
A Survey on Multimodal Large Language Models for Anomaly Detection

김규형

Overview of Previous Presentation

- VMAD: Visual-enhanced Multimodal Large Language Model for Zero-Shot Anomaly Detection, **IEEE 2024**
- Towards Zero-Shot Anomaly Detection and Reasoning with Multimodal Large Language Models, **CVPR 2025**

VMAD: Visual-enhanced Multimodal Large Language Model for Zero-Shot Anomaly Detection

Huilin Deng, Hongchen Luo, Wei Zhai, Yang Cao, *Member, IEEE*, Yu Kang, *Senior Member, IEEE*

Abstract—Zero-shot anomaly detection (ZSAD) recognizes and localizes anomalies in previously unseen objects by establishing feature mapping between textual prompts and inspection images, demonstrating excellent research value in flexible industrial manufacturing. However, existing ZSAD methods are limited by closed-world settings, struggling to unseen defects with predefined prompts. Recently, adapting Multimodal Large Language Models (MLLMs) for Industrial Anomaly Detection (IAD) presents a viable solution. Unlike fixed-prompt methods, MLLMs exhibit a generative paradigm with open-ended text interpretation, enabling more adaptive anomaly analysis. However, this adaption faces inherent challenges as anomalies often manifest in fine-

(a) Conventional Zero-shot Anomaly Detection Models

(b) Multimodal Large Language Models Zero-shot Anomaly Detection

Towards Zero-Shot Anomaly Detection and Reasoning with Multimodal Large Language Models

Jiacong Xu^{1*} Shao-Yuan Lo² Bardia Safaei¹ Vishal M. Patel¹ Isht Dwivedi²
¹Johns Hopkins University ²Honda Research Institute USA
{jxu155, bsafaei1, vpatel36}@jhu.edu {shao-yuan.lo, idwivedi}@honda-ri.com

Abstract

Zero-Shot Anomaly Detection (ZSAD) is an emerging AD paradigm. Unlike the traditional unsupervised AD setting that requires a large number of normal samples to train a model, ZSAD is more practical for handling data-restricted real-world scenarios. Recently, Multimodal Large Language Models (MLLMs) have shown revolutionary reasoning capabilities in various vision tasks. However, the reasoning of

Dataset	BTAD	ELPV	AITEX	VisA	MVTec	MPDD
Model 1	88.0	80.0	77.0	91.3	94.0	81.2
Model 2	88.0	80.0	77.0	91.3	94.0	81.2
Model 3	88.0	80.0	77.0	91.3	94.0	81.2
Model 4	88.0	80.0	77.0	91.3	94.0	81.2
Model 5	88.0	80.0	77.0	91.3	94.0	81.2
Model 6	88.0	80.0	77.0	91.3	94.0	81.2



Overview of Previous Presentation

- 1) 앞서 언급한 2개의 논문 자세히 알아보기
- 2) 2개의 논문에서 아쉬운 점으로 무엇이 있는지 생각하기
- 3) 연구 방향 생각하기
- 4) classification과 segmentation 중 어느 것에 집중할지 생각하기
- 5) 어떤 dataset을 사용할 것인지 생각하기

CONTENTS

01

VMAD : Visual-enhanced Multimodal Large Language Model for Zero-Shot Anomaly Detection

- Overall Architecture
- Projector
- Locality-enhanced Token Compression

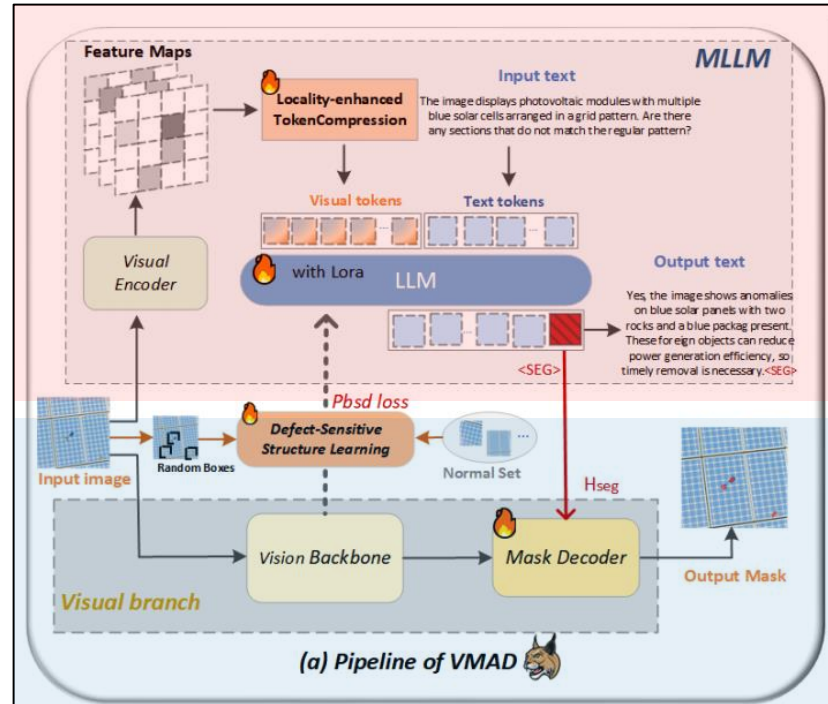
02

Towards Zero-Shot Anomaly Detection and Reasoning with Multimodal Large Language Models

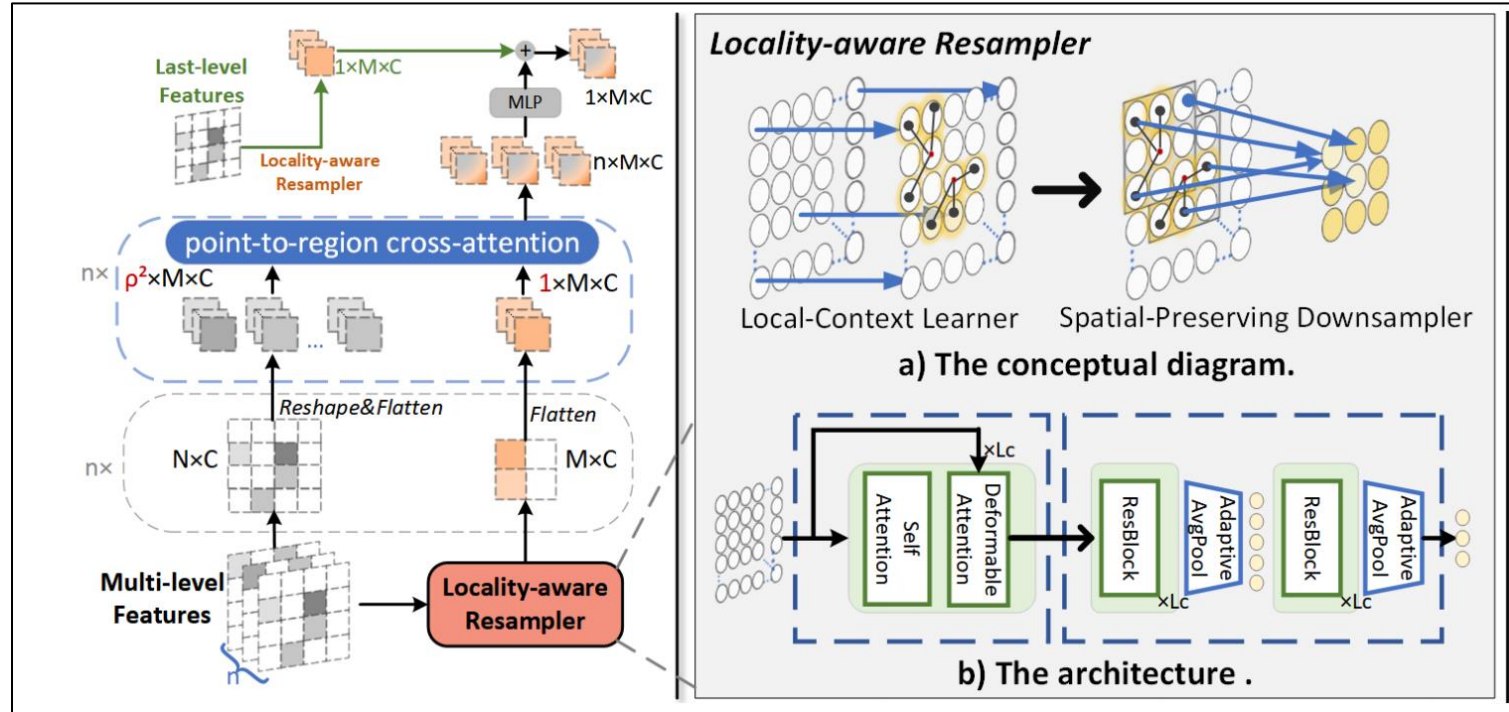
- Overall Architecture
- Look-Twice Feature Matching
- Visual Token Selector

03

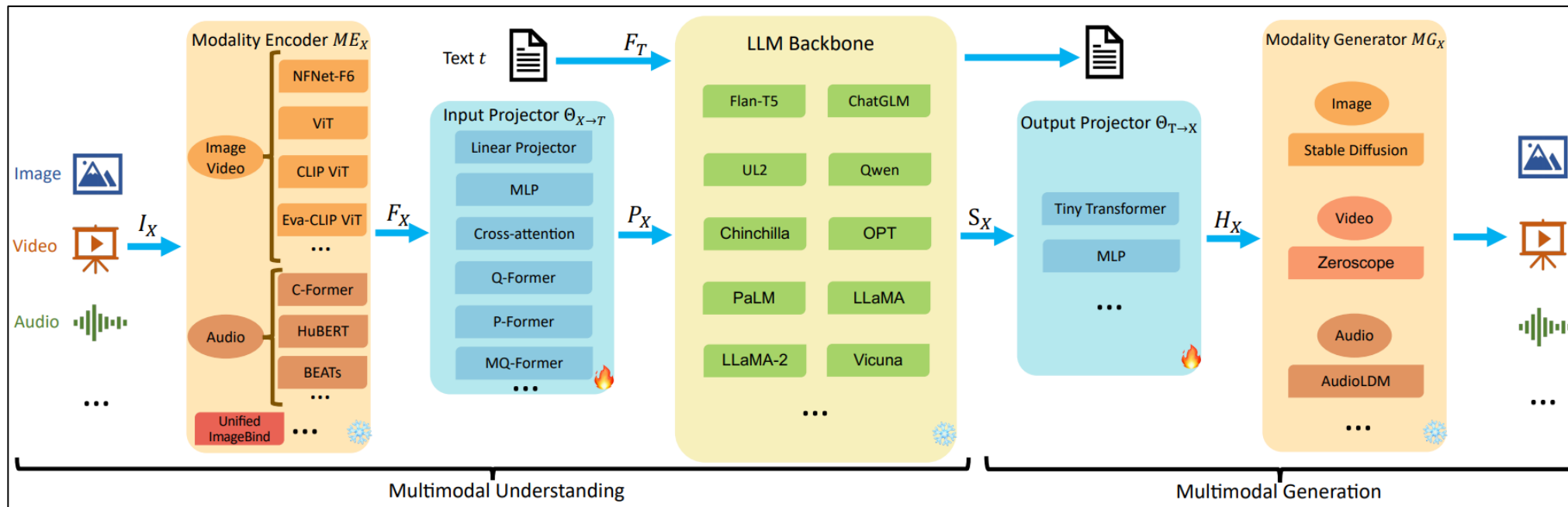
References



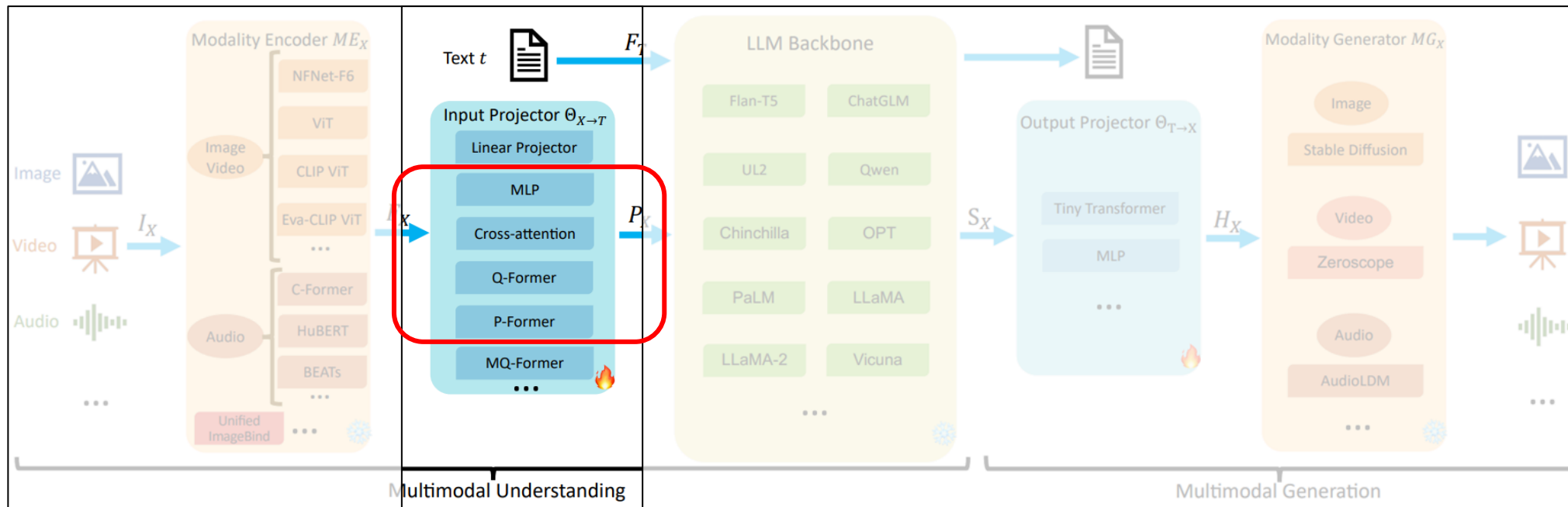
- Visual branch는 이미지의 visual feature를 효과적으로 추출하기 위해 구축함
- MLLM은 visual branch에서 추출한 visual feature와 text 정보를 효과적으로 결합하여 사용자에게 anomaly detection 결과를 보여줌
- Visual encoder : CLIP-ViT-L/14
- Vision Backbone : SAM



- 입력 이미지의 visual feature 정보를 해치지 않기 위해 Locality-enhanced Token Compression을 제안함
- Local-context에서 multi-level feature를 탐색함으로써, locality를 향상하는 downsampling을 수행함
- 여러 anomaly 모양과 크기를 처리하기 위해 convolution과 deformable attention을 결합함

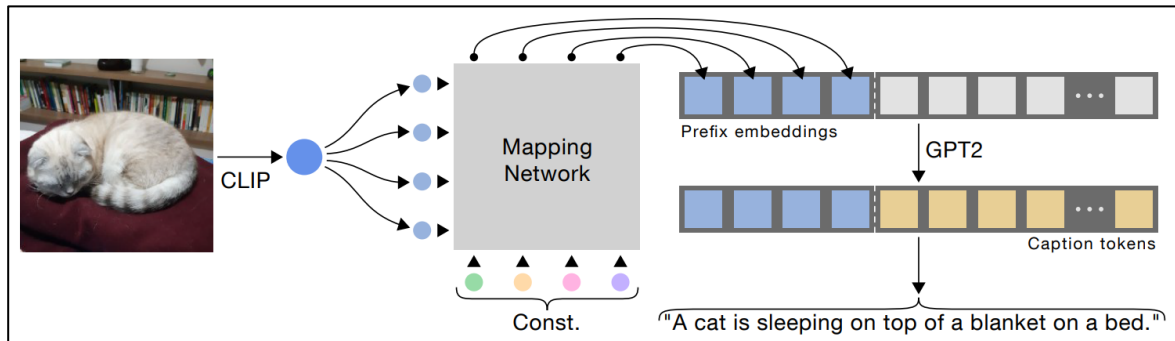


MM-LLMs: Recent Advances in MultiModal Large Language Models, arXiv 2024



MM-LLMs: Recent Advances in MultiModal Large Language Models, arXiv 2024

- ClipCap: CLIP Prefix for Image Captioning, arXiv 2021
- ClipCap은 상황에 따라 Mapping Network Architecture로 MLP와 Transformer-architecture 사용
 1. GPT-2를 fine-tuning 할 수 있을 때, MLP architecture 사용
 2. frozen 상태로 CLIP과 GPT-2를 사용할 때, Transformer-based architecture 사용
- 2개 모두 Mapping 하기 위해 사용 (Projector 역할)



ClipCap : transformer-based architecture

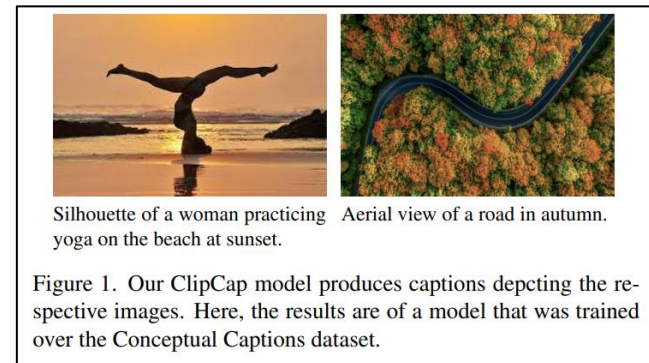
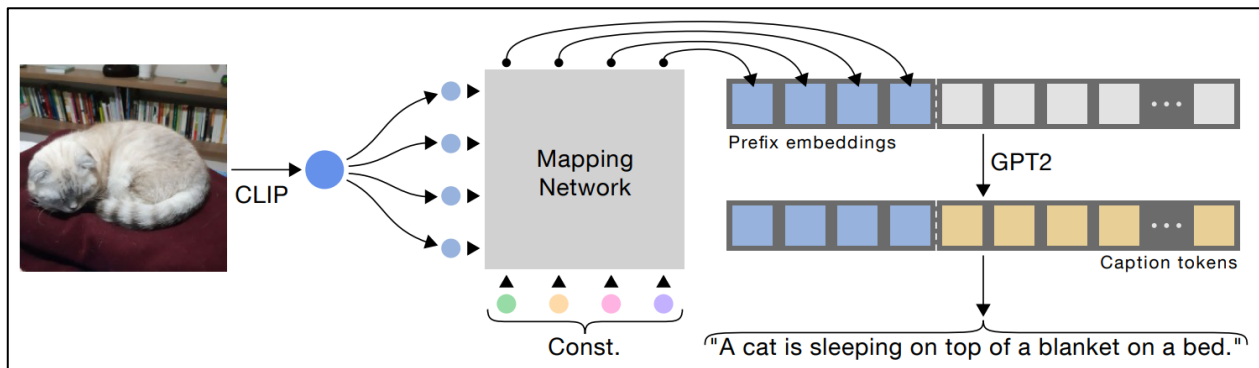
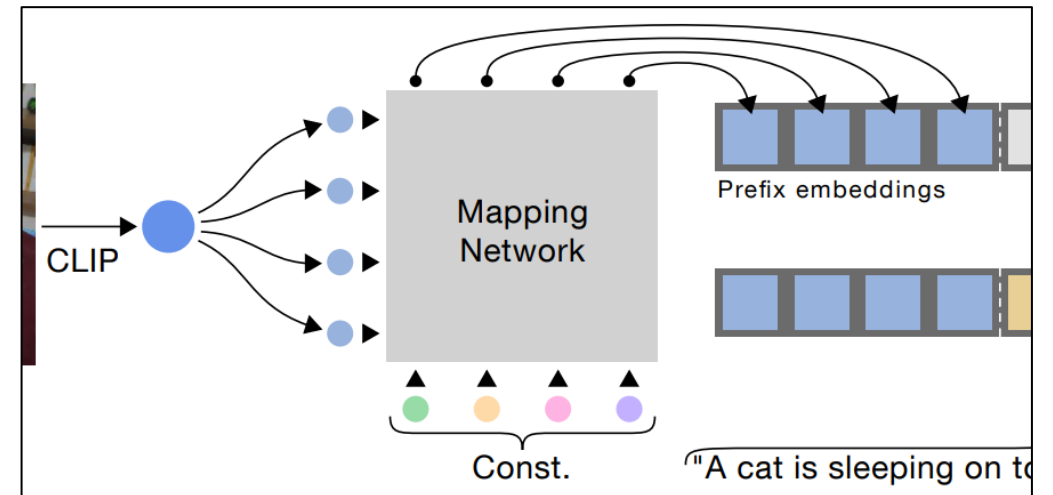


Image captioning

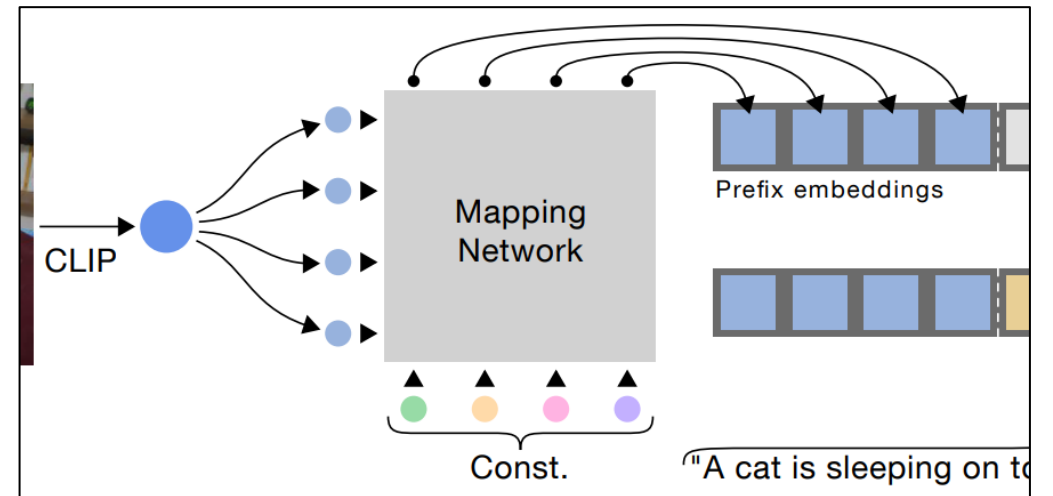
- Transformer-based architecture
 - frozen 상태의 CLIP과 GPT-2를 사용
 - Input : visual embedding, learned constant
 - GPT-2가 caption을 생성할 때 도움을 주는 prefix(fixed length)를 추출
 - prefix는 이미지의 정보를 담고 있고, 모델은 prefix를 통해 이미지를 이해
 - 학습할 때, CLIP과 GPT-2는 frozen 상태이기에 Mapping Network만 학습하면 됨
 - Learned constant를 key와 value로 사용하고, visual embedding을 query로 사용



- Learned constant를 사용한 이유
 - 숫자로 이루어진 벡터 형태이고, LLM embedding dimension과 동일한 차원을 가짐
 - Frozen 상태인 Language Model을 새로운 데이터에 잘 적응시키기 위해 사용
 - Constant는 학습을 통해 특정 차원을 가지게 됨
 - 'Prefix-Tuning: Optimizing Continuous Prompts for Generation' 논문에서 영감을 받음
- Transformer-based architecture를 사용한 이유
 - 각 입력 토큰 간에 global attention을 적용하여 서로 다른 입력 간의 관계를 학습
 - Multi-head attention을 통해 visual embedding에서 의미 있는 정보를 수집
 - 최종 output은 Language Model의 space로 들어갈 수 있는 벡터 시퀀스를 출력

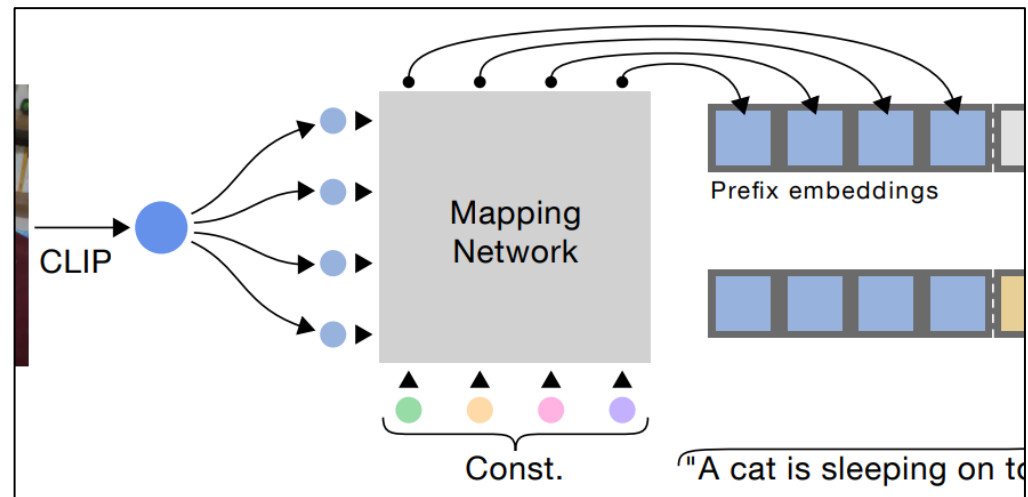


- MLP (Mapping Network)
 - Frozen 상태의 Language Model이 아닌, fine-tuning이 가능한 LM일 때 MLP 사용
 - Single hidden layer만 사용해도 현실적이고 의미 있는 caption을 생성할 수 있음
 - MLP 목적 또한, Image embedding을 GPT-2 모델이 이해할 수 있는 형태로 변환하는 것

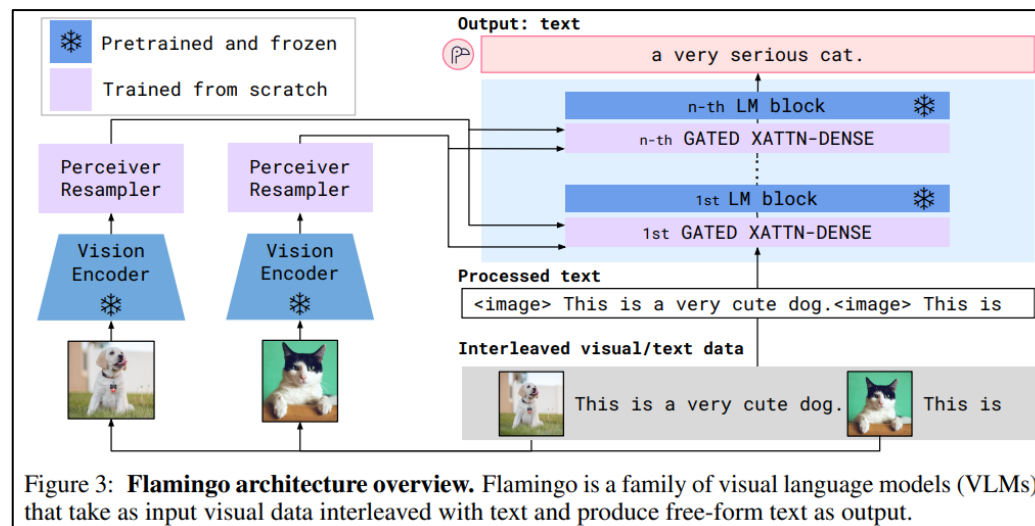


- MLP 단점

- 단순하지만, 복잡한 관계를 학습할 때 어려움이 있고 유연성이 부족함



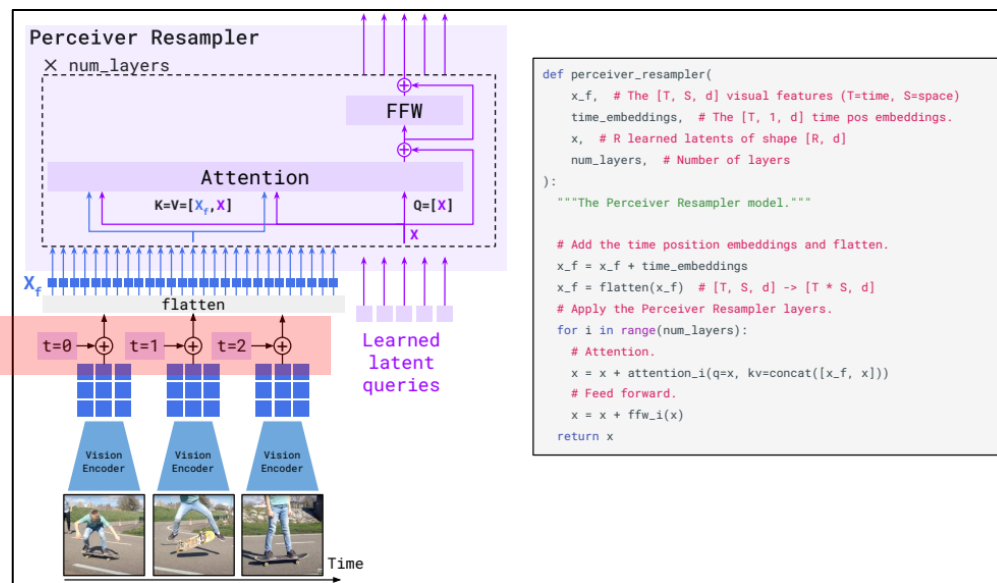
- Flamingo: a Visual Language Model for Few-Shot Learning, NeurIPS 2022
- Multimodal LLM의 시초로, few-shot learning을 통해 새로운 작업에 대해 빠르게 적응할 수 있는 모델을 구축
- Flamingo는 classification, captioning, question-answering 작업이 가능함
- 이미지를 인식할 수 있는 encoder, 기본적인 추론을 수행하는 LLM, 2개의 모델을 연결하는 architecture로 구성
- 32개의 예제만으로 fine-tuning 된 Flamingo는 최근 모델보다 좋은 성능을 보여줌



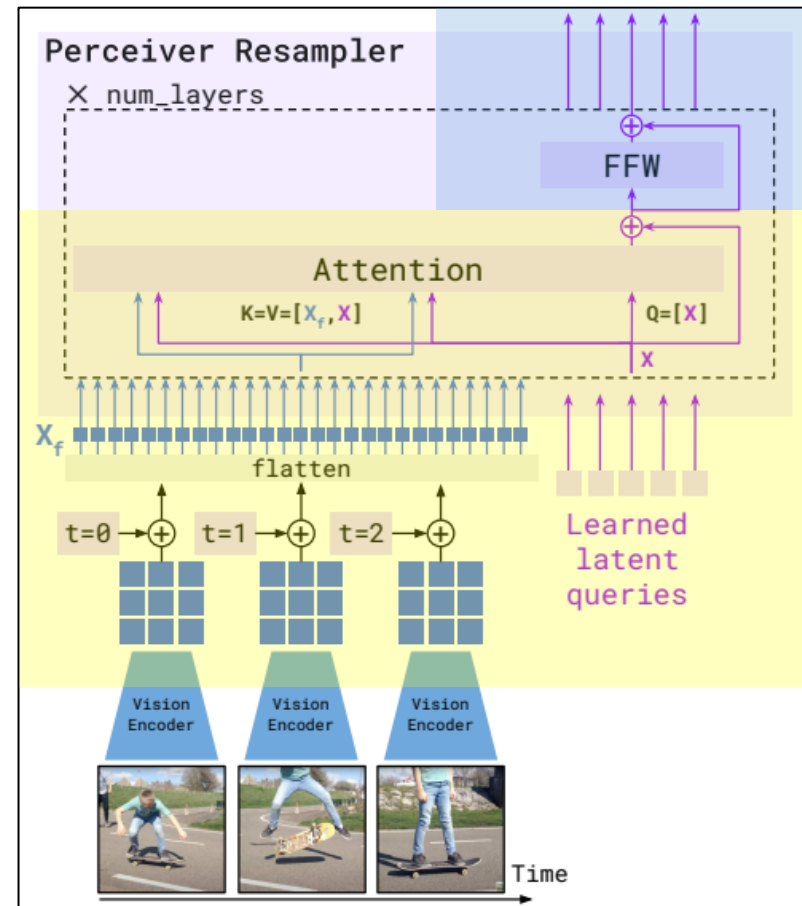
Resampler (cross-attention)

- 입력 데이터가 이미지일 때, Resampler는 1차원으로 flatten 된 visual feature와 학습된 latent query를 input으로 받음
- Visual feature와 학습된 latent vector를 결합하여 key와 value로 사용하고, 학습된 latent vector는 query로 사용함
- 2개의 input을 사용해 cross-attention을 수행함으로써, 고정된 visual output을 생성함 (논문에서는 64개를 생성)
- 고정된 visual output을 생성함으로써, vision-text cross-attention의 계산 복잡성을 줄임
- Ablation study를 통해, plain Transformer와 MLP보다 성능이 좋다는 것을 확인함

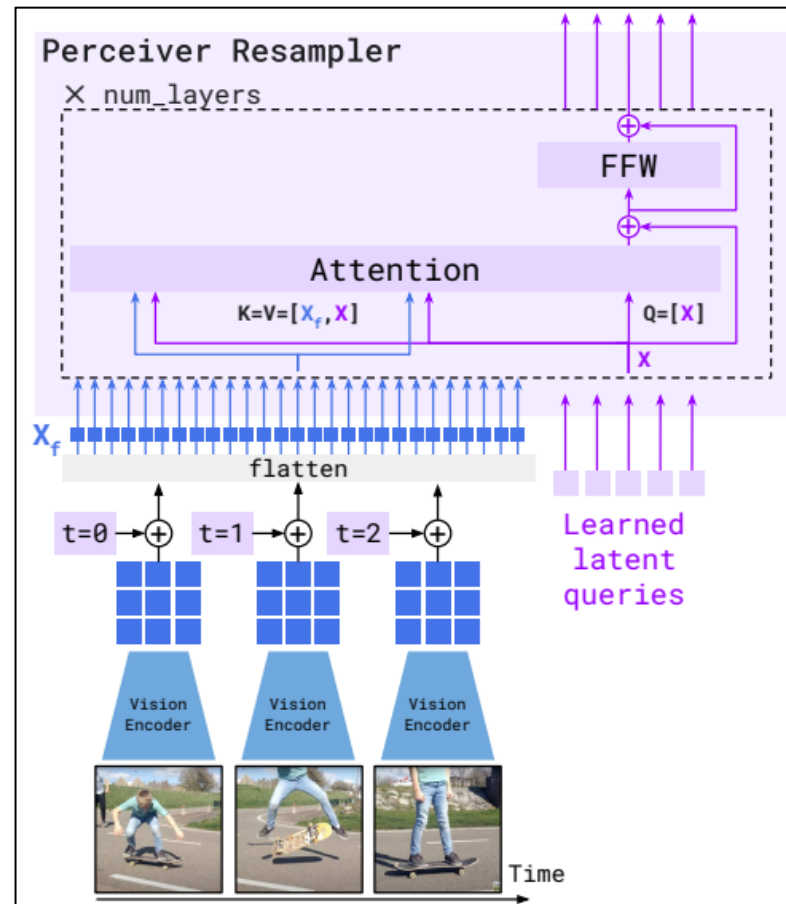
입력 데이터가 video일 때
추가적으로 진행되는 과정

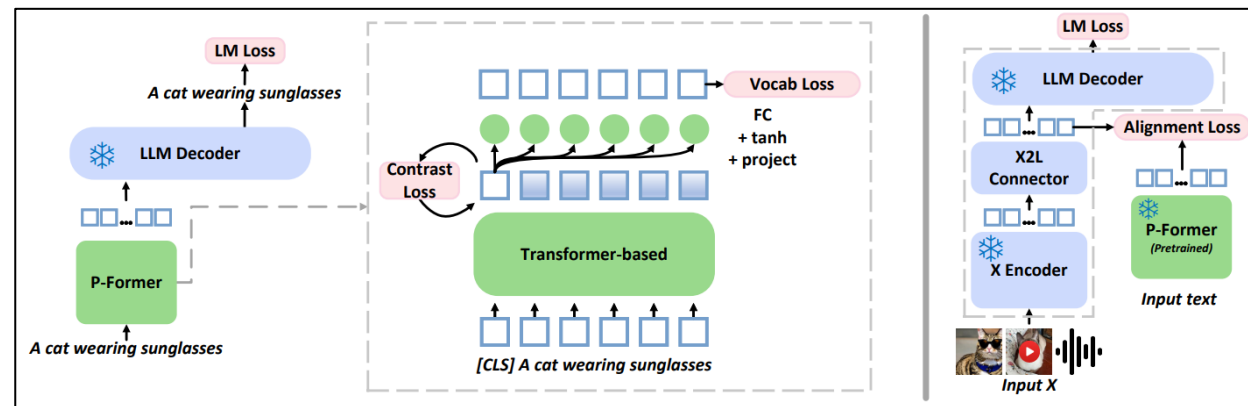
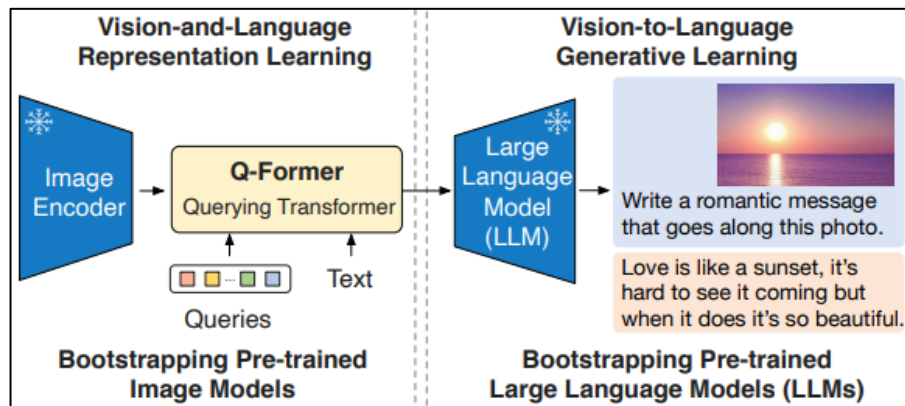


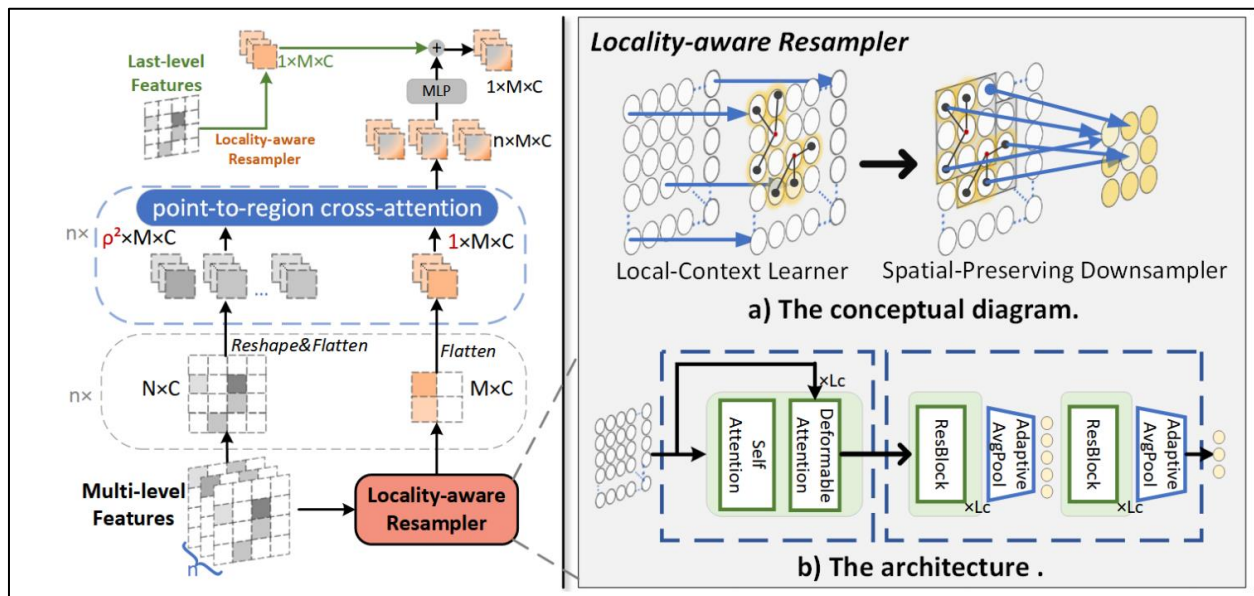
- Learned latent query
 - 입력 이미지의 중요한 정보를 추출하고 resampler의 output 토큰 수를 고정하기 위해 사용함
 - Latent query의 크기는 고정되어 있기에(그림에서는 5개), resampler의 output 토큰 수와 학습된 latent query의 수는 같음
 - 이러한 구조 덕분에, 입력 이미지의 크기와 관계없이 resampler는 항상 같은 개수의 토큰을 출력할 수 있게 됨
- Feed Forward Network
 - Resampler는 attention 연산을 수행하고 Feed Forward Network를 통과함
 - Feed Forward Network는 input layer, hidden layer, output layer로 구성됨
 - Activation function을 사용하여 input에 non-linearity를 추가하기 위해 사용함

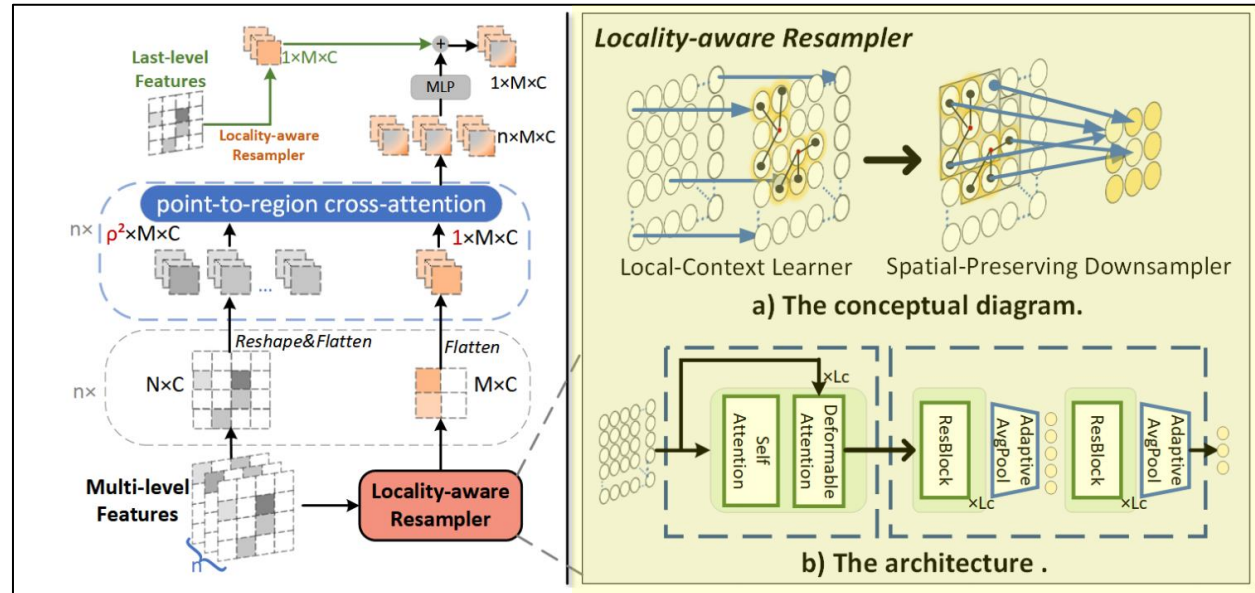


- Resampler의 단점
 - 다른 방법에 비해 classification 성능이 떨어짐
 - 유연하지만, image feature 정보를 손실할 가능성이 있음





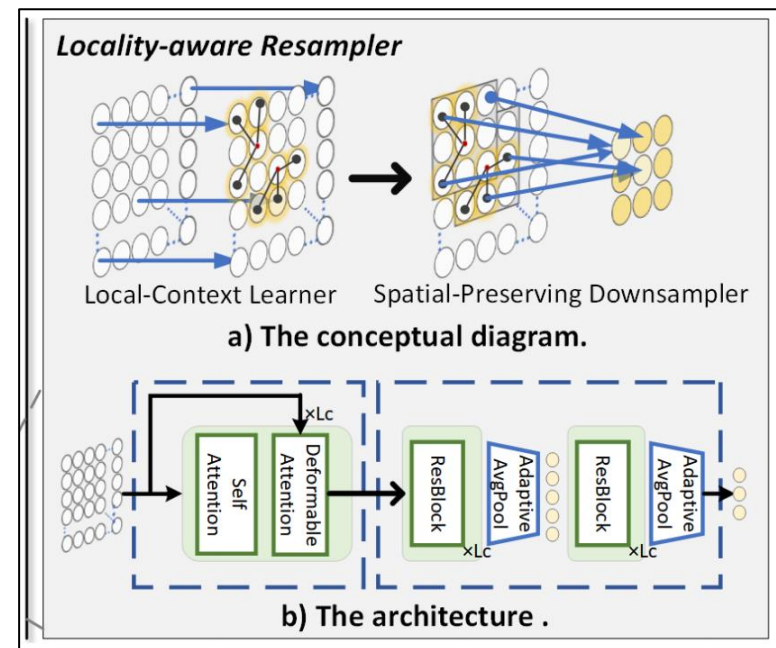




- visual feature 정보를 해치지 않으면서, 다양한 anomaly 모양과 크기를 처리하기 위해 LTC를 제안함
- LTC는 공간적 정보를 유지하기 위해 convolution을 사용했고, 다양한 형태와 크기의 객체를 탐지하기 위해 deformable attention을 사용함

- Deformable attention
 - Geometric variation에 유연하게 대응하기 위해 deformable attention을 사용함
 - Offset을 통해 query의 위치를 조정하기에, 다양한 형태와 크기의 객체를 탐지할 수 있음
 - 20번의 식은 deformable attention에서 특정 지역의 정보를 더 잘 포착하기 위해 사용되는 방식임

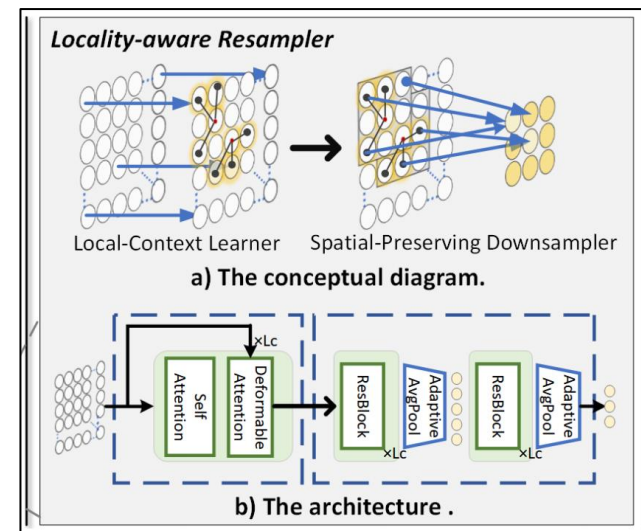
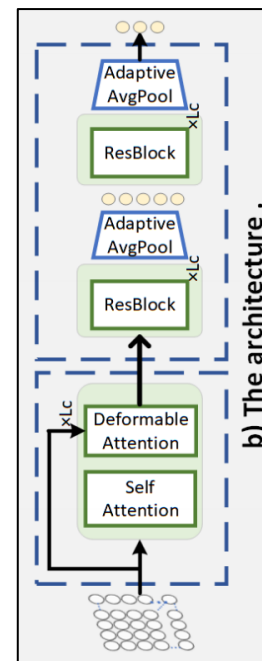
- Convolution
 - 이미지의 local 패턴을 효과적으로 처리하고, 이미지의 세부 정보를 보존하면서 정보를 압축하기 위해 convolution을 사용함
 - Convolution만으로는 물체의 크기나 비율 변화에 대응하기 어렵기에 deformable attention을 같이 사용함



$$p' = p + \Delta p, \quad (19)$$

$$O(i, j) = \sum_{k=0}^{K-1} W_k * X(i + s * (p' + \Delta p), j), \quad (20)$$

- 구조를 보면, deformable을 먼저 적용함으로써, 입력 이미지의 중요한 부분에 집중하고 지역 정보를 파악하는 것을 확인할 수 있음
- 이를 통해 이미지 내의 다양한 객체나 패턴에 따라 유연하게 처리할 수 있음
- 이러한 과정을 통해, convolution이 처리해야 할 정보가 줄어들고 연산량이 줄어듦
- 마지막에 convolution을 통해 더욱 명확한 local 패턴을 찾고 정보를 압축하여 모델의 연산량을 줄임

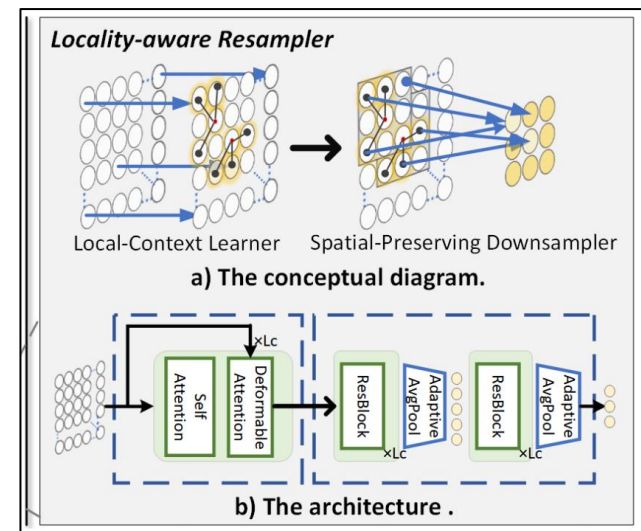
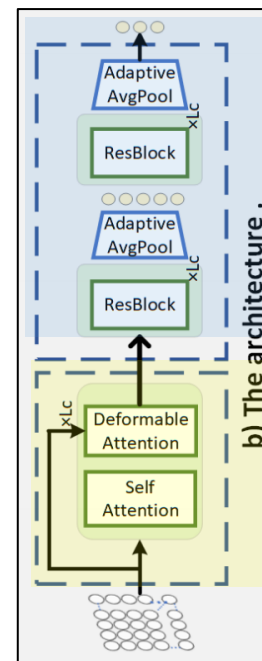


- Lc Block(local-context learner block)

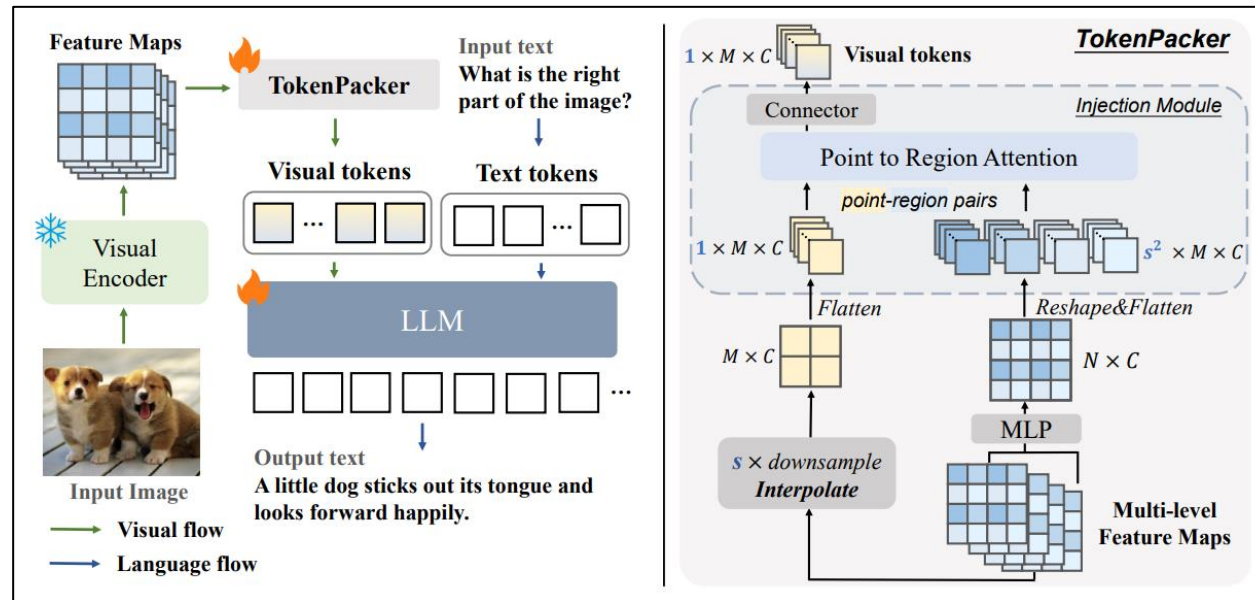
- Lc block은 self-attention과 deformable attention으로 구성됨
- Self-attention과 deformable attention의 상호 보완적인 특성을 활용함
- Self-attention은 모든 요소 간의 관계를 학습하여 중요한 정보를 강조하지만, 불필요한 정보를 무시함
- 이러한 self-attention의 단점을 다양한 형태와 크기의 객체를 처리하는 것에 유리한 deformable attention을 사용해 보완함

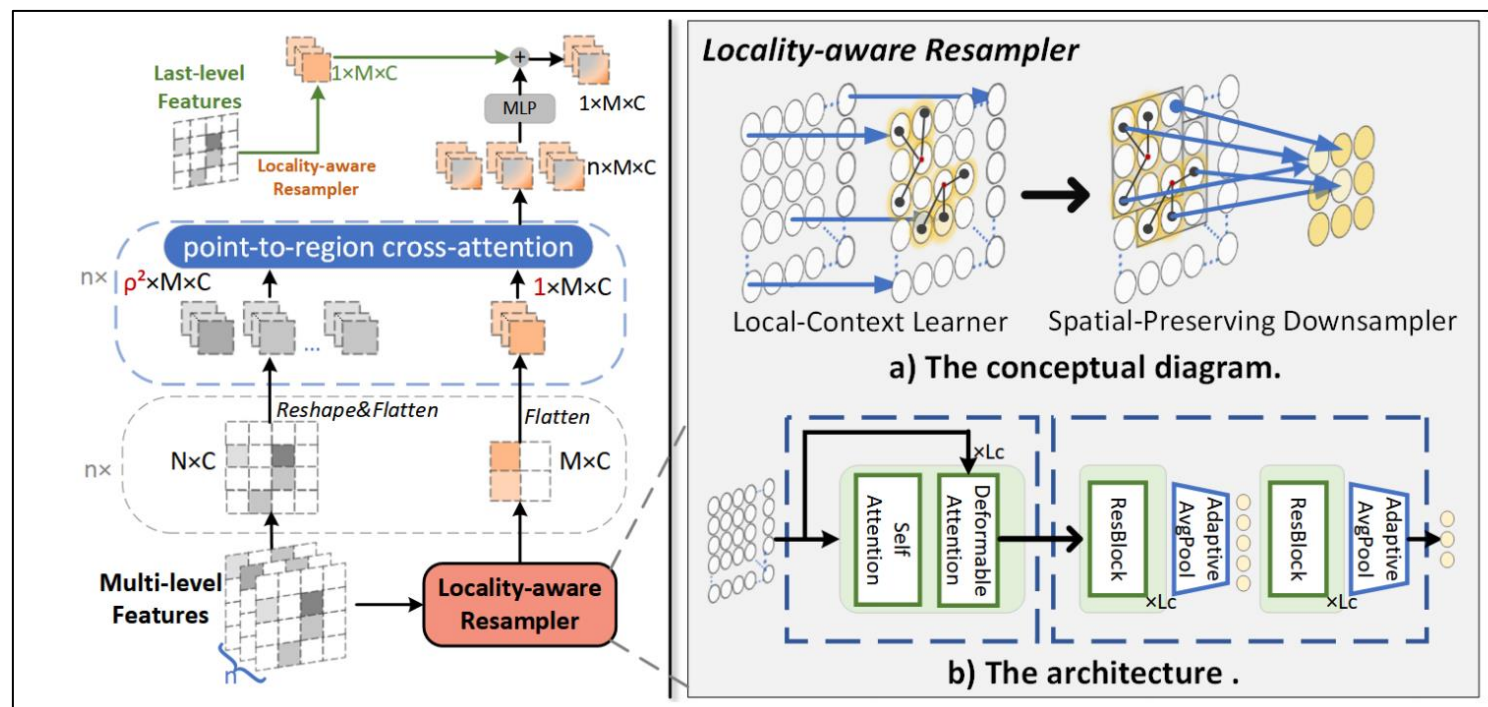
- Adaptive average pooling

- 고정된 출력을 생성하고 유연성을 가지기 위해 사용함
- 입력의 크기가 다르더라도 최종 결과의 크기는 고정되어 있기에, 다양한 크기와 형태의 이미지를 처리할 수 있음
- 네트워크의 마지막 부분에서 중요한 특징을 효과적으로 보존하는 특징을 가지고 있음



- TokenPacker: Efficient Visual Projector for Multimodal LLM, arXiv 2024
- 해당 논문의 아이디어를 반영함





Overall Architecture

- GPT-4o와 같은 최근 MLLM은 anomaly detection과 추론 과정에서 아쉬운 성능을 보인다고 함
- 아래의 그림에서 볼 수 있듯이 탐지는 정확하지만, anomaly에 대한 정확한 설명이 부족하다는 것을 알 수 있음
- 논문에서 제안한 Anomaly-OV는 visual anomaly에 대한 정확한 설명과 논리적 판단 근거를 제공하고, 이를 바탕으로 모델은 이상을 발생시킨 잠재적인 원인을 짐작하고 개선 사항을 제공함
- Anomaly-OV는 기존 MLLM에 anomaly detection을 잘 수행하기 위한 보조 전문가 모델을 추가하여, 기존 MLLM의 일반화 능력을 유지하고 AD 성능을 높임
- 기존 MLLM 중, 코드가 공개되었고 다른 상업적 모델들과 유사한 성능을 보이는 LLaVA-OneVision을 기반으로 모델을 구축함

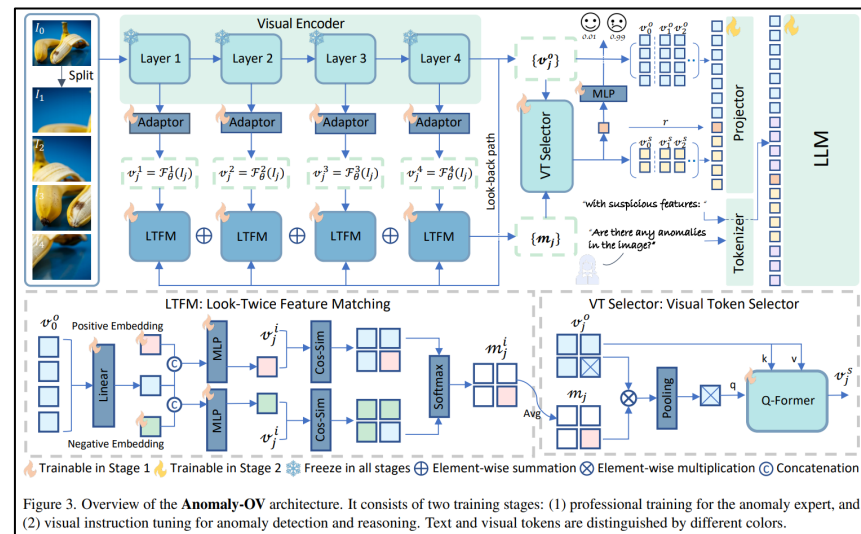
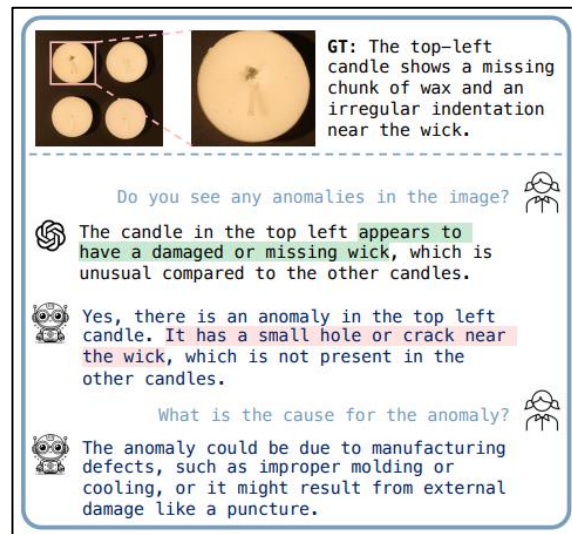
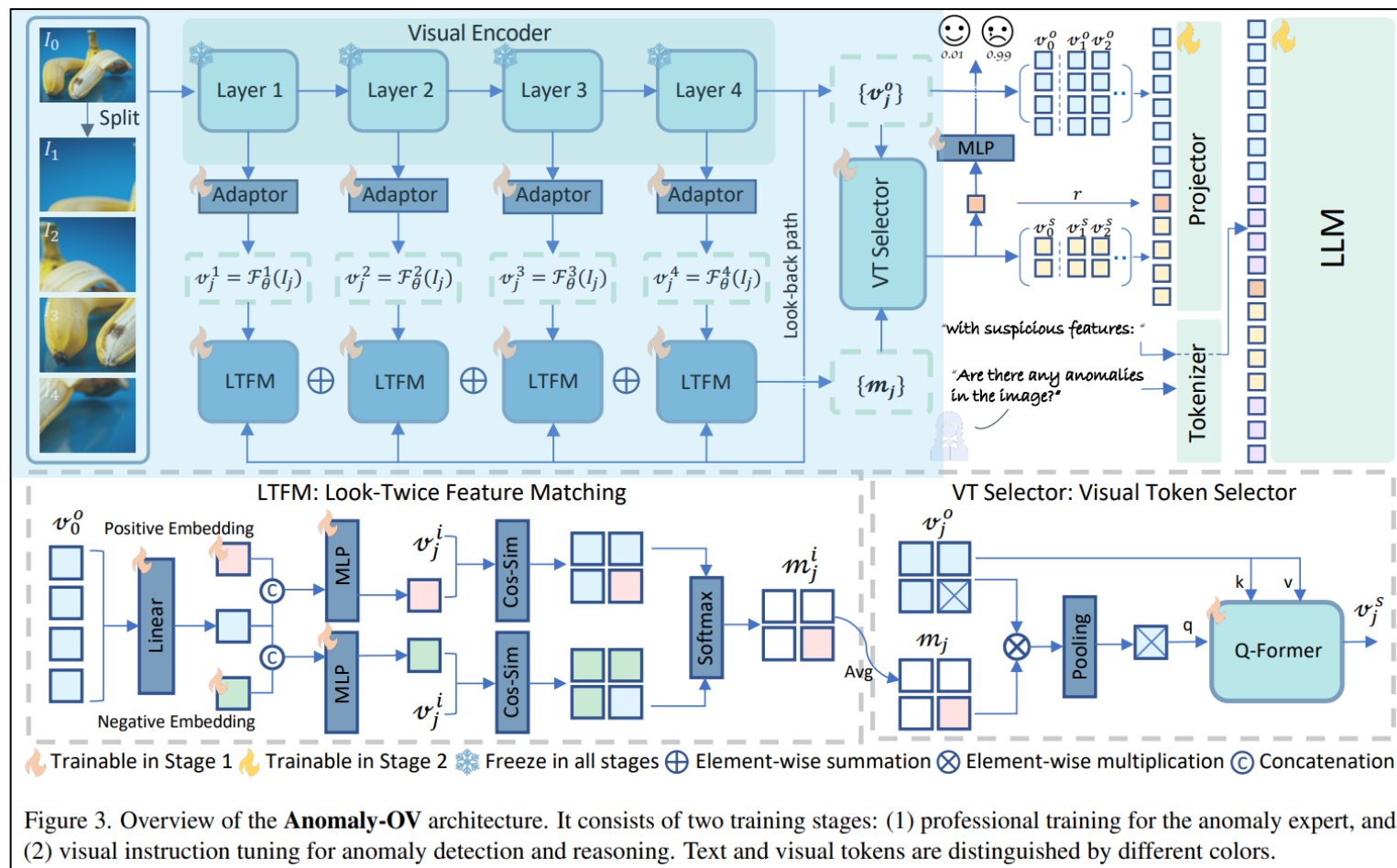
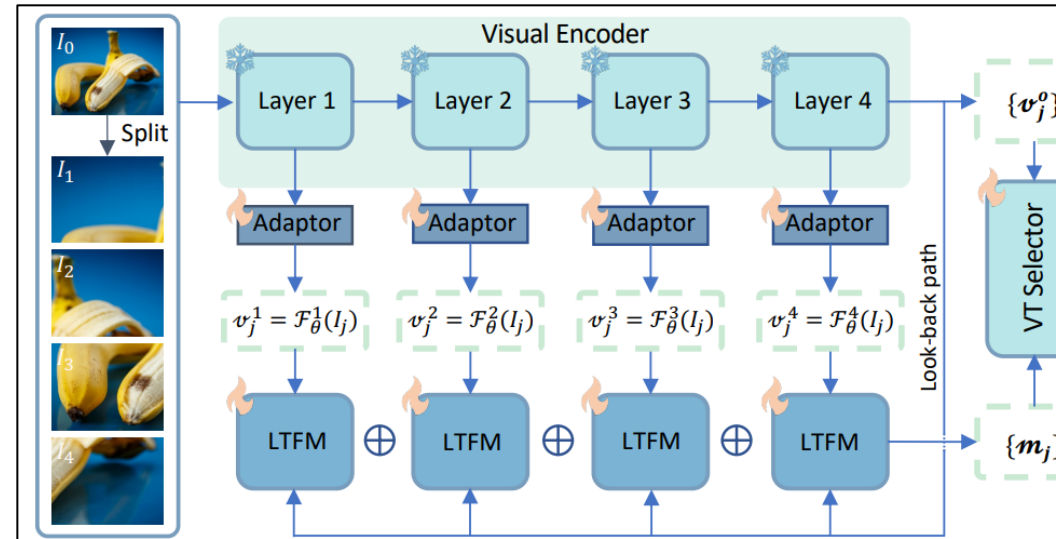
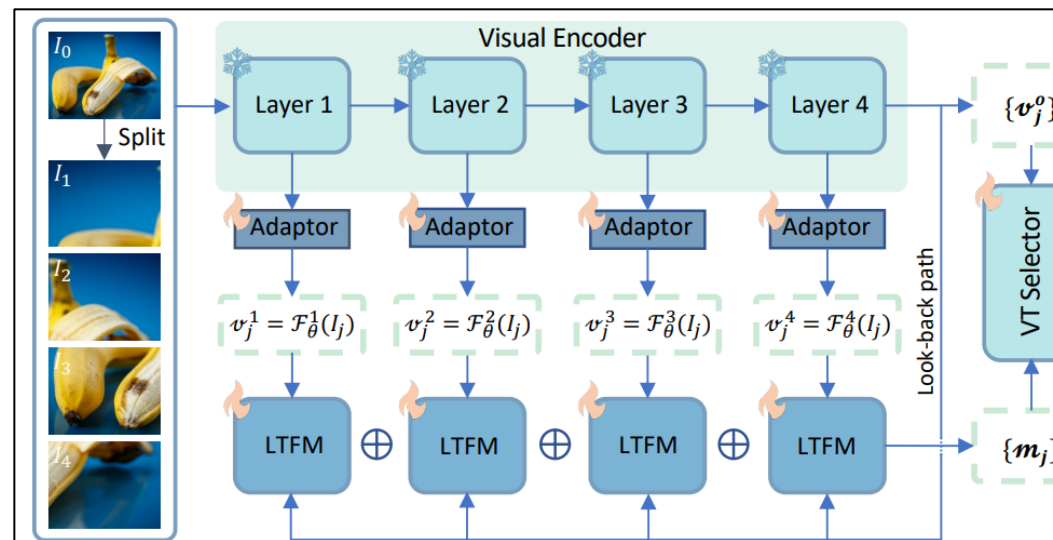


Figure 3. Overview of the **Anomaly-OV** architecture. It consists of two training stages: (1) professional training for the anomaly expert, and (2) visual instruction tuning for anomaly detection and reasoning. Text and visual tokens are distinguished by different colors.





- Anomaly-OV는 fine-grained local semantic을 효과적으로 처리하기 위해 multi-level feature 방식을 사용함
- LLaVA-OneVision의 전략인 AnyRes를 사용하여 고해상도 입력 이미지를 여러 조각으로 나눔 (그림에서는 4개)
- AnomalyCLIP과 유사한 방법으로, 각기 다른 level에서 여러 이미지 표현을 포착하기 위해 ViT에서 4개의 layer를 선택하고 선택된 layer의 output을 저장함



- 계산량을 줄이고 모델의 효율성을 높이기 위해, Adaptor를 사용하여 여러 layer에서 추출된 visual feature의 dimension을 축소함
- Adaptor는 추출된 visual feature를 압축하여 중요한 정보는 유지하고 불필요한 정보는 제거하는 역할을 함

$$\mathcal{I} = \{I_0, I_1, I_2, \dots, I_{n-1}\}$$

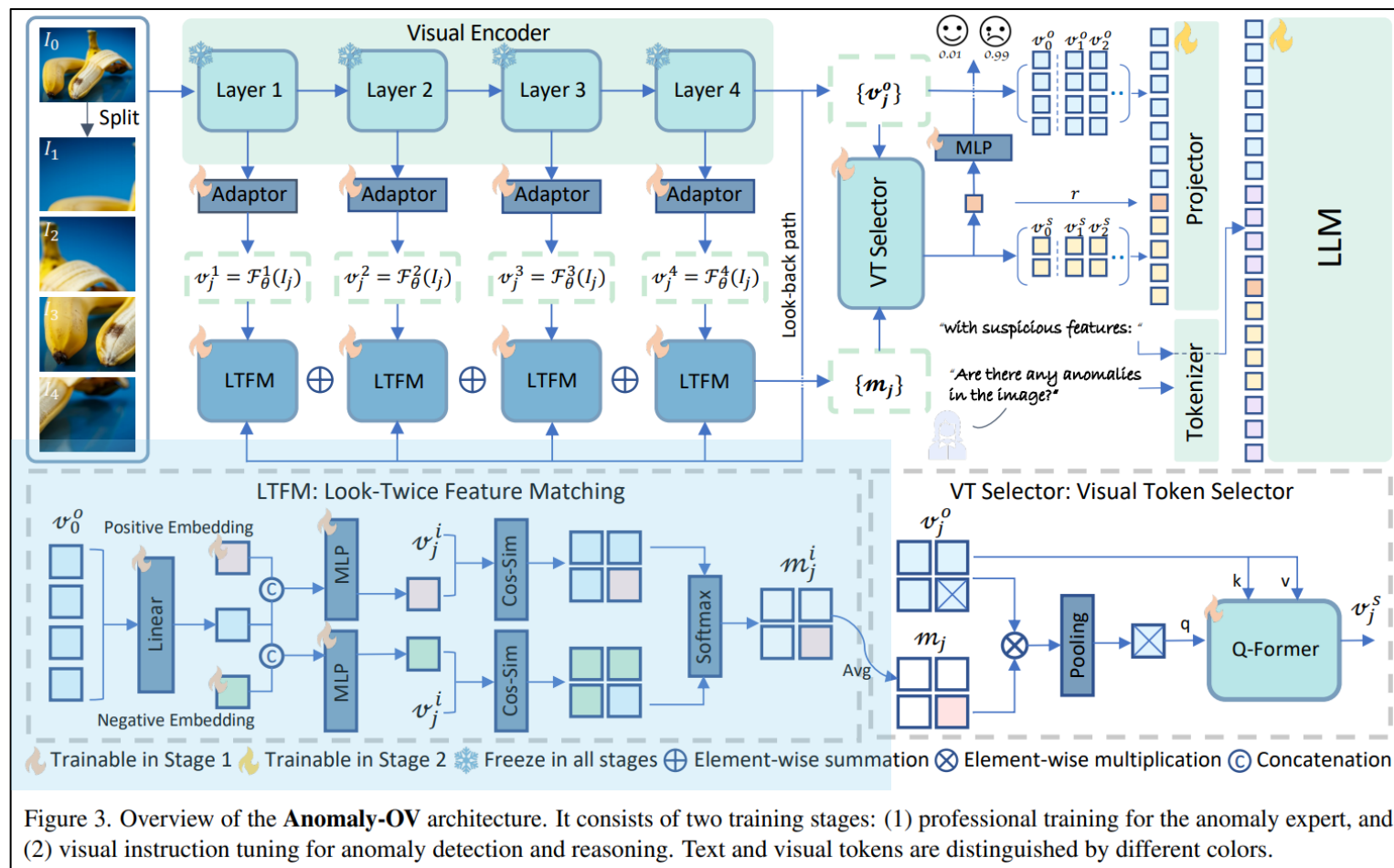
I_0 : 크기를 조정한 원본 이미지

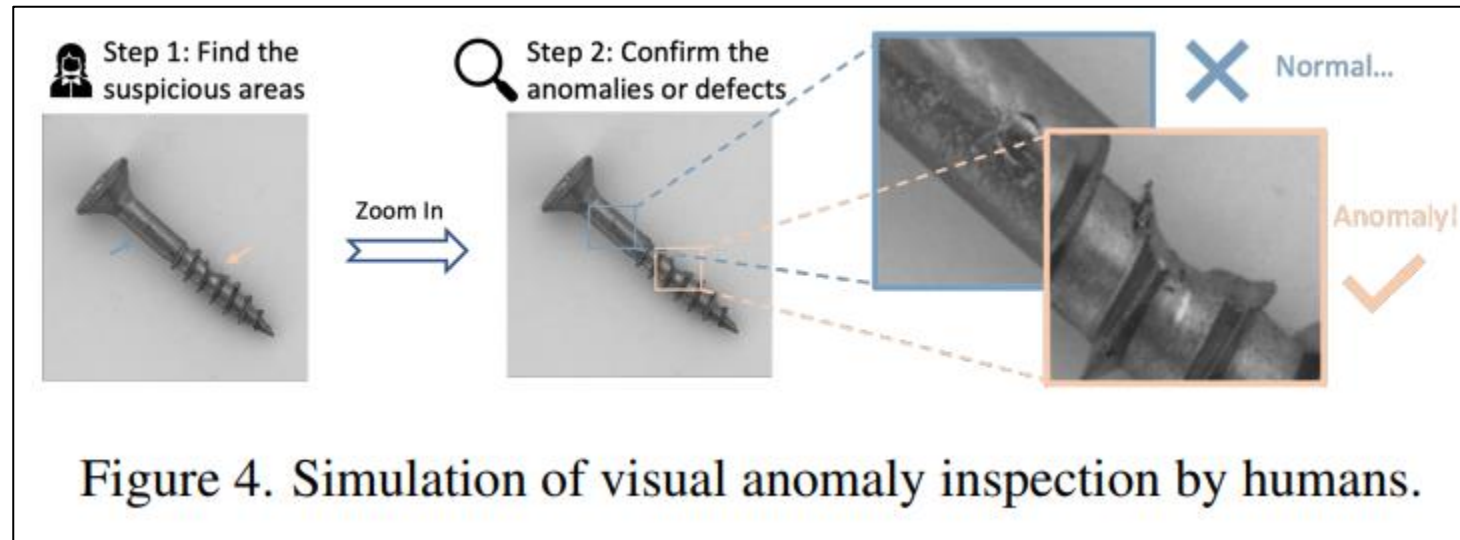
$$\mathbf{v}_j^i = \mathcal{F}_\theta^i(I_j)$$

i : i -th level

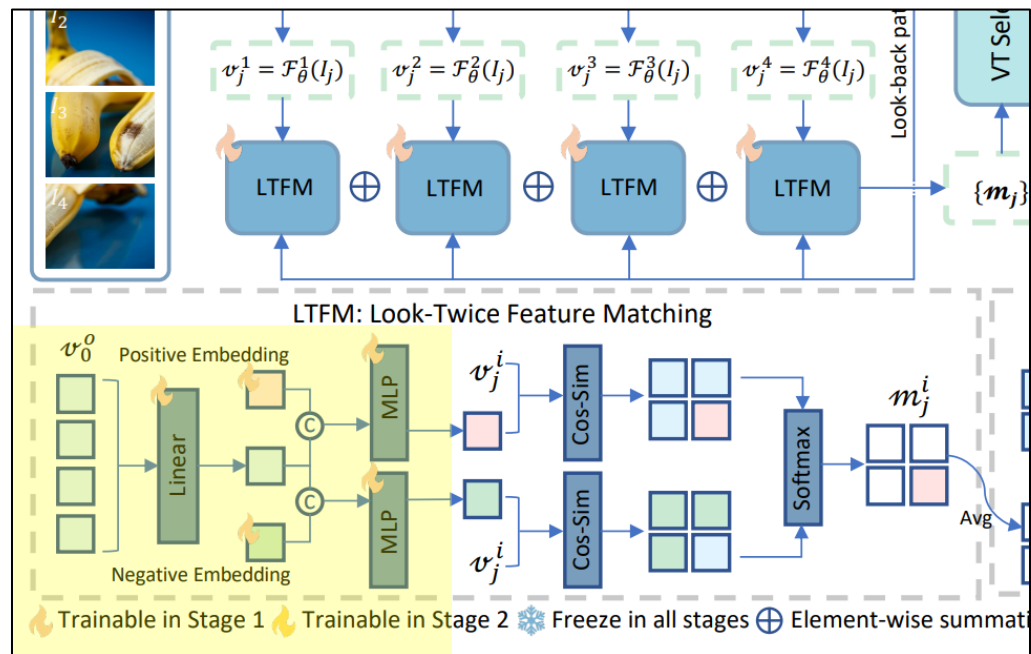
j : i -th level에 상응하는 이미지의 Index

\mathcal{F} : visual encoder

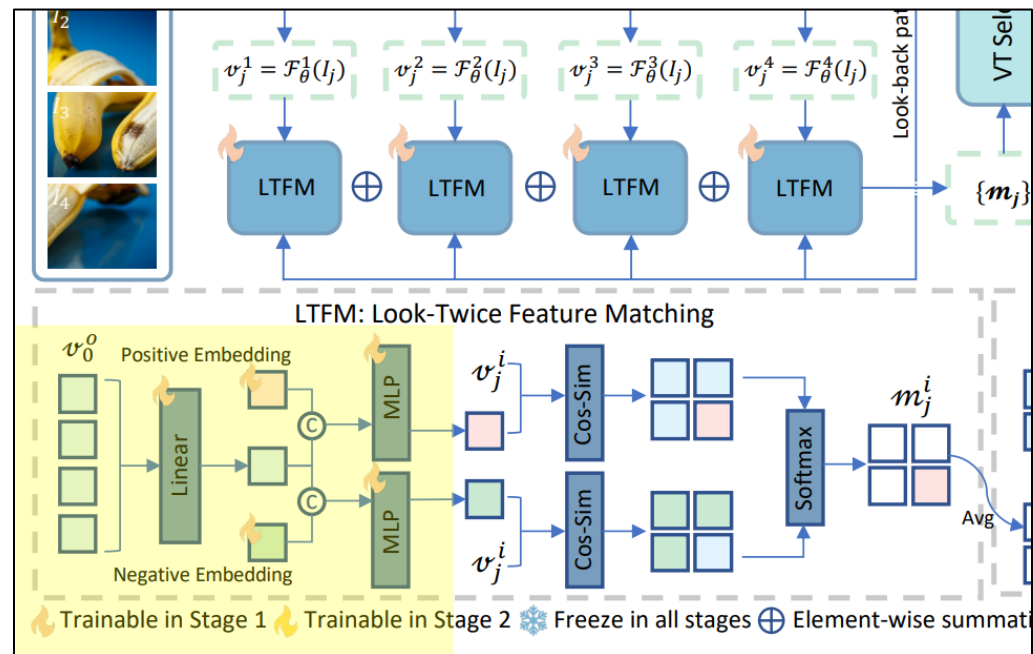




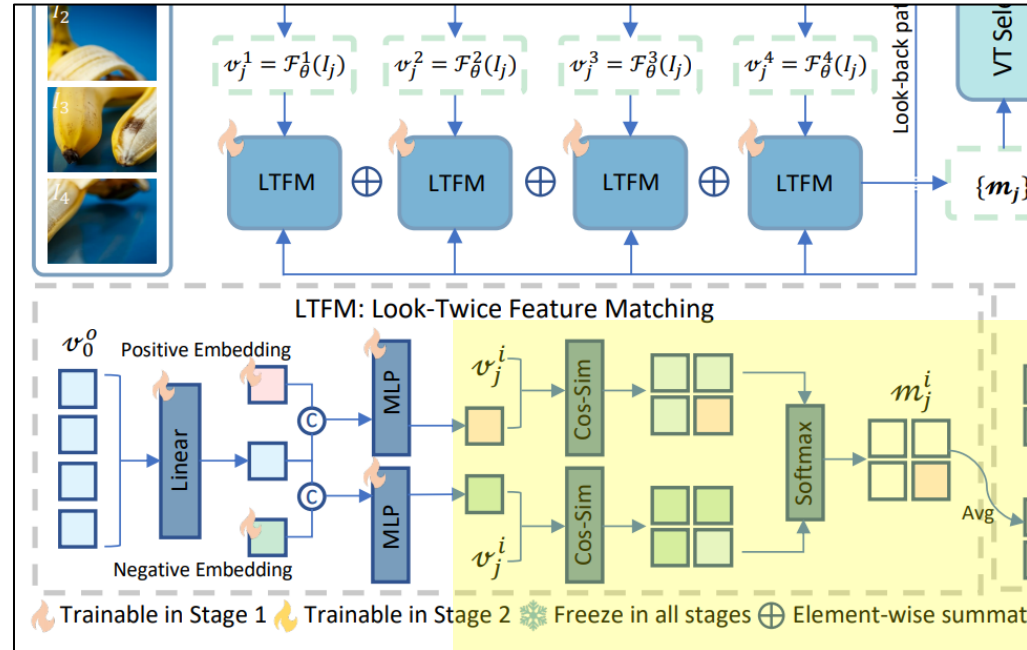
- 사람들은 anomaly를 검사할 때, 물체를 전체적으로 훑어보고 의심스러운 부분이 발견되면 해당 부분을 더 자세히 확대하여 검사함
- 이에 영감을 받아 이미지 전체 맥락을 고려하면서 의심스러운 부분을 집중적으로 분석하는 LTfM을 제안함



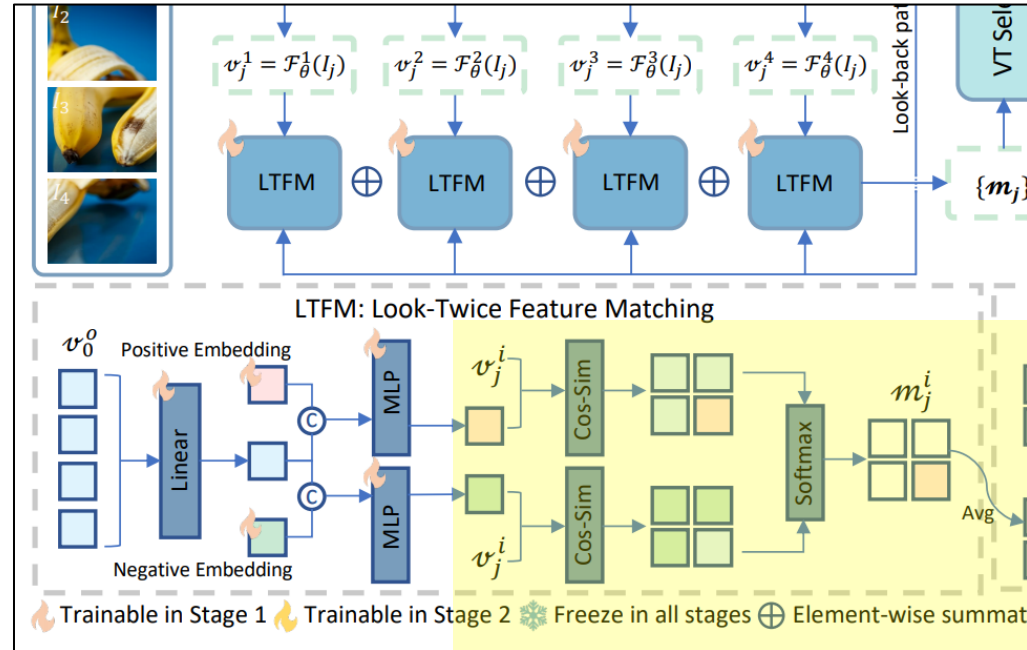
- AD에 도움이 되는 class-awareness abnormality description을 생성하기 위해, positive(anomalous) embedding과 negative(normal) embedding을 원본 이미지의 token과 각각 결합함
- 이후에 MLP를 통과하여 abnormality description and normality description을 생성함



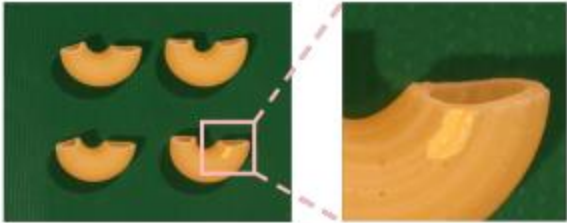
- 유용한 토큰들을 선택하고 결합하기 위해 원본 이미지의 토큰들을 linear layer에 통과시킴
- 이후에 2개의 learnable embedding과 결합함, 이를 MLP에 넣어 description을 생성함



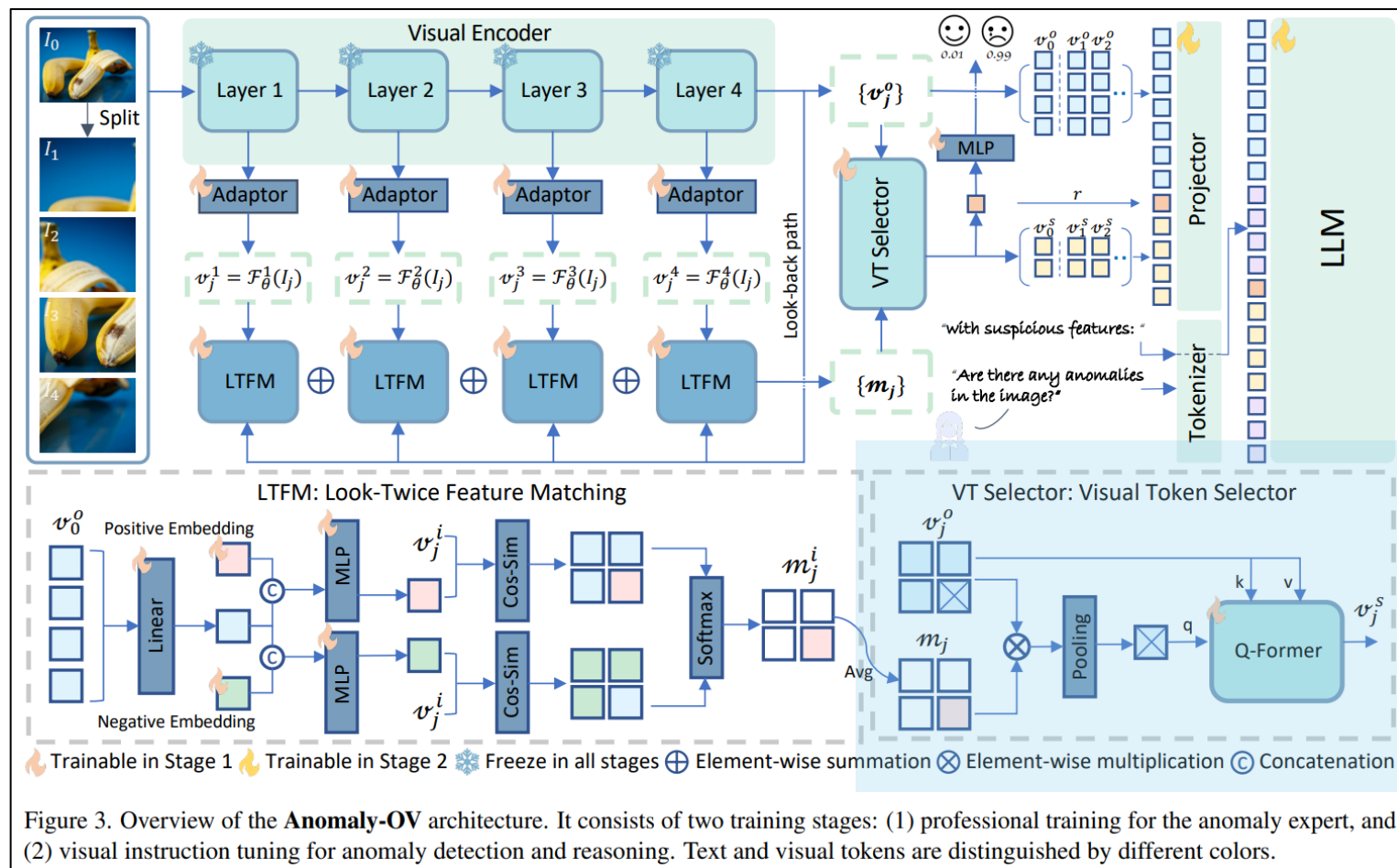
- CLIP의 zero-shot classification mechanism과 비슷하게 코사인 유사도와 softmax 연산을 결합하여 각각의 패치 토큰이 이상 패턴에 속할 가능성을 계산함
- 하나의 token이 abnormality/normality description과 각각 코사인 유사도를 계산함
- 코사인 유사도 계산을 통해 나온 2개의 결과를 가지고 softmax를 적용해 visual token이 이상 패턴과 얼마나 유사한지 보여줌

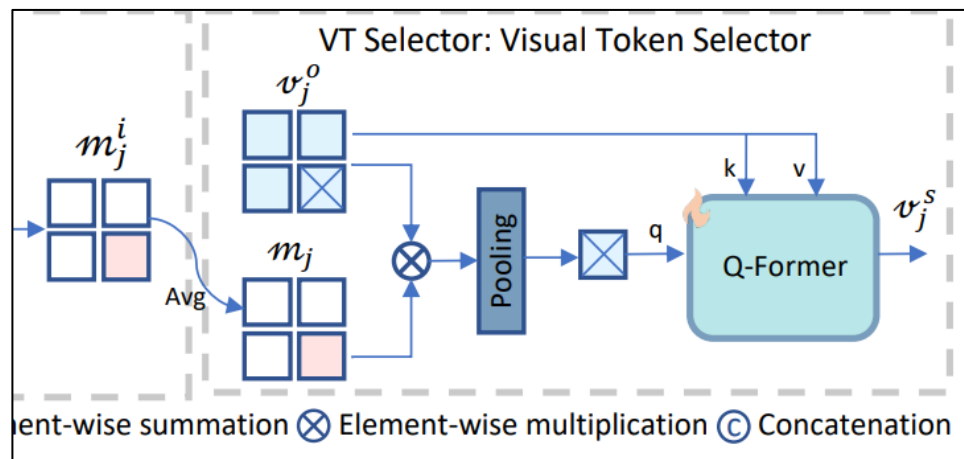


- LTFM의 최종 결과인 m 은 visual token들의 significance map을 나타냄
- LTFM의 최종 결과인 significance map을 사용자에게 보여줌으로써, 사용자가 모델의 판단 근거를 이해하고 모델의 신뢰도가 높아지게 됨

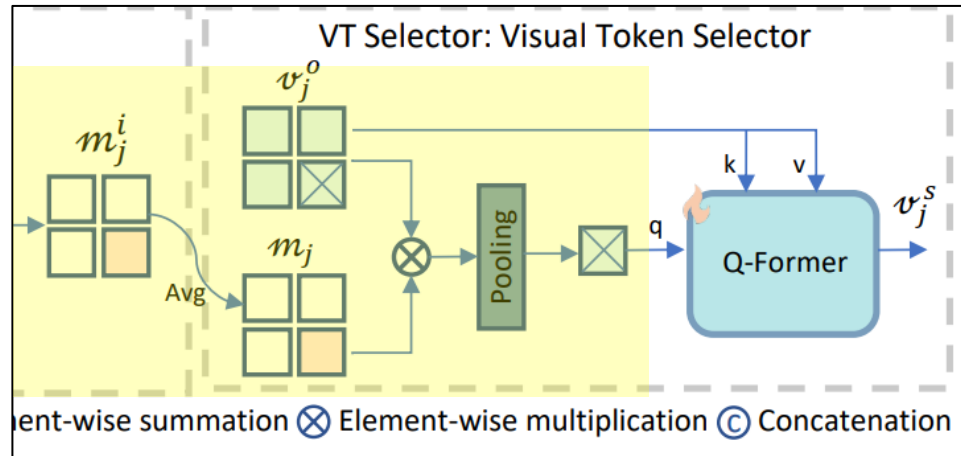
Macaroni Example: Yellowish Spot	
	
User	Do you see any anomalies in the image?
GPT-4o	The image shows four pieces of elbow macaroni on a green background. The anomaly is that the macaroni pieces are not whole; they are cut in half.
LLaVA-OV	The image shows four pieces of pasta, specifically macaroni shells, arranged on a green textured surface. The pasta appears to be uniformly colored and shaped, with no visible defects or anomalies.
Ours	Yes, there is an anomaly in the image. The bottom right pasta piece has a noticeable yellowish discoloration on its surface.

Anomaly가 있는 image patch를 사용자에게 보여줌

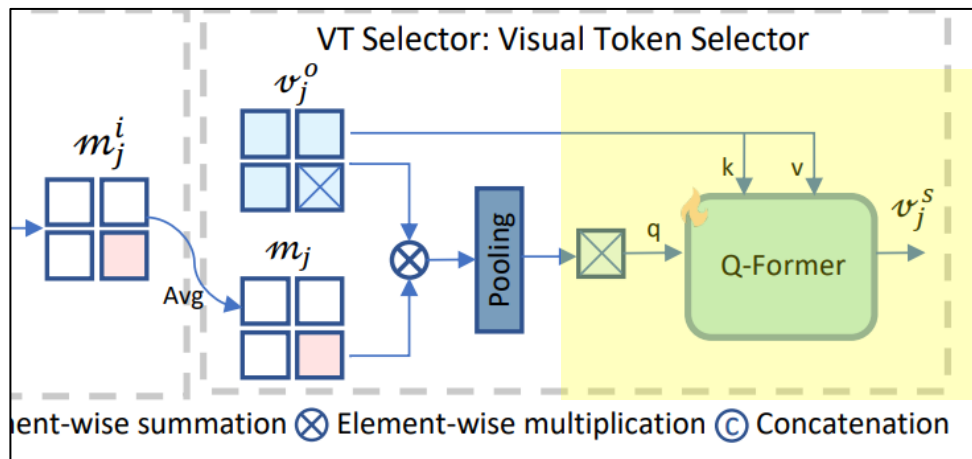




- Anomaly detection의 목적성에 맞게, anomaly 정보가 담겨있는 token을 강조하기 위해 VT selector를 사용함
- VT selector를 사용함으로써, 의심스러운 토큰을 선택하고 강조함

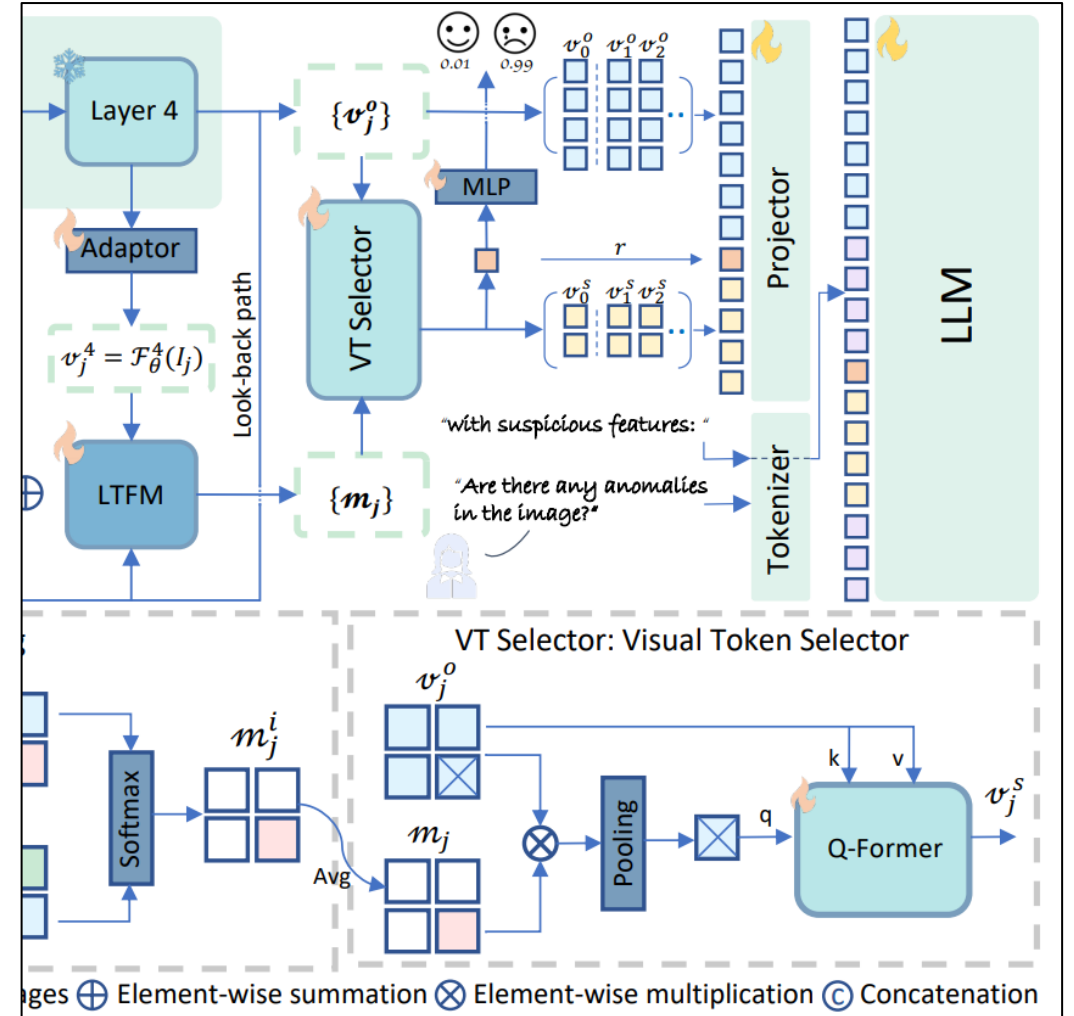


- Low level에서 high level까지의 token 중요성을 포착하기 위해 m 의 평균을 계산함
- 이후에 의심스러운 token과 평균을 낸 significance map m 을 곱함
- 곱하게 되면, m 에서 normal token은 0으로 조정되고 anomalous token은 유지됨
- 마지막으로 spatial average pooling을 사용하여 token의 수를 줄임

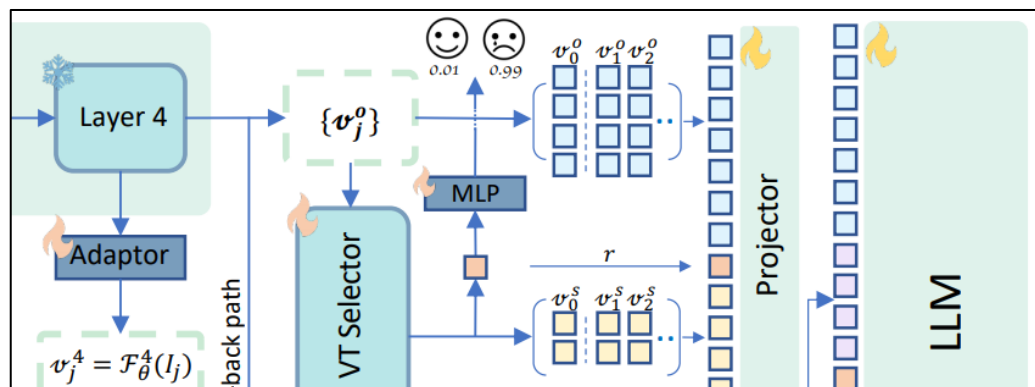


- 원본 이미지의 visual token 내에서 이상 패턴과 상관관계가 있는 token을 집계하고 정보를 압축하기 위해 Q-Former를 사용함
- 이를 통해 LLM은 AD 목적과 관련성이 높은 token에 집중할 수 있게 됨
- q : pooling이 적용된 token
- k, v : visual encoder에서 추출된 원본 이미지의 feature

- 일반적인 이미지 이해 능력도 유지하기 위해 원본 이미지의 visual token을 LLM에 전달
- VT selector를 통해 얻은 visual token들을 바탕으로, visual token의 중요도를 판단함
- 중요도를 판단함으로써, LLM이 어떤 token에 집중해야 하는지 명확하게 알려줌
- Anomaly expert(LTFM, VT selector)를 통해 입력 이미지에 대한 anomaly score를 계산함
(해당 점수는 이미지가 얼마나 비정상인지를 나타냄)
- 미리 정의된 임계값을 사용하여 score 값에 따라 {highly, moderately, slightly} 중 하나를
선택하고 <adv>에 입력함
- 최종적으로 모델이 생성하는 prompt는 다음과 같음
with <adv> suspicious feature: [Visual Tokens]
- 해당 prompt를 통해 LLM이 더욱 정확하고 맥락에 맞는 답변을 생성함



- 기존 MLLM과는 달리, Anomaly-OV에서는 text encoder를 사용하지 않았음
- visual encoder만 사용했기에, 모델의 복잡성이 줄어들었음
- LTFM과 VT selector를 사용하여 모델 자체가 의심스러운 class나 객체에 대한 정보를 분석하게 만든 것이 인상 깊음
- 코드 공개가 안 되어있고, 전체 architecture 그림에서 표정 그림에 대한 설명이 없어서 아쉬웠음
- LLaVA-OneVision model을 그대로 사용했기에, 모델이 가지고 있는 한계를 그대로 상속받음
- 논문에서 Anomaly-OV가 가끔 target object를 정확하게 분류하지 못한다고 언급함
- anomaly를 설명할 때, Anomaly-OV가 자세한 설명이 아닌 포괄적인 단어를 사용하려는 경향이 있음 (예 : 'crack')



MM-LLMs: Recent Advances in MultiModal Large Language Models, arXiv 2024

VMAD: Visual-enhanced Multimodal Large Language Model for Zero-Shot Anomaly Detection, IEEE 2024

Towards Zero-Shot Anomaly Detection and Reasoning with Multimodal Large Language Models, CVPR 2025

ClipCap: CLIP Prefix for Image Captioning, arXiv 2021

Flamingo: a Visual Language Model for Few-Shot Learning, NeurIPS 2022

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, ICML 2023

Bootstrapping Vision-Language Learning with Decoupled Language Pre-training, NeurIPS 2023

TokenPacker: Efficient Visual Projector for Multimodal LLM, arXiv 2024

감사합니다
