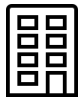
『CVSP 2026 Summer Seminar』

Optimizing Texts for an Open-Vocabulary:
From Iterative LLM Feedback to Vocabulary-Aware Fine-Tuning



Pusan National University
Dept. of Electronics Engineering
Computer Vision & Signal Processing Lab.



Presented By
김준수

Outline

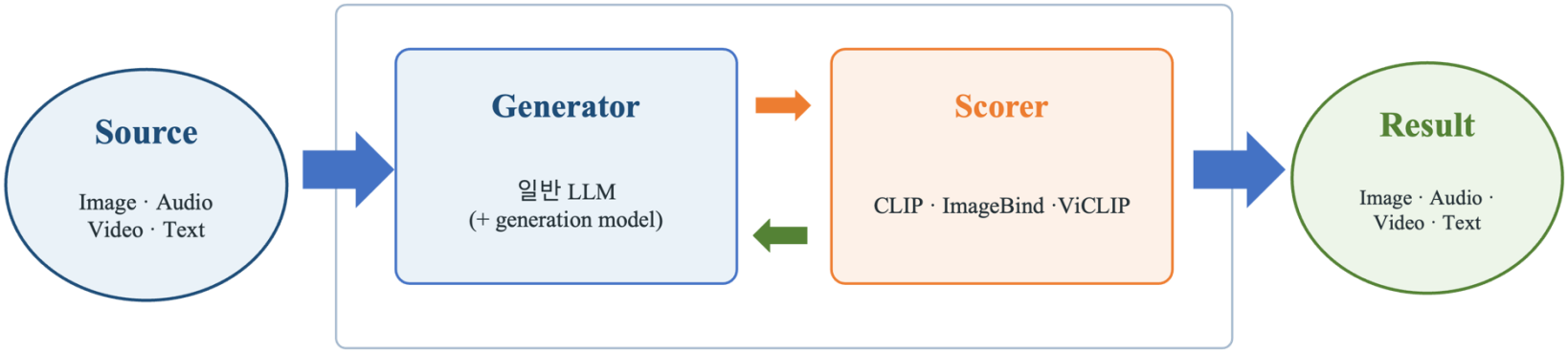
- 주제 선정 이유
- 논문 소개
  - LLMs can see and hear without any training [ICML 2025]
  - Vocabulary Scaling Law: Tuning Open-vocabulary Predictors for Their Openness [CVPR 2026 (Highlight)]
- 추후 계획

- **주제 선정 이유**
  - **LLM의 추론 능력을 이용해 원하는 텍스트를 찾을 때까지 반복하는 파이프라인을 구성.**
    - 데이터를 먼저 보고, 그것을 잘 설명하는 텍스트를 얻을 수 있다면, 추후 데이터셋 구축이나 모델의 전처리 과정에 도움이 될 것 같다고 생각하기 때문.

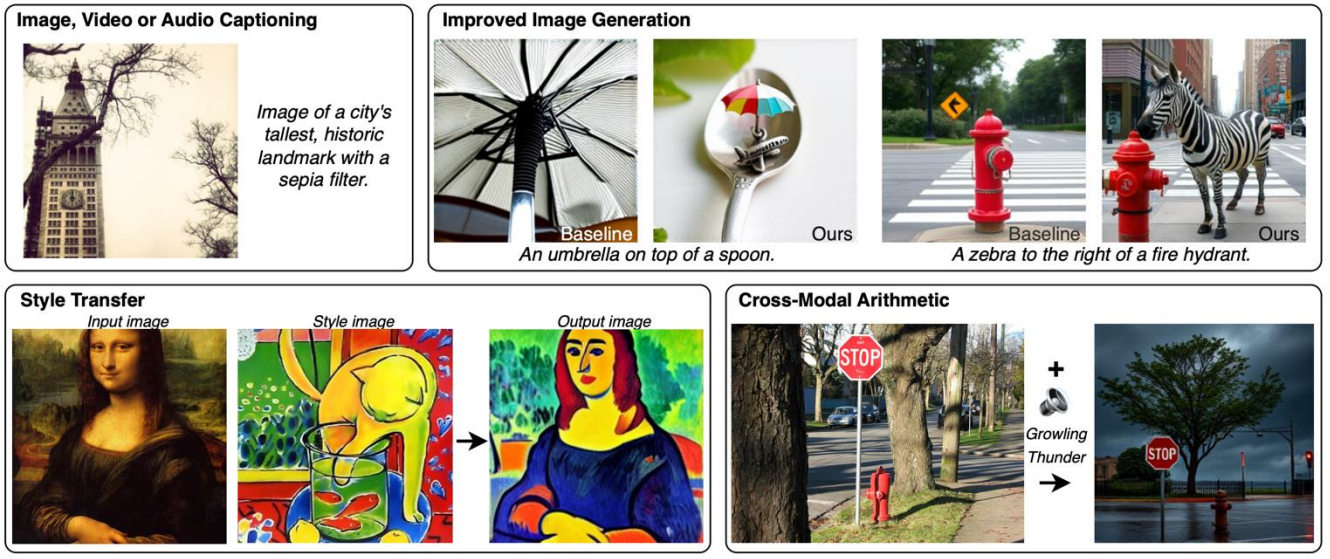
LLMs can see and hear without any training [ICML 2025]

- **Motivation**
  - **Multimodal-LLM(MLLM)이 아닌, 순수한 LLM만으로, Multimodal task를 수행가능한가?**
    - LLM이 발전함에 따라, 많은 것을 추론할 수 있게 추론능력이 향상되었다.
    - 따라서 그 추론 능력을 활용하면, MLLM과 같은 Multimodal-task에 특화된 모델없이, LLM만 이용하여 task를 수행 가능할지 확인하고자 함.

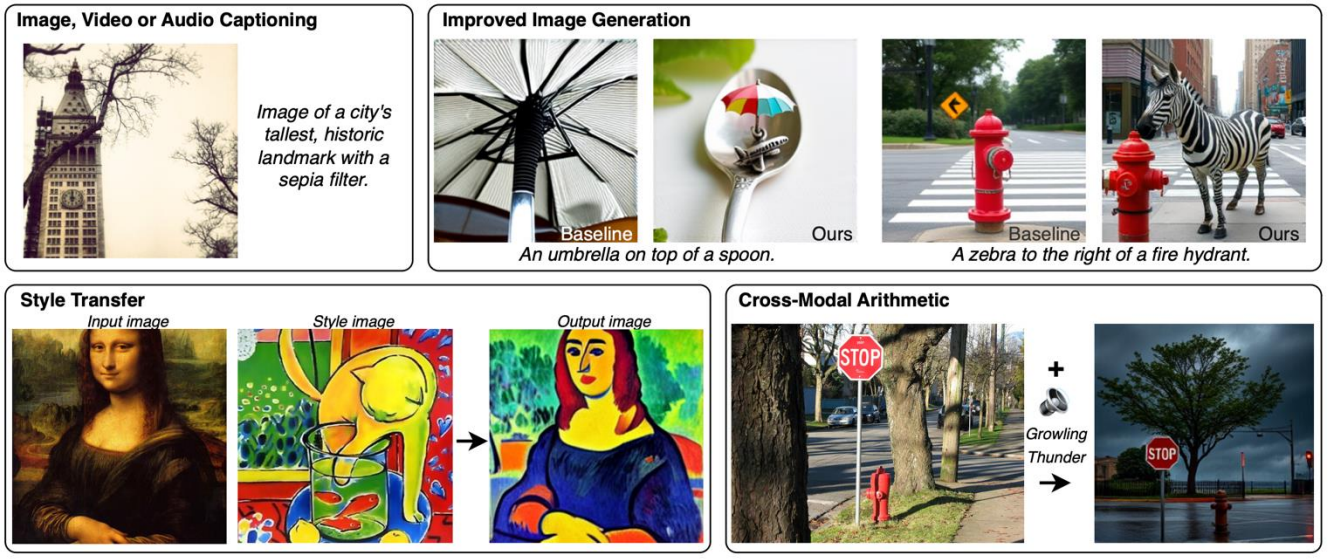
- **Contribution**
  - **Multimodal task 해결에 LLM의 reasoning 활용.**
    - 특수하게 학습된 멀티모달 모델 없이, 멀티모달 테스크를 수행한다.
      - 별도의 학습된 MLLM 모델 사용 없이, 일반 LLM, 그리고 CLIP과 같은 학습된 멀티모달 scorer를 주로 활용한다.
    - 그리고 LLM의 추론능력을 이용해 최적화 된 텍스트를 찾을 때까지 추론을 반복한다.



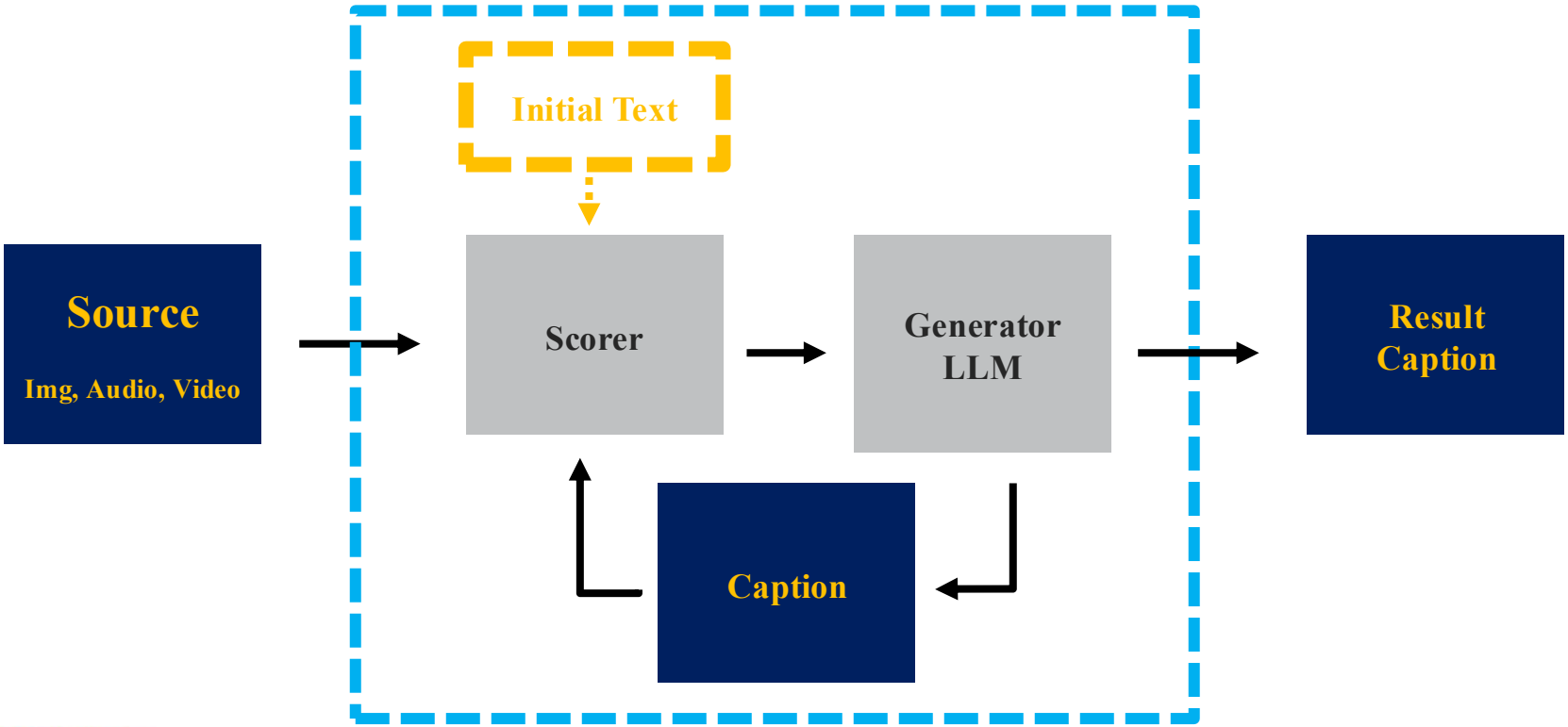
- **Architecture**
  - **4가지의 task를 목적으로 함.**
    - Source Captioning
      - 이미지, 비디오, 오디오에 대해 가장 잘 설명하는 텍스트를 찾는 것.
    - Improved Image Generation
      - 초기의 텍스트에 대해서 가장 잘 나타내는 이미지를 생성하는 것.



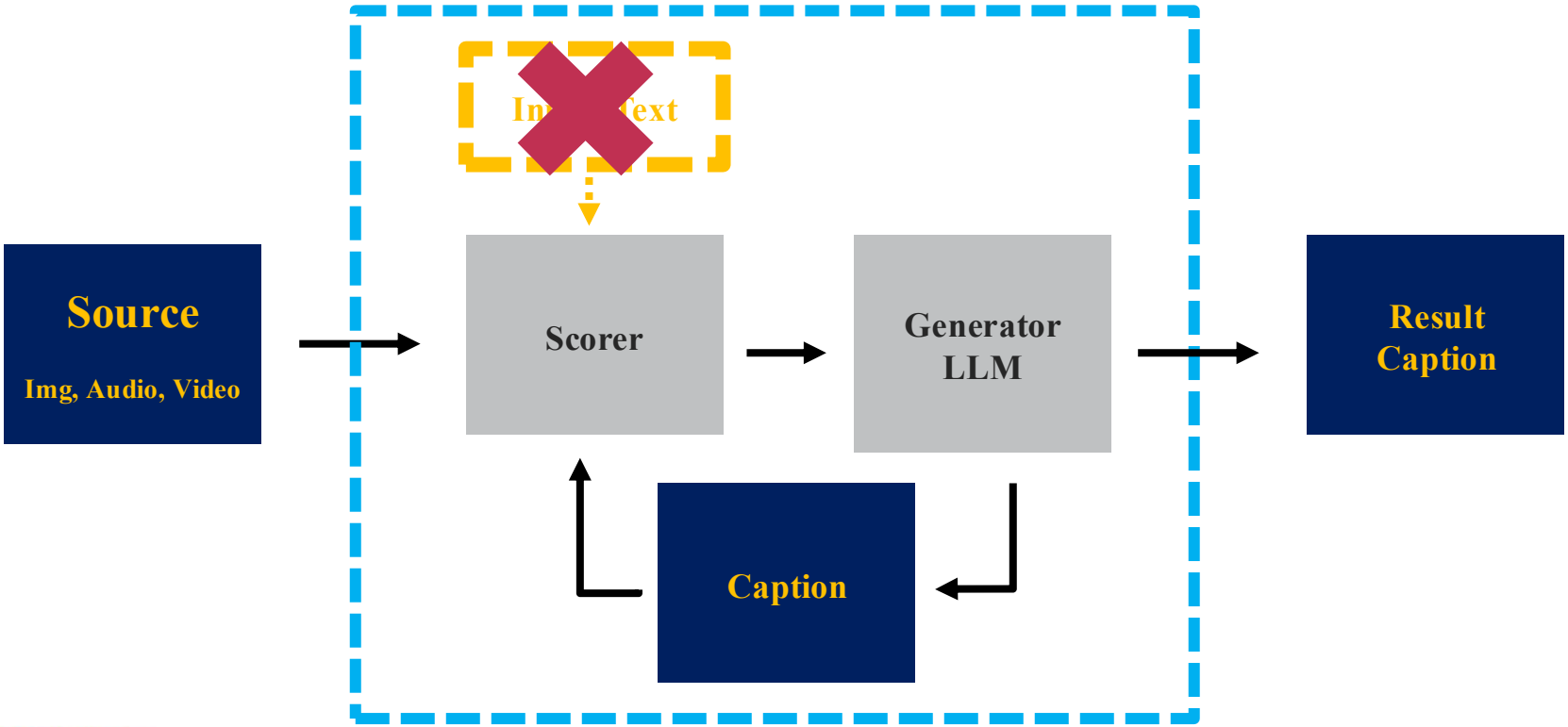
- **Architecture**
  - **4가지의 task를 목적으로 함.**
    - Style Transfer
      - 원본 이미지를 원하는 스타일의 이미지처럼 변형시키는 작업.
    - Cross—Modal Arithmetic
      - 이미지와 오디오를 텍스트로 설명하고, 그 둘의 정보를 담은 이미지 생성.



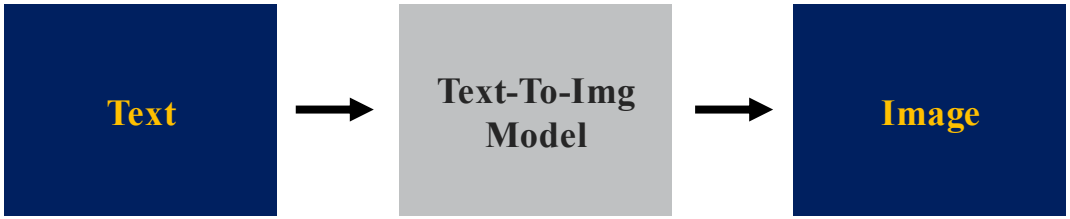
- **Architecture**
  - **Source Captioning.**
    - Source를 설명하는 캡션 텍스트를 LLM이 출력하는 것이 목표.
    - 초기에 저장된 다량의 텍스트셋과 Source를 scorer를 통해 유사도를 계산.



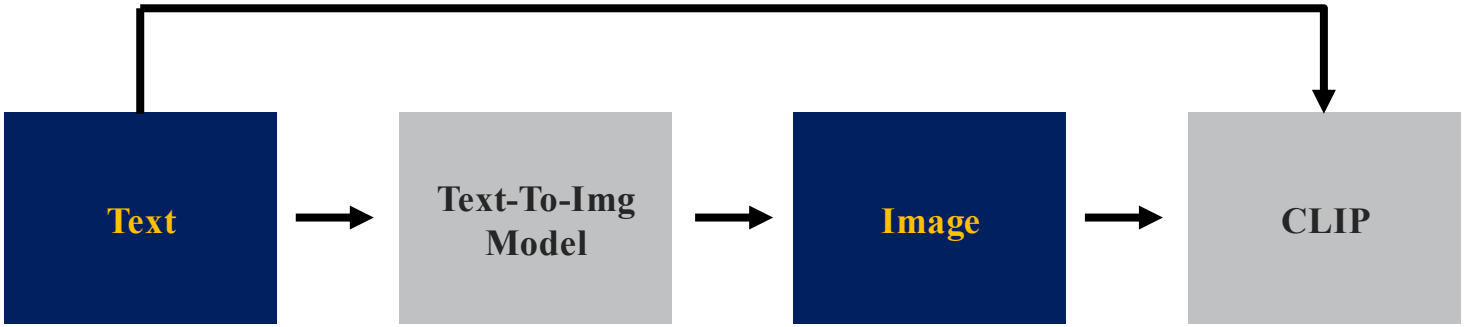
- **Architecture**
  - **Source Captioning.**
    - 이후 유사도와 텍스트를 LLM에게 입력하여, 현재 상황이 이러하므로, 유사도 점수를 더 올릴 수 있는 캡션 텍스트를 여러 개 만들어 달라고 요청.
    - 이후, 유사도 계산과 LLM 출력을 반복하여 최종적으로 캡션 텍스트를 획득.



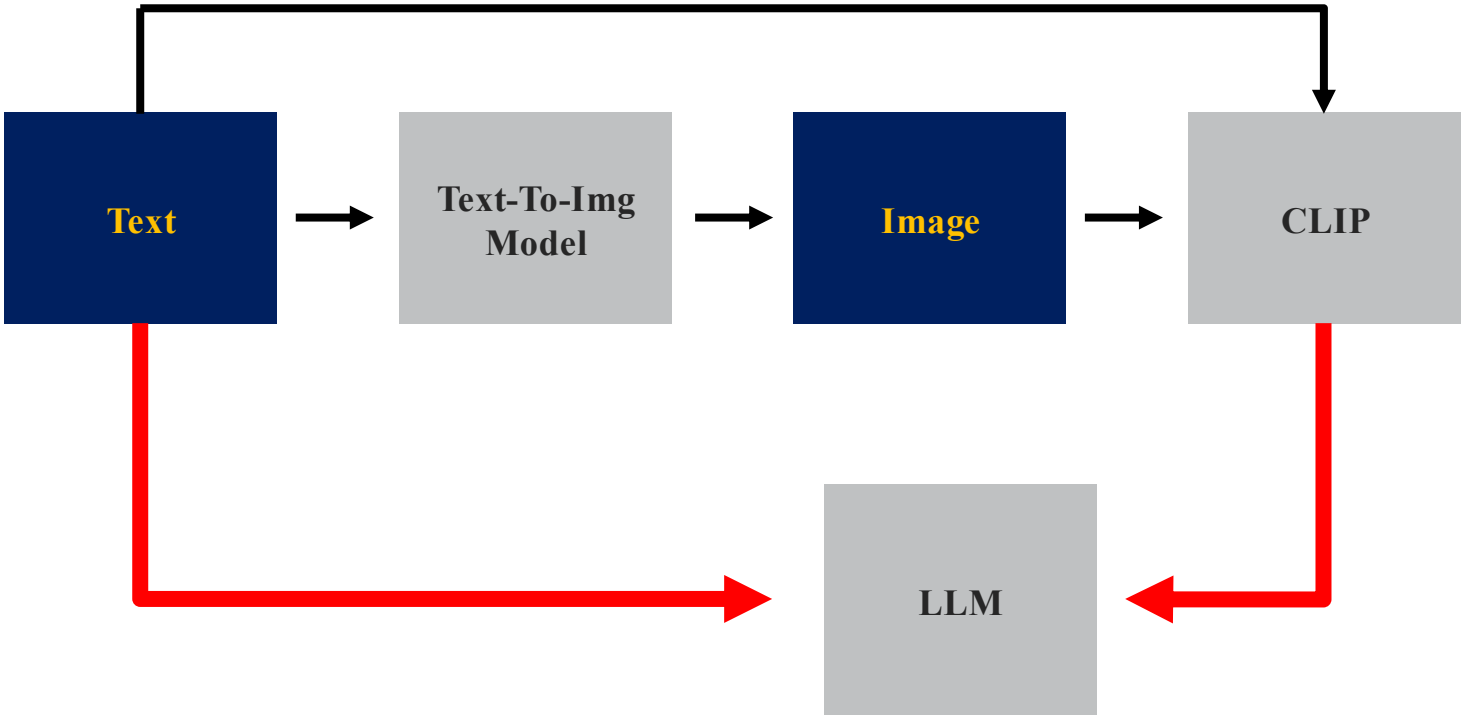
- **Architecture**
  - **Image Generation.**
    - 먼저, 유저가 초기에 원하는 이미지를 생성하기 위해 텍스트 프롬프트를 작성.
    - Text-To-Image 모델로 이미지 생성.



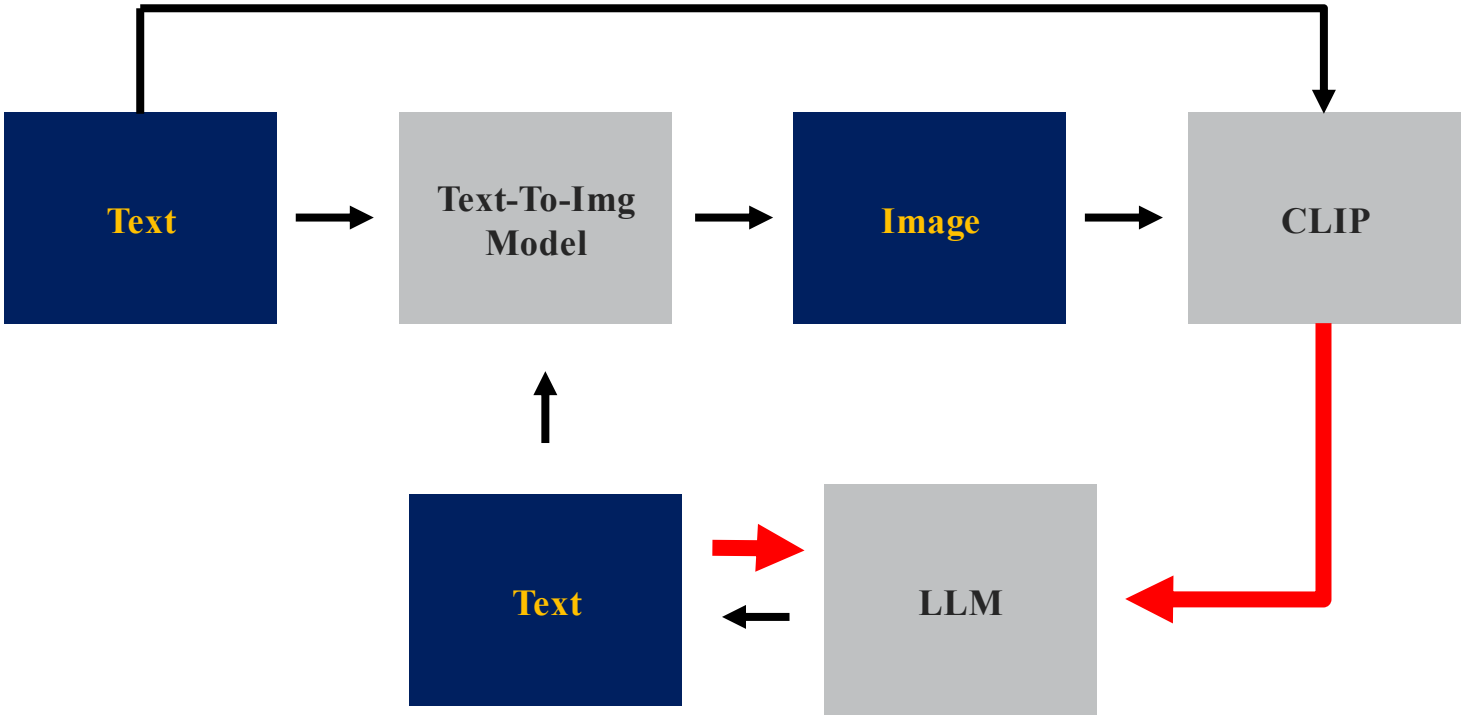
- **Architecture**
  - **Image Generation.**
    - 이후, 생성된 이미지와 텍스트 프롬프트 간의 유사도 점수를 구하여, 얼마나 원하는 대로 나왔는지 점수를 매김.



- **Architecture**
  - **Image Generation.**
    - 이후 유사도 점수와 텍스트 프롬프트를 LLM에게 전달.
    - 그리고 추가적으로 “현재 상황이 이러하므로 더 나은 텍스트를 생성하라” 라고 명령.



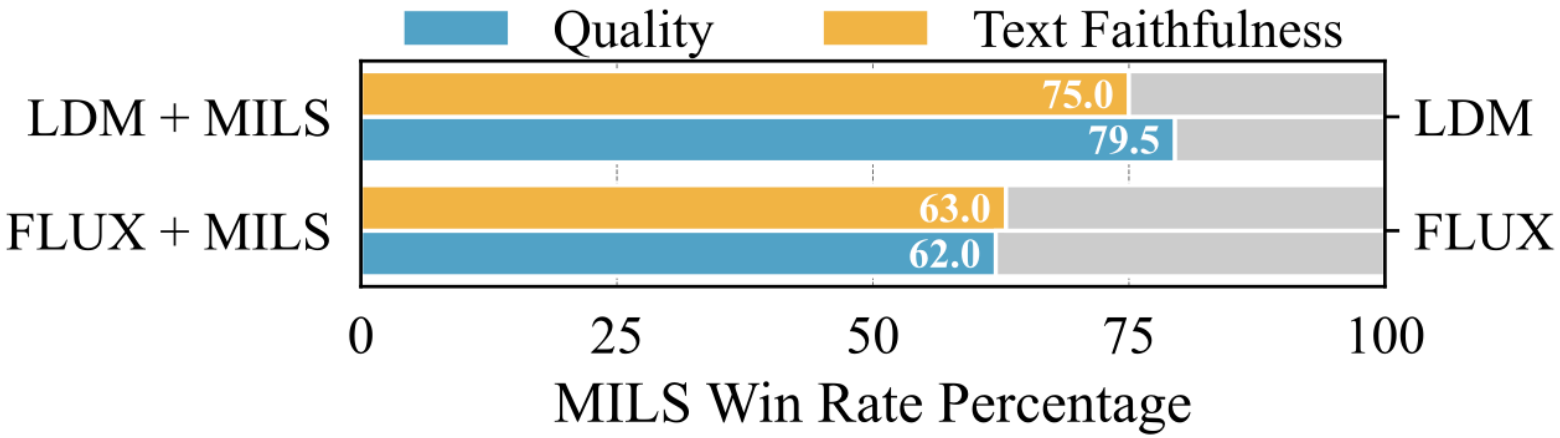
- **Architecture**
  - **Image Generation.**
    - 생성된 텍스트는 Text-To-Img 모델에 들어가게 되어, 루프를 반복.
    - 새롭게 생성된 텍스트를 CLIP에 넣으면 매번 기준이 고정되지 않으므로 초기 텍스트만 주입.



- **Result**
  - 결과 예시

#	1	4	8	12	16	20
Prompt rewrite	*A red car and a white sheep.*	*A white sheep and a red car in a stunning, natural landscape with a few mountains and a lake.*	*A red car and a white sheep standing in a stunning, rural landscape with a red barn, a few trees, and a few birds.*	*A red car and a white sheep standing in a stunning, rural landscape with a red barn, a few trees, and a few birds.*	*A red car and a white sheep in a stunning, natural landscape with a few mountains, a few wildflowers, and a serene lake.*	*A red car and a white sheep in a stunning, natural landscape with a few mountains, a few wildflowers, and a serene lake.*
Generated Image						

- **Result**
  - **Image Generation의 지표.**

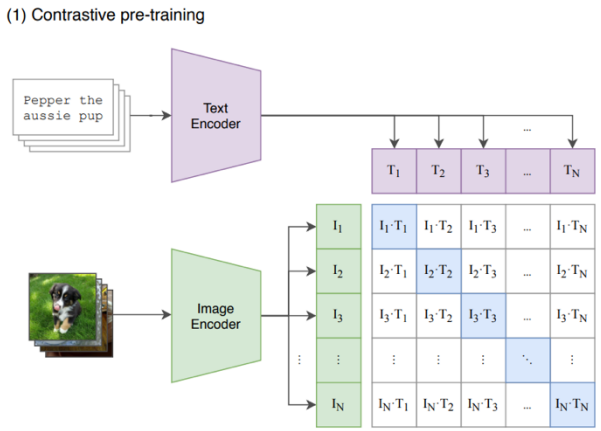


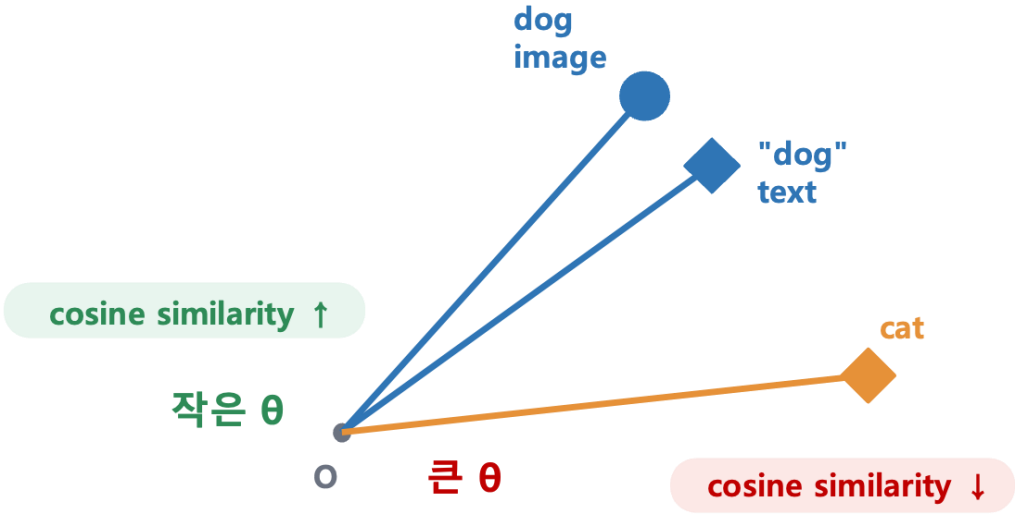
- **Limitation**
  - **Scorer의 의도와 사람의 의도가 같다는 보장이 없음.**
    - Scorer의 점수가 가장 높은 방향으로 수렴하지만, 그것이 사람이 가장 원하는 결과물이라고 보기 어려움.
    - 즉, Scorer가 보기에 가장 좋은 텍스트가 나오는 것이지, 사람이 보기에 최적의 텍스트라고 하기엔 어려움이 있음.

#	1	4	8	12	16	20
Prompt rewrite	*A red car and a white sheep.*	*A white sheep and a red car in a stunning, natural landscape with a few mountains and a lake.*	*A red car and a white sheep standing in a stunning, rural landscape with a red barn, a few trees, and a few birds.*	*A red car and a white sheep standing in a stunning, rural landscape with a red barn, a few trees, and a few birds.*	*A red car and a white sheep in a stunning, natural landscape with a few mountains, a few wildflowers, and a serene lake.*	*A red car and a white sheep in a stunning, natural landscape with a few mountains, a few wildflowers, and a serene lake.*
Generated Image						

Vocabulary Scaling Law:
Tuning Open-Vocabulary Predictors For Their Openness
[CVPR 2026 Highlight]

- **Background**
  - **CLIP.**
    - CLIP은 원하는 텍스트와 원하는 이미지의 임베딩 공간이 가까워 지고 다른 텍스트 임베딩은 원래 이미지 임베딩에서 멀어지게 하는 것을 목표.

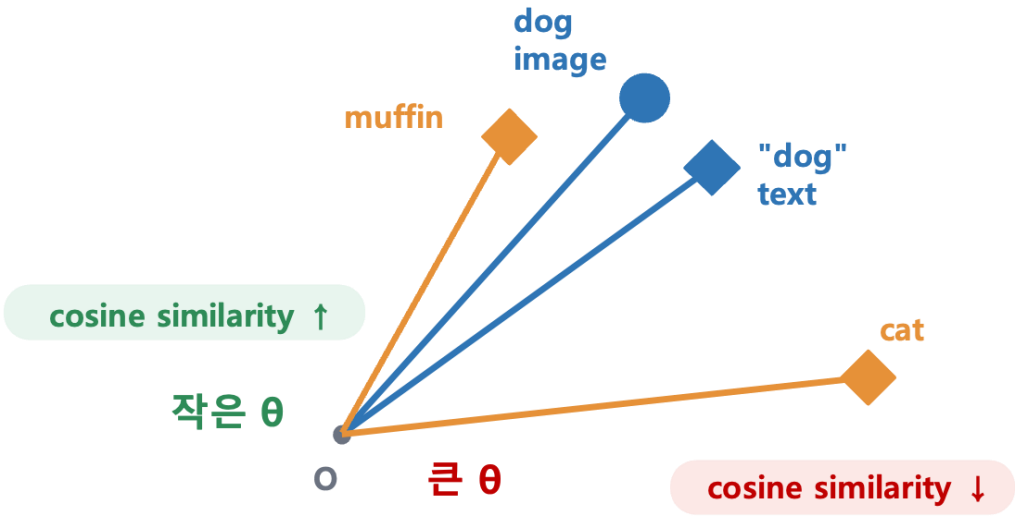




- **Background**
  - **CLIP 메커니즘의 약점.**
    - 그러나, classification 과정에서 원래 맞춰야 할 클래스가 아니라 다른 유사도가 높은 클래스가 선택되는 경우도 존재함.
      - 예를 들어, 머핀의 이미지를 치와와로 classification 할 수 있다.
    - 그러한 이유가 무엇인지 분석이 필요했음.

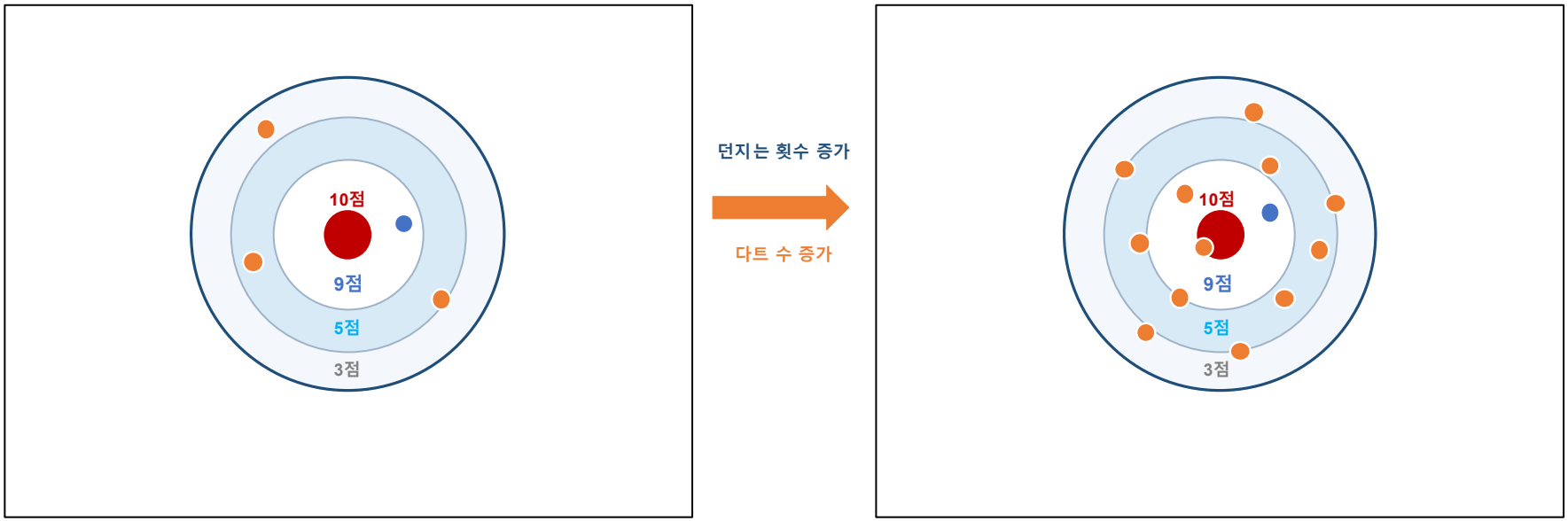


- **Background**
  - **CLIP 메커니즘의 약점.**
    - 이는 CLIP의 관심사는 오로지 텍스트와 이미지 임베딩 사이의 거리이기 때문이다.
      - Dog의 이미지 임베딩이 dog의 텍스트 임베딩과 가까워지고, 머핀과 cat의 텍스트 임베딩은 Dog의 이미지 임베딩으로부터 멀어지게 하는 것이 맞다.
      - 하지만, dog 텍스트 임베딩과 muffin 텍스트 임베딩끼리 멀어지게 하는 제약은 없으므로, 두 텍스트 임베딩이 가깝게 존재할 가능성이 존재한다.

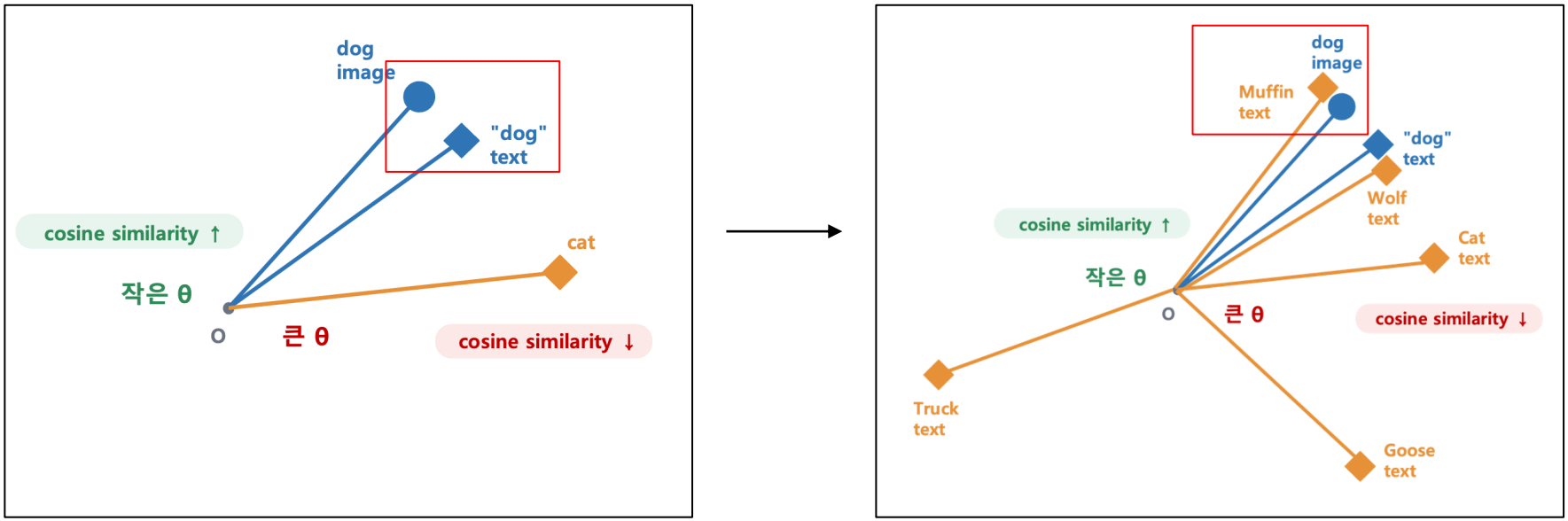


- **Motivation**
  - **CLIP의 성능은 Open-Vocabulary 환경의 클래스 수가 늘어날 수록 성능이 하락함.**
    - Open-Vocabulary 환경에서 CLIP의 Classification 성능이 떨어지는 경우가 존재함.
    - 따라서 현재 논문은 클래스 수가 늘어나더라도 성능이 떨어지지 않게 하는 것을 목표로 하여 문제의 원인을 제기하고 해결책을 제안한다.

- **Previous Limitation**
  - **클래스 수가 늘어날 수록 성능이 하락하는 이유.**
    - 먼저 간단한 예시를 들고자 함.
    - 다트판에 다트를 던질 때, 10점에 맞을 확률은 시행횟수가 많아질 수록 높아짐.
      - 현재 자신은 9점인데, 다른 사람이 10점을 맞출 확률은 시행횟수가 많아질 수록 커짐.



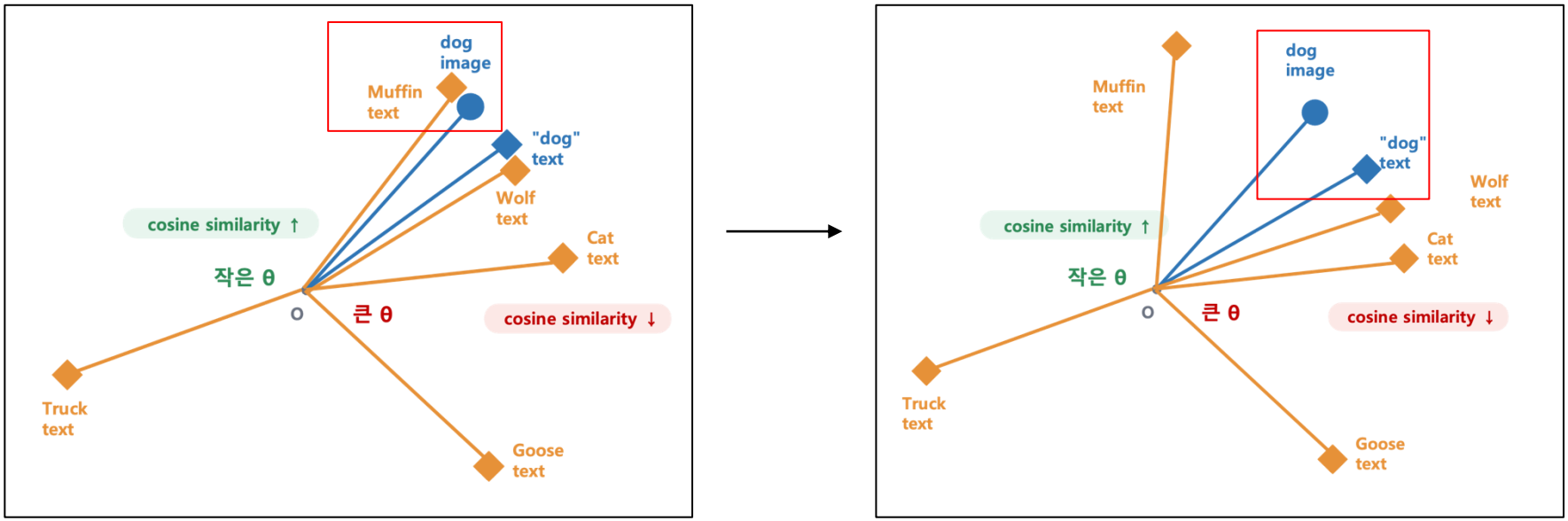
- **Previous Limitation**
  - 클래스 수가 늘어날 수록 성능이 하락하는 이유.
    - 이처럼, Open-Vocabulary에서 문제는 클래스 수가 매우 크게 늘어나게 될 경우 정답 클래스보다 similarity가 큰 클래스가 하나쯤은 생길 가능성이 생기게 된다.



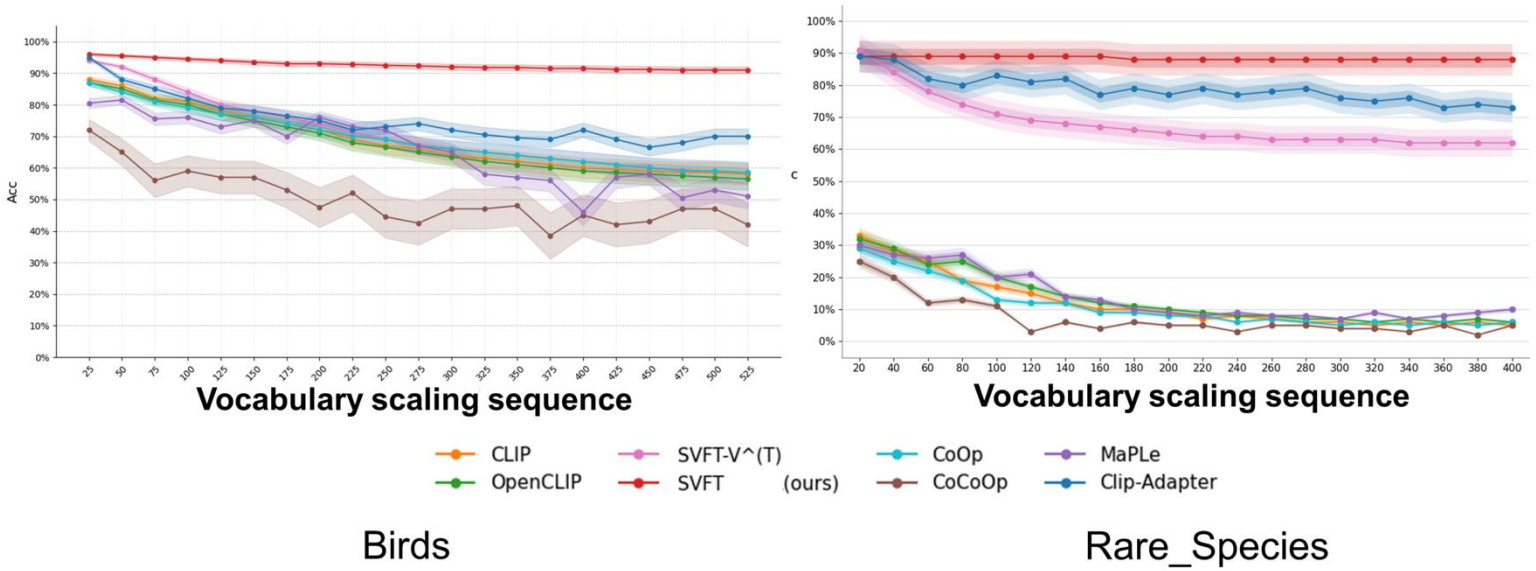
$$F(\{\bar{e}(w)\}, \hat{S}) \geq (1 - \frac{1}{e}) F(\{\bar{e}(w)\}, S^*). \quad (11)$$

$$\min_{\{e(w)\}_{\forall w \in S \cup \mathcal{V}^{(T)}}} \max_{S \subset U / \mathcal{V}^{(T)}, |S| \leq K} F(\{e(w)\}, S)$$
$$:= \mathbb{E}_{\substack{<\mathbf{x},\mathbf{y}> \\ \sim \mathcal{D}^{\text{train}}_{\mathcal{V}^{(T)}}}} \Big[-\log P^{(w_\mathbf{y})}_{f,g}(\mathbf{x}, S \cup \mathcal{V}^{(T)})$$
$$+\lambda \mathbb{E}_{\substack{w \in \mathcal{V}^{(T)}, \\ w' \in S}} \Big(\langle g(\boldsymbol{T}(e(w))), g(\boldsymbol{T}(e(w'))) \rangle + 1 \Big) \Big] \quad (10)$$

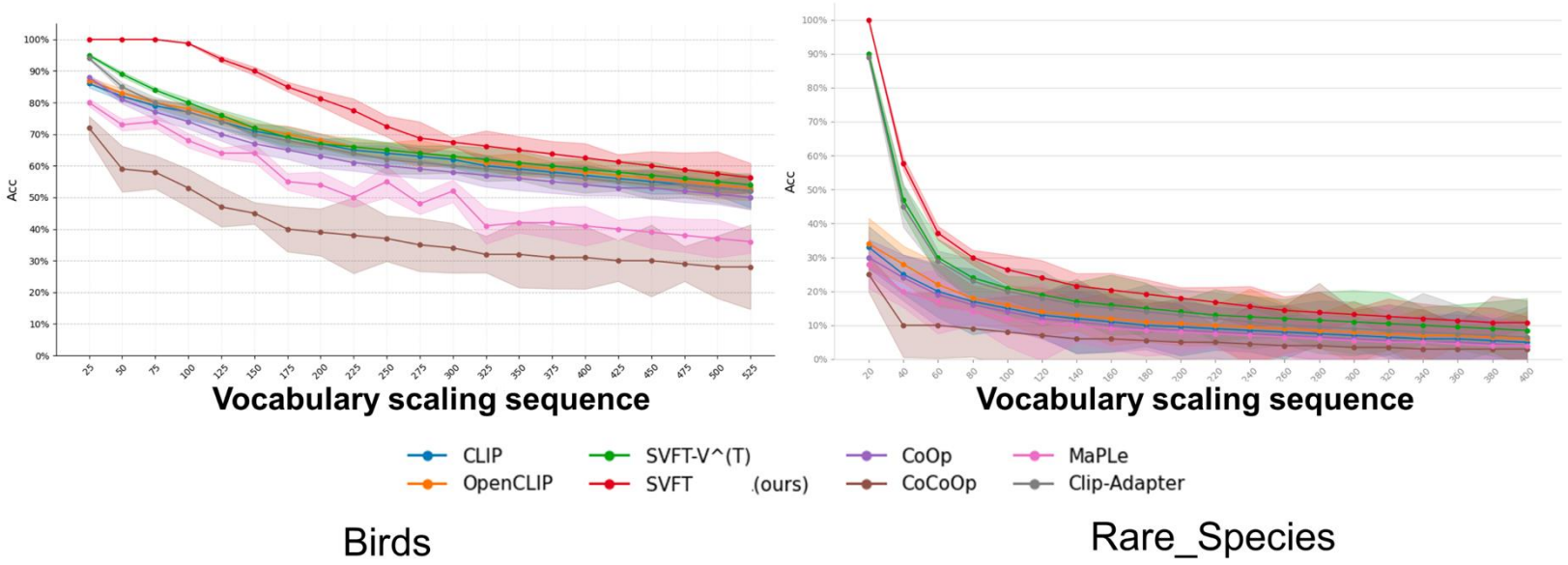
- **Method**
  - 논문의 해결책.
    - 이 논문은 클래스가 늘어나더라도, 헷갈릴 수 있는 클래스들을 찾아내는 수식과, 클래스 이름 임베딩끼리만 거리를 벌려 놓도록 새로운 로스를 제안함.



- **Result**
  - **Target Vocabulary 클래스들은 다른 클래스들이 추가되어도 잘 맞춘다.**
    - 기존의 모델들은 클래스 수가 늘어날 수록 Target Vocabulary에 대해 classification 성능이 떨어지는 반면, 이 논문의 모델은 대비하여 높은 성능을 유지한다.



- **Limitation**
  - **Target Vocabulary에 대해 잘 맞추는 것만 목적으로 한다.**
    - Fine-tuning 이후, Target Vocabulary에 대해선 잘 맞추지만, 그 외의 클래스에 대해서는 새롭게 이미지를 넣어서 맞추도록 하면 여전히 성능 저하가 존재하는 한계가 있음.

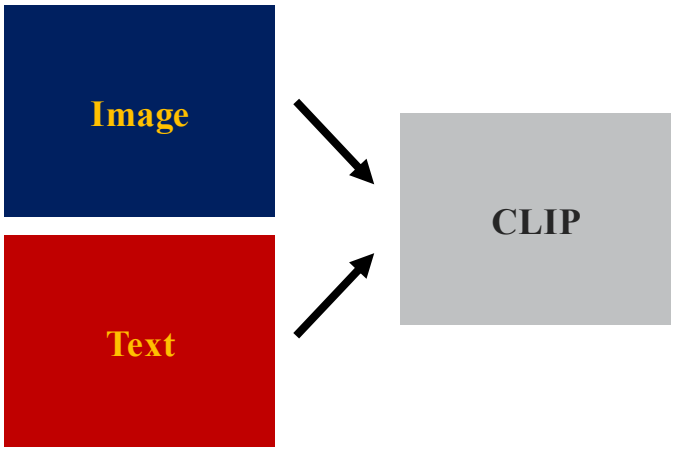


- **추후 계획**
  - **오디오 내에서 여러 인스턴스가 나오는 것을 최대한 구별하는 것을 목표로 하고 있음.**
    - LLM을 통해 오디오 이벤트를 찾아낼 수 있는 텍스트를 루프를 돌려 뽑을 예정.
    - 다량으로 뽑아낸 텍스트 중 헷갈리는 텍스트들은 두번째 논문 기반으로 정제할 가능성이 있다고 판단하고 있음.

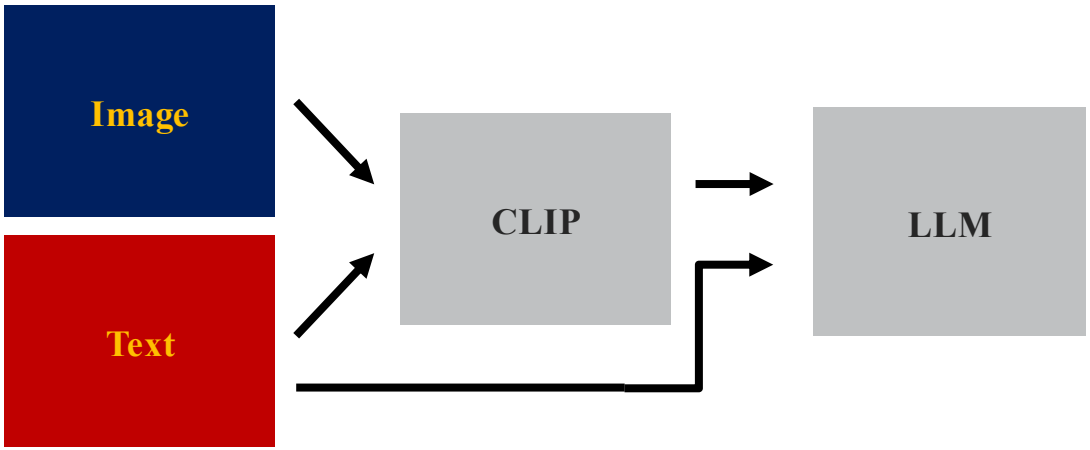
감사합니다

Appendix

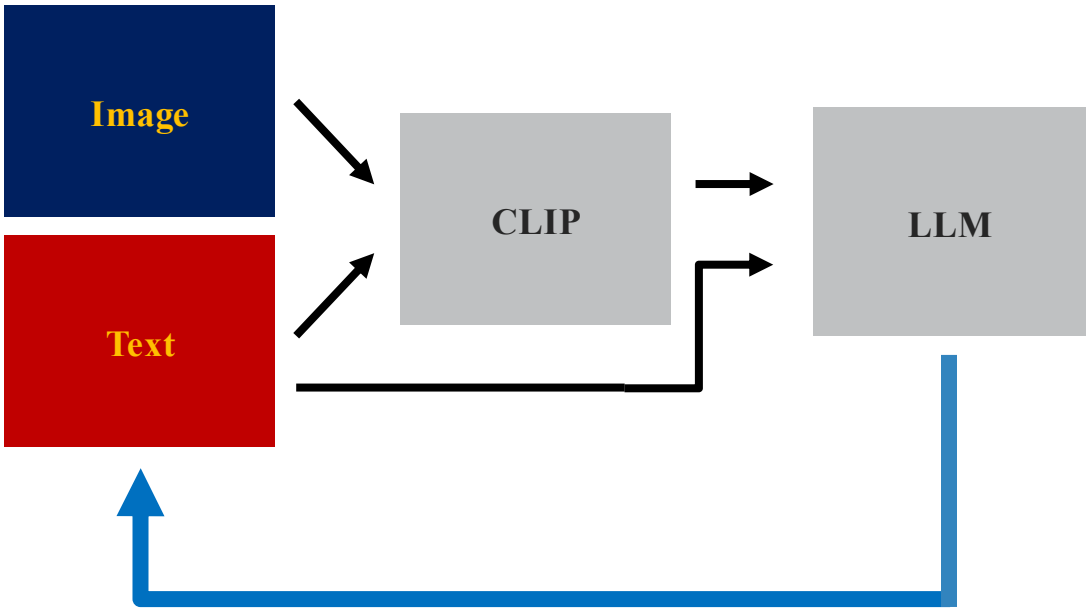
- **Architecture**
  - **Generator와 Scorer의 반복 구조.**
    - 이미지를 설명하는 텍스트가 초기에 존재하고, 그것을 CLIP을 통해 일차적으로 유사도 도출.



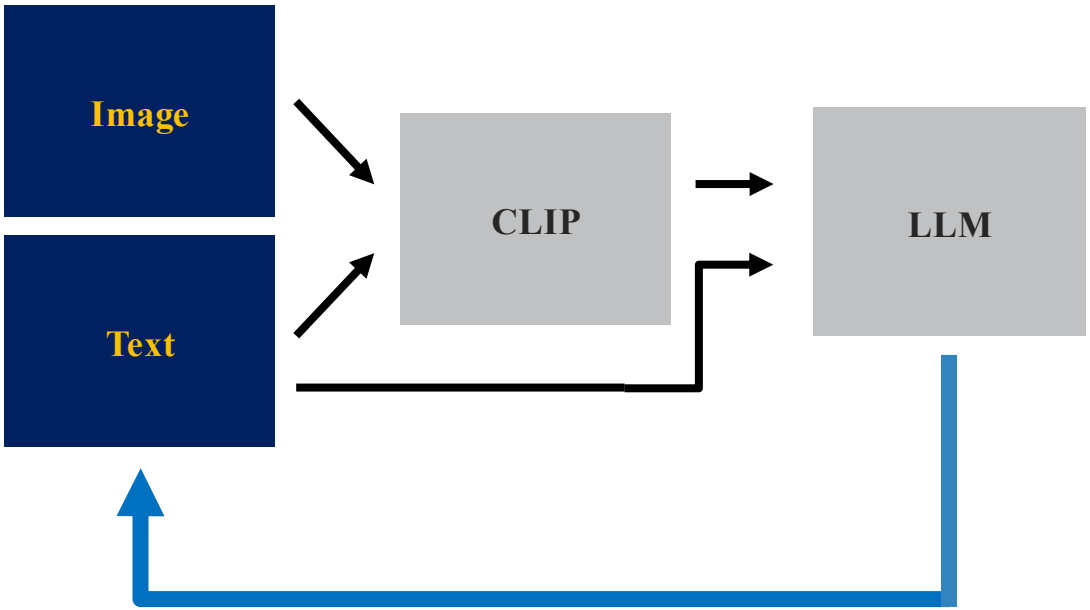
- **Architecture**
  - **Generator와 Scorer의 반복 구조.**
    - CLIP을 통해 얻어낸 유사도 결과와 CLIP에 들어간 텍스트를 LLM에 입력하여, 개선할 수 있는 텍스트를 도출하라고 명령.



- **Architecture**
  - **Generator와 Scorer의 반복 구조.**
    - LLM은 4개의 새로운 텍스트들을 출력하고, 이는 다시 CLIP으로 들어간다.
    - 그 4개 텍스트의 결과를 다시 LLM에게 전달하여 다시 4개의 텍스트를 뱉는 루프 형태.



- **Architecture**
  - **Generator와 Scorer의 반복 구조.**
    - 루프가 최종적으로 돌았을 때, 이미지를 잘 설명하는 텍스트를 얻어낼 수 있음.



- **Method**
  - **논문은 alignment 성능을 유지하기 위해 여러 조건을 내세움.**

$$\min_{\{e(w)\}_{\forall w \in S \cup \mathcal{V}^{(T)}}} \max_{S \subset U/\mathcal{V}^{(T)}, |S| \leq K} F(\{e(w)\}, S)$$
$$:= \mathbb{E}_{\substack{<\mathbf{x},\mathbf{y}> \\ \sim D^{\text{train}}_{\mathcal{V}^{(T)}}}} \Big[-\log P^{(w_\mathbf{y})}_{f,g}(\mathbf{x}, S \cup \mathcal{V}^{(T)})$$
$$+\lambda \mathbb{E}_{\substack{w \in \mathcal{V}^{(T)}, \\ w' \in S}} \Big(\langle g(\boldsymbol{T}(e(w))), g(\boldsymbol{T}(e(w'))) \rangle + 1 \Big) \Big] \quad (10)$$

    - 마음대로 텍스트 임베딩 공간을 조정하게 되면, CLIP의 이미지와 텍스트 간 공통점을 찾는 alignment 성능을 저해할 수 있다.
      - 이미지 임베딩이 고정된 상태에서 마땅한 기준없이, 텍스트 임베딩끼리 거리를 멀게 만든다면, 다른 엉뚱한 클래스의 텍스트 임베딩이 목표의 이미지 임베딩과 매칭될 수 있다.
    - 따라서, alignment 성능을 저해하지 않기 위해 두 분류로 클래스를 나누어 각각에 제약을 다르게 주고자 한다.
      - Align이 잘 되길 원하는 Target Vocabulary, 그리고 Target Vocabulary가 아닌 클래스로 나뉨.
      - Target Vocabulary는 이미지와 클래스 라벨링이 되어, 이미지와 텍스트의 임베딩이 가까워지게 함.
        - ✓Target Vocabulary의 alignment 성능은 유지되도록 유도.
      - 동시에, Target Vocabulary와 나머지 클래스들은 서로가 아닌, 다른 클래스의 텍스트 임베딩은 멀어지도록 제약을 추가함.
        - ✓Target Vocabulary외의 클래스들은 관심사가 아니기에, alignment가 깨어져도 상관이 없다.

- **Method**

$$F(\{\bar{e}(w)\}, \hat{S}) \geq (1 - \frac{1}{e})F(\{\bar{e}(w)\}, S^*). \quad (11)$$

  - 논문은 텍스트 임베딩을 조정하는 데에 다음과 같은 조건을 중요시한다.
    - 그러나, Open-Vocabulary 전체의 클래스 수가 많아질 수록, 모든 클래스를 조정하는 것은 비용을 많이 필요로 함.
    - 따라서, Target Vocabulary를 가장 헷갈리게 하는 몇몇개의 클래스들만 서브셋으로 골라서 조정함.
    - 추가적으로, alignment 저하를 최소화 하기 위해 비주얼 인코더와 텍스트 인코더는 fine-tuning 하지 않고, 오로지 텍스트 임베딩 내 클래스 토큰에 해당하는 임베딩만 fine-tuning한다.
    - 클래스 토큰에 해당하는 임베딩만 fine-tuning한다.
      - 텍스트 임베딩 :" a photo of {class}" 전체에 대한 임베딩
      - context 임베딩 : "a photo of"에 해당하는 임베딩
      - 클래스 토큰 임베딩 : "{class}" 에 해당하는 임베딩