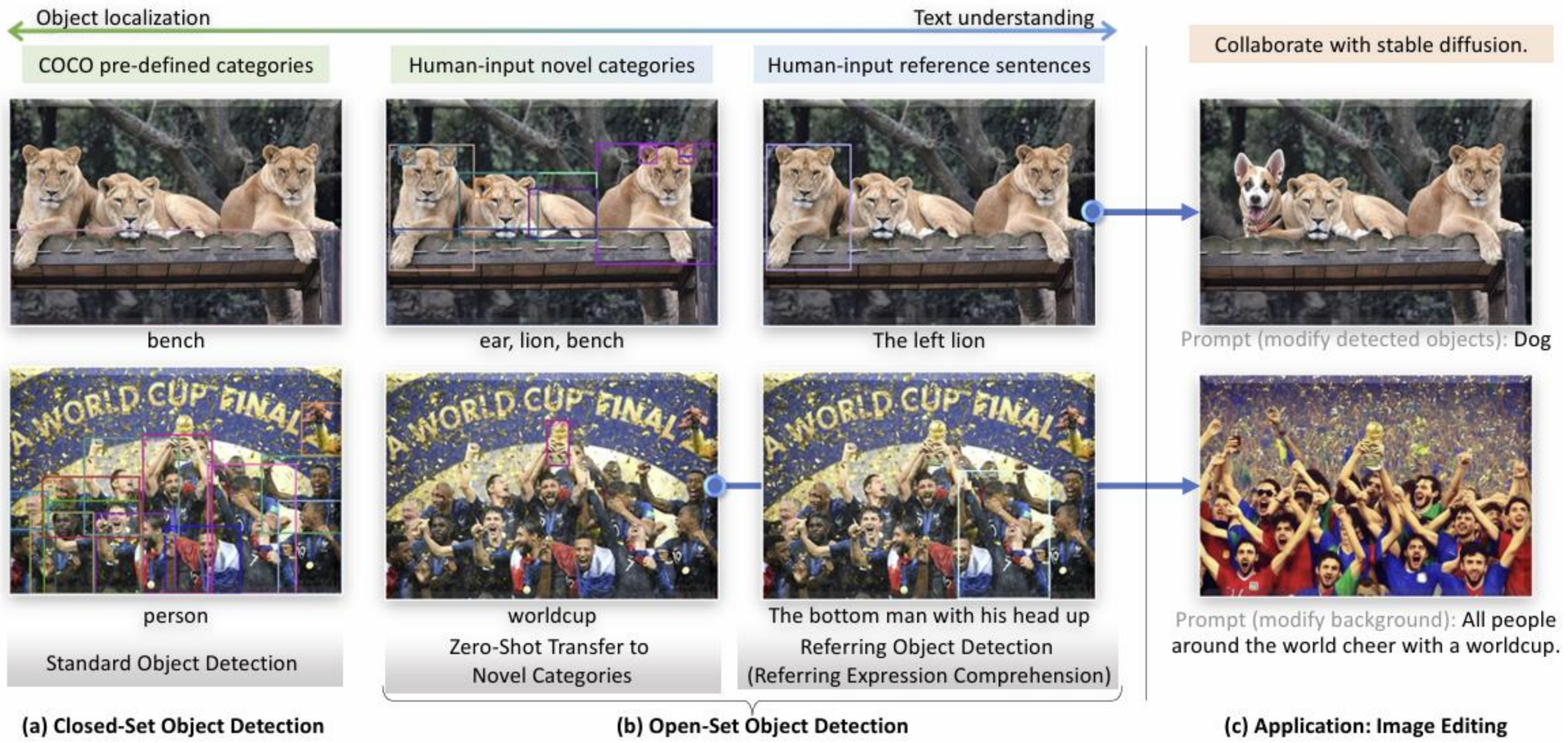
Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection

Shilong Liu et al., IDEA Research & Tsinghua & MSR Asia (CVPR 2023)

Closed-Set vs. Open-Set



Closed-Set vs. Open-Set

Text를 Image의 특정 영역과 정확하게 연결하는 게 핵심

Closed-Set Object Detection

미리 정의된 또는 고정된 카테고리 목록에 대해서만 물체 탐지 수행.

학습 데이터에 존재하지 않았던 새로운 카테고리의 물체는 탐지 불가능.

Open-Set Object Detection

사용자가 Language Input으로 입력하는 모든 카테고리를 탐지.

구체적이고 새로운 지시어 수용

기존 연구들의 딜레마



어떻게 강력한 시각적 특징 + 언어 시각 연결 능력 을
모두 가질 수 있을까?

기존 연구들의 딜레마

Vision-Language Models (예: GLIP, VL-DETR)

대규모 텍스트-이미지 쌍으로 학습하여 언어-시각 연결(Grounding) 능력 우수

시각적 특징 추출(Visual Feature Representation) 능력이 상대적으로 부족.
(ImageNet 등에서 Pre-training된 모델보다 성능이 낮음)

Self-Supervised Models (예: DINO, DINOv2)

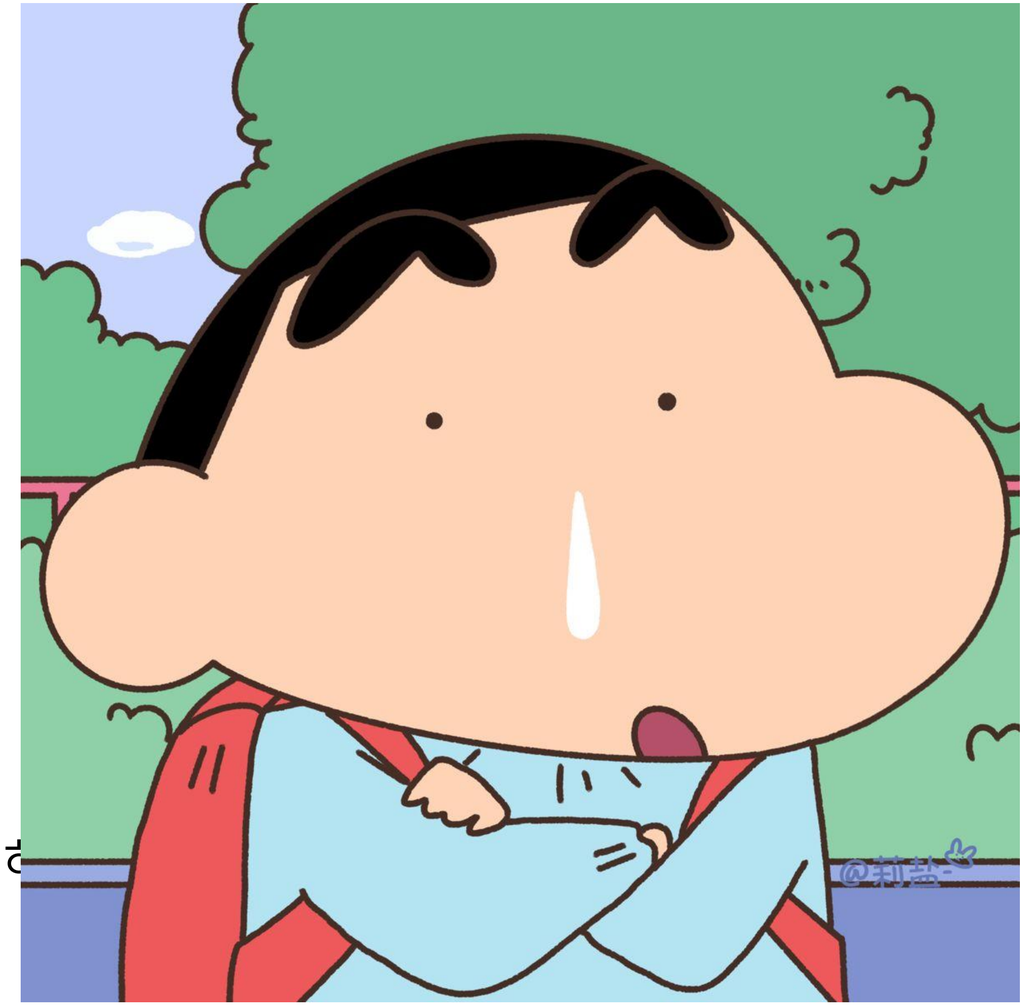
레이블 없는 데이터로 학습하여 매우 풍부하고 강력한 시각적 특징을 학습함. (Closed-Set SOTA)

'언어'와 연결되어 있지 않아 텍스트 지시어를 이해하고 Grounding하는 능력 부재.

기존 연구들의 딜레마



'언어'와 연결되어 있지 않아 텍스트 지시어를 이해



The Marriage

DINO 계열

GLIP 계열

강력한 시각 특징 학습 아키텍처 (e.g., DETR-like, DINO)

Closed-Set Detection SOTA 달성. 강력한 시각적 표현력.

언어-시각 연결 학습 방식 (Grounded Pre-training)

Open-Set Detection SOTA 달성. 뛰어난 Language Grounding 능력

⇒ **Grounding DINO**
Dual-Encoder, Cross-Modality Decoder 구조
텍스트와 이미지를 Encoder부터 Decoder까지 Deep Fusion하여 성능 극대화.

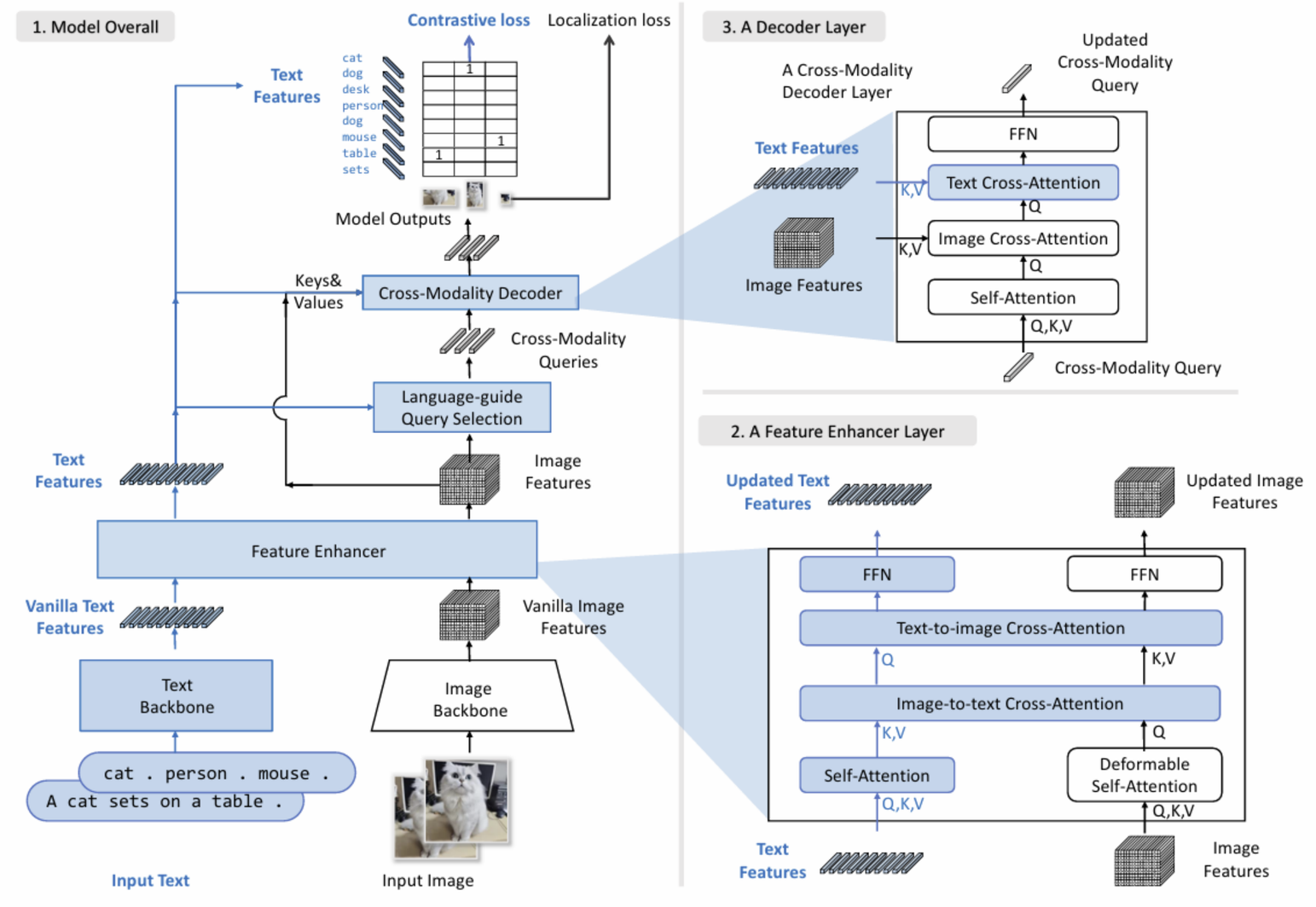
Related Work

GLIP (Grounded Language-Image Pre-training)

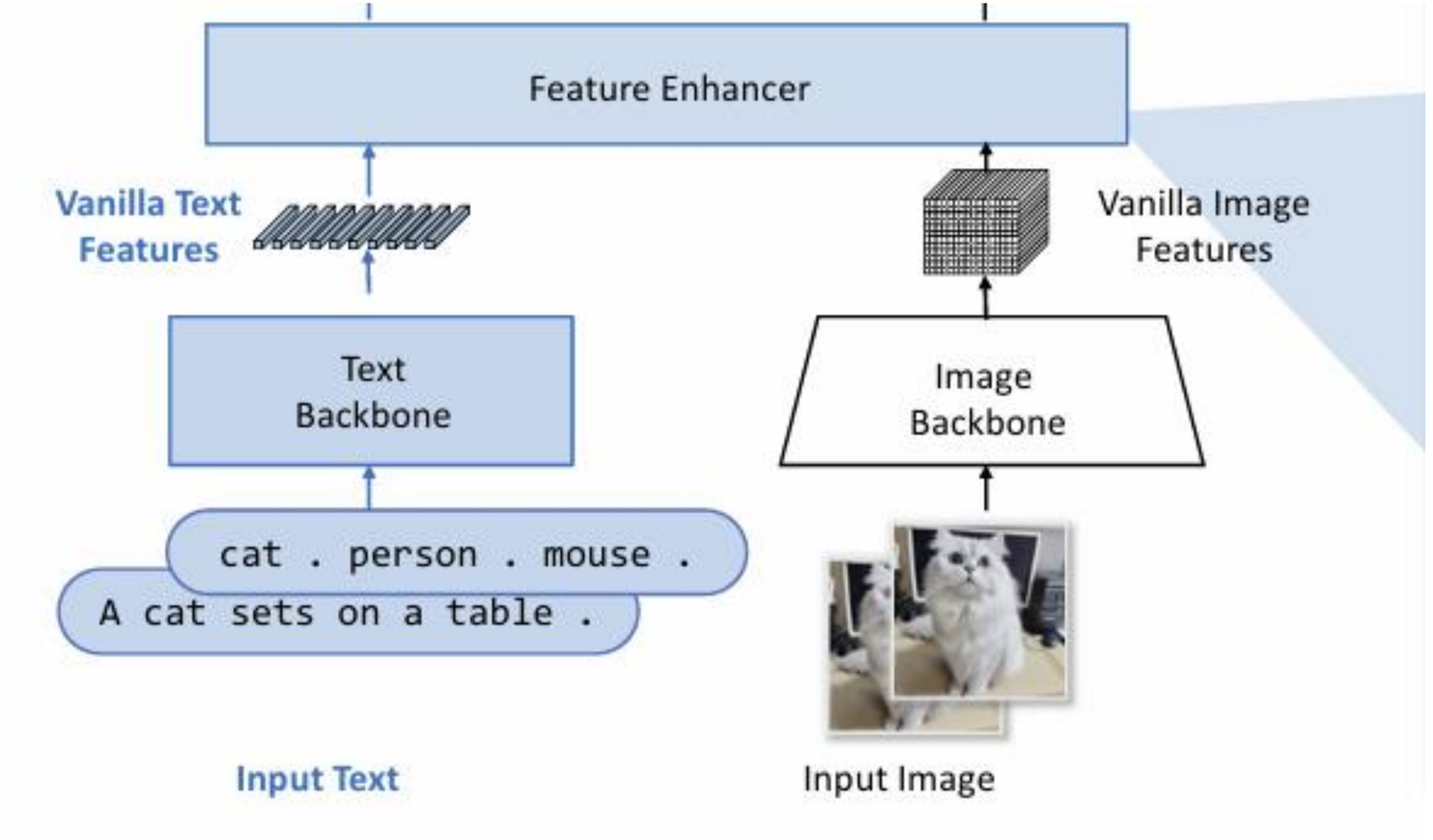
Transformer-based Detectors (DETR Family): DETR, DAB-DETR, DN-DETR

DINO (DETR with Improved denoising)

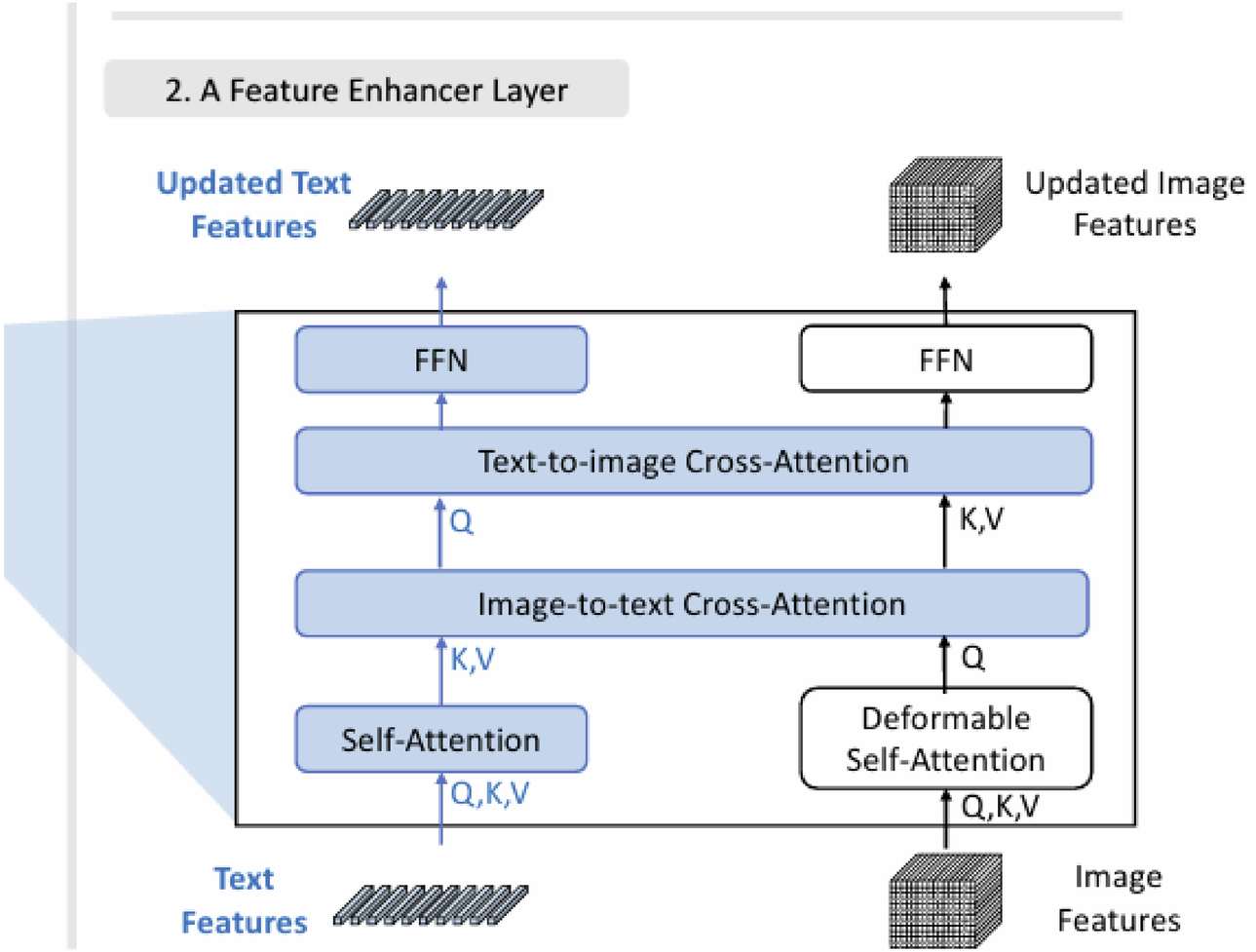
Overall Architecture



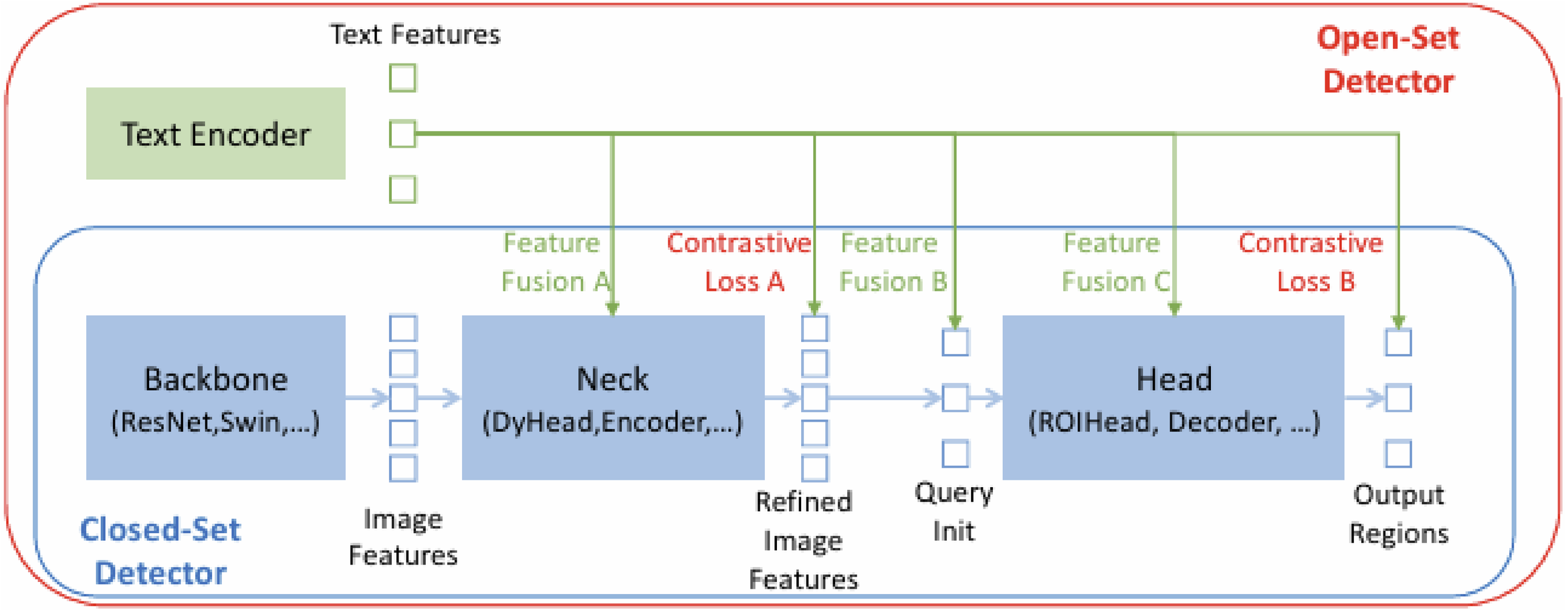
Dual Encoder



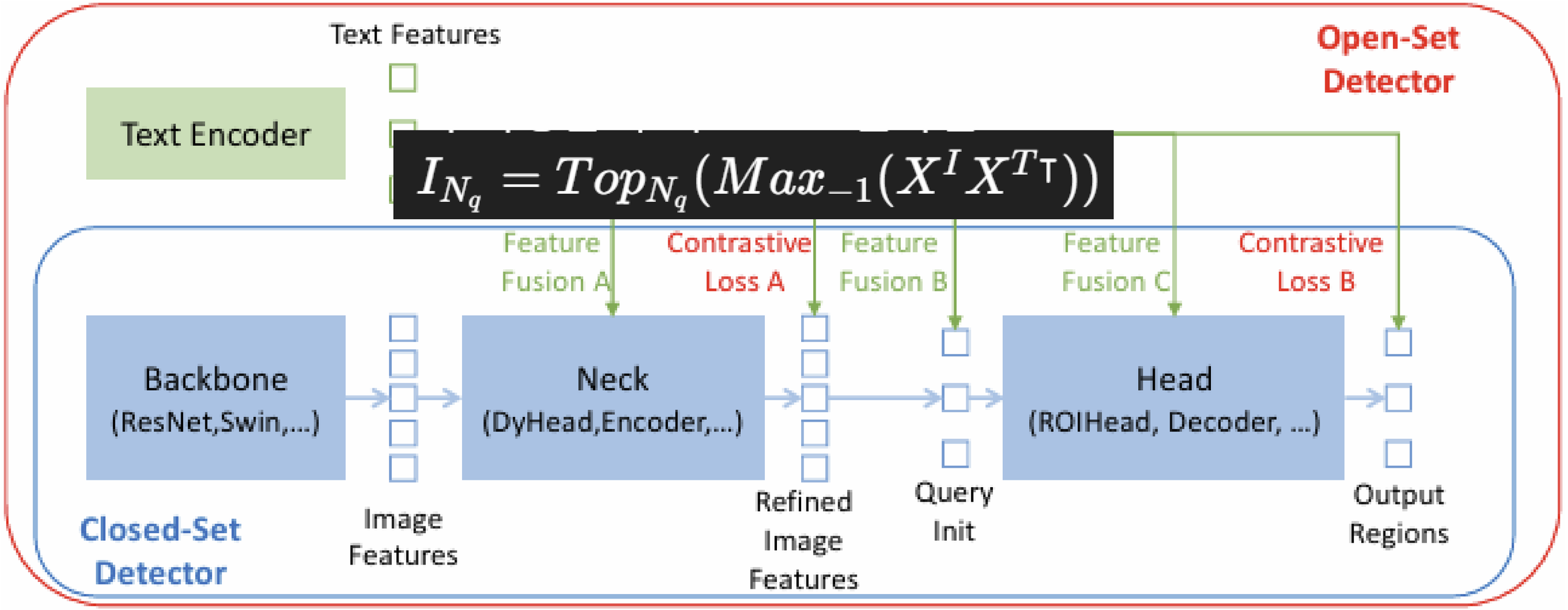
Feature Enhancer



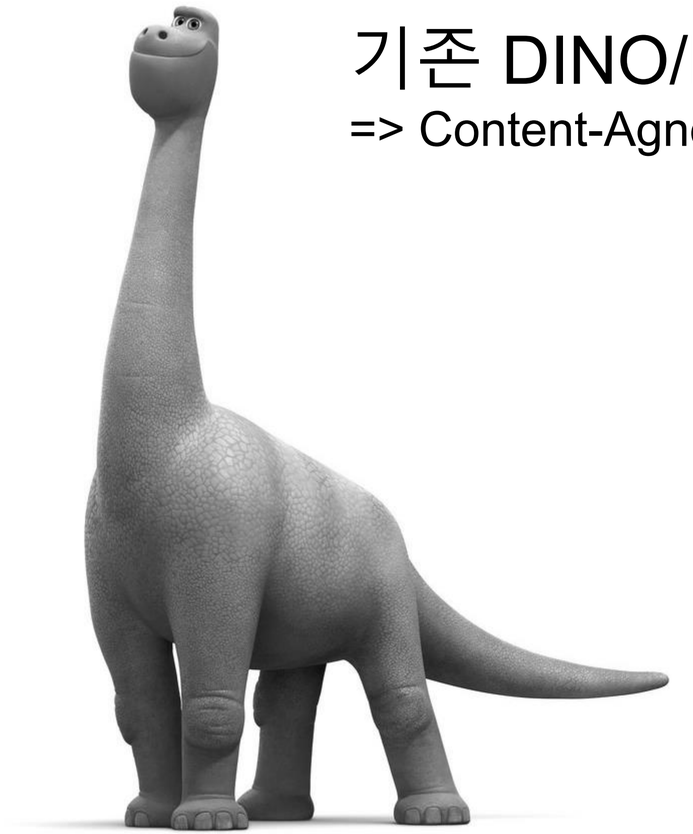
Language-Guided Query Selection



Language-Guided Query Selection



Language-Guided Query Selection



기존 DINO/DETR

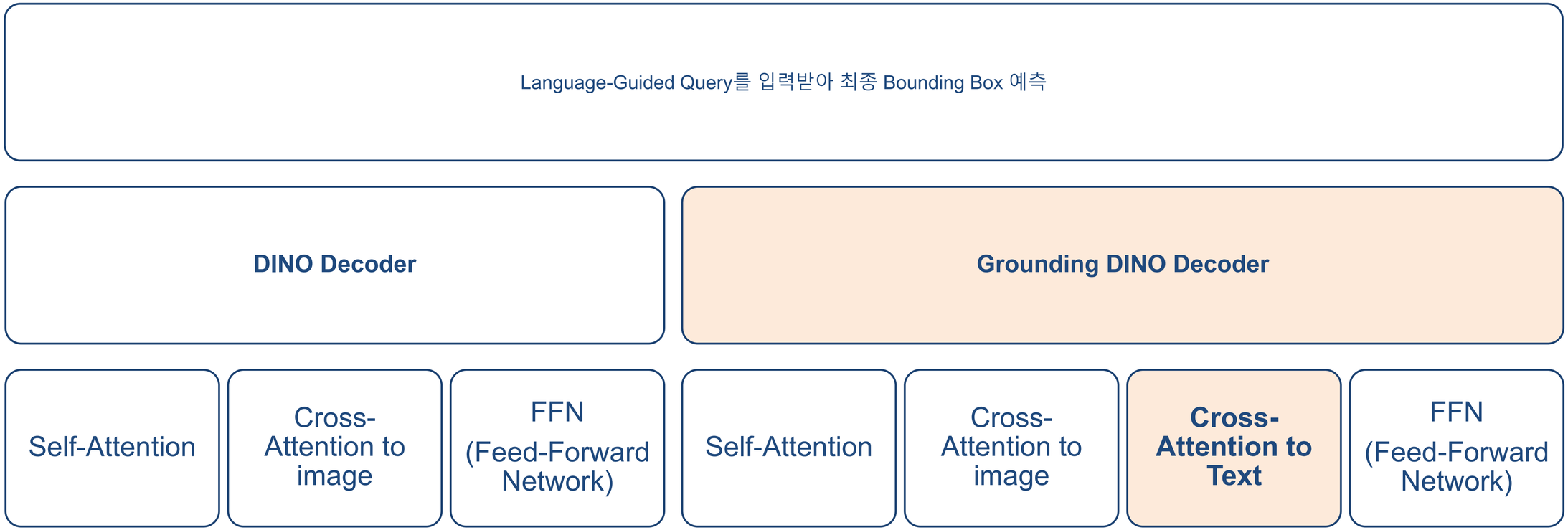
=> Content-Agnostic



Grounding DINO

=> Content-Aware

Cross-Modality Decoder



Decoder의 각 Layer에서 Query가 시각 정보와 언어 정보를 Cross-Modal Fusion
=> 텍스트 의미와 시각적 위치를 더욱 정교하게 매칭

Training & Loss Functions

Text-Box pairs
대규모 data sets

Loss Functions

1. Box Loss (L1 loss + GIOU loss)
예측된 Box와 정답 Box 간의 위치 및 크기 오차를 줄임.
(Regression)

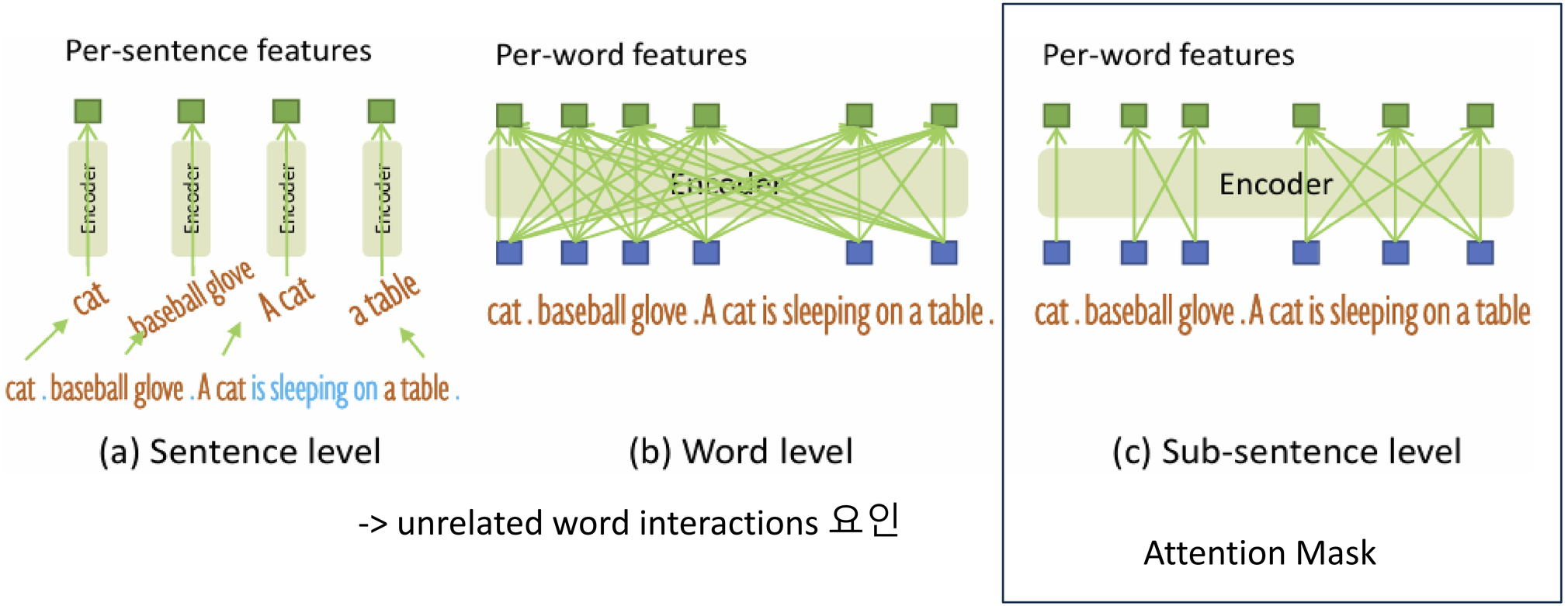
2. Contrastive loss
예측된 Box의 featur가 정답 텍스트의 feature에는 가까워지고,
오답 텍스트의 feature와는 멀어지도록 분류함.

3. DN loss
정답 Box에 일부러 노이즈를 넣고 모델이 노이즈가 없는
원래의 정답 Box와 텍스트를 복원하도록 학습.
⇒ **학습 안정성을 높이고 속도를 향상**

Grounding DINO vs. DINO

		Input	Query	Encoder	Decoder	Output
DINO	Closed-set detector	Image only	Content-Agnostic Queries 학습된 고정값	Single Modality 이미지 특징만 처리	Single Modality Cross-Attention Query가 이미지 특징만 참조	고정된 class ID
Grounding DINO	Open-set detector	Image + text prompt	Language-Guided Queries 언어 기반 동적 생성	Cross-Modality Fusion 이미지와 텍스트를 융합하여 처리	Cross-Modality Decoder Query가 이미지와 텍스트 feature를 동시에 참조	입력된 텍스트와의 유사도 및 Box 좌표

Decoder 각 단계에서
Text feature를
참고하는가



-> unrelated word interactions 요인

Experimental Setup

Zero-Shot (Open-Set) Object Detection
학습에 사용된 데이터셋(O365, GoldG 등)과 *겹치지 않는* 카테고리로 구성된 데이터셋에서 성능 평가
ODinW (Object Detection in the Wild)
➔Generalization

평가 Data set
COCO: 80개 카테고리 (가장 표준적인 벤치마크)
LVIS: 1203개 카테고리 (특히 Rare 카테고리 성능이 중요)

비교 모델
GLIP, GLIPv2: 이전 Open-Set 모델
OV-DETR, DINO-5scale: 다른 Open-Set / Closed-Set 모델
Backbones: Swin-T (Tiny), Swin-B (Base), Swin-L (Large) -> 비교 모델과 동일한 backbone 이용

1. Zero-Shot COCO

Model	Backbone	Pre-Training Data	Zero-Shot 2017val	Fine-Tuning 2017val/test-dev
Faster R-CNN	RN50-FPN	-	-	40.2 / -
Faster R-CNN	RN101-FPN	-	-	42.0 / -
DyHead-T [5]	Swin-T	-	-	49.7 / -
DyHead-L [5]	Swin-L	-	-	58.4 / 58.7
DyHead-L [5]	Swin-L	O365,ImageNet21K	-	60.3 / 60.6
SoftTeacher [50]	Swin-L	O365,SS-COCO	-	60.7 / 61.3
DINO(Swin-L) [57]	Swin-L	O365	-	62.5 / -
DyHead-T† [5]	Swin-T	O365	43.6	53.3 / -
GLIP-T (B) [25]	Swin-T	O365	44.9	53.8 / -
GLIP-T (C) [25]	Swin-T	O365,GoldG	46.7	55.1 / -
GLIP-L [25]	Swin-L	FourODs,GoldG,Cap24M	49.8	60.8 / 61.0
DINO(Swin-T)† [57]	Swin-T	O365	46.2	56.9 / -
Grounding DINO T (Ours)	Swin-T	O365	46.7	56.9 / -
Grounding DINO T (Ours)	Swin-T	O365,GoldG	48.1	57.1 / -
Grounding DINO T (Ours)	Swin-T	O365,GoldG,Cap4M	48.4	57.2 / -
Grounding DINO L (Ours)	Swin-L	O365,OI [19],GoldG	**52.5**	**62.6** / **62.7** (**63.0** / **63.0**)*
Grounding DINO L (Ours)	Swin-L	O365,OI,GoldG,Cap4M,COCO,RefC	**60.7**‡	**62.6** / -

2. Zero-Shot LVIS

Model	Backbone	Pre-Training Data	MiniVal [18] AP	MiniVal [18] APr/APc/APf
Zero-Shot Setting				
GLIP-T (C)	Swin-T	O365,GoldG	24.9	17.7/19.5/31.0
GLIP-T	Swin-T	O365,GoldG,Cap4M	26.0	20.8/21.4/31.0
DetCLIPv2	Swin-T	O365,GoldG,CC15M	40.4	36.0/41.7/40.0
Grounding DINO T	Swin-T	O365,GoldG	25.6	14.4/19.6/32.2
Grounding DINO T	Swin-T	O365,GoldG,Cap4M	27.4	18.1/23.3/32.7
Grounding DINO L	Swin-L	O365,OI,GoldG,Cap4M, COCO,RefC	33.9	22.2/30.7/38.8
Fine-Tune Setting				
MDETR	RN101	GoldG,RefC	24.2	20.9/24.9/24.3
Mask R-CNN	RN101	-	33.3	26.3/34.0/33.9
DetCLIPv2 [51]	Swin-T	O365,GoldG,CC15M	50.7	44.3/52.4/50.3
Grounding DINO T	Swin-T	O365,GoldG	**52.1**	35.4/51.3/55.7

3. ODinW Benchmark

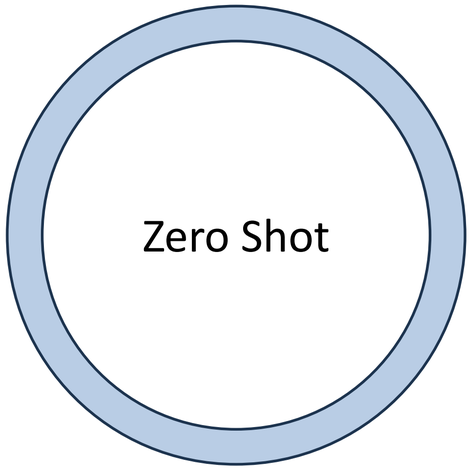
Model	Language Input	Backbone	Model Size	Pre-Training Data	Test $AP_{average}$	Test AP_{median}
Zero-Shot Setting						
MDETR [18]	✓	ENB5 [46]	169M	GoldG,RefC	10.7	3.0
OWL-ViT [35]	✓	ViT L/14(CLIP)	>1243M	O365, VG	18.8	9.8
GLIP-T [25]	✓	Swin-T	232M	O365,GoldG,Cap4M	19.6	5.1
OmDet [60]	✓	ConvNeXt-B	230M	COCO,O365,LVIS,PhraseCut	19.7	10.8
GLIPv2-T [59]	✓	Swin-T	232M	O365,GoldG,Cap4M	22.3	8.9
DetCLIP [52]	✓	Swin-L	267M	O365,GoldG,YFCC1M	24.9	18.3
Florence [54]	✓	CoSwinH	≈841M	FLD900M,O365,GoldG	25.8	14.3
Grounding DINO T(Ours)	✓	Swin-T	172M	O365,GoldG	20.0	9.5
Grounding DINO T(Ours)	✓	Swin-T	172M	O365,GoldG,Cap4M	22.3	11.9
Grounding DINO L(Ours)	✓	Swin-L	341M	O365,OI,GoldG,Cap4M,COCO,RefC	**26.1**	**18.4**
Few-Shot Setting						
DyHead-T [5]	✗	Swin-T	≈100M	O365	37.5	36.7
GLIP-T [25]	✓	Swin-T	232M	O365,GoldG,Cap4M	38.9	33.7
DINO-Swin-T [57]	✗	Swin-T	49M	O365	41.2	41.1
OmDet [60]	✓	ConvNeXt-B	230M	COCO,O365,LVIS,PhraseCut	42.4	41.7
Grounding DINO T(Ours)	✓	Swin-T	172M	O365,GoldG	**46.4**	**51.1**
Full-Shot Setting						
GLIP-T [25]	✓	Swin-T	232M	O365,GoldG,Cap4M	62.6	62.1
DyHead-T [5]	✗	Swin-T	≈100M	O365	63.2	64.9
DINO-Swin-T [57]	✗	Swin-T	49M	O365	66.7	68.5
OmDet [60]	✓	ConvNeXt-B	230M	COCO,O365,LVIS,PhraseCut	67.1	71.2
DINO-Swin-L [57]	✗	Swin-L	218M	O365	68.8	70.7
Grounding DINO T(Ours)	✓	Swin-T	172M	O365,GoldG	**70.7**	**76.2**

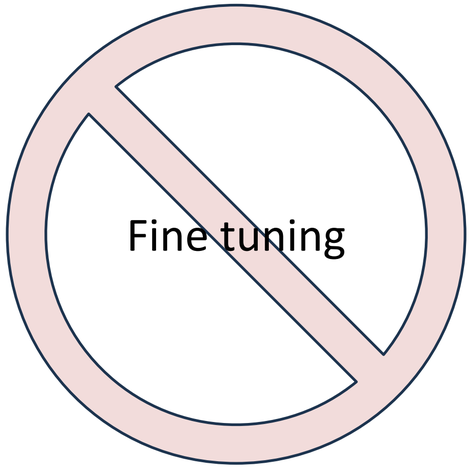
Referring Expression Comprehension (REC)

Method	Backbone	Pre-Training Data	Fine-tuning	RefCOCO			RefCOCO+			RefCOCOg	
				val	testA	testB	val	testA	testB	val	test
MAttNet [53]	R101	None	✓	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27
VGTR [10]	R101	None	✓	79.20	82.32	73.78	63.91	70.09	56.51	65.73	67.23
TransVG [7]	R101	None	✓	81.02	82.72	78.35	64.82	70.70	56.94	68.67	67.73
VILLA_L* [11]	R101	CC, SBU, COCO, VG	✓	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71
RefTR [26]	R101	VG	✓	85.65	88.73	81.16	77.55	82.26	68.99	79.25	80.01
MDETR [18]	R101	GoldG,RefC	✓	86.75	89.58	81.41	79.52	84.09	70.62	81.64	80.89
DQ-DETR [45]	R101	GoldG,RefC	✓	88.63	91.04	83.51	81.66	86.15	73.21	82.76	83.44
GLIP-T(B)	Swin-T	O365,GoldG		49.96	54.69	43.06	49.01	53.44	43.42	65.58	66.08
GLIP-T	Swin-T	O365,GoldG,Cap4M		50.42	54.30	43.83	49.50	52.78	44.59	66.09	66.89
Grounding DINO T (Ours)	Swin-T	O365,GoldG		50.41	57.24	43.21	51.40	57.59	45.81	67.46	67.13
Grounding DINO T (Ours)	Swin-T	O365,GoldG,RefC		73.98	74.88	59.29	66.81	69.91	56.09	71.06	72.07
Grounding DINO T (Ours)	Swin-T	O365,GoldG,RefC	✓	89.19	91.86	85.99	81.09	87.40	74.71	84.15	84.94
Grounding DINO L (Ours)*	Swin-L	O365,OI,GoldG,Cap4M,COCO,RefC	✓	**90.56**	**93.19**	**88.24**	**82.75**	**88.95**	**75.92**	**86.13**	**87.02**

Ablation Study

- Encoder Fusion
- Language-Guided Query
- Text Cross-Attention
- Sub-sentence Text Prompt





고찰

Limitations

- High Computational Costs
- Still dependent on Data set
- Complex realation / Negation

앞으로의 미래 연구 방향성

- 모델의 경량화
- Video-Grounding으로의 확장
- Few shot / efficient Tuning

결론

Grounding DINO: Open-Set Object Detection을 위한 새로운 모델을 제안.
With DINO의 Closed-Set Detector와 Grounded Pre-training (Open-Set Method)의 Marriage

Key factors
- Language-Guided Query Selection (텍스트 기반 동적 Query 생성)
- Deep Cross-Modality Fusion (Encoder의 Feature Enhancer & Decoder의 Cross-Attention)
- Sub-sentence-level 텍스트 특징 활용

결과: COCO, LVIS, ODinW 등 주요 Zero-Shot 벤치마크에서 이전 모델 GLIP을 매우 큰 폭으로 능가.

의의: Object Detection, Referring Expression Comprehension 등 다양한 Grounding Task를 하나의 통합된 모델로 해결하는 Foundation Model 제공.



Foundation Models for Vision

SAM

(Segment Anything Model)

산업경영공학과 22학번 박채원

Kirillov et al., Meta AI (2023) ICCV 2023

Segment Anything(SA) Project

:하나의 모델이 어떤 이미지든, 어떤 객체든 분할할 수 있게 하려는 시도.

모델	**SAM** (Segment Anything Model)	"Foundation Models for Vision"
데이터셋	**SA-1B**	**SA프로젝트를 위한 데이터셋.** **1B개 마스크와 11M개의 이미지를 포함한다.**
Task	**Promptable** **Segmentation Task**	**"유저가 입력한 프롬프트(클릭, 상자 등)에 따라** **즉시 세그멘테이션 수행"**

Foundation Model

: 거대 데이터로 학습하여 다양한 학습에 사용 가능한 '기반 모델'
파인 튜닝을 통해 세부적인 Task를 수행 가능함.

e.g.)
NLP: GPT, BERT → 번역, 요약, 질의 가능
Vision(+언어): CLIP → 이미지 분류, 검색



Imgage Segmentation을 위한
Foundation Model? SAM!

Imgage Segmentation



(a) Image



(b) Semantic Segmentation



(c) Instance Segmentation



(d) Panoptic Segmentation

- **Sementic Segmentation:** 같은 클래스는 하나로 분할 (사람 class, 자전거 class)
- **Instance Segmentation:** 개별 객체를 구분 (사람 1, 사람2...)
- **Panoptic Segmentation:** 위 두 가지를 결합

Interactive Segmentation: 사용자가 클릭, 박스 등을 제공시 해당 물체를 대상으로 분할

SAM 탄생 배경 +용어 설명

1. 언어 분야에서의 성공 사례

→ GPT, BERT 등 LLM 들이 웹 규모 데이터 학습으로
Zero-shot / Few-shot 일반화를 달성.

→ “prompt engineering”으로 다양한 작업을 수행 가능함.

2. 비전 분야로의 확장 필요성

→ CLIP, ALIGN 같이 텍스트-이미지 결합을 통해 일부 성공했지만,
비전 문제 전체를 포괄하기엔 한계가 있음.

3. SAM의 목표

→ “Build a foundation model for image segmentation.”
즉, 세그멘테이션을 위한 범용 파운데이션 모델을 구축.

4. 핵심 질문 세 가지

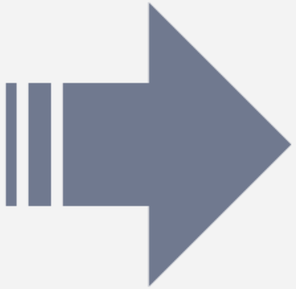
어떤 task가 zero-shot 일반화를 가능하게 할까?
어떤 모델 구조가 적합할까?
어떤 데이터가 이를 뒷받침할 수 있을까?

Data Engine Loop	모델이 데이터를 생성 → 그 데이터로 다시 모델을 개선하는 순환 구조
Zero-shot Transfer	학습에 없는 이미지나 객체도 새 학습 없이 분할

TASK: **Background**

기존 문제의 한계

- 전통적인 세그멘테이션은 “하나의 고정 목표”에 맞춰 훈련됨.
  (e.g. 사람만 찾기, 자동차만 분할하기 등)
  -> 학습하지 않은 객체나 새로운 이미지에서 잘 작동 x

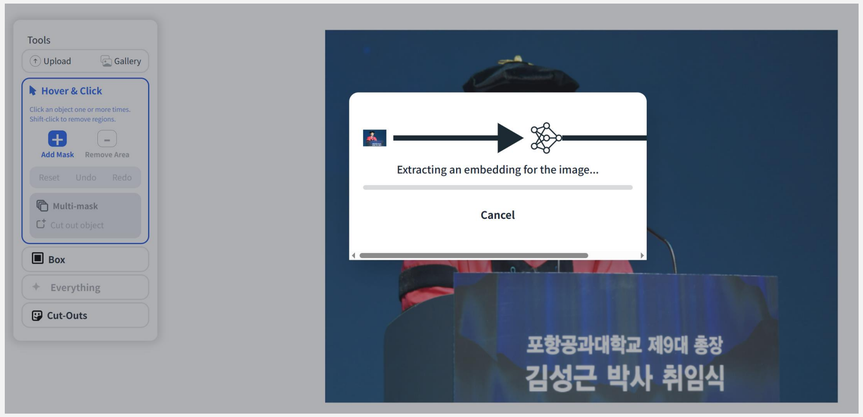


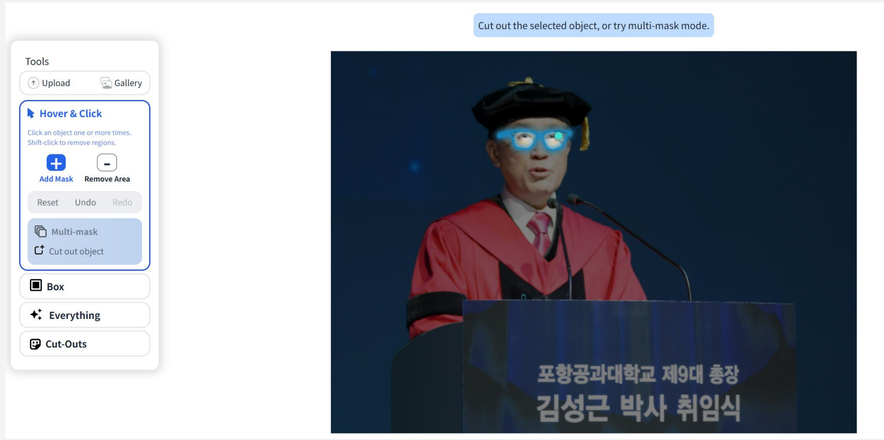
Promptable Segmentation

- 세그멘테이션을 프롬프트에 따라 달라지는 '일반화된 과제'로 정의
- 인터랙티브 + 제로샷 설계
- 항상 유효한 마스크를 내는 것이 목표

TASK: **Promptable Segmentation Task**

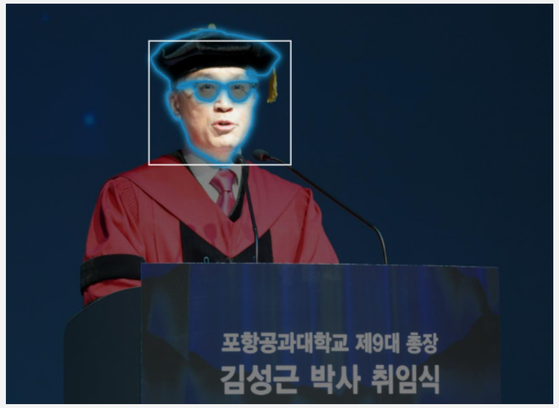
https://segment-anything.com/demo











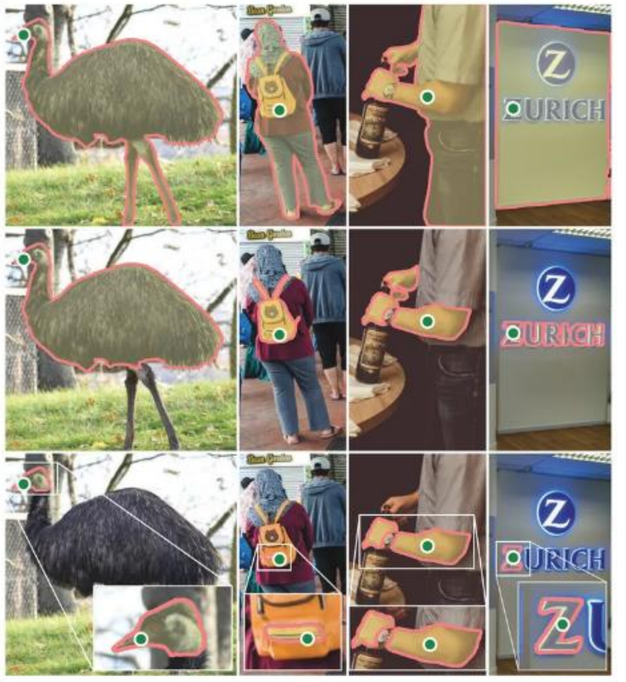


TASK: **Promptable Segmentation Task**

**Challange: "NLP가 프롬프트(질문/지시)에 답을 하듯,
이미지에서도 어떤 프롬프트(점, 박스, 텍스트, 러프한 마스크 등)가 오든지
모델은 항상 Valid*한 마스크를 출력해야 한다."**

Valid*: 프롬프트가 모호해서 여러 해석 가능성이 있더라도,
하나 이상의 타당한 해석에 한 제대로 된 마스크를 내놓으면 됨.
이때 재현가능성이 보장됨.

e.g. 옷의 패턴을 클릭한것인지, 옷 자체를 클릭한 것인지 애매함.
또는 텍스트 프롬프트 “밤” → 먹는 밤 / night



TASK: **Promptable Segmentation Task**

**Challange: "NLP가 프롬프트(질문/지시)에 답을 하듯,
이미지에서도 어떤 프롬프트(점, 박스, 텍스트, 러프한 마스크 등)가 오든지
모델은 항상 Valid*한 마스크를 출력해야 한다."**

1) Pre-training 전략

학습 때 프롬프트 시퀀스(점, 박스, 마스크 등)를 샘플마다 여러 번 던지고,
모델이 낸 마스크와 정답를 반복 비교하며 점점 개선되게 함. (지도학습의 일종)
'인터랙티브 세그멘테이션'에서 착안했으나, “사람이 충분히 클릭하면 결국 정답 마스크에 수렴”하는 것을
목표로 하는 기존 방식과 달리, 모호한 값에도 항상 유효한 마스크를 낼 수 있는 모델을 목표로 함.

모호성이 많은 실제 상황에서도 사람 도움 없이 쓸 수 있는 강한 사전학습 신경망이 필요하기 때문.

TASK: **Promptable Segmentation Task**

**Challange: "NLP가 프롬프트(질문/지시)에 답을 하듯,
이미지에서도 어떤 프롬프트(점, 박스, 텍스트, 러프한 마스크 등)가 오든지
모델은 항상 Valid*한 마스크를 출력해야 한다."**

2) Zero-shot transfer

pre-training을 통해 다양한 프롬프트–마스크 대응 패턴을 학습하여,
새 이미지나 도메인에서도 재학습 없이 일반화된 추론을 수행함.

이를 통해 '프롬프트 엔지니어링'이 가능해지며,
처음 보는 새로운 과제도 프롬프트 조합만으로 수행할 수 있는 범용성을 가지게 됨.

Model: **Segment Anything Model**

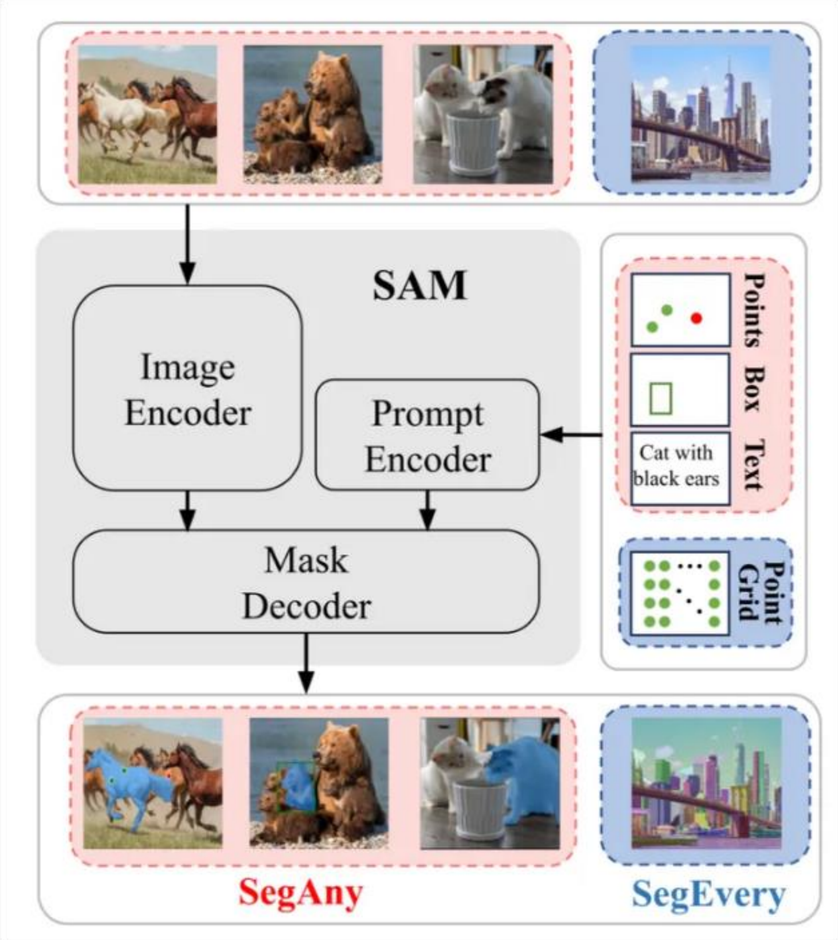


Image Encoder

- 복잡한 이미지 데이터를 숫자 형태의 압축된 정보로 바꾸는(임베딩) 부분

Prompt Encoder

- 사용자가 입력하는 “명령”을 이해하는 부분

Mask Decoder

- 인코더가 만든 이미지/프롬프트 정보를 받아서 실제로 “경계선(마스크)”을 만들어내는 부분

Model: **Segment Anything Model**

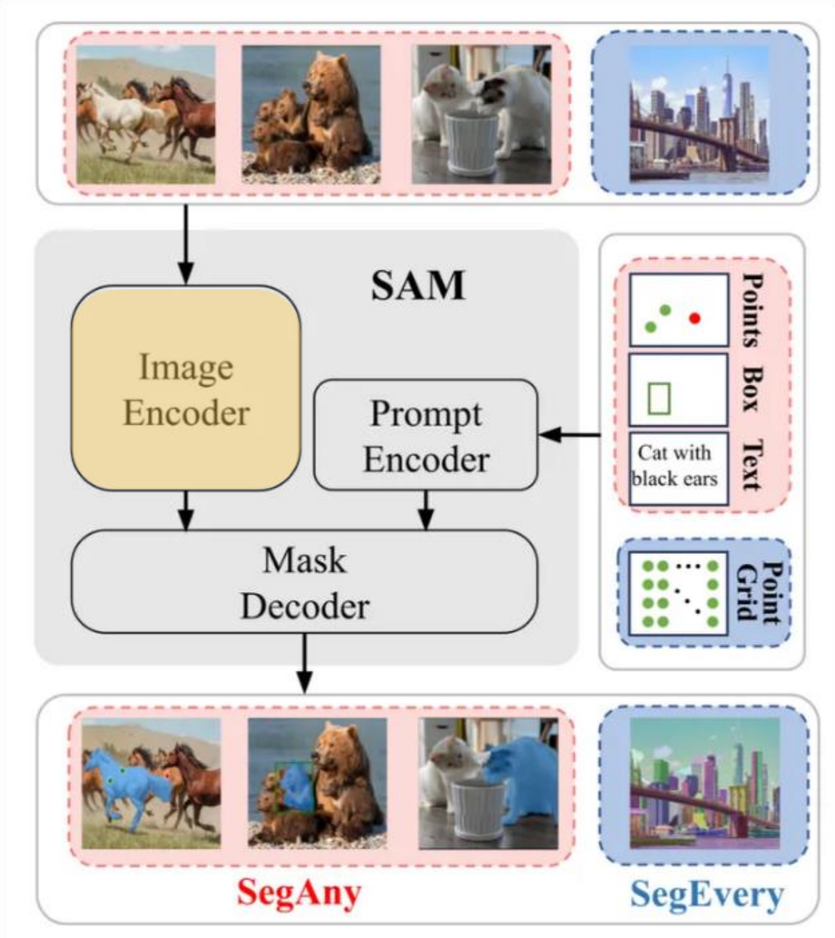
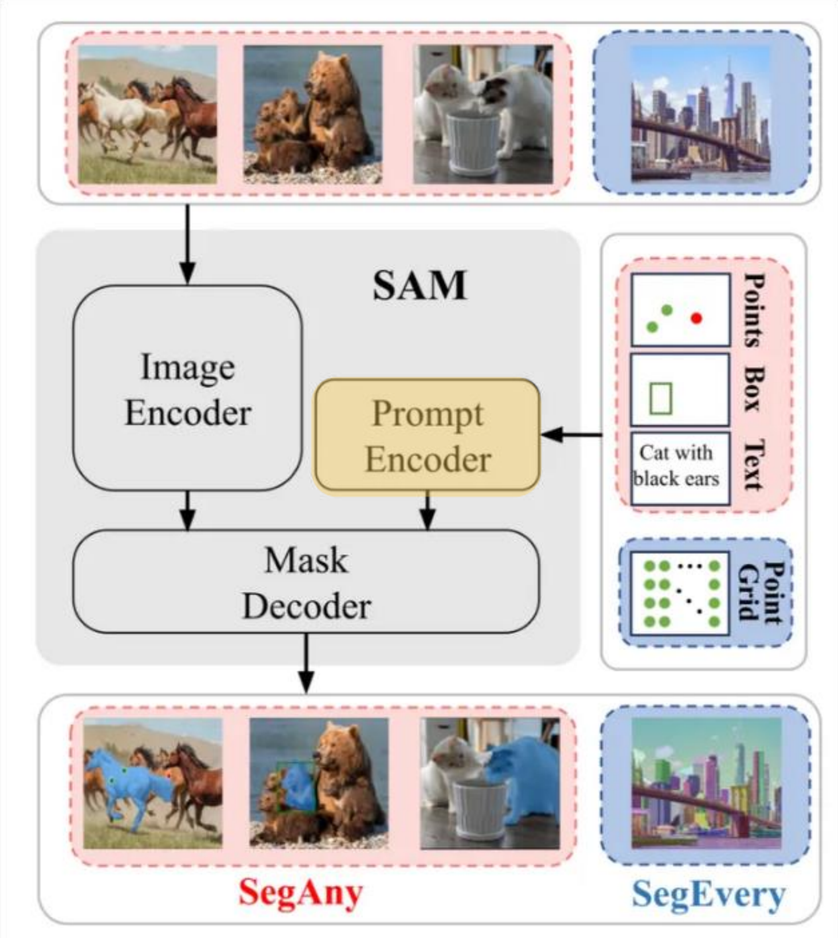


Image Encoder (1회)

- 이미지를 보고 특징(feature)을 추출해 시각적 임베딩으로 변환.
- 비전 트랜스포머(ViT) 기반으로 설계됨.
- 이미지를 한 번 인코딩하면 이후엔 다시 반복 계산할 필요x

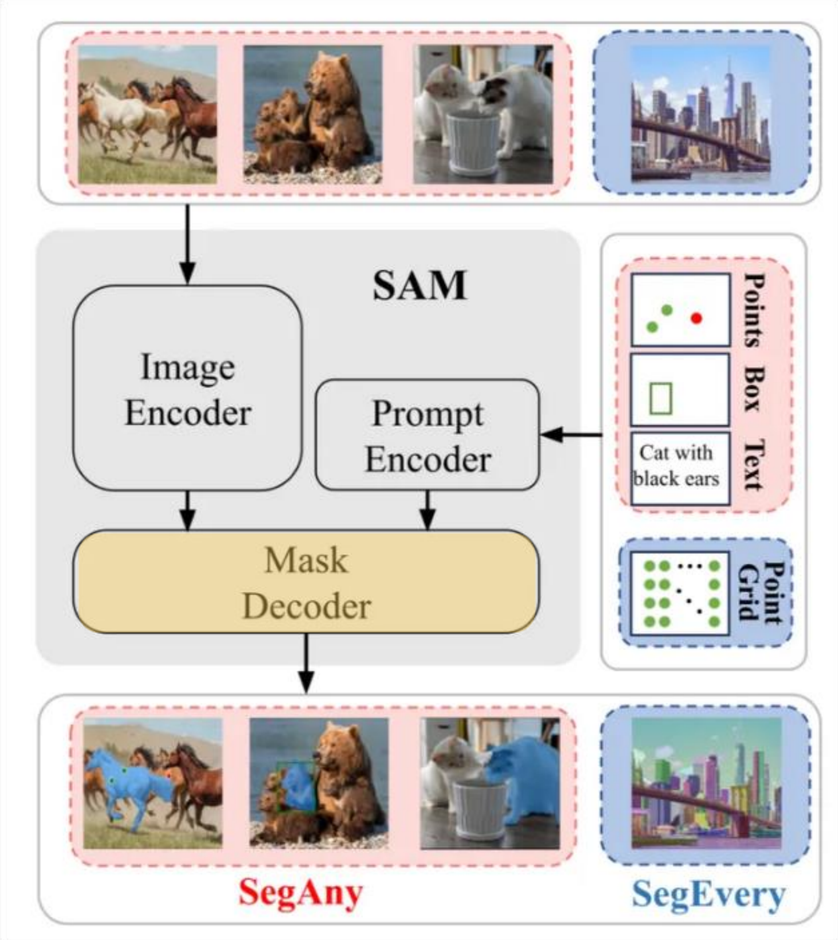
Model: **Segment Anything Model**



Prompt Encoder

- 사용자의 지시(텍스트, 점, 박스 등)를 숫자 벡터로 바꿔서 처리.
  e.g. '손바닥' 텍스트, 또는 클릭 지점의 좌표를 벡터 형태로 바꿔서 feature와 결합
- **두 가지 프롬프트**
  - Sparse (희소형) → 점, 박스, 텍스트
  - Dense (밀집형) → 기존 마스크
- 텍스트 인코딩엔 이미지-텍스트 연결해 학습한 **'CLIP' 모델의 텍스트 인코더** 사용
  -> CLIP의 한계 그대로 가짐.
  (텍스트를 이해x, **학습되지 않은 단어에는 랜덤에 가까운 임베딩**)
  -> 텍스트 프롬프트는 탐색적이고 완전히 robust하지 않음 (사실상 부가기능)

Model: **Segment Anything Model**



Mask Decoder

- 두 정보를 결합해서 "어떤 부분이 분리되어야 하는가?"를 실시간으로 계산.
- 웹브라우저에서도 50ms(0.05초) 정도로 빠르게 작동할 만큼 가볍게 설계됨
- Transformer 기반 구조

  -> 어떤 부분이 물체이고, 어떤 부분이 배경인지 확률로 계산
- 한 번의 실행으로 여러 가능한 마스크를 동시에 예측할 수 있도록 설계되어 있어, Valid 출력을 낼 수 있는 핵심임

DATA: **SA-B1**

Data Engine

- 좋은 모델을 만들려면 대규모의 다양한 데이터가 필요하나, 그런 이미지 분할 데이터셋은 기존에 존재x.
- Meta의 대안: 데이터 엔진 직접 만들기.

엔진의 데이터 수집 단계

1. Assisted-manual: 사람이 직접 마스크를 만들 때 SAM이 보조. (모델 불완전)
2. Semi-automatic: SAM이 자동으로 일부 마스크를 만들고, 사람이 수정.
3. Fully-automatic: SAM이 거의 전부 자동으로 생성.

SAM이 스스로
더 나은 데이터로
학습하는 순환구조

*데이터 수집 단계별 결과: Appendix

SA-B1

- Data Engine의 결과물. (10억 개 이상의 마스크, 1천1백만 장의 이미지)
- 기존 데이터셋보다 400배 크며, 모든 이미지는 라이선스 확인 완료 및 개인 정보 보호 준수

DATA: **SA-B1**

수집된 데이터셋 개요

1. 데이터 규모

- 기존 세그멘테이션 데이터셋 중 가장 큰 규모

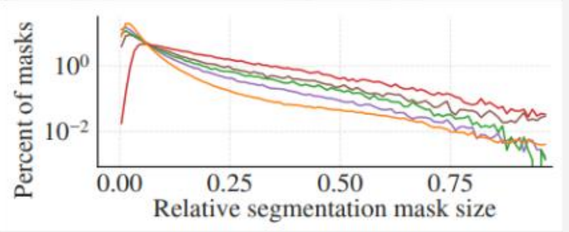
2. 이미지 품질

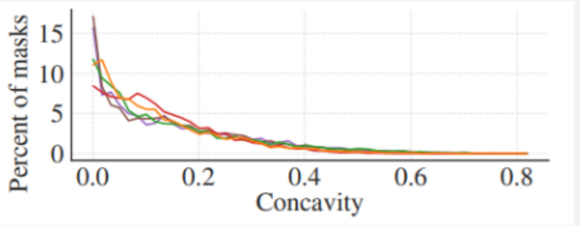
- 이미지는 전문 사진작가를 통해 라이선스를 확보하여 수집됨.
- 평균 해상도: 3300 × 4950 픽셀 (COCO모델의 약 7배)

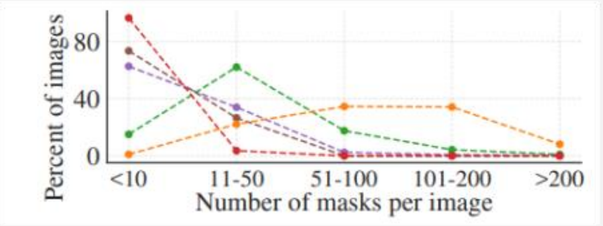
  -> 물체 경계·미세 질감 등을 더 정확히 분할

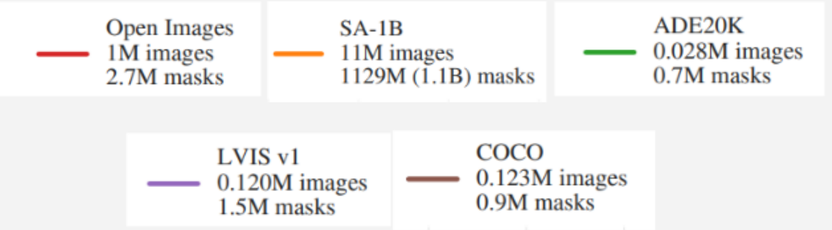
3. 마스크 생성 및 품질

- 전체 마스크 11억 개 중 99.1 %는 완전 자동 생성.
- 무작위 500장 이미지에 대해 사람이 직접 수정 후 기존vs수정간 IoU 계산 결과, 자동 마스크 품질이 기존 모든 모델과 동등하거나 더 좋음









SAM 검증 — **Responsible AI (RAI) 분석**

Segment Anything (SAM)과 SA-1B 데이터셋이
특정 지역, 인종, 성별, 나이 등에서 편향 없이 공정하게 작동하는지를 검증하기 위해 RAI 분석을 수행했다.

1) 지리적&소득적 분포

- 유럽(약 50%) 아시아·오세아니아(36%)
외에도 중간소득국의 이미지 비율이 기존
COCO/Open Images보다 월등히 높음.

- 모든 지역(아프리카 포함)에
최소 2,800만 개 이상의 마스크 포함
→ 데이터의 전 지구적 다양성 확보.

- 지역별 평균 마스크 수는 거의 동일
→ 지역별 품질 편차 없음.

2) 사람 이미지 분할시 공정성

- SAM이 사람을 분할할 때 성별·연령·피부색
에 따른 성능 차이를 측정
- 성별, 피부색, 연령별 성능 차이나 편향 없음
- 통계적으로 의미있는 편향 발견되지 않음

SAM 검증 — **Zero-Shot Transfer 실험**

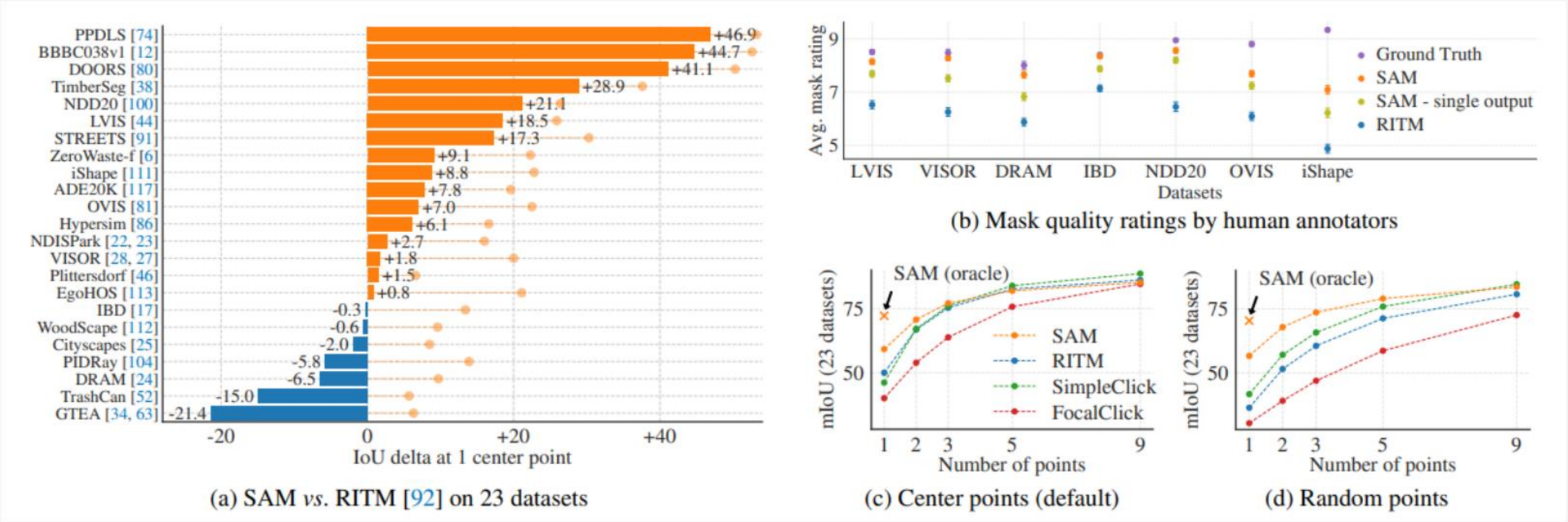
SAM이 학습되지 않은 "전혀 새로운 데이터셋과 과제"에서도
정상적인 마스크를 만들어낼 수 있는지 테스트.

중요한 점: SAM이 학습한 것은 “프롬프트가 주어졌을 때 해당 영역의 경계를 찾아내는 일반적 능력”임.
(즉, segmentation이라는 능력 자체)

SAM 검증 — **Zero-Shot Transfer 실험**

1) Zero-Shot Single Point Evaluation

- 프롬프트로 한 개의 모호한 point(학습된적x)만 주고, 객체의 마스크를 추론하도록 함.



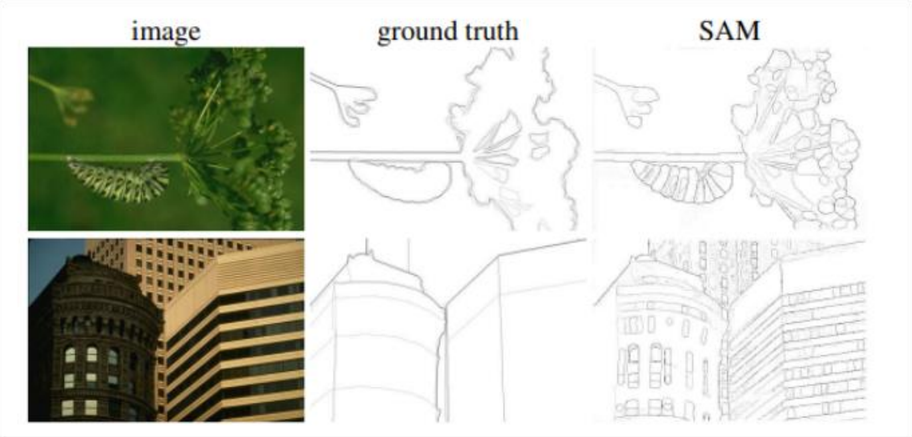
- Oracle: 모델의 여러 후보 결과 중 정답과 제일 가까운 것을 고를 때의 최고 잠재 성능
- RITM: SAM 이전의 대표적인 인터랙티브 세그멘테이션 모델. (왕복 상호작용)

SAM 검증 — **Zero-Shot Transfer 실험**

2) Zero-Shot Edge Detection

- SAM은 엣지 검출용으로 학습되지x
  →세그멘테이션 기능을 응용하여수행.

*SAM의 Zero-Shot Edge Detection 방법: Appendix



→ 사람이 놓친 합리적 경계선도 탐지
→ 일부 과잉탐지 성향

method	year	ODS	OIS	AP	R50
HED [108]	2015	.788	.808	.840	.923
EDETR [79]	2022	.840	.858	.896	.930
zero-shot transfer methods:					
Sobel filter	1968	.539	-	-	-
Canny [13]	1986	.600	.640	.580	-
Felz-Hutt [35]	2004	.610	.640	.560	-
SAM	2023	.768	.786	.794	.928

- SAM은 학습 없이도 HED(학습된 딥러닝 모델) 수준에 거의 근접 (.768 vs .788).
- R50(precision 50%에서의 recall)
  : 0.928로 HED보다 조금 더 우수.

  - precision: "찾은 것 중 정답의 비율". threshold조절해서 바꿀 수 있음
  - Recall: 정답 중에 실제로 찾아낸 비율 -> 보통 둘은 반비례함

- 최신 학습 기반 모델(EDETR)보다는 약간 낮으나, 전통적 엣지 검출보다 훨씬 우수

SAM 검증 — Zero-Shot Transfer 실험

3) Zero-Shot Object Proposals

- **Object Proposal:** 이미지 안에 객체가 있을 법한 영역을 미리 제안하는 것
  → 객체 인식, 탐지 모델 등 후속 단계에 넘겨주는 역할.
  Object Detection의 전처리로 중요한 작업.

- SAM은 Object Proposal 학습하지x
  →세그멘테이션 기능을 응용하여수행.

*SAM의 Zero-Shot Object Proposal 방법: Appendix

	mask AR@1000						
method	all	small	med.	large	freq.	com.	rare
ViTDet-H [62]	63.0	51.7	80.8	87.0	63.1	63.3	58.3
zero-shot transfer methods:							
SAM – single out.	54.9	42.8	76.7	74.4	54.7	59.8	62.0
SAM	59.3	45.5	81.6	86.9	59.1	63.9	65.8

객체 크기별 데이터셋 내 등장 빈도별

평가 기준

- **mask AR@1000 :**
  최대 1,000개의 후보를 낼 수 있을 때의 평균 recall
- **AR (Average Recall) :**
  모델이 Object '후보'를 얼마나 빠짐없이 찾아냈는지의 지표

→ 특정 조건들에서 학습 모델보다 더 우수한 성능을 보임

SAM 검증 — **Zero-Shot Transfer 실험**

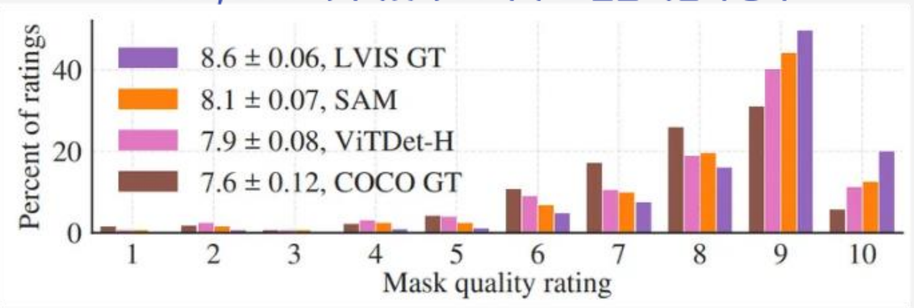
4) Zero-Shot Instance Segmentation

- SAM은 Instance 분할을을 직접 학습하지x
  →세그멘테이션 기능을 응용하여수행.
- 방법 (모듈 합성)
  탐지기(ViTDet)가 특정 부분에 물체가 있음을 알려
  주면 SAM이 박스 안 물체를 masking

<COCO, LVIS 데이터셋을 통한 성능비교>

	COCO [66]				LVIS v1 [44]			
method	AP	AP^S	AP^M	AP^L	AP	AP^S	AP^M	AP^L
ViTDet-H [62]	51.0	32.0	54.3	68.9	46.6	35.0	58.0	66.3
zero-shot transfer methods (segmentation module only):								
SAM	46.5	30.8	51.0	61.7	44.7	32.5	57.6	65.5

<COCO, LVIS 데이터셋의 GT와 두 모델을 사람이 평가>



<COCO에서 ViTDet의 AP가 SAM을 앞서는 이유>

- COCO의 GT 마스크가 완벽하지 않은데, ViTDet은 COCO 데이터로 학습했기 때문에 해당 bias까지 학습함.
  → COCO에서 점수는 높지만, 실제 품질은 떨어짐.
- SAM은 제로샷이라 더 일반적인 품질을 냄.
- LVIS는 라벨 품질이 높기 때문에 두 모델의 차이가 줄어듦.

결론

1. SAM 성능 최고!
2. 프롬프트 + 조합성(composition)은 모델을 확장 가능하게 만드는 강력한 도구.

e.g. Object Proposals(박스) + SAM(마스크) → 인스턴스 세그멘테이션 파이프라인.

Segment Anything Project의 의의:

사람-중심 설계였던 전통적 인터랙티브 세그멘테이션을 넘어,
알고리즘 시스템 속 컴포넌트로도 잘 작동하는 범용 파운데이션 블록을 제시.

Appendix

데이터 수집 단계별 결과

1. Assisted-manual Stage

- 평균 주석 시간이 COCO 데이터셋 대비 6.5배 빠름.
- 한 이미지에서 얻는 마스크 수도 20개 → 44개로 증가.
- 총 12만 장의 이미지에서 430만 개의 마스크 수집.

2. Semi-automatic Stage

- 평균 마스크 수: 44개 → 72개로 증가.
- 총 18만 장 이미지에서 4.19백만개의 마스크 수집

3. Fully-automatic Stage

- 총 1,100만 장의 이미지에서 1.1 b개 마스크 수집

Appendix

- AP: Precision-Recall 곡선 아래 면적
- ODS(Optimal Dataset Scale): (모든 이미지에)한 임계값에 대한 최고 점수. 일반화 성능
- OIS(Optimal Image Scale): 이미지별 최적 임계값 적용시 평균 성능

Appendix

SAM의 Zero-Shot Edge Detection 방식 (프롬프트 제공 방법)

1. 16×16 격자로 이미지를 나눠, 각 칸마다 중심점을 뽑아서 총 256개의 점을 입력

2. 각 점마다 SAM이 3개의 마스크를 예측 → 약 768×3개 후보 생성.

3. 중복된 마스크는 NMS(Non-Maximum Suppression)로 제거.
→ 겹치는 경계선을 정리.

4. 남은 마스크를 합쳐서 edge map 생성.
→ Sobel filter로 마스크 확률 맵을 미세하게 보정. (선을 부드럽게 다듬음)

Appendix

SAM의 Zero-Shot Object Proposal 방식 (프롬프트 제공 방법)

1. SAM이 원래 하던 자동 마스크 생성(auto mask generation) 방식을 사용함
2. 그 마스크들을 "이게 객체일 수도 있다"는 proposal 후보로 간주함
3. 각 후보를 평가할 때는 LVIS v1의 표준 AR (Average Recall) 지표를 사용
4. 임계점을 설정하고, AR을 기반으로 Object Proposal 과정 수행

Appendix

META는 자기학습 구조의 오류 누적과 편향 증폭을 어떻게 해결했나?

단순 자기학습처럼 보이지만, 실제로는 '단계적 자기개선(looped refinement)' 구조

1. 이미 사람이 개입한 수백만 개의 고품질 마스크가 존재

→ 교정 데이터(anchor)의 역할
⇒SAM이 후속 단계에서 만든 마스크는 완전히 자유롭게 생성되는 게 아니라,
이미 고정된 사람 주석 품질 범위 안에서 정규화된 분포를 유지

2. 모델은 정답 복제가 아닌 샘플 확장을 함.

같은 이미지에서도 다양한 프롬프트 조합을 주면서
특정 프롬프트에 맞는 마스크를 반복 학습.
-> 하나의 데이터로부터 수십개의 입력-정답 쌍을 만듦
=>프롬프트 다양성을 통해 모델이 세그멘테이션 규칙을 더 일반화하도록 유도함.
결국 데이터 수는 폭발적으로 늘지만, 오류 패턴은 분산