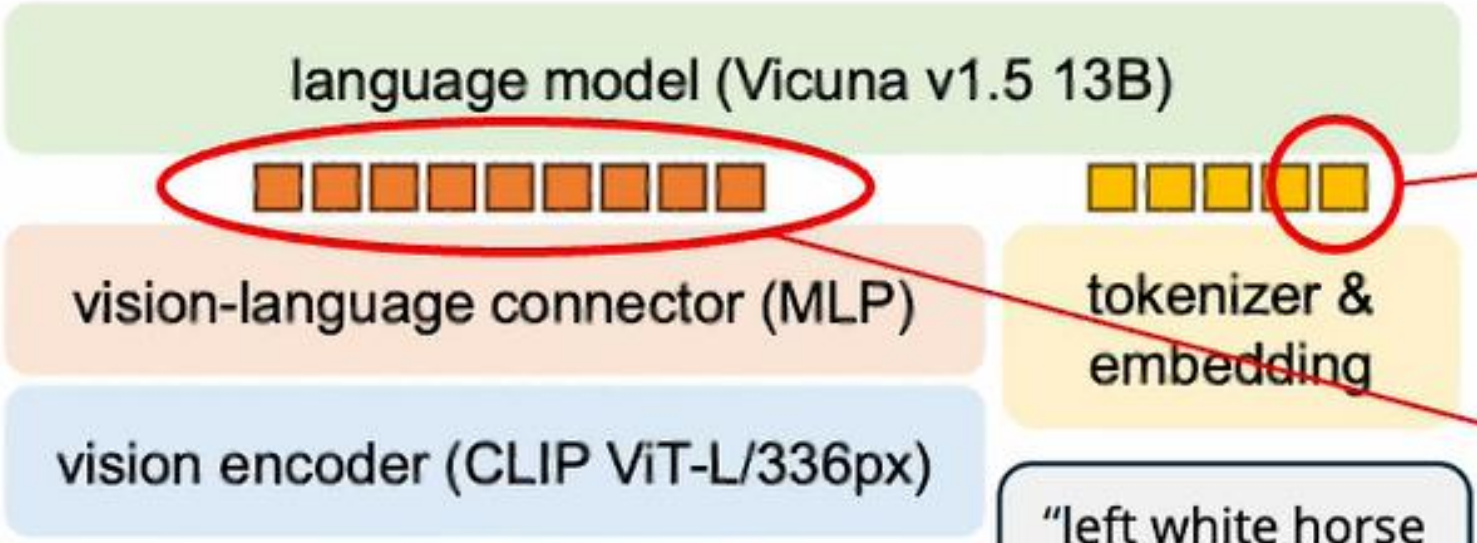
Measure the degree of “focusing on image” and “focusing on a localized region” for each attention head

3. Finding Localization Heads via Criteria

Select localization heads based on the quantitative criteria

4. Visual Grounding with Localization Heads

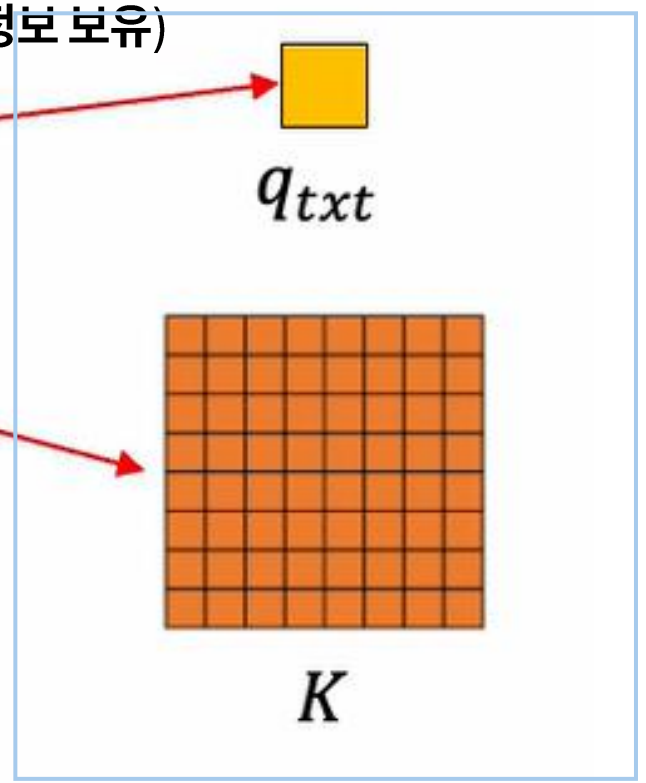
Perform visual grounding using the selected localization heads



Visual encoder -> patch로 분해
각 이미지 토큰이 K & V 생성

Tokenizer & embedding

마지막 입력 텍스트 토큰의
query vector (전체 문장의 맥락을 **캡슐화**, 이전 토큰과의 **어텐션 연산을 통해 많은 문맥 정보 보유**)



이미지 토큰의 key value들의
군집 K를 Key vectors로 활용

1. Investigation of Image – Text Interaction

텍스트의 query
(마지막 토큰)

이미지 패치의
key vector들

$$\boldsymbol{a}^{\ell,h} = \text{softmax}\left(\frac{\boldsymbol{q}_{\text{txt}}\boldsymbol{K}^{\top}}{\sqrt{d_h}}\right) \in \mathbb{R}^{P^2+L},$$

2. Criteria to Find Localization Heads

- 어텐션 헤드 별 “이미지에 집중하는 정도 " 를 정량화
  - 여러 어텐션 헤드 중 **이미지에 많이 집중하는 헤드**를 1차적 선별
  - 텍스트를 기반으로 이미지 전체에 골고루 vs 강하게 관심 갖는지 파악

2. Criteria to Find Localization Heads

- 사용 데이터
  - refCOCO training dataset 무작위 1000개의 이미지-텍스트 쌍 사용

- 계산 대상
  - 각 레이어 ℓ, 어텐션 헤드 h에 대해
  - 텍스트 query 와 이미지 토큰 key K 사이의 attention weight a 계산
  - 이 벡터는 크기 P^2 (=이미지 토큰 수)

2. Criteria to Find Localization Heads

$$S_{\text{img}}^{\ell,h} = \sum_{i=1}^{P^2} \boldsymbol{a}^{\ell,h}[i]$$

이미지의 모든 패치에 대해 어텐션 weight 총합하여
해당 head의 이미지 집중 정도를 수치화

전체 이미지에 대해 많은 Simg 을 가진 헤드 = 이미지 전체를 강하게 주의를 줌.
이를 기반으로 localization head 필터링 가능

"Can you segment the left white horse one next to black?"

Vision encoder로 들어가면
일정한 크기의 패치 = image token들로 나뉨.

텍스트 입력을 tokenizer를 통해 분해
마지막 토큰의 임베딩을 통해 query vector 생성

Image token – Transformer 구조 내에서
Key (**K**), Value (V) 벡터를 가짐.

$$a^{\ell,h} = \text{softmax}\left(\frac{q_{\text{txt}} K^{\top}}{\sqrt{d_h}}\right) \in \mathbb{R}^{P^2+L},$$

얼마나 주목
(attention) 했는가?

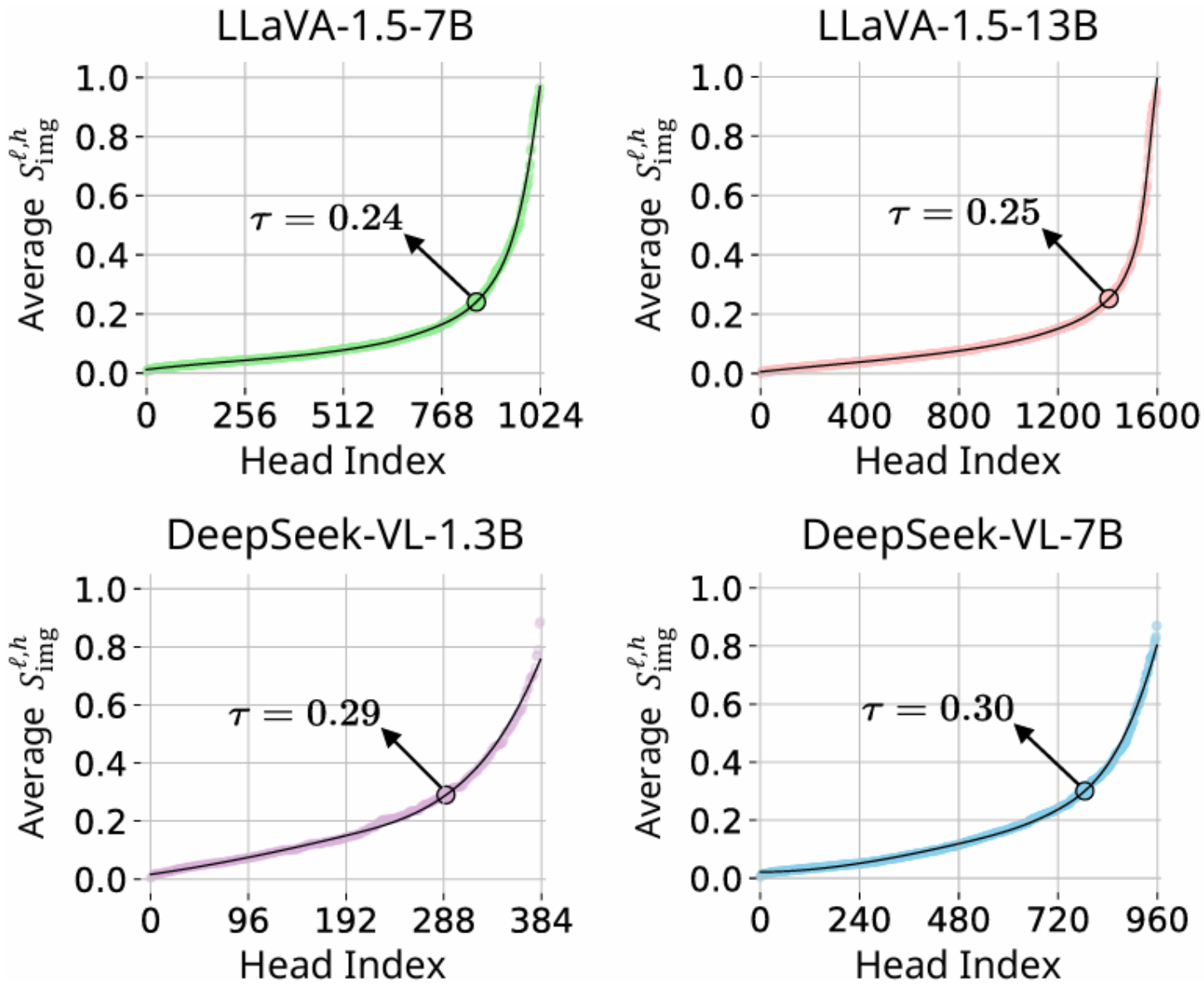
$$a^{\ell,h} = \text{softmax}\left(\frac{q_{\text{txt}} K^{\top}}{\sqrt{d_h}}\right) \in \mathbb{R}^{P^2+L},$$

얼마나 주목
(attention) 했는가?

$$S^{\ell,h}_{\text{img}} = \sum_{i=1}^{P^2} a^{\ell,h}[i]$$

어텐션 헤드가 텍스트에 반응하여 전체 이미지에 얼마나 집중했는가? 를 수치화.

[텍스트 → 이미지] 에만 쏟아지는 attention의 합만 봄.
합이 클수록 텍스트가 이미지 쪽에 더 많은 관심을 기울임. = 적합!



최대 곡률 지점 이후의 head 만 선별

Attention이 이미지에서 얼마나 국소적으로 모여있는지 수치화

- Attention이 여러 군데로 흩어져 있으면 안 좋은 head임.

-> 특정 위치로 집중된 정도를 수치로 나타내는 것이 **공간 엔트로피 H**

H가 작으면 한 곳에 집중, H가 크면 여러 군데 흩어짐

Attention이 이미지에서 얼마나 국소적으로 모여있는지 수치화

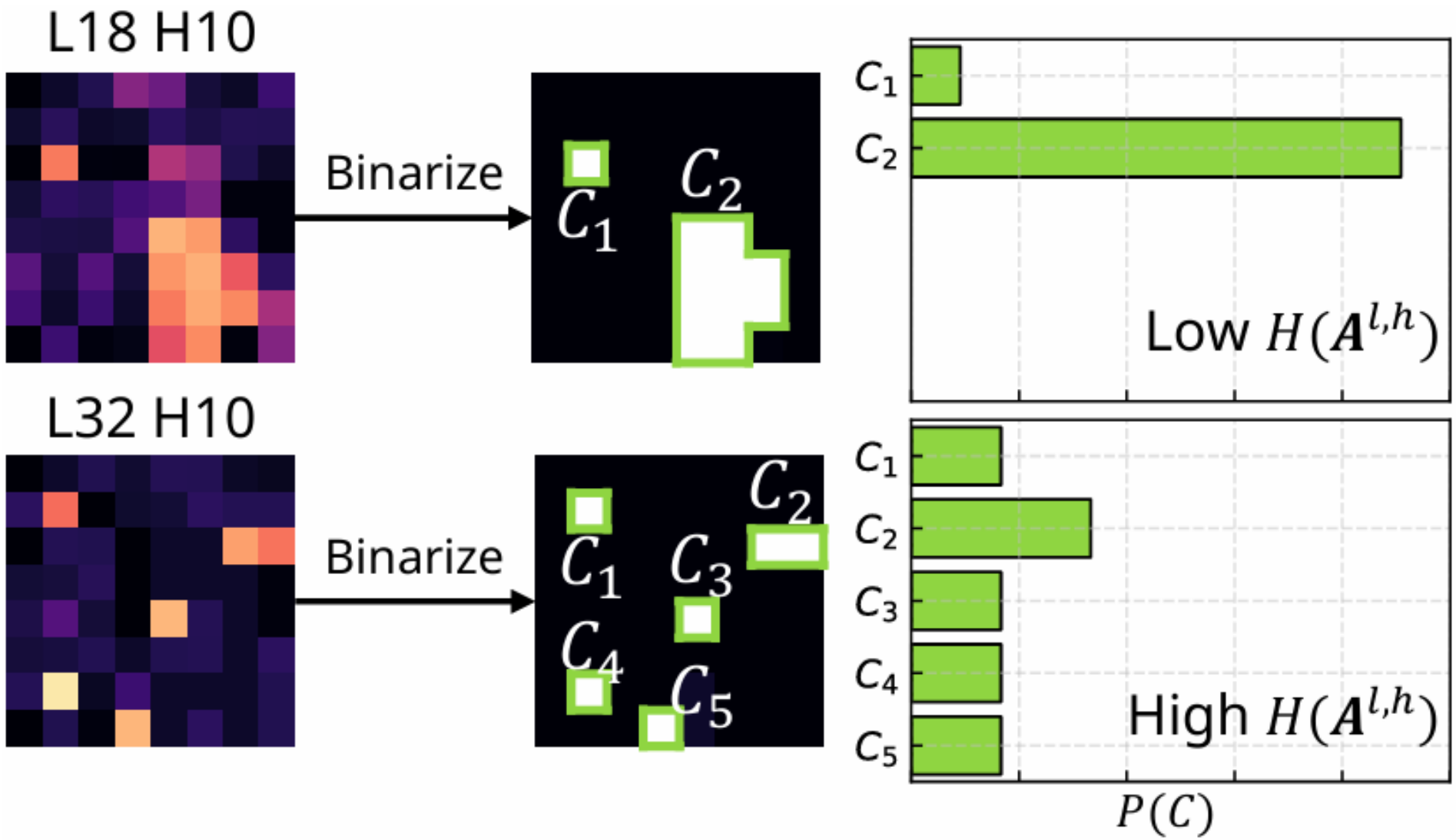
1. **Attention map -> Binarize**
   일정 threshold 이상만 1, 나머지는 0 -> 높은 attention 값이 하나의 blob

2. **Find Connected Component**
   서로 연결된 픽셀 그룹들 C1, C2, ..를 찾음 -> 각 그룹은 하나의 관심 영역

3. **각 컴포넌트의 크기 비율 -> 분포 P(Ci)**
   Ci의 전체 픽셀 수를 합친 뒤, 비율 계산
   분포를 바탕으로 엔트로피 계산

$$P(C_i) = |C_i| / \sum_{i=1}^{N} |C_i|$$

$$H(\boldsymbol{A}^{\ell,h}) = -\sum_{i=1}^{N} P(C_i) \log P(C_i),$$


L18 H10
Binarize
C_1
C_2
L32 H10
Binarize
C_1
C_2
C_3
C_4
C_5
C_1
C_2
Low $H(\boldsymbol{A}^{l,h})$
C_1
C_2
C_3
C_4
C_5
High $H(\boldsymbol{A}^{l,h})$
$P(C)$

3. Finding Localization Heads via Criteria

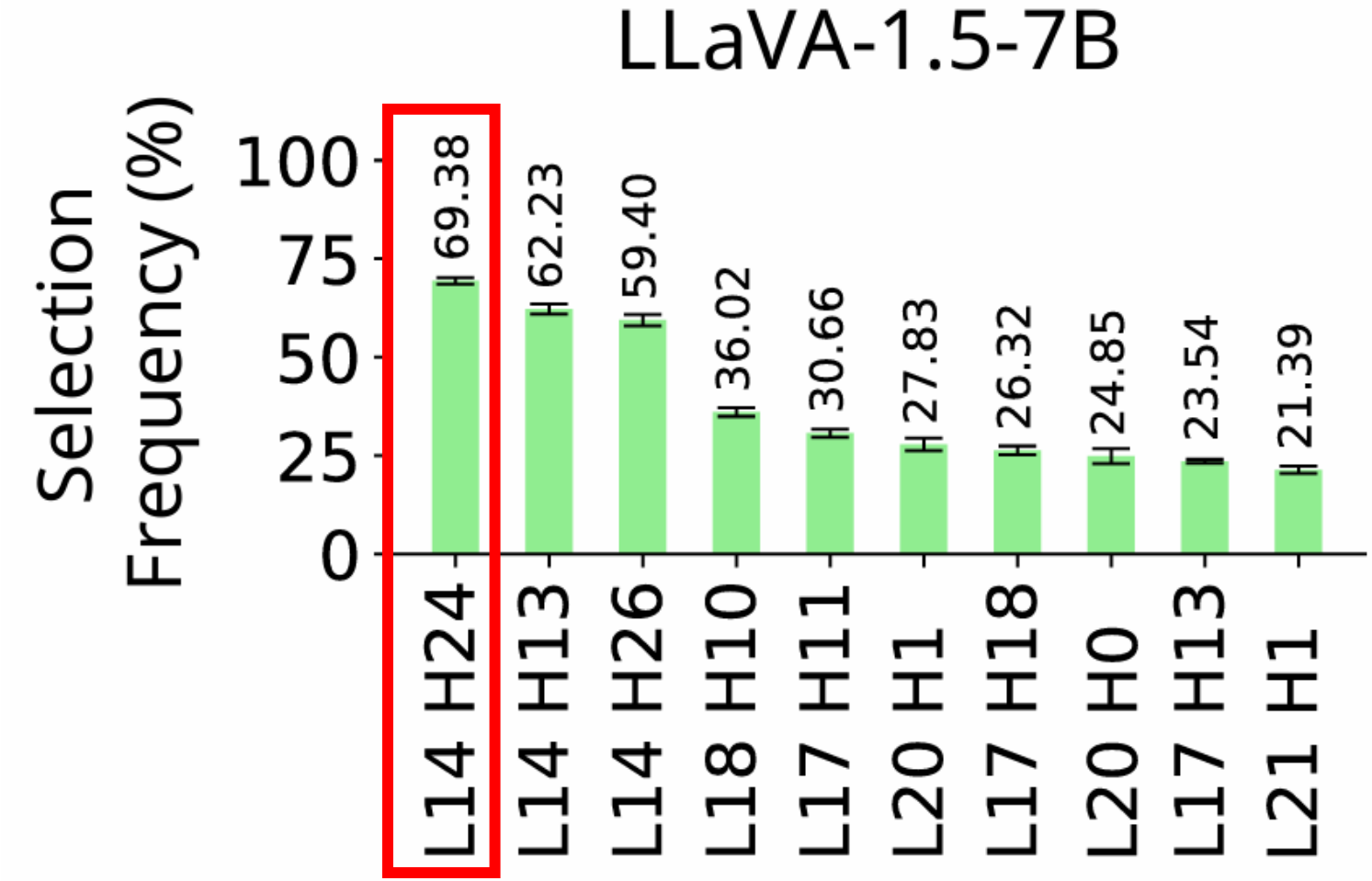
Selection Frequency 사용

- 같은 샘플링 과정을 여러 번 반복
- 어떤 head가 spatial entropy 기준으로 가장 집중도가 높았던 top10에 얼마나 자주 등장했는지를 퍼센트로 나타낸 값

3. Finding Localization Heads via Criteria

1. refCOCO trainset에서 1000개의 sample 랜덤 추출
2. Maximum curvature 이상인 head만 대상.
3. 각 샘플에 대해
   - 해당 head의 spatial entropy를 계산
   - 가장 낮은 entropy를 보인 10개의 head를 기록
4. 이 과정을 5회 반복
5. 각 헤드가 얼마나 자주 lowest 10에 포함되었는지 계산

3. Finding Localization Heads via Criteria



5번의 selection frequency 계산 과정에서 **14번 layer, 24번 head가 69.38%** 의 빈도로 spatial entropy 기준 lowest 10에 들었음.
일관성있게 군집화된 head임.

4. Visual Grounding with Localization Heads

1. Localization Attention Maps -> Pseudo Mask 생성
    1. 선택된 localization heads (예: L14-H24)의 attention map 추출
    2. 가우시안 스무딩 적용
    3. 일정 threshold 기준으로 이진화하여 pseudo segmentation mask 생성

    => SAM (Segment Anything Model)에 이 마스크 입력하여 정확한 segmentation 결과 출력

4. Visual Grounding with Localization Heads

2. Attention map -> Bounding box 생성
    - 이전 attention map에서 값이 1인 픽셀 좌표 추출
    - 이 점들을 감싸는 Convex Hull 계산 -> 최소한의 블록 형태
    - Bounding box로 변환하여 object 위치를 정함
    - box를 프롬프트로 사용하여 SAM에 다시 입력

    -> 최종 마스크 생성

Experiments

- Fine-tuning based methods
  - 기존 모델을 task-specific하게, 성능 높지만 학습 비용이 큼.
- Fine-tuning based methods w/ LVLMs
  - LVLM 위에 fine-tuning한 방식.
- Training-free methods
  - 추가 학습 없이 동작
- Training-free methods w/ LVLMs (Ours)
  - LVLM의 attention만 활용해 학습 없이 grounding 수행

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
Fine-tuning based methods								
MDETR [21]	86.8	89.6	81.4	79.5	84.1	70.6	81.6	80.9
SeqTR [77]	87.0	90.2	83.6	78.7	84.5	71.9	82.7	83.4
G-DINO [37]	89.2	91.9	86.0	81.1	87.4	74.7	84.2	84.9
ONE-PEACE [60]	92.6	94.2	89.3	88.8	92.2	83.2	89.2	89.3
UNINEXT [31]	92.6	94.3	91.5	85.2	89.6	79.8	88.7	89.4
Fine-tuning based methods w/ LVLMs								
Shikra-7B [6]	87.0	90.6	80.2	81.6	87.4	72.1	82.3	82.2
Ferret-7B [68]	87.5	91.4	82.5	80.8	87.4	73.1	83.9	84.8
Shikra-13B [6]	87.8	91.1	81.8	82.9	87.8	74.4	82.6	83.2
Ferret-13B [68]	89.5	92.4	84.4	82.8	88.1	75.2	85.8	86.3
CogVLM-17B [61]	92.8	94.8	89.0	88.7	92.9	83.4	89.8	90.8
Training-free methods								
ReCLIP [52]	45.8	46.1	47.1	47.9	50.1	45.1	59.3	59.0
Han et al. [15]	49.4	47.8	51.7	48.9	50.0	46.9	61.0	60.0
GroundVLP [48]	65.0	73.5	55.0	68.8	78.1	57.3	74.7	75.0
Training-free methods w/ LVLMs (Ours)								
DeepSeek-VL-1.3B	73.2	77.7	70.7	62.0	66.7	57.1	65.2	69.3
Mini-Gemini-2B	74.0	77.5	71.1	62.5	67.8	59.3	65.1	69.3
InternVL-6B	85.2	86.4	78.5	78.0	83.3	71.9	81.1	80.5
Yi-VL-6B	85.1	86.8	78.4	78.9	84.2	72.2	80.5	80.9
DeepSeek-VL-7B	85.3	87.2	81.0	77.8	83.9	73.5	81.1	82.8
ShareGPT4V-7B	86.1	87.1	80.5	79.7	86.2	71.3	82.4	82.9
LLaVA-7B	80.3	83.5	77.4	74.5	80.2	69.3	77.5	77.1
LLaVA-1.5-7B	86.5	89.8	80.2	80.1	86.3	71.9	82.3	83.0
LLaVA-13B	82.8	85.3	79.8	79.3	82.4	73.0	79.8	79.5
LLaVA-1.5-13B	87.2	90.0	83.3	82.7	88.5	74.0	84.3	85.5

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
Fine-tuning based methods								
LAVT [67]	72.7	75.8	68.8	62.1	68.4	55.1	61.2	62.1
ReLA [34]	73.8	76.5	70.2	66.0	71.0	57.7	65.0	66.0
UniRef++ [63]	79.1	82.1	77.5	68.4	74.0	61.5	71.4	72.8
UNINEXT [31]	82.2	83.4	81.3	72.5	76.4	66.2	74.4	76.4
Fine-tuning based methods w/ LVLMs								
LISA-7B [26]	74.1	76.5	71.1	62.4	67.4	56.5	66.4	68.5
GSVA-7B [65]	76.4	77.4	72.8	64.5	67.7	58.6	71.1	72.0
LISA-13B [65]	73.4	76.2	69.5	62.3	66.6	56.3	68.2	68.5
GSVA-13B [65]	77.7	79.9	74.2	68.0	71.5	61.5	73.2	73.9
GLaMM [45]	79.5	83.2	76.9	75.9	78.7	68.8	76.8	78.4
PSALM [74]	83.6	84.7	81.6	72.9	75.5	70.1	73.8	74.4
Training-free methods								
Yu et al. [71]	24.9	23.6	24.7	26.2	24.9	25.8	31.1	31.0
TAS [53]	29.5	30.3	28.2	33.2	38.8	28.0	35.8	36.2
Ref-Diff [40]	35.2	37.4	34.5	35.6	38.7	31.4	38.6	37.5
Training-free methods w/ LVLMs (Ours)								
DeepSeek-VL-1.3B	56.3	57.0	52.7	51.2	55.5	49.2	52.3	55.8
Mini-Gemini-2B	59.8	60.3	55.5	56.3	59.9	51.8	55.1	60.3
InternVL-6B	62.1	65.8	60.9	62.2	65.5	55.5	63.5	65.4
Yi-VL-6B	62.5	65.8	60.7	61.0	65.3	56.0	64.0	67.0
DeepSeek-VL-7B	73.9	76.6	70.7	63.1	66.1	56.5	64.0	68.9
ShareGPT4V-7B	73.5	76.7	70.1	59.4	63.8	55.9	60.7	65.1
LLaVA-7B	65.4	66.2	61.1	59.9	63.2	52.7	59.7	63.3
LLaVA-1.5-7B	74.2	76.5	70.4	62.5	65.2	56.0	64.2	68.1
LLaVA-13B	66.8	68.0	63.7	62.3	66.9	57.3	65.0	68.2
LLaVA-1.5-13B	76.1	78.9	72.8	64.1	67.1	57.3	67.7	69.0

Conclusion

- Training 없이 Visual Grounding 수행 가능
  - LVLM의 내부 attention 구조 분석만으로 추가 학습 없이 visual grounding 가능하게 하는 새로운 방법 제안 및 월등한 성능 개선.

- Localization Head 개념 최초 도입 및 식별 방법 제안
  - 일부 attention head가 텍스트-이미지 매핑에서 특히 강력한 위치 추정 능력을 보인다는 사실 입증, 이를 localization head로 정의함.
  - Head들을 선별할 수 있는 2단계 criteria 도입 (Attention Sum + Spatial Entropy)

Thank You