CLIP(Contrastive Language-Image Pre-training)

2025 Summer seminar 08.01

Sogang University
Vision & Display Systems Lab, Dept. . of Artificial Intelligence

Presented By
Heedong Seo

Outline

- Papers
  - Learning Transferable Visual Models From Natural Language Supervision [ICML 2021]
  - Is CLIP ideal? No. Can we fix it? Yes [arxiv 2025]

- Reason for selection
  - 여러가지 방법론(counting, anomaly detection..)이 CLIP 기반으로 개발 중
  - CLIP 개념을 이해하기 위해 기초 논문과 구조적 문제점을 지적한 논문 선정

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford[1)] Jong Wook Kim Chris Hallacy Aditya Ramesh Gabriel Goh Sandhini Agarwal Girish Sastry Amanda Askell
Pamela Mishkin Jack Clark Gretchen Krueger Ilya Sutskever

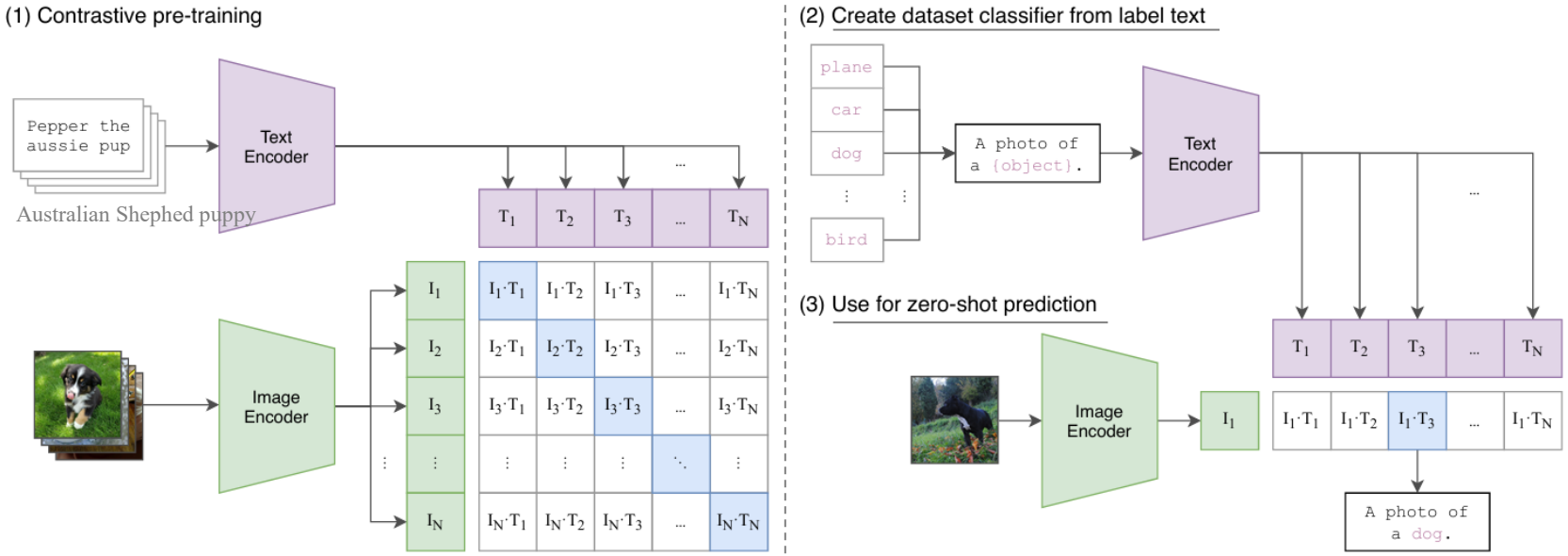
Background

- General Computer Vision
  - 일반적인 Computer Vision에서는 미리 정해진 고정된 객체 범주를 예측
    - 새로운 이미지에 대해서는 추가 레이블 데이터가 필요
    - 일반성과 유용성이 제한적

- Contrastive Language-Image Pre-training (CLIP)[1] 개념 도입
  - 인터넷에서 수집된 4억 개의(이미지, 텍스트) 쌍 데이터셋을 사용
    - 캡션(텍스트)이 어떤 이미지와 짝을 이루는지 예측하는 간단한 사전 학습 작업 진행
    - 추가 레이블 데이터 학습 없이 최신 이미지 표현을 효율적으로 학습 가능
      - 자연어 설명(캡션)을 통해 학습된 시각 개념을 참조하여 다양한 downstream task에 zero-shot transfer 가능

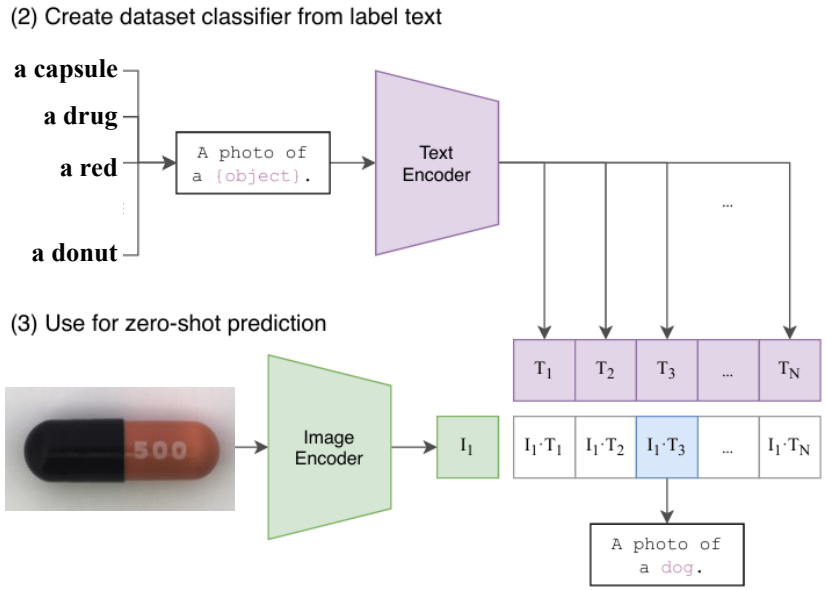
Methodology

- CLIP[1] Model
  - Image와 Text 간의 관계를 학습하고 유사도를 바탕으로 Image 예측 할 수 있는 모델
    - Contrastive pre-training
      - Text Encoder : Text의 특징 추출 (Transformer)
      - Image Encoder : Image의 특징 추출 (ResNet, ViT..)
    - Create dataset classifier from label text & Use for zero-shot prediction
      - 입력되는 Image와 Texts (e.g., car, plane..)들 사이에 유사도를 수치로 표현 (최대 유사도 대상 추출)



Methodology

- CLIP Code 동작 예시
  - Create dataset classifier from label text & Use for zero-shot prediction



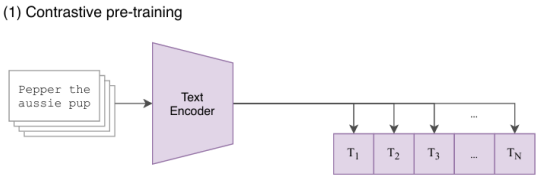

```python
import os
import torch
import clip
from PIL import Image


def test_clip(image_path, word_list):
    device = "cuda" if torch.cuda.is_available() else "cpu"
    model, preprocess = clip.load("ViT-B/32", device=device)
                                  # ViT-B/32 parameter download
    image = preprocess(Image.open(image_path)).unsqueeze(0).to(device)
    text = clip.tokenize(word_list).to(device)

    with torch.no_grad():
        image_features = model.encode_image(image)
        text_features = model.encode_text(text)
        logits_per_image, logits_per_text = model(image, text)
        probs = logits_per_image.softmax(dim=-1).cpu().numpy()

    print("Label probs:", probs)
    ranked_items = sorted(zip(word_list, probs[0]), key=lambda x: x[1],
reverse=True)
    print("Ranked list:", [(f"{i+1}. {word}, {prob:.4f}") for i, (word,
prob) in enumerate(ranked_items)])

select_image = "003.png"  # Image (capsule)
word_list = ["a drug", "a capsule", "a red", "a donut"] # Texts

if __name__ == "__main__":
    image_path = os.path.join(os.path.dirname(__file__), select_image)
    test_clip(image_path, word_list)
```

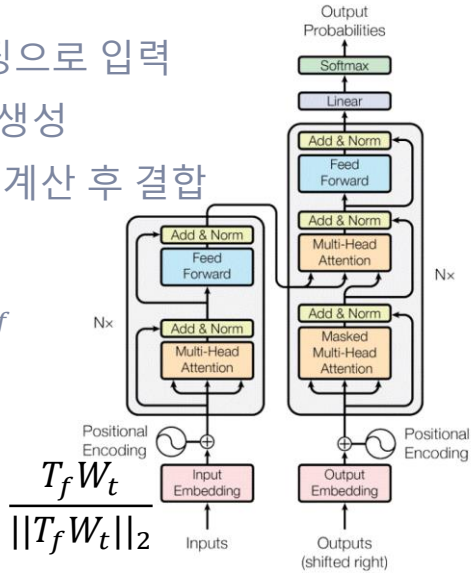
Ranked list: ['1. a capsule, 0.9834', '2. a drug, 0.0144', '3. a red, 0.0022', '4. a donut, 0.0000']

Ranked list: ['1. capsule, 0.8705', '2. drug, 0.1277', '3. red, 0.0018', '4. donut, 0.0000']

Methodology

- Contrastive pre-training
  - Text Encoder : 입력된 텍스트를 고차원 특징 벡터로 변환
    - 소문자 변환 & Byte Pair Encoding(BPE)
      - BPE: 자주 등장하는 문자열 쌍을 하나의 새로운 토큰으로 변환하여 어휘 크기 효과적 관리
        - ✓ e.g., "photograph → ["photo", "graph"]
      - 특수 토큰 추가 : 시작 [SOS] (Start Of Sequence), 끝에는 [EOS] (End of Sequence)
      - 최대 시퀀스 길이 제한 : 계산 효율성을 위해 최대 시퀀스 길이 77로 설정
        논문에서는 76, 공개 코드에서는 77
    - Transformer 모델에 입력
      - 토큰을 임베딩 벡터로 변환하여 위치 임베딩과 같이 입력 임베딩으로 입력
      - 다른 선형 변화을 적용하여 Query(Q), Key(K), Value(V) 행렬을 생성
      - Multi-Head Attention (MHA) 여러 어텐션 헤드의 출력을 병렬로 계산 후 결합
      - Add & Norm → Feed-Forward → Add & Norm …..
      - 최종 레이어의 출력에서 [EOS] 토큰에 해당하는 행 벡터 추출 T_f
    - Layer Normalization[1)] : 학습 안정성 향상 $\gamma \frac{x_i - u}{\sqrt{\sigma^2 + \varepsilon}} + \beta$
    - Linear Projection : 공통 임베딩 공간의 차원 d_e 로 변환 $T_f W_t$
    - L2 Normalization : 단위 벡터화, 코사인 유사도와 내적의 일치 $T_e = \frac{T_f W_t}{||T_f W_t||_2}$

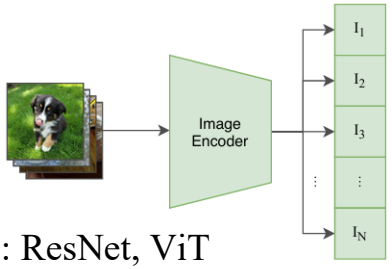




63M-parameter 12-layer 512-wide model with 8 attention head

Methodology

- Contrastive pre-training
  - Image Encoder : 입력된 Image을 고차원 특징 벡터로 변환
    - CLIP에서는 기본적으로 2가지 base 기준으로 feature extractor를 진행 : ResNet, ViT
      - ResNet 기반 모델 (5개 제공)
        - ✓ RN50 (ResNet-50 기반), RN101(ResNet-101 기반), RN50x4 (ResNet-50을 4배 확장),
        - ✓ RN50x16 (ResNet-50을 16배 확장), RN50x64(ResNet-50을 64배 확장)
      - Vision Transformer(ViT) (4개 제공)
        - ✓ ViT-B/32 (Base, 32x32 패치), ViT-B/16 (Base, 16x16 패치),
        - ✓ ViT-L/14 (Large, 14x14 패치), ViT-L/14@336px (Large, 336px 입력 해상도)
      - ResNet or ViT (선택 기준)을 통해 최종 특징 맵 추출 I_f
    - Linear Projection : 공통 임베딩 공간의 차원 d_e 로 변환 $I_f W_i$
    - L2 Normalization : 단위 벡터화, 코사인 유사도와 내적의 일치 $I_e = \frac{I_f W_i}{||I_f W_i||_2}$



very large minibatch size of 32,768

Methodology

- Contrastive pre-training
  - 유사도 계산

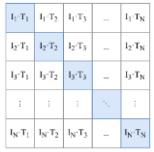
    – Text Encoder와 Image Encoder을 통과한 특징 벡터들의 코사인 유사도[1] 계산

    ⁘ 스케일링된 쌍별 코사인 유사도(Logits) 계산 $logits = I_e T_e^T \cdot \exp(\tau)$ ▶
    learnable temperature parameterr was initialized to the equivalent of 0.07

    ✓ $\exp(\tau)$: 학습 가능한 스칼라 온도 파라미터, softmax 확률 분포의 날카로움을 제어

    ⁘ 정답 레이블 : labels = np.arrange(n)

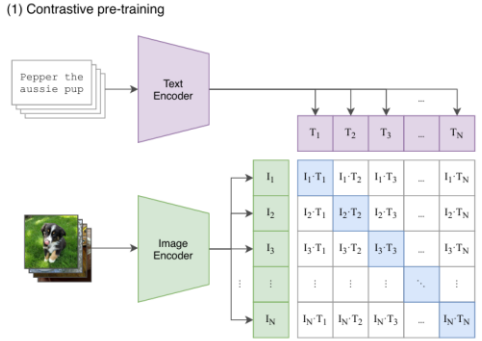
$$\begin{pmatrix} I_1 \cdot T_1 & I_1 \cdot T_2 & I_1 \cdot T_3 \\ I_2 \cdot T_1 & I_2 \cdot T_2 & I_2 \cdot T_3 \\ I_3 \cdot T_1 & I_3 \cdot T_2 & I_3 \cdot T_3 \end{pmatrix}$$



Real pairs : N
Incorrect Pairings : N^2 - N

  - 대칭 교차 엔트로피 손실 계산

$loss_i$ (Image에서 Text로의 손실)	$loss_t$ (Text에서 Image로의 손실)
$P(T_k \mid I_i) = \frac{\exp(logits_{ik})}{\sum_{j=1}^{N} \exp(logits_{ij})}$ 확률 함수	$P(I_k \mid T_j) = \frac{\exp(logits_{kj})}{\sum_{i=1}^{N} \exp(logits_{ij})}$
$L_{I \to T}^{(i)} = -\log\left(\frac{\exp(logits_{ii})}{\sum_{j=1}^{N} \exp(logits_{ij})}\right)$ 손실 함수	$L_{T \to I}^{(j)} = -\log\left(\frac{\exp(logits_{jj})}{\sum_{i=1}^{N} \exp(logits_{ij})}\right)$
$loss_i = \frac{1}{N} \sum_{i=1}^{N} L_{I \to T}^{(i)}$ 손실 평균	$loss_t = \frac{1}{N} \sum_{j=1}^{N} L_{T \to I}^{(j)}$



$$loss = \frac{loss_i + loss_t}{2}$$
최종 대칭 손실

Methodology

- Why is the word “contrastive” used?
  - 일반적인 Deep Learning 모델은 Input 과 label과의 matching 계산
  - CLIP은 Real pairs (N)은 최대화, Incorrect Pairings($N^2 - N$) 최소화 하는 대조 학습 모델

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

$$\begin{pmatrix} I_1 \cdot T_1 & I_1 \cdot T_2 & I_1 \cdot T_3 \\ I_2 \cdot T_1 & I_2 \cdot T_2 & I_2 \cdot T_3 \\ I_3 \cdot T_1 & I_3 \cdot T_2 & I_3 \cdot T_3 \end{pmatrix}$$

Real pairs : N
Incorrect Pairings : N^2 - N

행렬의 대각선(Real Pairs)을 최대화,
비대각선(Incorrect Pairings) 최소화

Experiments

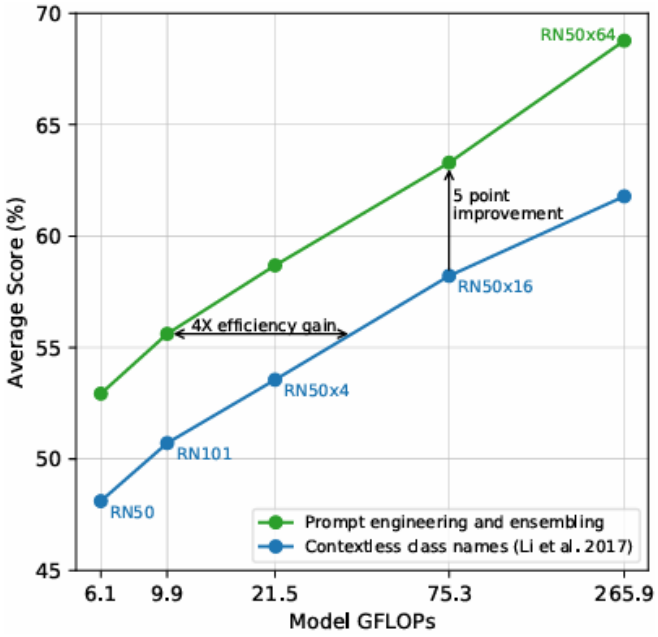
- Zero-shot transfer image classification
  - Visual N-Grams 대비 더 높은 성능을 보이는 것을 확인

	aYahoo	ImageNet	SUN
Visual N-Grams	72.4	11.5	23.0
CLIP	**98.4**	**76.2**	**58.5**

    - Visual N-Grams이 나온 지 오래되었기 때문에 직접적인 비교로 보면 안됨
      - Visual N-Grams 대비 10x dataset, 100x compute pre prediction, 1000x training compute…
  - Prompt engineering에 따라서 성능 변화 확인
    - Task에 맞춰 prompt 텍스트를 조정함으로써 zero-shot 성능 향상 가능
      - ImageNet : Prompt template “A photo of a {label}” 사용 시 정확도 1.3% ↑
      - Oxford-IIIT Pets : "A photo of a {label}, a type of pet.“ 사용 시 효과적
      - Food101 : 음식 유형 지정 시 성능 향상
      - FGVC Aircraft : 항공기 유형 지정 시 성능 향상
      - OCR : 텍스트 또는 숫자 주위에 따옴표를 넣으면 성능 향상
      - 위성 이미지 분류 : "a satellite photo of a {label}.“ 사용 시 효과적

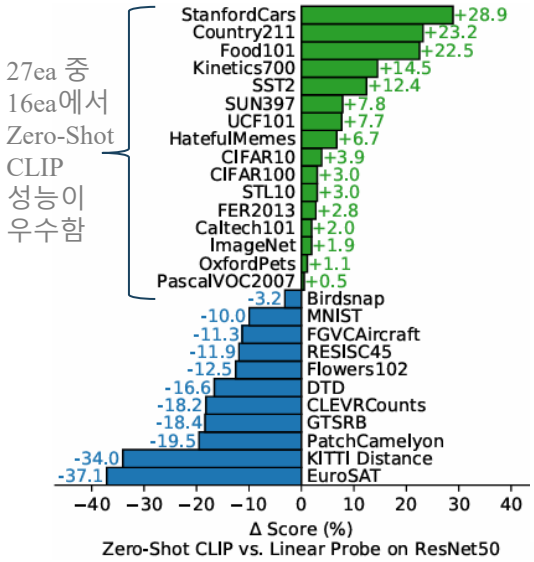
Experiments

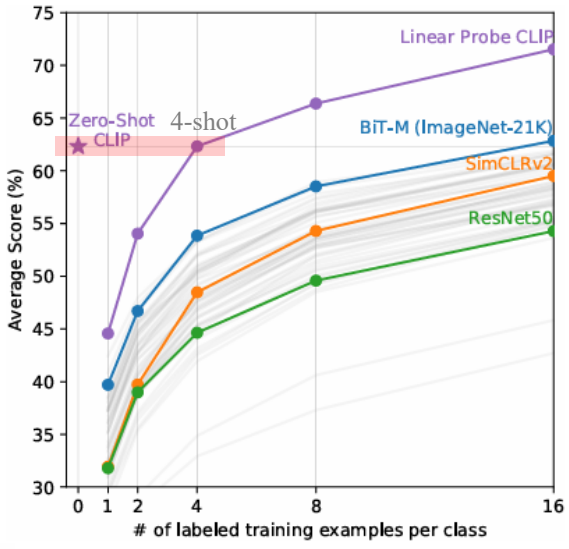
- Zero-shot transfer image classification
  - Prompt engineering and ensembling : ImageNet 성능 3.5% ↑
    - 80개의 서로 다른 context prompt을 ensemble하면 단일 prompt 보다 3.5% 성능 향상 (ImageNet)
      - "A photo of a big {label}.", "A photo of a small {label}." 등을 사용하여 여러 개의 제로샷 분류기/임베딩을 생성

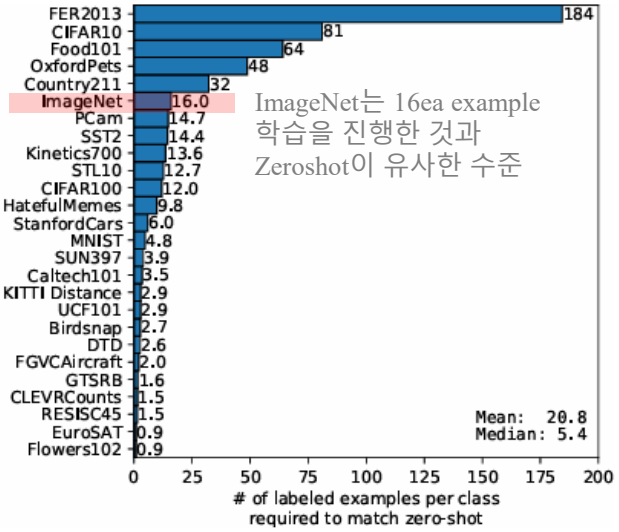


Experiments

- Zero-shot transfer image classification
  - Zero-Shot CLIP이 Fully Supervised Learning로 학습된 ResNet-50 대비 16ea 더 나은 성능
    - 여러 특수하고 복잡하거나 추상적인 Task에서는 ResNet-50대비 성능이 낮음 (11ea)
  - Zero-Shot 과 4-Shot logistic regression 성능이 일치
  - Dataset 별로 효율성이 다양함
    - class당 1개 미만의 labeled example ~ 184개까지 나타날 수 있음

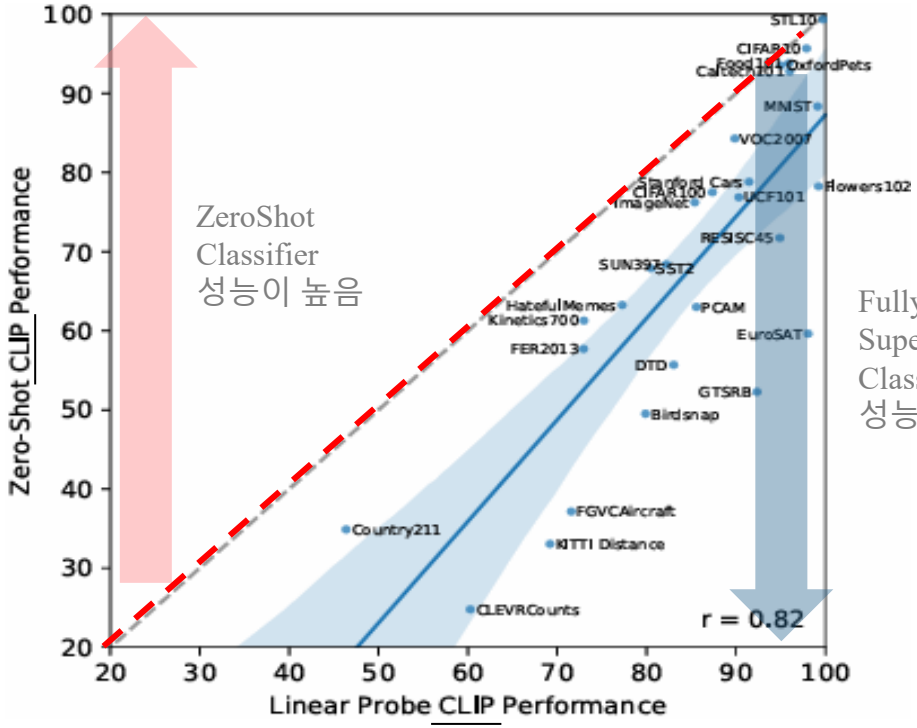






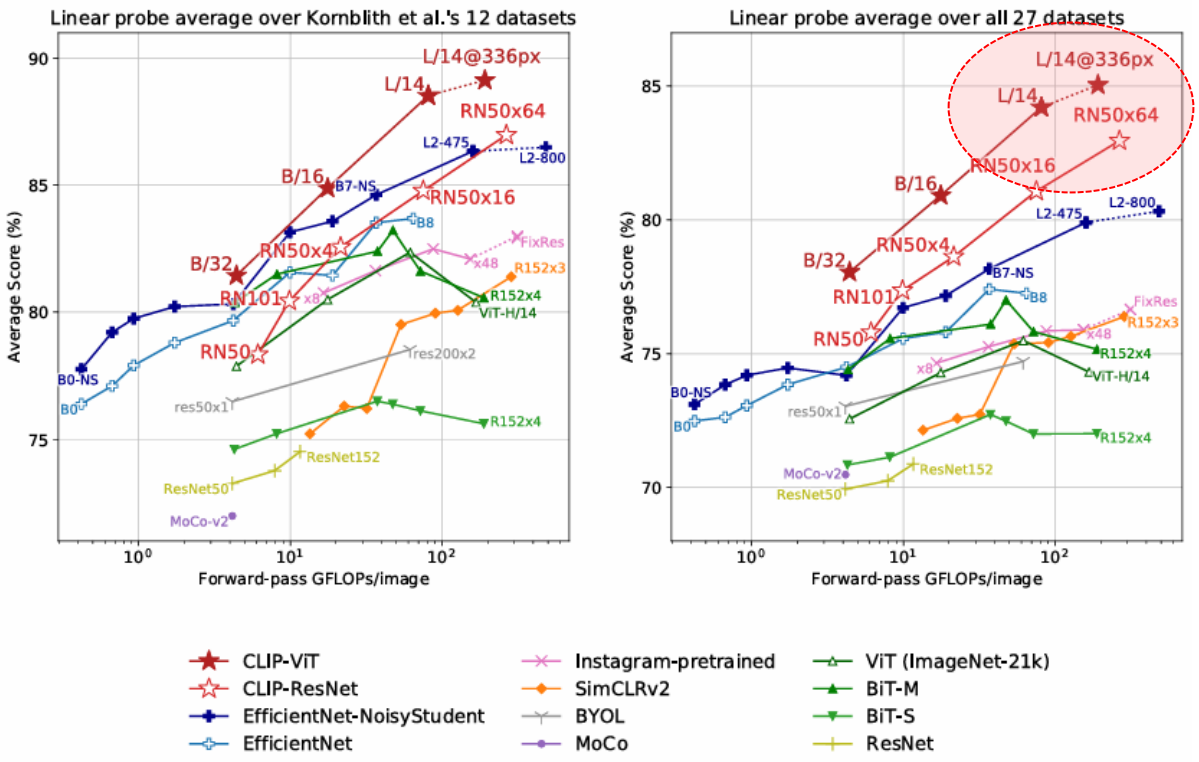
Experiments

- Zero-shot transfer image classification
  - 대부분의 dataset : Zero-Shot Classifier은 Fully Supervised Classifier 보다 성능 낮음 (10~20% ↓)



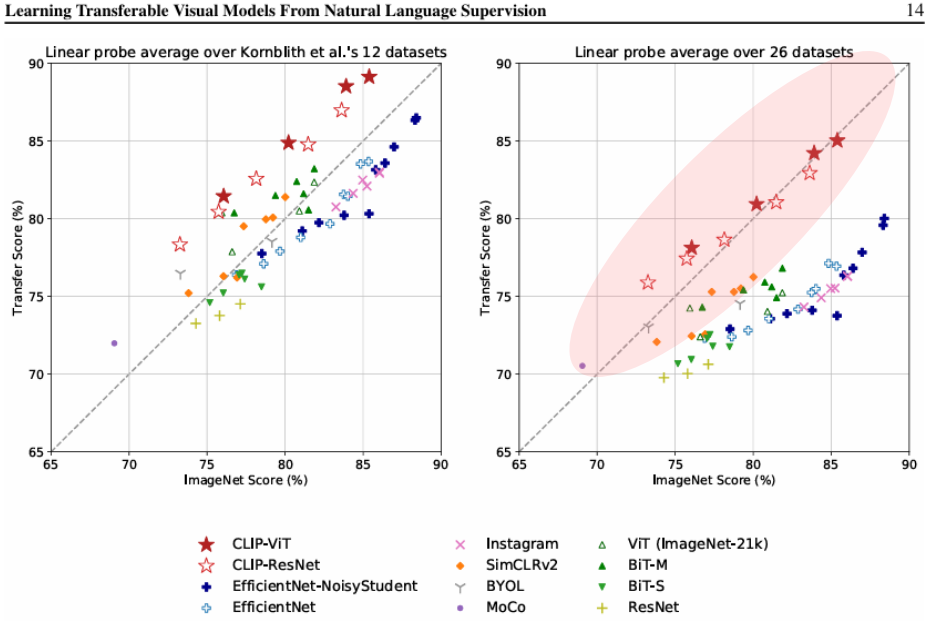
Experiments

- Transfer Learning 관점에서 모델 성능 비교
  - Pre-training을 통해 학습된 모델에서 특징 추출기(feature extractor)을 고정하고, 이 특징들 위에 간단한 선형 분류기를 훈련하여 성능을 측정한 결과 CLIP은 우수한 성능을 보임
    - CLIP 중의 최고의 모델은 ViT-L/14@336px



Experiments

- Robustness to Natural Distribution Shift
  - CLIP은 Transfer Learning에서 우수한 Robustness[1] 실력을 보여줌 (Robustness)



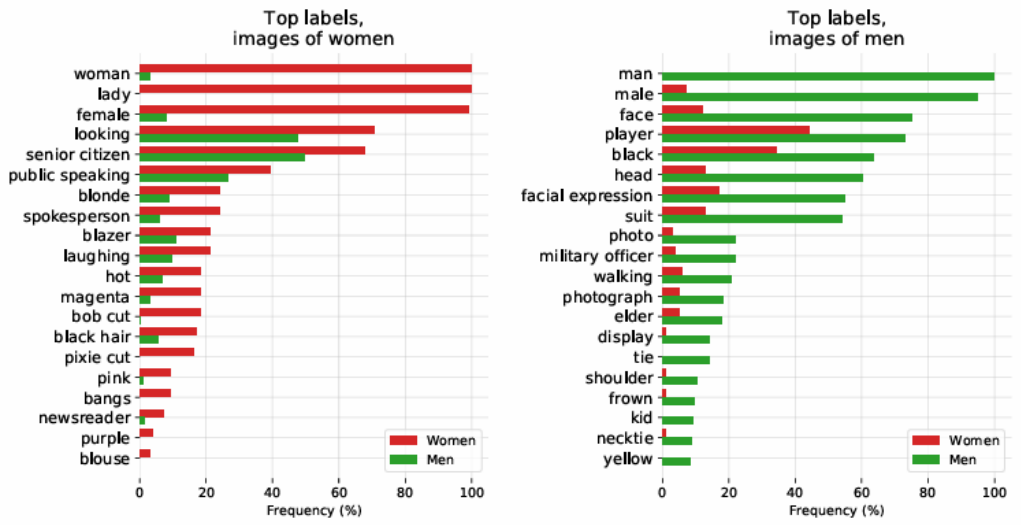
점선 (y=x) : 모델의 데이터 포인트가 이 선 위에 위치할수록, ImageNet 성능에 비해 다른 작업으로의 전이 성능이 더 우수하다는 것을 의미

Experiments

- Bias
  - Data 수와 유명도 등에 따라서 편향을 가질 수 있음
    - Black 계열에서 Non-human categories가 높게 나타남

Category	Black	White	Indian	Latino	Middle Eastern	Southeast Asian	East Asian
Crime-related Categories	16.4	24.9	24.4	10.8	19.7	4.4	1.3
Non-human Categories	14.4	5.5	7.6	3.7	2.0	1.9	0.0

- men vs women와 상관없지만 각자 다르게 연관도가 높게 제시되는 단어들이 있음



Is CLIP ideal? No. Can we fix it? Yes

Raphi Kang Yue Song Gerogia Gkioxari Pietro Perona

1) Raphi Kang et al. “Is CLIP ideal? No. Can we fix it? Yes” in arXiv, 2025

2) Semanitic Topology : image 내의 시각적 요소와 text내의 언어적 요소 사이의 구조적이고 의미적인 관계 ▶

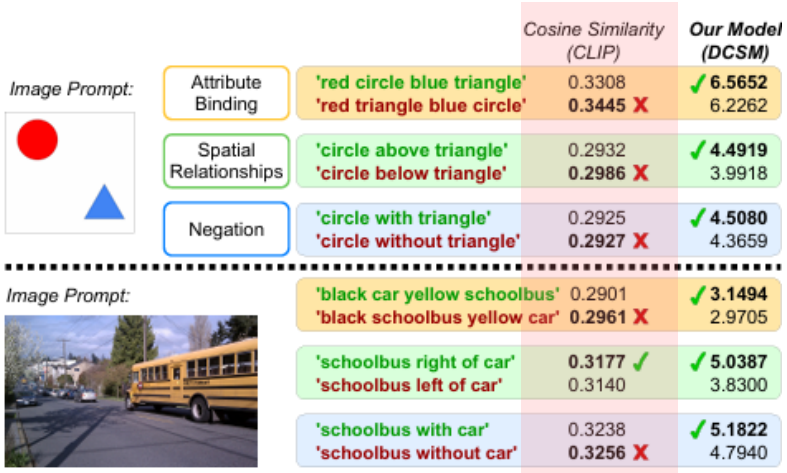
Background

- CLIP Issue
  - CLIP의 잠재공간(latent space)는 복잡한 시각-텍스트 상호 작용을 처리하기 어려움(실패함)
    - 복잡한 시각-텍스트 상호 작용 처리가 어려운 근본적인 원인은 CLIP의 기하학적 구조에 있음
    - CLIP과 유사한 공동 임베딩 공간인 다음 중 두 가지를 동시에 올바르게 수행할 수 없음
      - 기본 설명 및 이미지 콘텐츠 표현 (Represent basic descriptions and image content)
        - ✓ image와 image에 대한 간단한 설명(text)을 정확하게 매칭하는 능력
      - 속성 바인딩 표현(Represent attribute binding)
        - ✓ 특정 객체에 정확한 속성(e.g., 색상, 크기)를 연결하는 능력
      - 공간적 위치 및 관계 표현(Represent spatial location and relationships)
        - ✓ 객체들의 공간적인 위치나 객체들 간의 관계를 이해하는 능력
      - 부정 표현(Represent negation)
        - ✓ “X가 아니다” 와 같이 부정적인 개념을 포함하는 텍스트 설명을 정확하게 이해하는 능력

- Dense Cosine Similarity Maps(DCSMs)[1]제안
  - Image patches와 text tokens의 semantic topology[2]을 유지함으로써 CLIP의 근본적인 한계 해결

Background

- Two questions about CLIP
  - Cosine 유사도를 참조 metric로 사용하는 Vision-Language Model(VLM) 작업에 성공하는 데 필요한 모든 속성을 가진 CLIP과 유사한 잠재 공간이 존재하는가?
    - 주요 VLM 작업에 필요한 속성
      - 기본 설명 및 이미지 콘텐츠 표현
      - 속성 결합 표현
      - 공간적 위치 및 관계 표현
      - 부정 표현

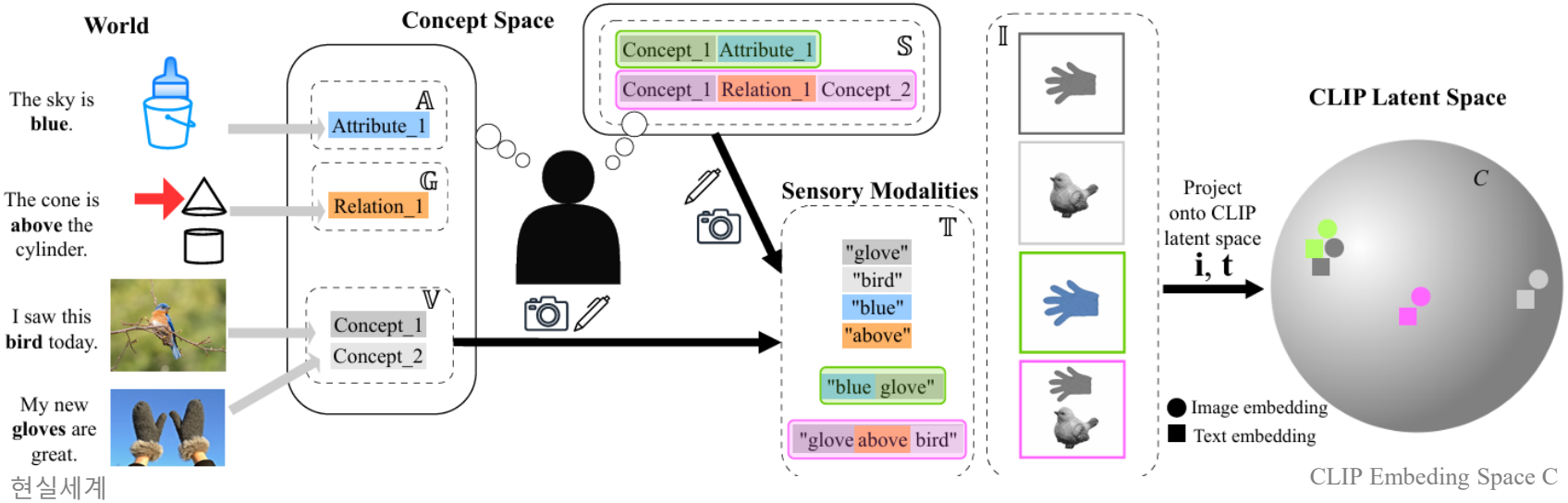


- 그러한 공간이 존재하지 않는다면, CLIP을 완전히 포기하지 않고 이 잠재 공간을 "구제"할 방법이 있는가?

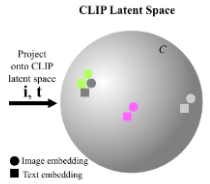
Background

- CLIP Model Concept Flowchart
  - CLIP의 기하학적 구조의 문제점을 이해하기 위해 CLIP Latent Space 개념 정리

Notation	Explanation
$\mathbf{i}(\cdot), \mathbf{t}(\cdot)$	CLIP image and text embeddings
$\mathbb{I}, \mathbb{T}$	Set of images and texts
$\mathbb{V}$	Set of atomic object concepts bird, glove….
$\mathbb{A}$	Set of atomic attributes blue, big, smooth
$\mathbb{G}$	Set of compositional relations up, left, right…
$\mathbb{S}$	Finite ordered combinations of elts from $\mathbb{V}, \mathbb{A}, \mathbb{R}$ bird over glove
$\mathbf{i}(x_a)$	Image embedding of object x with attribute a



Background



- Geometric Requirements for an Ideal CLIP

※ CLIP encoder (image, text encoder)가 만약 '이상적인 위치'에 존재한다면, 어떤 이미지나 텍스트라도 해당 이상적인 위치로 정확하게 투영 할 수 있다고 가정 (oracle encoder 가정)

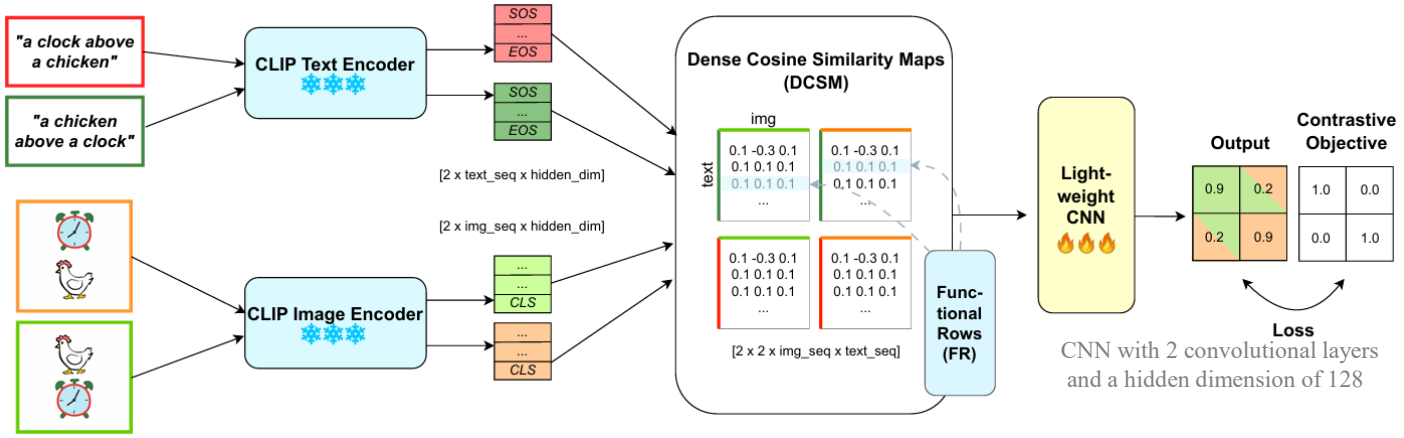
Condition	Formula	Contents
categorization	$i(x)\cdot t(x) > i(x)\cdot t(y)$ $i(x,y)\cdot t(x) > i(x,y)\cdot t(z)$	• 특정 image에 대한 올바른 text 설명의 cosine similarity가 다른 text 설명 보다 높음 (e.g., 사과 image, 사과 text > 사과 image , 바나나 text)
	$i(x_a)\cdot i(x_b) > i(x)\cdot i(y)$ $i(x, g_{loc1})\cdot i(x, g_{loc2}) > i(x)\cdot i(y)$	• 동일한 의미론적 개념을 포함하지만 속성이나 장면 구성이 다른 image들은 다른 의미론적 개념을 포함하는 image보다 더 높은 cosine similarity을 가짐 (e.g., 빨간 사과, 파란 사과 > 사과, 바나나)
attribute binding	$i(x_a)\cdot i(x_b) < 1$	• 다른 속성을 가진 개념은 CLIP 공간에서 평행하지 않음 (e.g., 빨간 자동차, 파란 자동차는 CLIP 잠재 공간에서 다른 위치에 임베딩)
	$i(x_a)\cdot t(a) < i(x_b)\cdot t(a)$	• 특정 속성을 가진 개념을 나타내는 이미지는 해당 속성의 텍스트 임베딩에 더 가까움 (e.g., 빨간 사과, 빨간 > 파란 사과, 빨간)
	$i(x_a, y_b)\cdot i(x_b, y_a) < 1$	• 동일한 개념과 속성을 가지지만 서로 다른 방식으로 짝지어진 이미지들은 CLIP 공간에서 평행하지 않음 (e.g., 빨간 공과 파란상자, 파란 공과 빨간 상자 <1)
spatial relationship	$i(x, g_{loc1})\cdot i(x, g_{loc2}) < 1$	• 동일한 객체가 다른 위치에 있는 이미지들은 동일한 임베딩을 가지지 않음 (e.g., 빨간공이 상자 안, 빨간 공이 상자 밖 < 1)
	$i(x, g_{rel3}, y)\cdot i(x, g_{rel4}, y) < 1$	• 동일한 객체들이 다른 공간전 관계가 있는 이미지들은 동일한 임베딩을 가지지 않음 (e.g., 고양이가 상자 옆, 상자가 고양이 옆 < 1)
	$i(x, g_1, y)\cdot i(x, g_1, z) > i(x, g_1, y)\cdot i(x, g_2, z)$	• 객체가 동일한 위치나 관계에 있는 이미지들은 다른 위치나 관계에 있는 이미지보다 의미론적으로 더 가까움 (e.g., 테이블 위 책, 테이블 위 컵 > 테이블 위 책, 테이블 아래 컵)
negation	$t(x)\cdot t(-x) < t(y)\cdot t(-x)$	• 텍스트 임베딩과 그 부정어는 다른 어떤 쌍보다 낮은 cosine similarity를 가짐 (e,g, 개, 개아님 < 개, 고양이)
	$i(x)\cdot t(-x) < i(x)\cdot t(y)$	• 한 개념의 긍정적 버전은 부정어보다 다른 어떤 개념과도 낮은 cosine similarity를 가짐 (e.g., 개 image, 개 아님 text < 개 image, 고양이 text)
	$t(x)\cdot t(y) < t(-y)\cdot t(-x)$	• 두 개의 뚜렷이 부정된 개념은 뚜렷이 부정되지 않은 개념보다 더 높은 cosine similarity를 가짐 (e.g., 개, 고양이 < 고양이 아님, 개 아님)

✓ 상기 Ideal CLIP 조건을 만족하지 못하면 CLIP 잠재 공간으로 투영 가능하다는 가정에 모순 발생

➔ 두 가지 조건을 동시에 만족하지 못한다는 것을 확인 ▶

Methodology

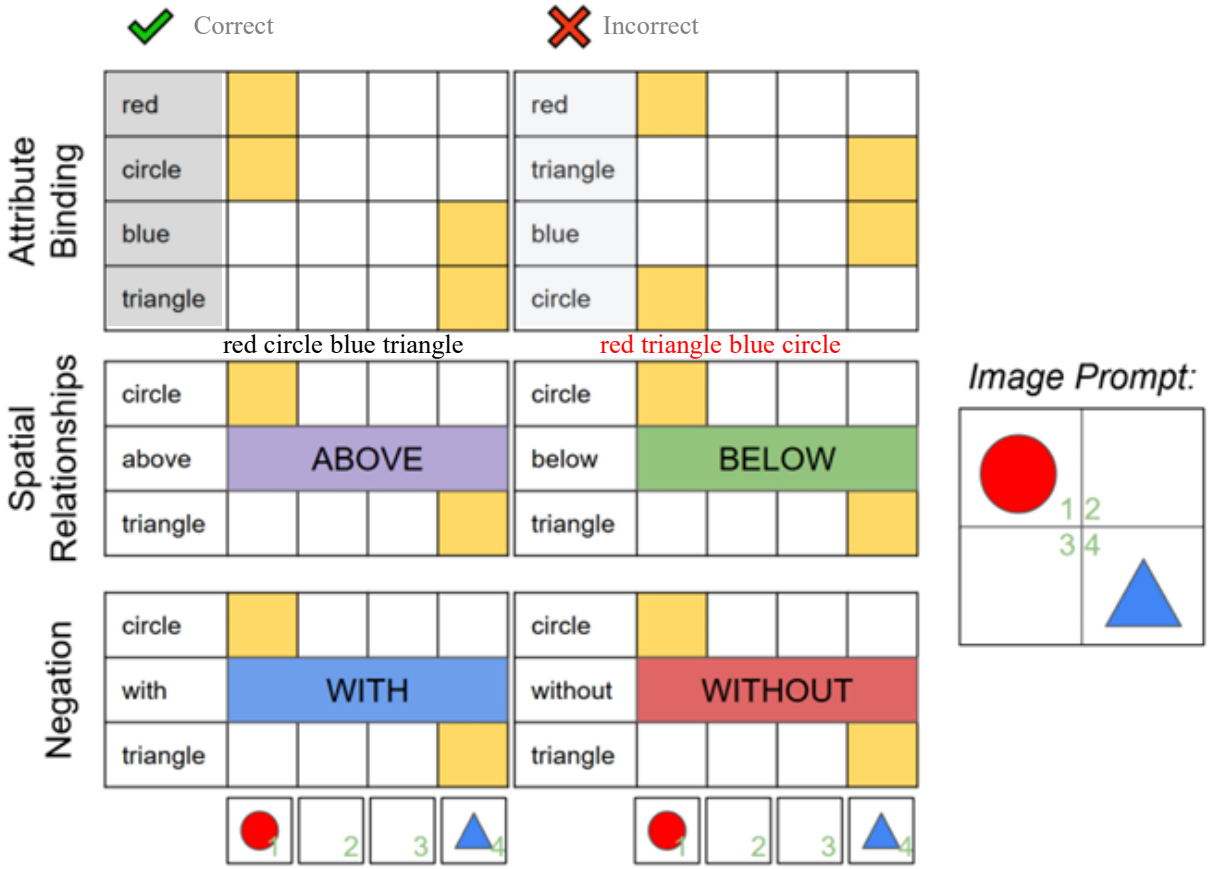
- DCSMs [1)]
  - 기존 CLIP의 근본적인 기하학적 한계, 특히 속성 바인딩, 공간 관계, 부정 표현 어려움 해결
    - 기존 CLIP : 전체 Image vs 전체 text의 global cosine similarity
      - 최종 레이어의 출력에서 [EOS] 토큰에 해당하는 행 벡터 추출 T_f
      - ResNet or ViT (선택 기준)을 통해 최종 특징 맵 추출 I_f
    - DCSMs : image patch vs text token 간 dense cosine similarity map 계산 후 CNN을 통해 종합 판단
      - 모든 토큰/패치 임베딩 활용: CLIP 인코더에서 추출된 모든 텍스트 토큰과 모든 이미지 패치 임베딩을 사용
      - 쌍별 코사인 유사도 계산: 모든 토큰/패치 임베딩 쌍에 대해 코사인 유사도 계산 (Dense Cosine Similarity Map)
      - Functional Rows(FR) 삽입: 공간 관계나 부정 등을 나타내는 특정 '기능어'에 해당하는 DCMS의 행을 미리 정의된 상수 벡터로 교체하는 커스텀 로직 구현
      - 경량 CNN 훈련: 최종 유사도 점수를 출력하는 새로운 경량 CNN모듈을 별도로 설계하고 훈련



DCSMs는 Parameter Efficient Tunning(PET) 계열로 일부 Parameter 학습 ▶

Methodology

- Retained topology in DCSMs



1) Attribute Binding : 객체에 속성(색상, 크기등)을 정확하게 연결하는 모델의 능력 평가
2) Spatial Reasoning : 이미지 내 객체들의 공간적 위치나 관계를 이해하는 모델의 능력 평가
3) Negation : 문장에서 부정의 의미를 정확하게 이해하는 모델의 능력을 평가

Experiments

- 성능 비교 평가
  - 속성 바인딩, 공간, 부정 개념에서 우수한 성능 확인

	Attribute Binding [1]			Spatial Reasoning [2]			Negation [3]	
Model Name	**CLEVR$_{bind}$**	**NCD**	**VG_attr**	**WhatsUp**	**COCO$_{1\&2obj}$**	**VG$_{1\&2obj}$**	**NegBench$_{coco}$**	**NB$_{voc}$**
CLIP$_{ViTB/32}$	22.2	71.3	61.3	31.9	47.0	47.1	39.2	38.3
CLIP$_{ViTB/16}$	20.2	63.1	61.8	30.5	48.9	51.5	41.5	37.9
NegCLIP	21.8	79.2	**72.4**	33.2	46.1	47.2	31.4	26.5
CoCa	19.2	48.7	50.8	24.5	48.6	49.5	21.6	20.8
BLIP	11.1	56.2	59.4	24.4	48.5	50.4	20.5	20.7
SigLIP	13.3	53.6	48.4	26.0	47.4	51.1	26.3	29.7
DCSM$_{synth}$	31.0	93.6	60.9	62.6	65.6	64.4	38.8	35.2
DCSM$_{coco}$	**39.9**	**95.7**	68.1	**63.7**	**72.4**	**67.0**	**48.6**	**49.0**
Random Chance	25.0	50.0	50.0	25.0	50.0	50.0	25.0	25.0

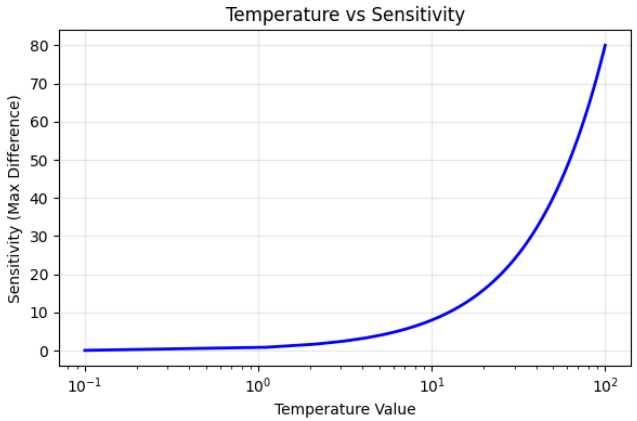
- CLIP ViT B/32, CLIP ViT B/16 : CLIP 모델 32patch, 16patch
- NegCLIP : OpenCLIP 기반 finetuned model, 특히 부정(negation)과 같은 어려운 네거티브 샘플을 통해 학습
- CoCa : 이미지와 텍스트를 인코딩하고 캡션 생성도 가능한 모델
- BLIP : 웹에서 수집된 노이즈가 많은 데이터를 합성 및 필터링된 캡션으로 보강하여 학습
- SigLIP : CLIP과 유사한 구조를 가지지만, 시그모이드 손실 함수를 사용하여 학습 효율을 높인 모델
- DCSMsynth : 합성 데이터로 훈련
- DCSMcoco : COCO 데이터셋으로 훈련, DCSM은 CLIP ViT B/16을 기본 모델로 사용
- Random Chance: 무작위로 추측했을 때 얻을 수 있는 평균적인 정확도

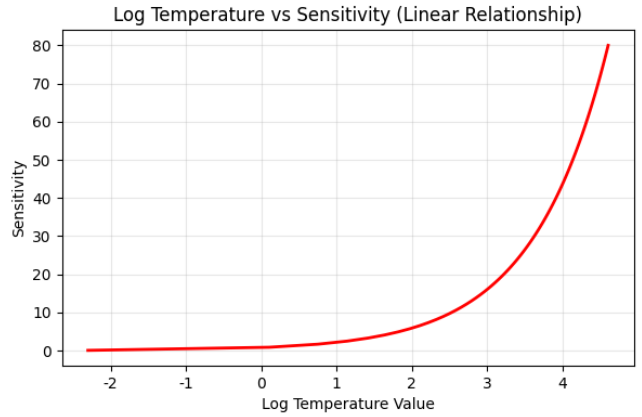
  - CLEVRbind: Does CLIP Bind Concepts? Probing Compositionality in Large Image Models 논문에서 제안된 벤치마크
  - NCD (Natural Colors Dataset): 자연스러운 색상과 관련된 속성 바인딩을 평가
  - VG attr (VG attribution): Attribute, Relation, Order (ARO) 데이터셋의 속성 관련 부분
  - WhatsUp: What's "up" with vision-language models? Investigating their struggle with spatial reasoning 논문에서 제시된 공간 추론 벤치마크
  - COCO1&2obj, VG1&2obj: COCO 및 VG 데이터셋 중 1개 또는 2개의 객체를 포함하는 이미지에 대한 질문을 통해 공간 관계를 평가
  - NegBenchcoco, NegBenchvoc: Vision-language models do not understand negation 논문에서 개발된 부정 이해를 위한 벤치마크

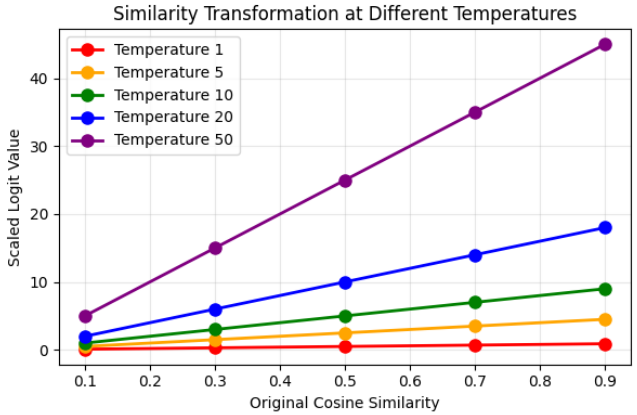
감사합니다

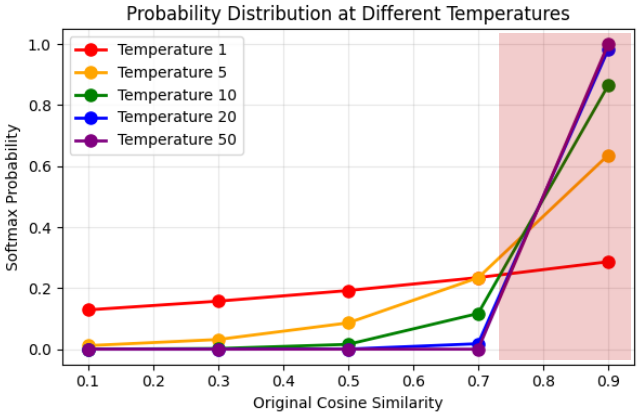
Temperature(τ) concept

- $logits = I_e T_e^T \cdot \exp(\tau)$에서 $\exp(\tau)$을 곱하는 이유
  - $\exp(\tau)$로 곱하면 temperature τ가 커질수록 Sharp해져서 구분되는 민감도가 커짐









Cosine similarity에 대비하여 softmax probability가 temperature 값이 커질수록 급격한 변화가 일어나는 것을 확인 가능
$\exp(\tau)$로 곱에 의해 나타나는 현상

1) in-distribution : 모델이 학습한 데이터 분포와 같은 분포에서 나온 데이터
2) Out-of-distribution : 모델이 학습한 데이터 분포와 다른 분포에서 나온 데이터

Robustness

- Effective Robustness (효과적인 강건성)

$$\mathrm{ER}(Effective\ Robustness) = Accuracy_{OOD} - f(Accuracy_{ID})$$

$f(Accuracy_{ID})$는 다양한 모델들에서의 ID-OOD관계를 회귀분석한 기준선(line of best fit)

  - ID[1] (학습한 데이터 분포)와 OOD[2] (학습한 데이터와 분포가 다른 데이터)에서의 정확도 간에 알려진 관계를 넘어서는 개선분을 측정
    - 기존에 학습한 것을 바탕으로 예상했던 것보다 더 뛰어난 응용력과 적응력을 보여주는 것

- Relative Robustness (상대적인 강건성)

$$\mathrm{RR}(\text{Relative } Robustness) = Accuracy_{OOD}$$

  - 학습 데이터 분포와 다른 데이터에서의 정확도가 얼마나 향상되었는지를 포괄적으로 측정
    - 전반적인 절대적인 성능 수준

✓ 모델이 학습 데이터셋의 특성에 지나치게 맞춰지면 상대적인 강건성은 높아지지만, 효과적인 강건성은 낮아질 수 있음

- Spurious Correlation
  - 두 가지 현상에서 통계적으로 강한 관계가 있는 것처럼 보이지만, 실제로는 인과 관계가 없거나 제3의 요인에 의한 우연히 발생한 것처럼 보이는 관계
    - 상대적인 강건성을 너무 높이다가 효과적인 강건성이 낮아질 수 있음

Semantic Topology

*Semantic Topology*는 이미지 내의 시각적 요소와 텍스트 내의 언어적 요소 사이의 구조적이고 의미적인 관계를 의미

- 시각적 토폴로지 (Image Patch Level)
  - 이미지 내에서 객체들이 어떻게 배열되어 있는지, 속성이 어떤 객체에 속하는지, 그리고 객체들 간의 공간적 관계(~위에, ~옆에 등)가 어떻게 형성되는지에 대한 정보
  - Image Patch의 인덱스를 통해 공간 정보를 보존하는 방식과 관련됨

- 텍스트 토폴로지 (Text Token Level)
  - 텍스트 프롬프트 내에서 단어들이 어떤 순서로 배열되어 있고, 이 단어들이 조합되어 어떤 복합적 의미 (“빨간색 자동차”와 “자동차 빨간색”의 차이, 또는 부정 표현)을 형성하는지에 대한 정보
  - Text Token의 인덱스를 통해 의미론적 순서를 보존하는 방식과 관련됨

CLIP usage method

- Feature Extractor ✓ 가장 기본적인 접근 방식, Parameter 고정
  - CLIP의 image encoder, text encoder fix → 추출된 특징을 downstream task의 입력으로 사용
    → 그 위에 새로 추가된 작은 network만 학습
    - CLIP의 방대한 Parameter는 전혀 update되지 않아 효율성은 극대화되지만, specific task에 대한 적응 능력이 제한될 수 있음

- Full Fine-tuning ✓ Parameter 학습
  - CLIP 모델의 모든 parameter을 downstream task에 맞게 조정
    - 가장 높은 성능을 기대할 수 있지만, 엄청난 계산 자원과 대규모 데이터셋이 필요
    - Pre-training knowledge을 잃어버리는 위험이 있음

- Parameter-Efficient Tunning (PET) ✓ 일부 Parameter 학습
  - CLIP의 Parameter 대부분은 FIX, 추가적인 소량의 Parameter을 도입 or 기존 Parameter 중 일부만 선택적으로 학습시켜 효율성을 높이는 방식

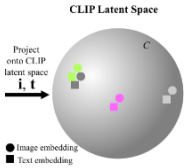
Geometric Contradictions in CLIP

❖ CLIP embedding space (C)가 condition 1을 만족한다면, 이상적인 상태가 되기 위한 다른 어떤 조건도 만족할 수 없음

Condition1

$$i(x^1, x^2) = \underset{i(x^1,x^2)}{argmax}[i(x^1, x^2) \cdot i(x^1) + i(x^1, x^2) \cdot i(x^2) - \sum_{j=3}^{M} i(x^1, x^2) \cdot i(x^j)] \quad s.t. \left|\left|i(x^1, x^2)\right|\right| = 1$$

$i(x^1, x^2)$: 두 객체 이미지의 *CLIP* 임베딩



이상적인 위치 : 해당 이미지가 포함되는 객체들과는 의미론적으로 가깝고, 포함하지 않는 다른 객체들과는 멀어진 상태

→ 이 최적화 문제를 풀면 $i(x^1, x^2)$가 개별 객체 임베딩 $i(x^1)$과 $i(x^2)$의 정규화된 선형 중첩 (linear superposition)이 된다

$$i(x^1, x^2) = \frac{i(x^1) + i(x^2)}{||i(x^1) + i(x^2)||}$$

Condition2

$i(x_a, y_b) \neq i(x_b, y_a)$ 빨간 자동차와 파란 공 ≠ 파란 자동차와 빨간 공

$i(x_a) \cdot t(a) > i(x) \cdot t(a)$

$$i(x_a) = (1 - \delta)i(x) + v, \qquad \delta \ll 1, \qquad \left|\left|i(x_a)\right|\right| = 1, \qquad i(x_a) \cdot i(x) \geq 1 - \delta$$

$$v = pt(a)$$

$$i(x_a) = (1 - \delta)i(x) + pt(a)$$
$$i(y_a) = (1 - \delta)i(y) + pt(a)$$
$$i(x_b) = (1 - \delta)i(x) + qt(b)$$
$$i(y_b) = (1 - \delta)i(y) + qt(b)$$

$$i(x_a, y_b) = \frac{(1 - \delta)\big(i(x) + i(y)\big) + pt(a) + qt(b)}{2}$$
$$= i(x_b, y_a)$$

객체 이미지 / 텍스트 속성

$$i(x) \cdot t(a) = i(y) \cdot t(a) = \cos(\theta)$$
$$i(x) \cdot t(b) = i(y) \cdot t(b) = \cos(\omega)$$

DCSM Visualization

❖ 각 DCSM에서 가장 높은 점수는 EOS 토큰과 CLS 패치 임베딩 사이에 있음