DMQA Open Seminar

Industrial Image Anomaly Detection ②

2025. 09. 19

Korea University

Data **M**ining & **Q**uality **A**nalytics Lab.

최지형

Data Mining
Quality Analytics

발표자 소개

❖ 최지형 (Jihyung Choi)

- 고려대학교 산업경영공학과 대학원 재학
- Data Mining & Quality Analytics Lab. (김성범 교수님)
- M.S Student (2024.09 ~ Present)

❖ Research Interest

- Federated Learning
- Semi-supervised Learning
- Robotics

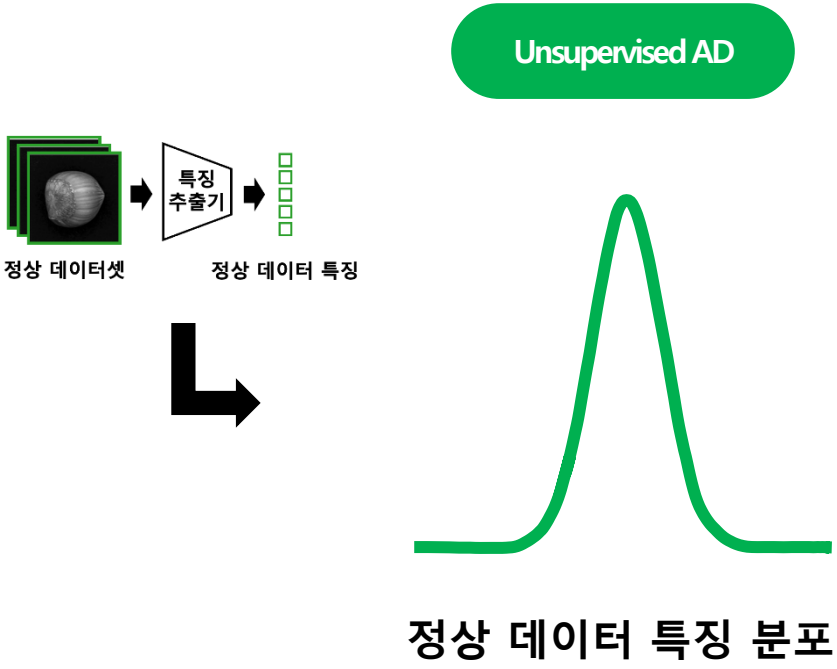
❖ Contact

- jibro@korea.ac.kr

Introduction

❖ Anomaly Detection (AD)

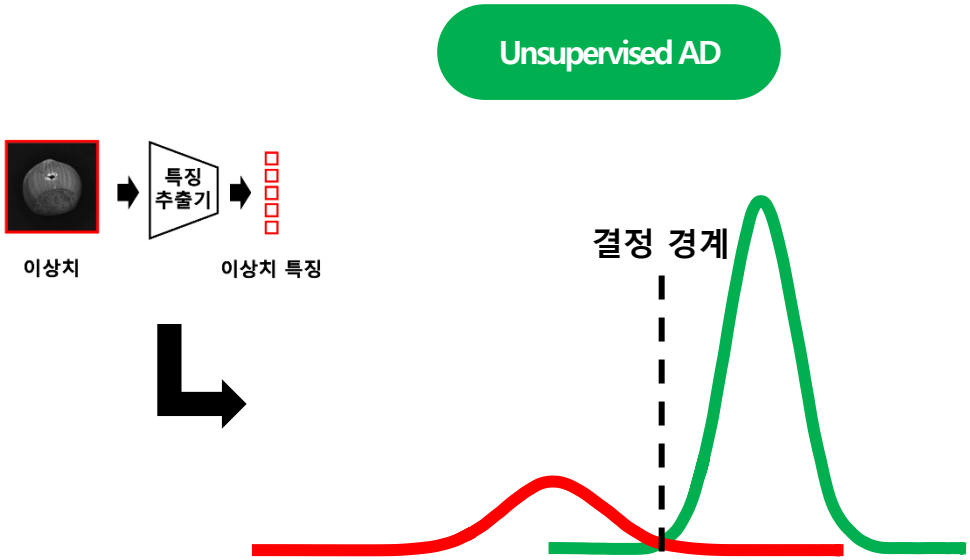
- Goal: 정상 데이터와 다른 특성을 가지는 이상치를 분류
- 이상치는 정상 데이터에 비해 매우 희귀 → 학습에 활용이 어려움
- **Challenge: 이상치가 극히 적은 상황에서, 이상치를 분류할 수 있을까?**



Introduction

❖ Anomaly Detection (AD)

- Goal: 정상 데이터와 다른 특성을 가지는 이상치를 분류
- 이상치는 정상 데이터에 비해 매우 희귀 → 학습에 활용이 어려움
- **Challenge: 이상치가 극히 적은 상황에서, 이상치를 분류할 수 있을까?**

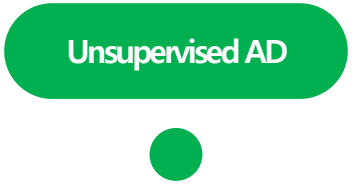


정상 데이터로만 학습한 모델은, 이상치에 대해서는 정상과 다른 결과를 도출

Introduction

❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

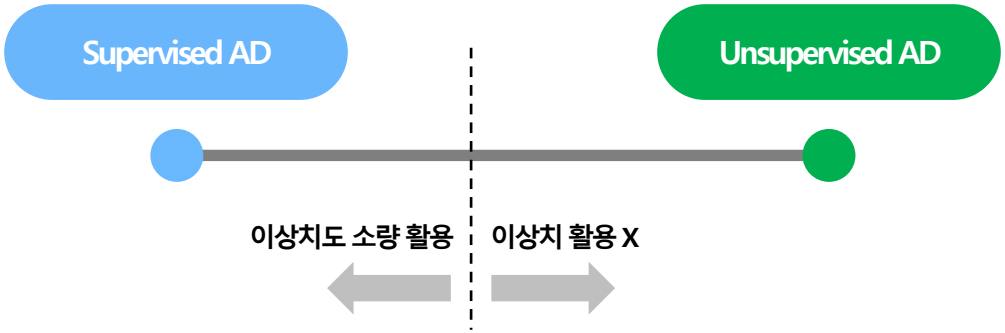
- Topic 1: 무엇보다 AD 정확도가 중요 → Supervised AD
- Topic 2: 정상 데이터도 없음 → Zero-shot AD
- Topic 3 : 이미지 정보만으로는 탐지할 수 없는 이상치 → Multi-Modal AD



Introduction

❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

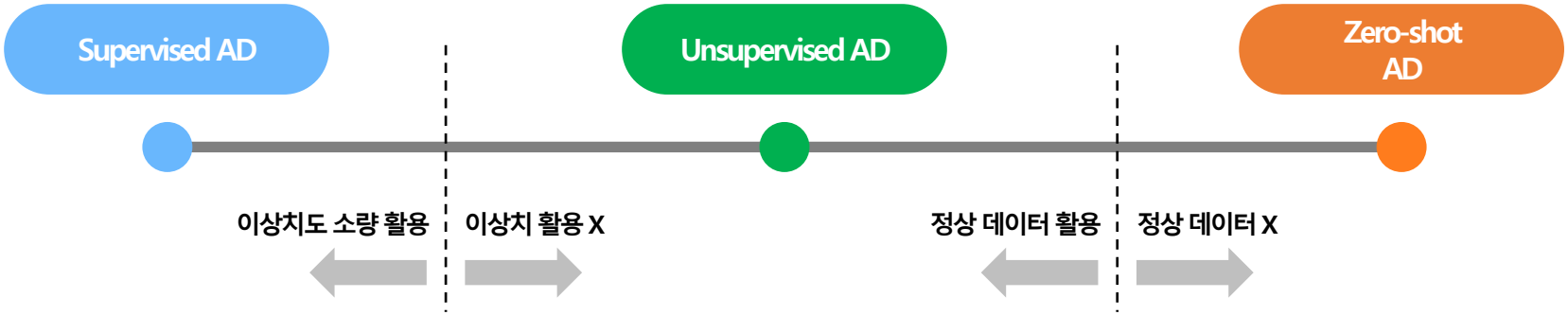
- **Topic 1: 무엇보다 AD 정확도가 중요 → Supervised AD**
- Topic 2: 정상 데이터도 없음 → Zero-shot AD
- Topic 3 : 이미지 정보만으로는 탐지할 수 없는 이상치 → Multi-Modal AD



Introduction

❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

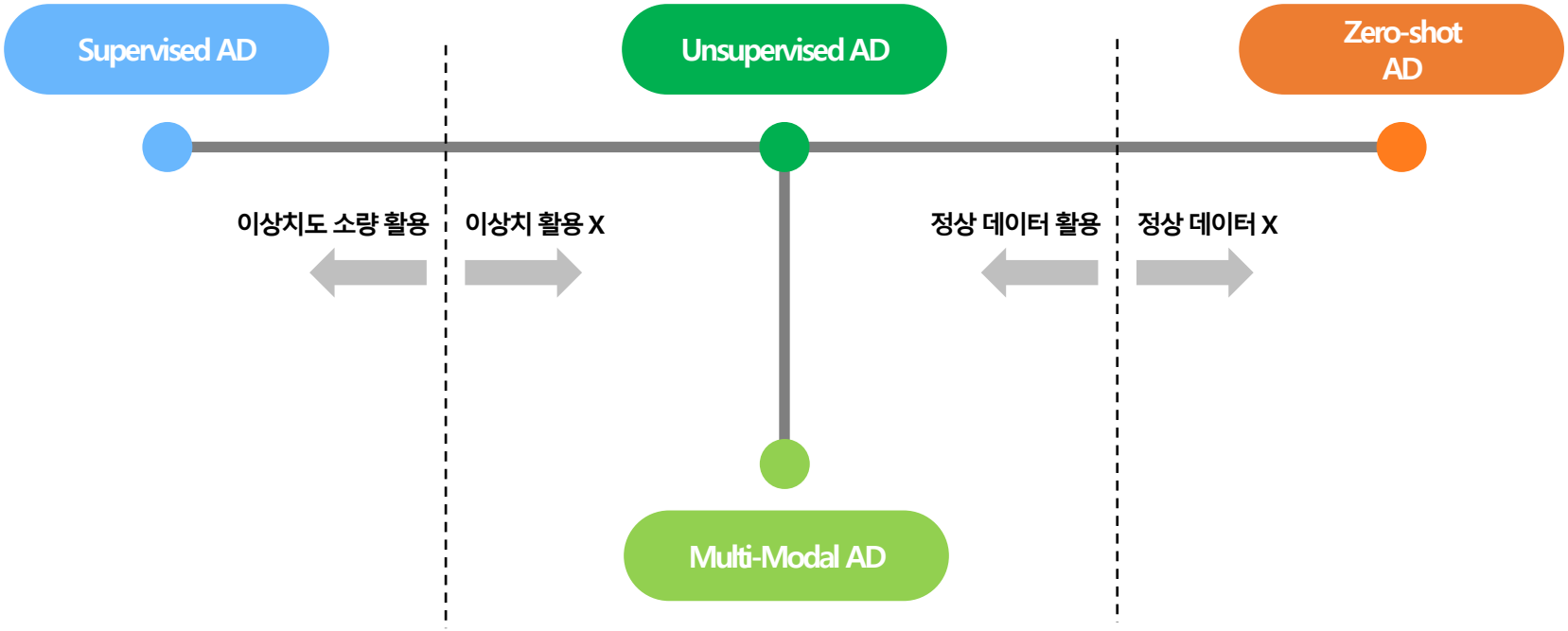
- Topic 1: 무엇보다 AD 정확도가 중요 → Supervised AD
- **Topic 2: 정상 데이터도 없음 → Zero-shot AD**
- Topic 3 : 이미지 정보만으로는 탐지할 수 없는 이상치 → Multi-Modal AD



Introduction

❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

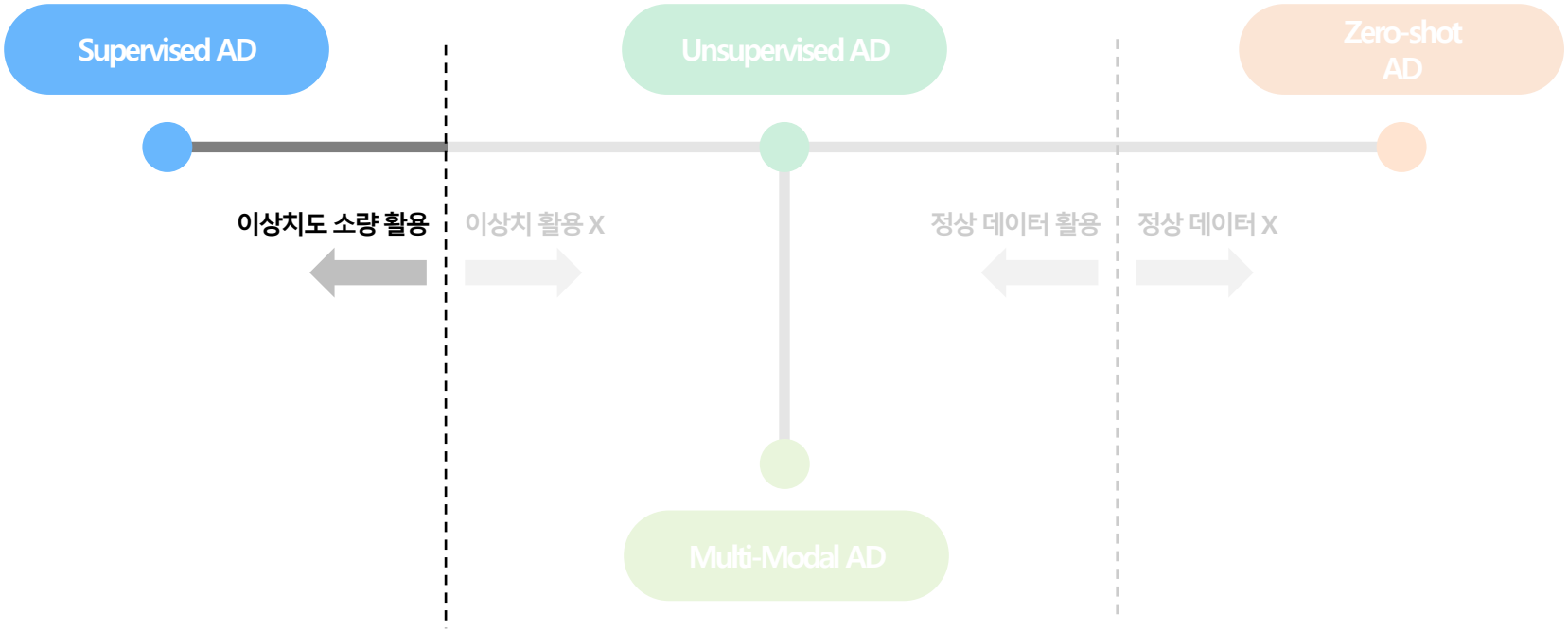
- Topic 1: 무엇보다 AD 정확도가 중요 → Supervised AD
- Topic 2: 정상 데이터도 없음 → Zero-shot AD
- **Topic 3 : 이미지 정보만으로는 탐지할 수 없는 이상치 → Multi-Modal AD**



Supervised AD

❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

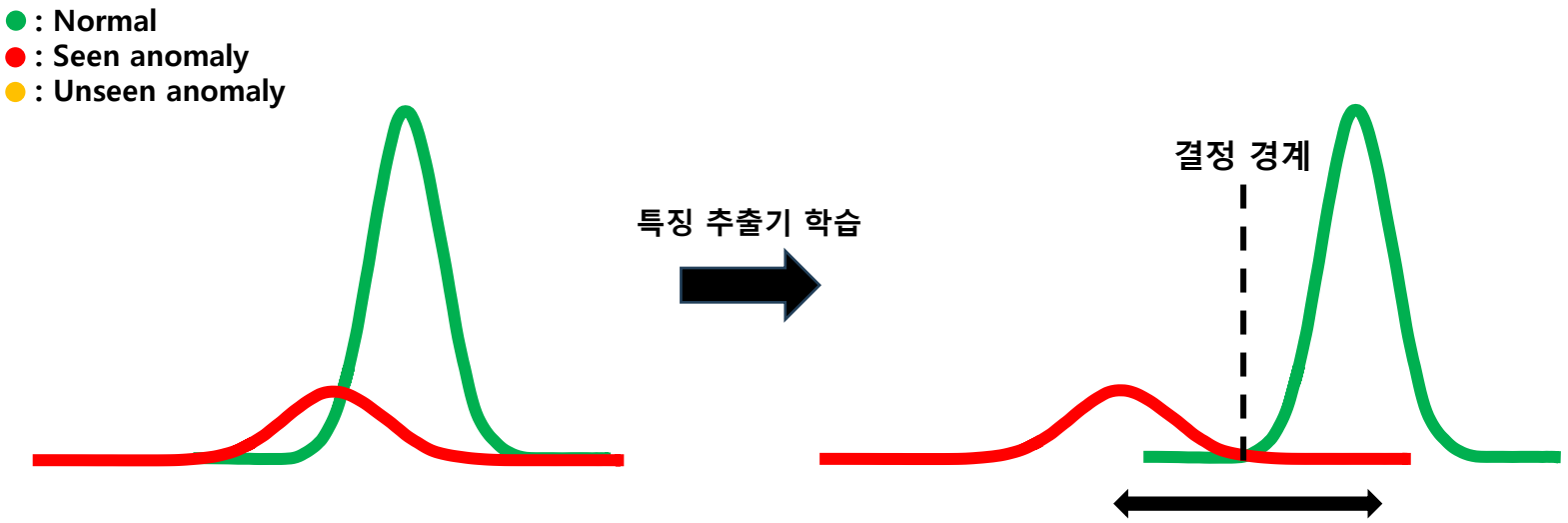
- **Topic 1: 무엇보다 AD 정확도가 중요 → Supervised AD**
- Topic 2: 정상 데이터도 없음 → Zero-shot AD
- Topic 3 : 이미지 정보만으로는 탐지할 수 없는 이상치 → Multi-Modal AD



Supervised AD

❖ **Background**

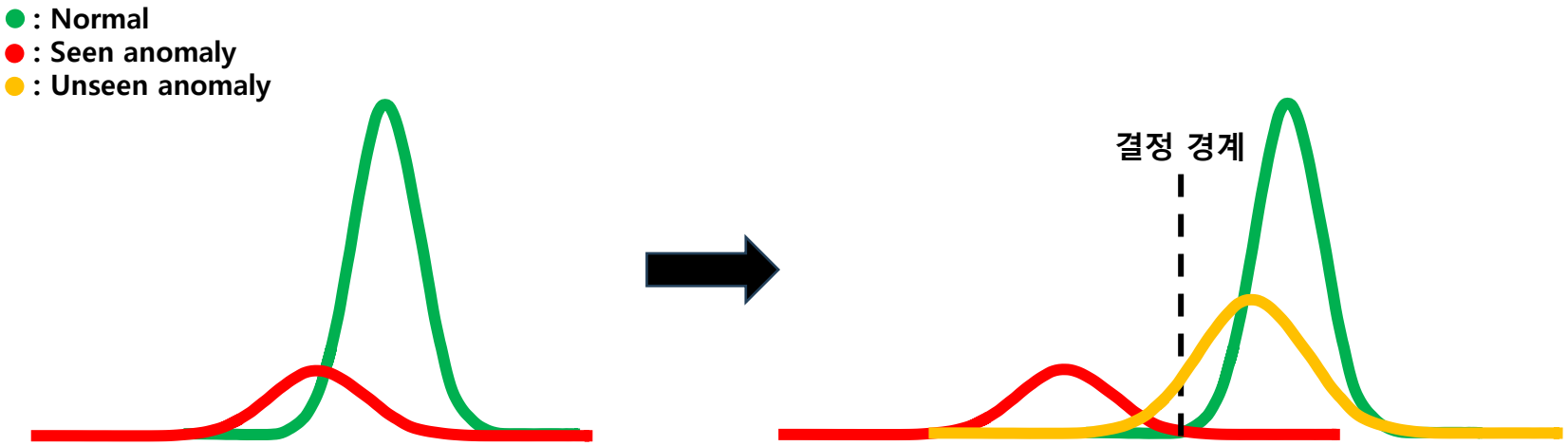
- 정상 데이터와 이상치 소량을 함께 학습
- 정상 데이터와 이상치, 각각에서 추출한 특징 간 차이가 두드러지도록 학습
- 한계: 학습한 이상치에 대해서만 과적합 될 가능성 ↑



Supervised AD

❖ **Background**

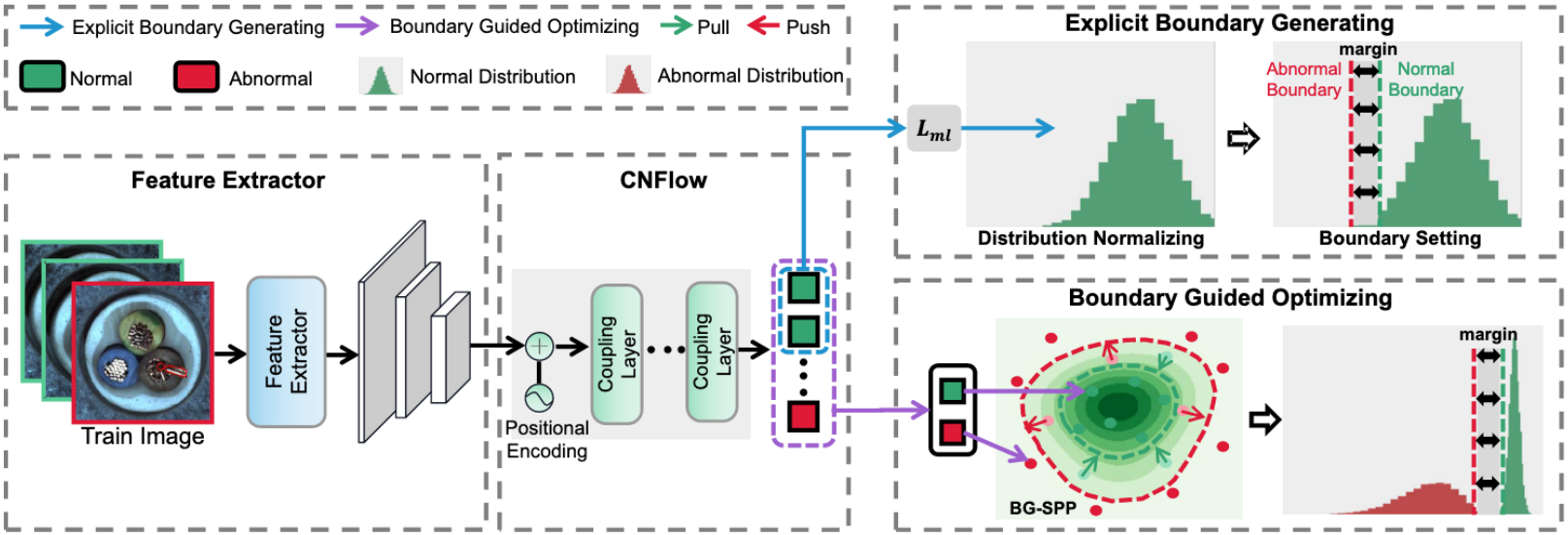
- ➢ 정상 데이터와 이상치 소량을 함께 학습
- ➢ 정상 데이터와 이상치, 각각에서 추출한 특징 간 차이가 두드러지도록 학습
- ➢ 한계: 학습한 이상치에 대해서만 과적합 될 가능성 ↑



Supervised AD

❖ Explicit Boundary Guided Semi-Push-Pull Contrastive Learning for Supervised Anomaly Detection (CVPR'23)

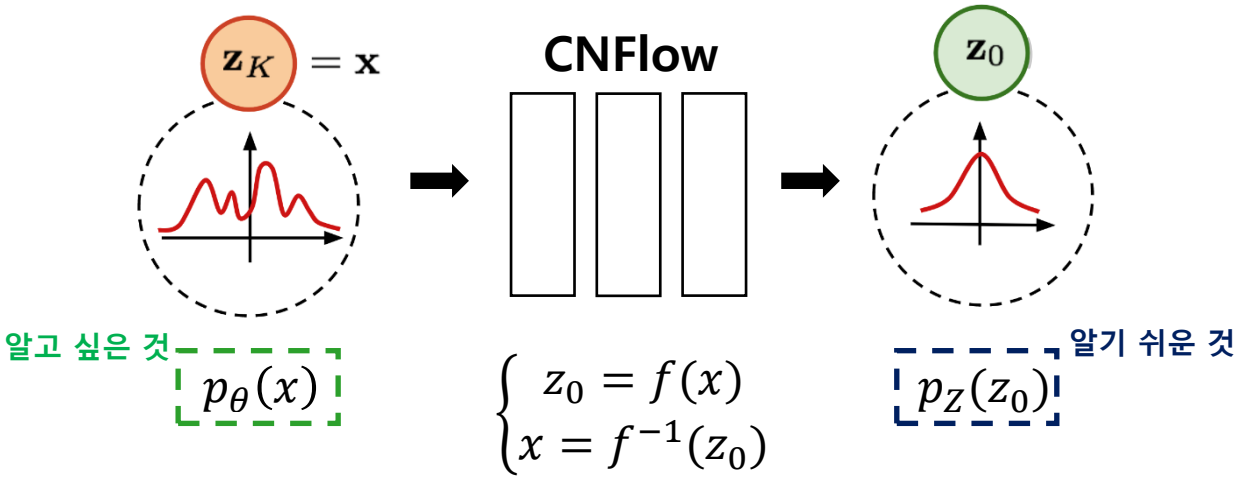
- ➢ 이상치 소량 활용 → 이상치에 대한 변별력 향상
- ➢ BGAD(Proposed Method): 학습하지 않은 이상치에 대해서도 높은 일반화 성능 확보



BGAD

❖ Step 1. Explicit Boundary Generating

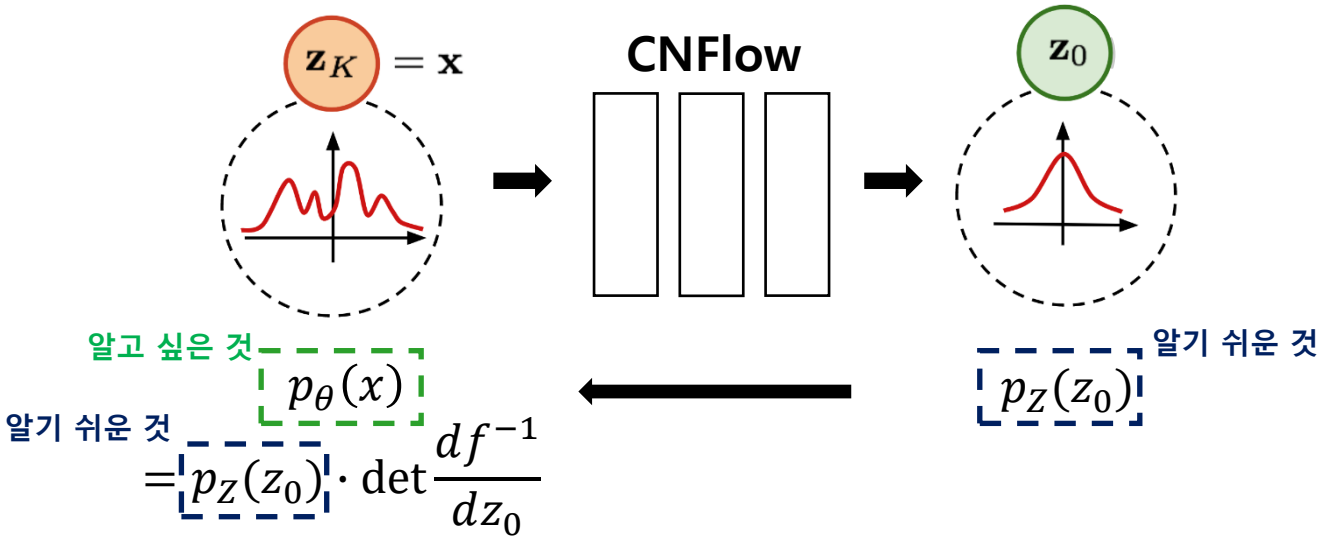
- ➢ Normalizing flow: 경계 설정을 위한 정상 데이터에 대한 분포 추정 단계
- ➢ 입력 데이터 분포 $p_\theta(x)$를 표준 정규 분포 $N(0,1)$에 대한 식으로 표현
  - f가 학습 대상 (CNFlow, 가역 연산으로 구성)
- ➢ 손실 함수: $\sum_x -log\, p_\theta(x)$ (MLE)
  - Goal: CNFlow가 입력 정상 데이터를 잘 나타내는 분포를 찾도록 학습



BGAD

❖ Step 1. Explicit Boundary Generating

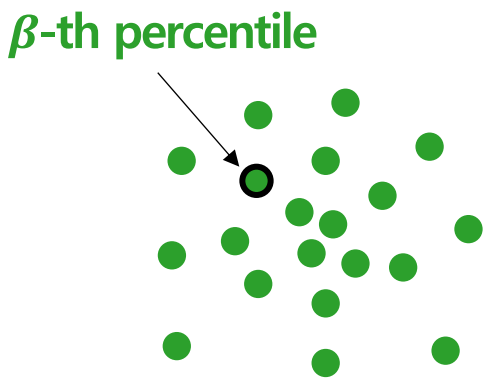
- ➢ Normalizing flow: 경계 설정을 위한 정상 데이터에 대한 분포 추정 단계
- ➢ 입력 데이터 분포 $p_\theta(x)$를 표준 정규 분포 $N(0,1)$에 대한 식으로 표현
  - f가 학습 대상 (CNFlow, 가역 연산으로 구성)
- ➢ 손실 함수: $\sum_x -log\, p_\theta(x)$ (MLE)
  - Goal: CNFlow가 입력 정상 데이터를 잘 나타내는 분포를 찾도록 학습



BGAD

❖ **Step 1. Explicit Boundary Generating**

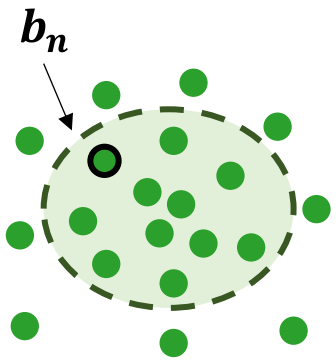
- Finding Boundary: 정상 데이터 분포에 기반하여 이상치 탐지 경계 설정
- 이상치 점수: $s(x) = 1 - p_\theta(x)$
- 하이퍼 파라미터 두 가지 사용
  - β: $s(x)$ 기준 β-th percentile에 해당하는 $log\, p_\theta(x)$를 정상 데이터 경계로 설정 → b_n
  - τ: 강건한 이상치 탐지를 위한 추가 여백 / 이상치 경계 $b_a = b_n - \tau$



BGAD

❖ Step 1. Explicit Boundary Generating

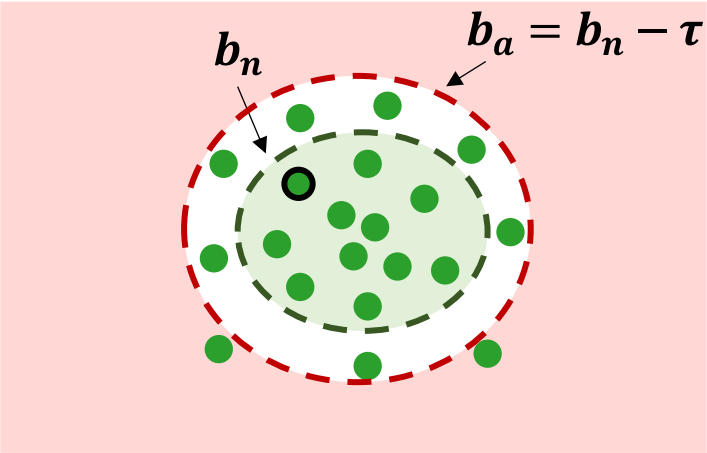
- ➢ Finding Boundary: 정상 데이터 분포에 기반하여 이상치 탐지 경계 설정
- ➢ 이상치 점수: $s(x) = 1 - p_\theta(x)$
- ➢ 하이퍼 파라미터 두 가지 사용
  - β: $s(x)$ 기준 β-th percentile에 해당하는 $log\, p_\theta(x)$를 정상 데이터 경계로 설정 → b_n
  - τ: 강건한 이상치 탐지를 위한 추가 여백 / 이상치 경계 $b_a = b_n - \tau$



BGAD

❖ Step 1. Explicit Boundary Generating

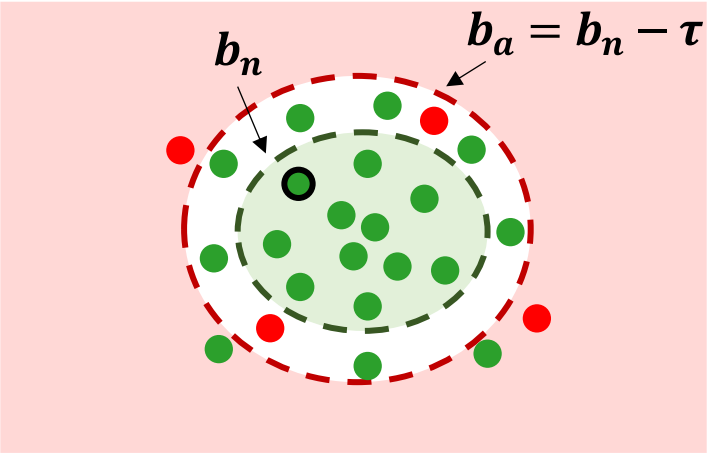
- ➢ Finding Boundary: 정상 데이터 분포에 기반하여 이상치 탐지 경계 설정
- ➢ 이상치 점수: $s(x) = 1 - p_\theta(x)$
- ➢ 하이퍼 파라미터 두 가지 사용
  - β: $s(x)$ 기준 β-th percentile에 해당하는 $log\, p_\theta(x)$를 정상 데이터 경계로 설정 → b_n
  - τ: 강건한 이상치 탐지를 위한 추가 여백 / 이상치 경계 $b_a = b_n - \tau$



BGAD

❖ Step 2. Boundary guided semi-push-pull

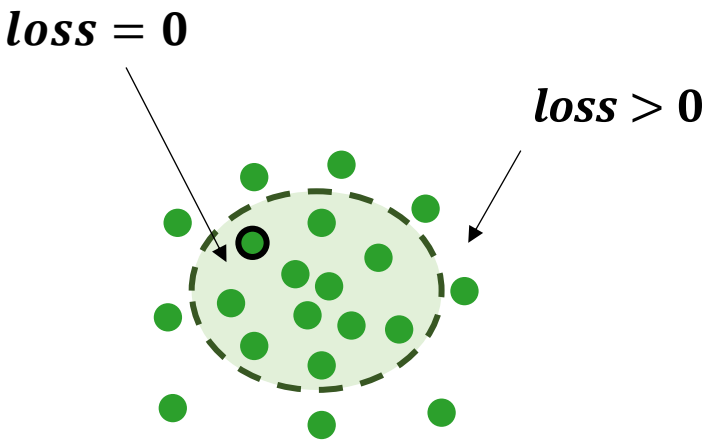
- Step 1에서 설정한 경계를 명시적으로 이용 (Boundary guided)
- Step 1과 달리 이상치를 학습에 함께 사용
- 정상 데이터는 b_n 안으로 당기고! (Pull)
- 이상치는 b_a 밖으로 밀도록 학습! (Push)



BGAD

❖ **Step 2. Boundary guided semi-push-pull**

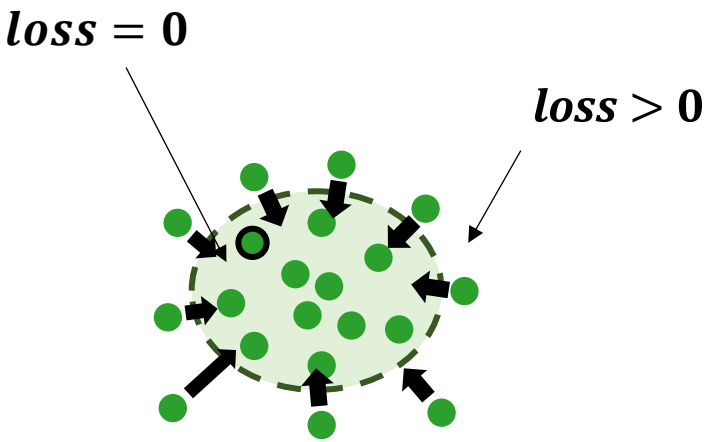
- 전체 손실 함수: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)| + \sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$
- 정상 데이터 안쪽으로 당기기: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)|$
- 이상치 바깥으로 밀기: $\sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$



BGAD

❖ **Step 2. Boundary guided semi-push-pull**

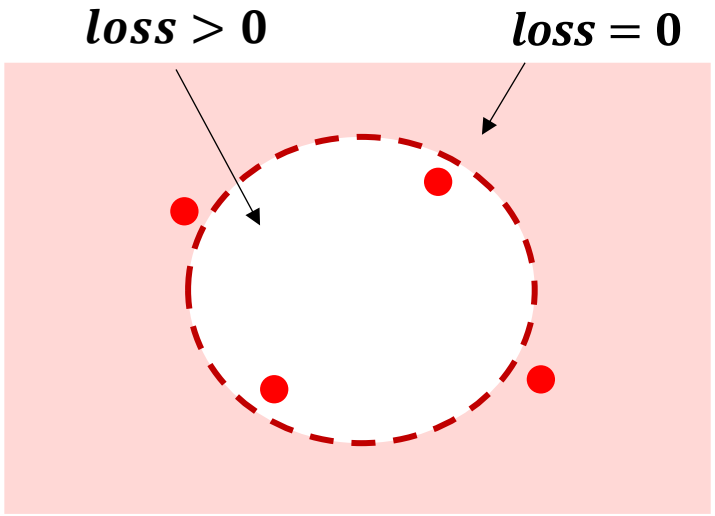
- 전체 손실 함수: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)| + \sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$
- 정상 데이터 안쪽으로 당기기: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)|$
- 이상치 바깥으로 밀기: $\sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$



BGAD

❖ **Step 2. Boundary guided semi-push-pull**

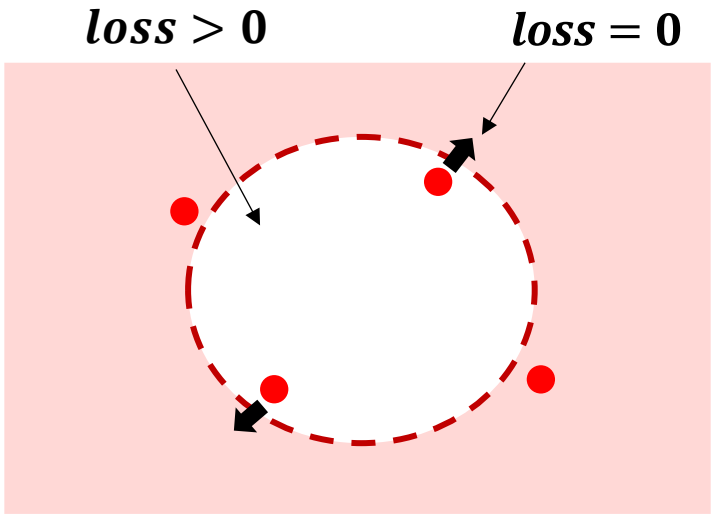
- 전체 손실 함수: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)| + \sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$
- 정상 데이터 안쪽으로 당기기: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)|$
- 이상치 바깥으로 밀기: $\sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$



BGAD

❖ **Step 2. Boundary guided semi-push-pull**

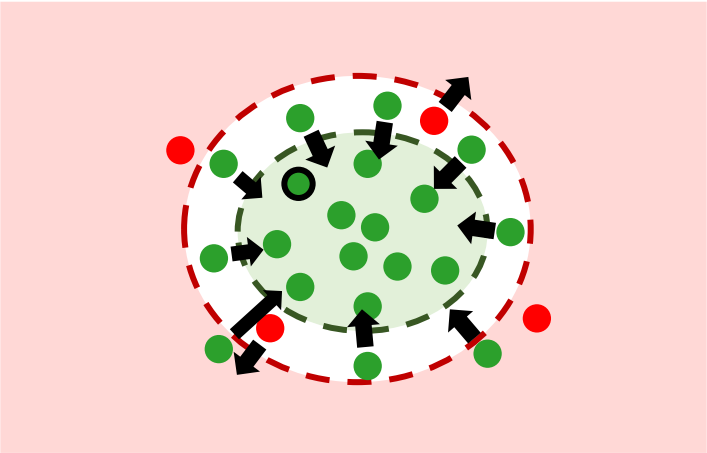
- 전체 손실 함수: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)| + \sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$
- 정상 데이터 안쪽으로 당기기: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)|$
- 이상치 바깥으로 밀기: $\sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$



BGAD

❖ **Step 2. Boundary guided semi-push-pull**

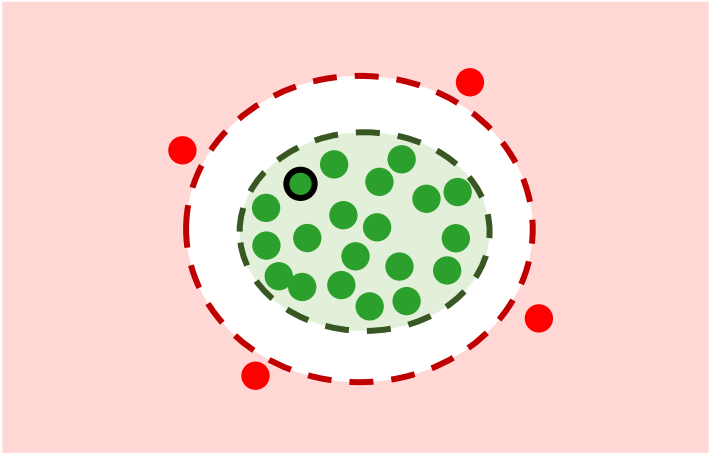
- 전체 손실 함수: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)| + \sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$
- 정상 데이터 안쪽으로 당기기: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)|$
- 이상치 바깥으로 밀기: $\sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$



BGAD

❖ **Step 2. Boundary guided semi-push-pull**

- 전체 손실 함수: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)| + \sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$
- 정상 데이터 안쪽으로 당기기: $\sum_{i=1}^{N} |min((log\, p_i - b_n), 0)|$
- 이상치 바깥으로 밀기: $\sum_{j=1}^{M} |max((log\, p_j - b_n + \tau), 0)|$



BGAD

❖ Experiments

- ➢ 데이터셋: MVTecAD Dataset 사용 (Multi-category Image AD Dataset)
- ➢ Supervised AD: 카테고리 별 이상치 10장을 학습에 활용
- ➢ 정량 평가: 세 가지 성능 지표 사용
  - Image level AUROC : 이미지에 대한 anomaly detection 성능 평가
  - Pixel level AUROC : 픽셀에 대한 anomaly detection 성능 평가 (Anomaly localization)
  - Per-Region-Overlap (PRO) Curve : Pixel level AUROC에 비해 이상 영역 크기에 강건한 평가 지표

	Category	Unsupervised AD Methods							Supervised AD Method
		DRAEM* [52]	PaDiM* [10]	MSFD* [47]	PatchCore* [32]	CFA* [20]	NFAD‡	BGAD$^{w/o}$ (Ours)	BGAD (Ours)
Textures	Carpet	0.954/0.947	0.983/0.946	0.990/0.958	0.985/0.959	0.989/0.943	0.994/0.983	0.994/0.982	**0.996**±0.0002/**0.989**±0.0004
	Grid	0.997/0.984	0.963/0.894	0.986/0.937	0.974/0.891	0.977/0.932	0.993/0.980	0.994/0.980	**0.995**±0.0002/**0.986**±0.0001
	Leather	0.992/0.981	0.984/0.966	0.978/0.924	0.992/0.974	0.991/0.958	0.997/0.994	0.997/0.994	**0.998**±0.0001/**0.994**±0.0003
	Tile	0.994/0.949	0.958/0.884	0.952/0.841	0.960/0.939	0.960/0.860	0.969/0.929	0.968/0.927	**0.994**±0.0077/**0.978**±0.0021
	Wood	0.962/0.935	0.963/0.891	0.953/0.925	0.968/0.857	0.948/0.882	0.969/0.957	0.970/0.957	**0.982**±0.0053/**0.970**±0.0007
Objects	Bottle	0.993/0.955	0.978/0.936	0.985/0.940	0.986/0.956	0.987/0.944	0.988/0.965	0.989/0.964	**0.994**±0.0009/**0.971**±0.0011
	Cable	0.961/0.910	0.979/0.973	0.972/0.922	0.986/**0.980**	**0.987**/0.931	0.975/0.944	0.980/0.968	0.986±0.0010/0.977±0.0030
	Capsule	0.869/0.901	0.980/0.924	0.979/0.878	0.990/0.946	0.989/0.943	0.989/0.952	0.992/0.959	**0.992**±0.0021/**0.964**±0.0033
	Hazelnut	0.997/0.985	0.980/0.951	0.982/0.968	0.988/0.924	0.986/0.953	0.984/0.976	0.985/0.976	**0.995**±0.0040/**0.982**±0.0028
	Metal nut	0.992/0.935	0.979/0.929	0.972/0.985	0.986/0.935	0.987/0.918	0.971/0.942	0.976/0.948	**0.996**±0.0003/**0.970**±0.0012
	Pill	0.979/0.959	0.978/0.957	0.971/0.929	0.983/0.947	0.986/0.965	0.976/0.978	0.980/0.980	**0.996**±0.0002/**0.988**±0.0005
	Screw	0.992/0.965	0.974/0.923	0.983/0.924	0.984/0.928	0.985/0.944	0.988/0.945	0.992/0.960	**0.993**±0.0003/**0.968**±0.0010
	Toothbrush	0.970/0.940	0.980/0.894	0.986/0.877	0.987/0.939	0.989/0.894	0.983/0.904	0.986/0.938	**0.995**±0.0003/**0.961**±0.0026
	Transistor	0.970/0.935	0.983/0.967	0.886/0.781	0.964/0.967	**0.985**/0.960	0.923/0.788	0.940/0.830	0.983±0.0005/**0.972**±0.0015
	Zipper	0.984/0.966	0.978/0.948	0.981/0.935	0.986/0.963	0.988/0.944	0.986/0.957	0.987/0.957	**0.993**±0.0003/**0.977**±0.0002
	Mean	0.969/0.947	0.976/0.932	0.970/0.915	0.981/0.940	0.982/0.931	0.979/0.946	0.982/0.955	**0.992**±0.0007/**0.976**±0.0006
	Image-level Mean	0.978	0.975	0.964	0.988	0.989	0.968	0.974	**0.993**±0.0012

BGAD

❖ Experiments

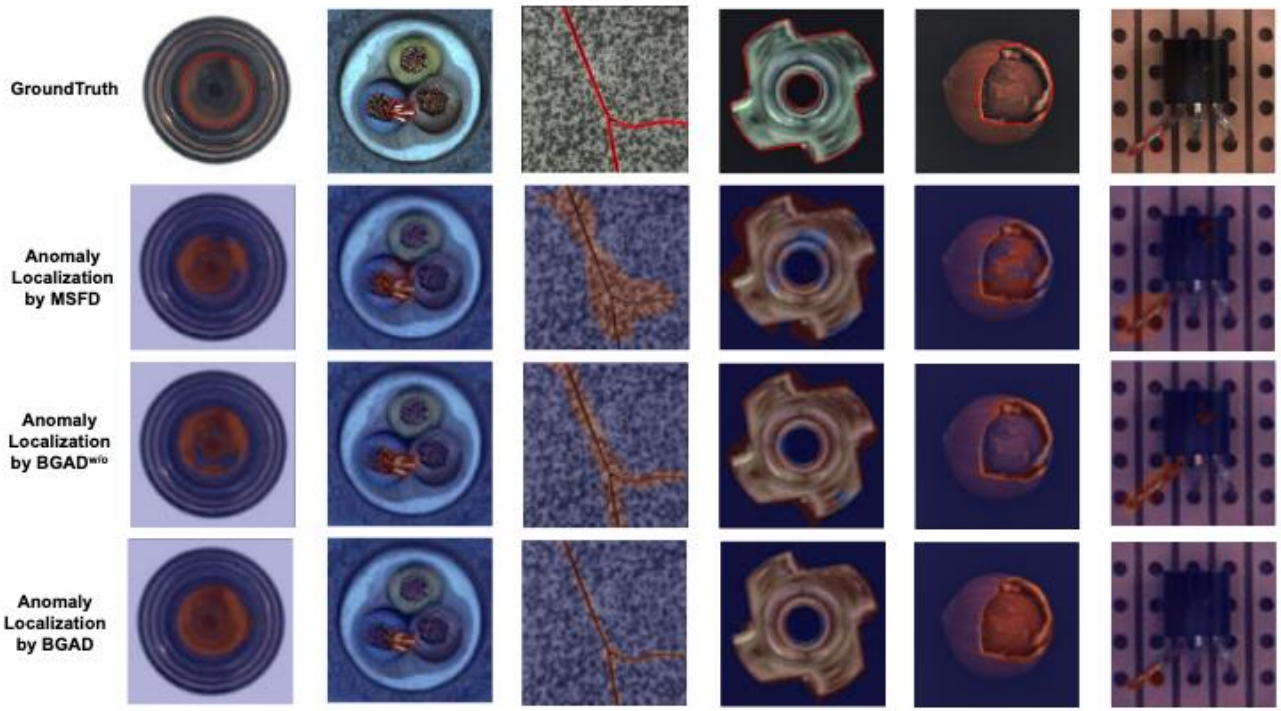
- ➢ 데이터셋: MVTecAD Dataset 사용 (Multi-category Image AD Dataset)
- ➢ Supervised AD: 카테고리 별 이상치 10장을 학습에 활용
- ➢ 정량 평가: 세 가지 성능 지표 사용
  - Image level AUROC : 이미지에 대한 anomaly detection 성능 평가
  - Pixel level AUROC : 픽셀에 대한 anomaly detection 성능 평가 (Anomaly localization)
  - Per-Region-Overlap (PRO) Curve : Pixel level AUROC에 비해 이상 영역 크기에 강건한 평가 지표

Category	Supervised AD Methods (Ten Abnormal Samples)			
	FCDD* [24]	DevNet* [27]	DRA* [12]	BGAD (Ours)
Carpet	0.981/0.952	-/-	-/-	**0.996**±0.0002/**0.989**±0.0004
Grid	0.949/0.897	-/-	-/-	**0.995**±0.0002/**0.986**±0.0001
Leather	0.984/0.973	-/-	-/-	**0.998**±0.0001/**0.994**±0.0003
Tile	0.977/0.938	-/-	-/-	**0.994**±0.0077/**0.978**±0.0021
Wood	0.950/0.901	-/-	-/-	**0.982**±0.0053/**0.970**±0.0007
Bottle	0.966/0.939	-/-	-/-	**0.994**±0.0009/**0.971**±0.0011
Cable	0.963/0.980	-/-	-/-	0.986±0.0010/0.977±0.0030
Capsule	0.970/0.922	-/-	-/-	**0.992**±0.0021/**0.964**±0.0033
Hazelnut	0.970/0.958	-/-	-/-	**0.995**±0.0040/**0.982**±0.0028
Metal nut	0.966/0.934	-/-	-/-	**0.996**±0.0003/**0.970**±0.0012
Pill	0.975/0.960	-/-	-/-	**0.996**±0.0002/**0.988**±0.0005
Screw	0.963/0.925	-/-	-/-	**0.993**±0.0003/**0.968**±0.0010
Toothbrush	0.967/0.907	-/-	-/-	**0.995**±0.0003/**0.961**±0.0026
Transistor	0.942/0.935	-/-	-/-	0.983±0.0005/**0.972**±0.0015
Zipper	0.968/0.948	-/-	-/-	**0.993**±0.0003/**0.977**±0.0002
Mean	0.966/0.938	-/-	-/-	**0.992**±0.0007/**0.976**±0.0006
Image-level Mean	0.965	0.948	0.961	**0.993**±0.0012

BGAD

❖ **Experiments**

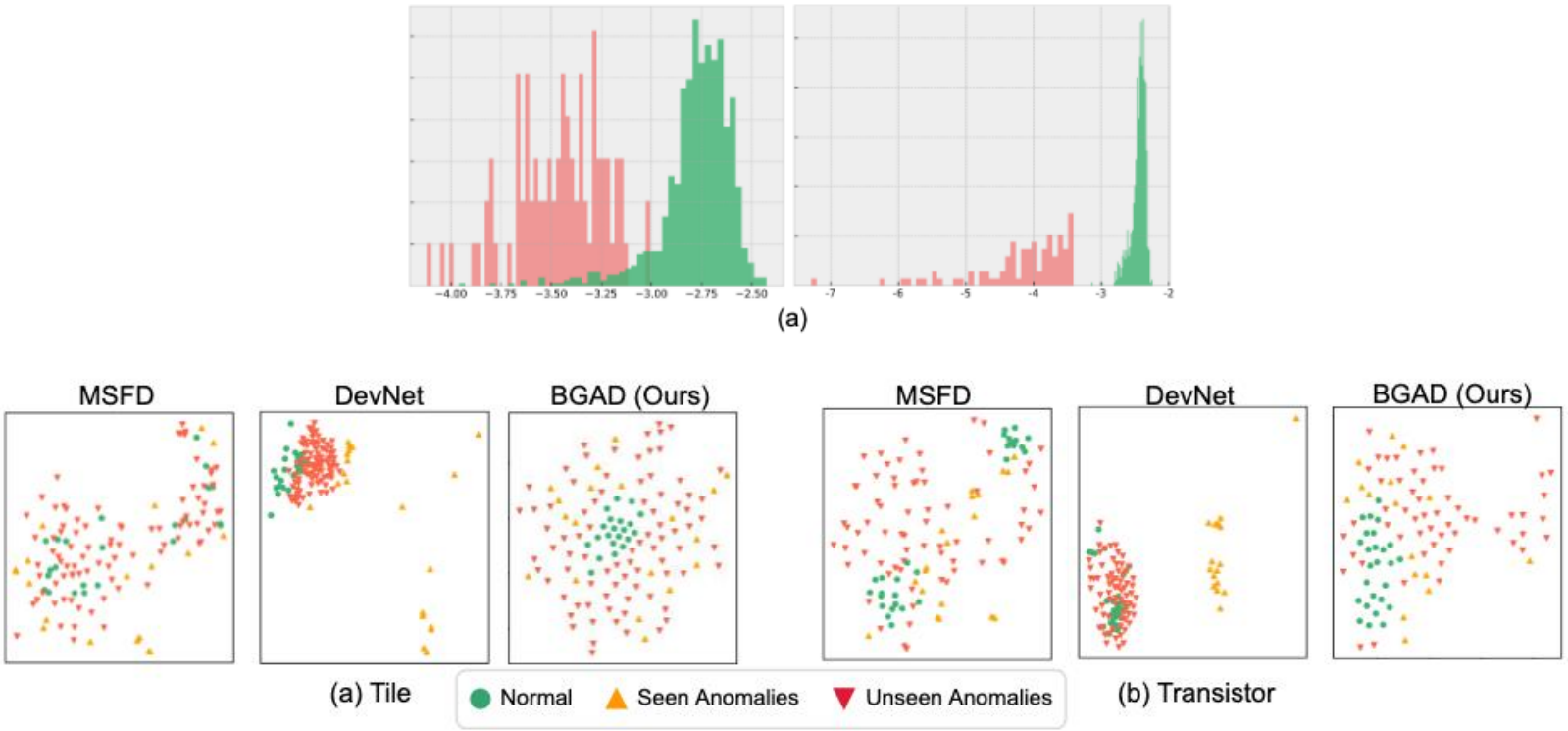
- ➢ PRO Curve 값이 타 평가 지표 대비 눈에 띄게 향상 → Anomaly localization에 유용
- ➢ 정성 평가 역시 뛰어난 localization 성능을 보임을 확인



BGAD

❖ **Experiments**

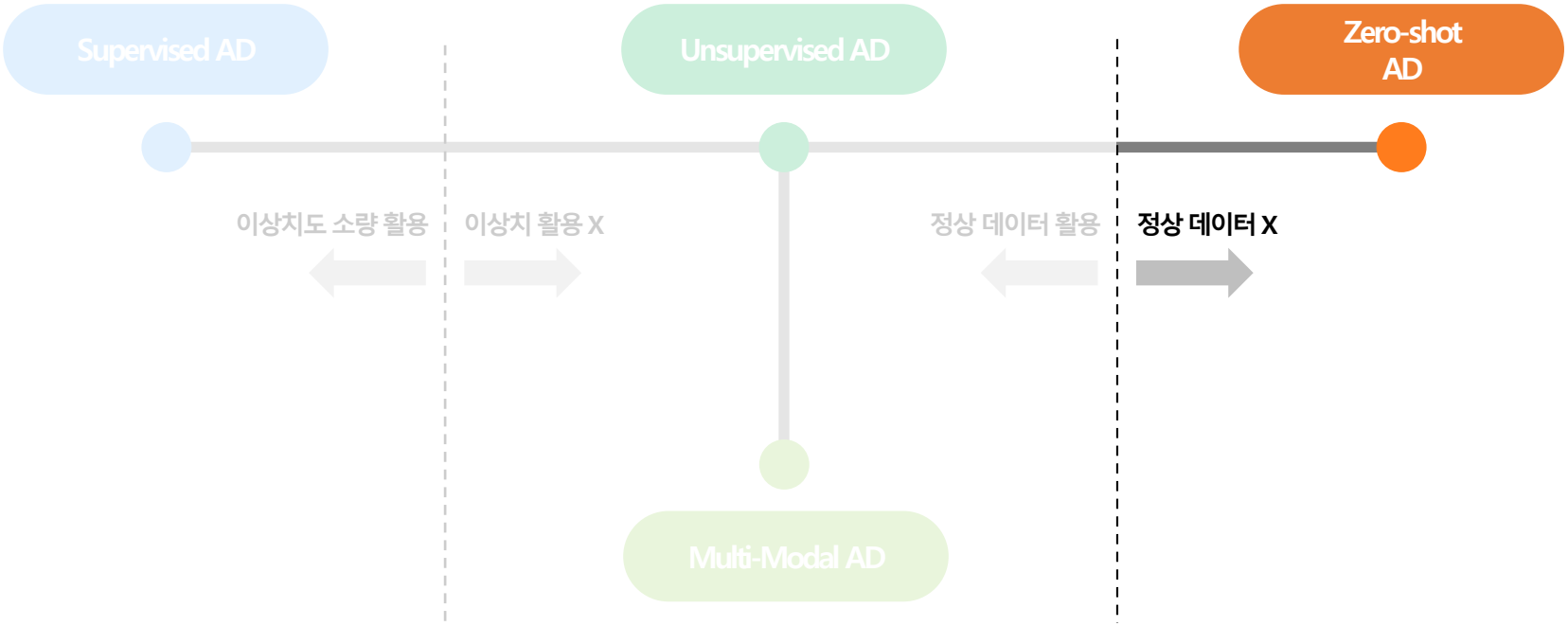
- ➢ 이상치를 소량 사용하였을 때, 정상 데이터와 이상치 간 차이가 두드러짐을 확인
- ➢ 그러면서도 학습에 사용하지 않은 이상치에 대해서도 일반적임을 확인



Zero-shot AD

❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

- Topic 1: 무엇보다 AD 정확도가 중요 → Supervised AD
- **Topic 2: 정상 데이터도 없음 → Zero-shot AD**
- Topic 3 : 이미지 정보만으로는 탐지할 수 없는 이상치 → Multi-Modal AD

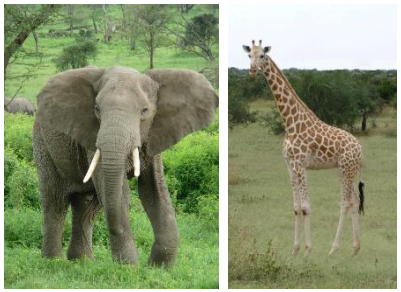


Zero-shot AD

❖ **Zero-shot Learning**

- Goal: 모델이 학습하지 않은 과제도 수행할 수 있어야 함
- ① **방대한 사전지식**과 ② **뛰어난 응용 능력**을 요구
- 위와 같은 모델이 있다는 전제 하에, 이를 활용하여
  - ➢ **Zero-shot Classification**: 새로운 분류 상황에 대한 추가 학습 없이 바로 적용
  - ➢ **Zero-shot AD**: 새로운 AD 상황에 대한 추가 학습 없이 바로 적용

사전학습: 코끼리와 기린을 분류해주세요



코끼리는 이렇고, 기린은 저렇구나.
둘은 이렇게 분류하면 되겠다.

Zero-shot AD

❖ **Zero-shot Learning**

- Goal: 모델이 학습하지 않은 과제도 수행할 수 있어야 함
- ① **방대한 사전지식**과 ② **뛰어난 응용 능력**을 요구
- 위와 같은 모델이 있다는 전제 하에, 이를 활용하여
  - ➢ **Zero-shot Classification**: 새로운 분류 상황에 대한 추가 학습 없이 바로 적용
  - ➢ **Zero-shot AD**: 새로운 AD 상황에 대한 추가 학습 없이 바로 적용

지식 응용: 어딘가 이상한 코끼리를 골라주세요

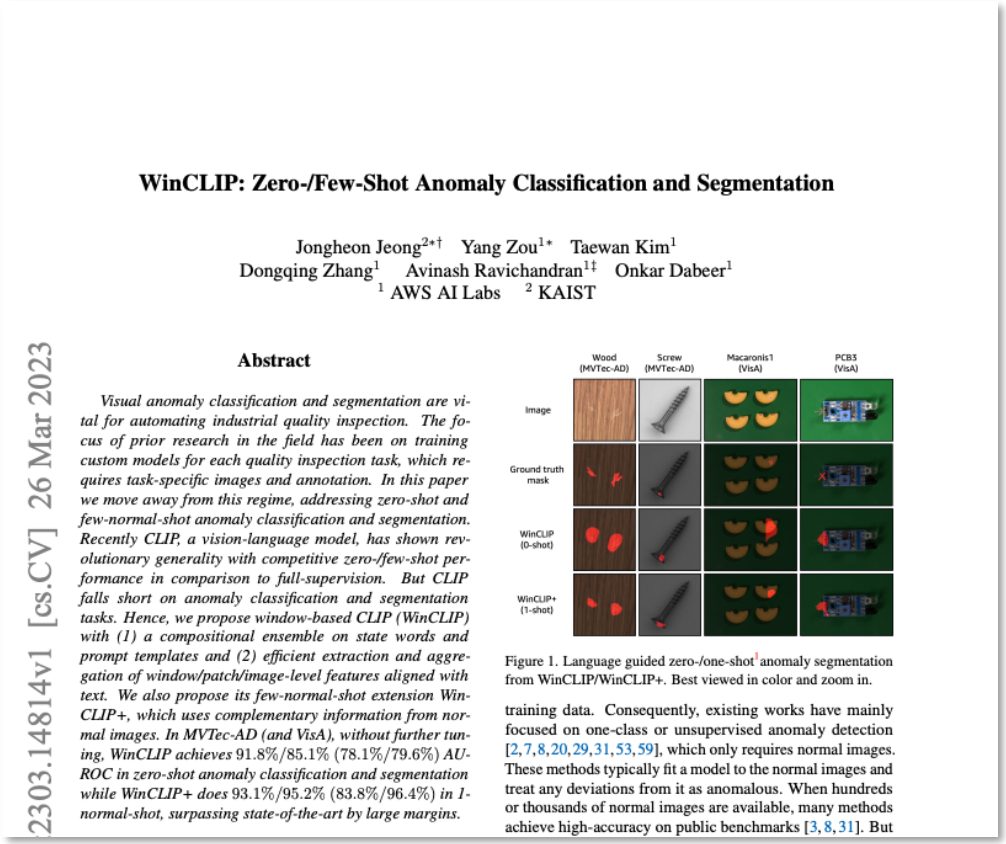
Zero-shot AD

❖ **WinCLIP: Zero-Shot Anomaly Classification and Segmentation (CVPR'23)**

- 대규모 데이터로 학습한 vision-language model (VLM)인 CLIP 사용 (Zero-shot Classification)

① **Compositional Prompt Ensemble**: 부가적 학습 없이 프롬프트 변화만으로 zero-shot AD 수행

② **Window-based CLIP**: Segmentation을 위한 지역 정보 추출을 위해 windowing 적용



WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation

Jongheon Jeong[2*†] Yang Zou[1*] Taewan Kim[1]
Dongqing Zhang[1] Avinash Ravichandran[1‡] Onkar Dabeer[1]
[1] AWS AI Labs [2] KAIST

Abstract

Visual anomaly classification and segmentation are vital for automating industrial quality inspection. The focus of prior research in the field has been on training custom models for each quality inspection task, which requires task-specific images and annotation. In this paper we move away from this regime, addressing zero-shot and few-normal-shot anomaly classification and segmentation. Recently CLIP, a vision-language model, has shown revolutionary generality with competitive zero-/few-shot performance in comparison to full-supervision. But CLIP falls short on anomaly classification and segmentation tasks. Hence, we propose window-based CLIP (WinCLIP) with (1) a compositional ensemble on state words and prompt templates and (2) efficient extraction and aggregation of window/patch/image-level features aligned with text. We also propose its few-normal-shot extension WinCLIP+, which uses complementary information from normal images. In MVTec-AD (and VisA), without further tuning, WinCLIP achieves 91.8%/85.1% (78.1%/79.6%) *AUROC in zero-shot anomaly classification and segmentation while WinCLIP+ does* 93.1%/95.2% (83.8%/96.4%) *in 1-normal-shot, surpassing state-of-the-art by large margins.*

Figure 1. Language guided zero-/one-shot anomaly segmentation from WinCLIP/WinCLIP+. Best viewed in color and zoom in.

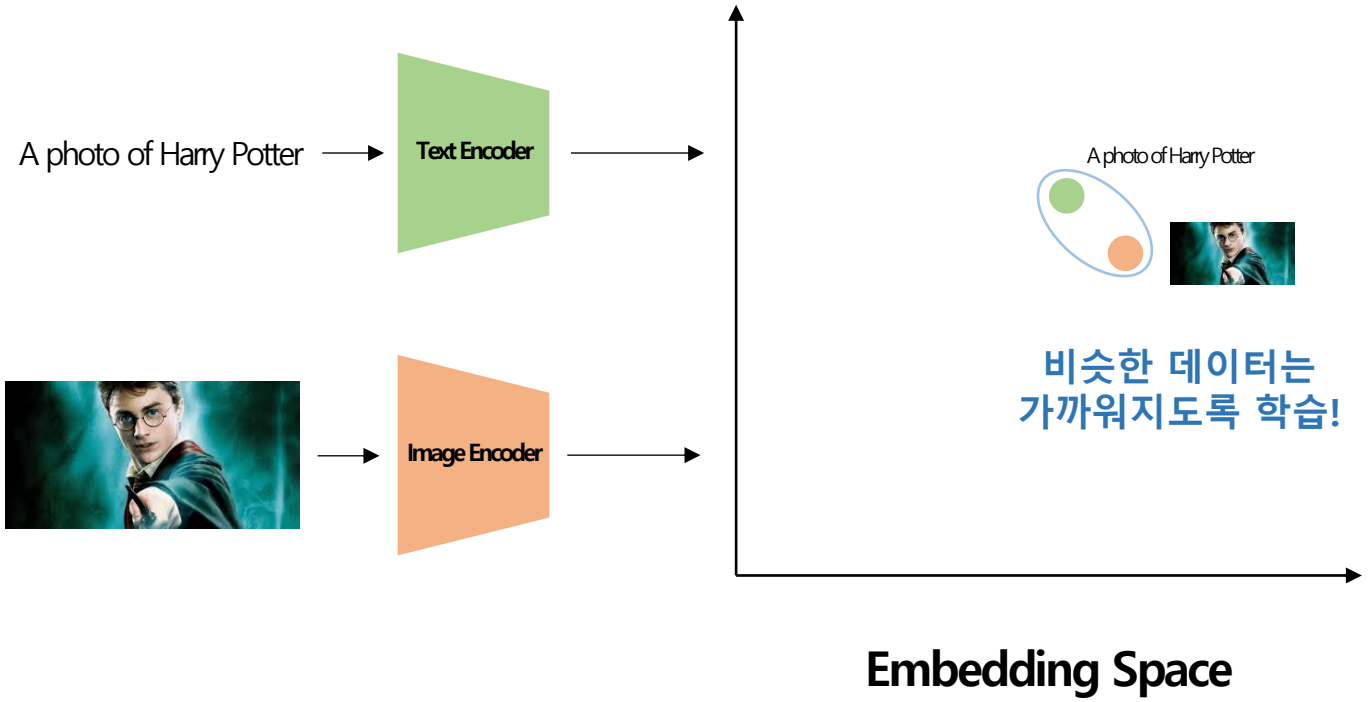
training data. Consequently, existing works have mainly focused on one-class or unsupervised anomaly detection [2,7,8,20,29,31,53,59], which only requires normal images. These methods typically fit a model to the normal images and treat any deviations from it as anomalous. When hundreds or thousands of normal images are available, many methods achieve high-accuracy on public benchmarks [3, 8, 31]. But

WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

- Zero-shot Classification: 어떤 레이블이 들어와도 분류가 가능한 모델
- 기존 분류 모델: 이미지를 고정된 레이블에 대응시켜 학습
- **CLIP: 이미지 4억 장과 이에 대한 설명 (자연어) 간 Contrastive Learning 수행**

  → 이미지와 자연어 간 **의미 관계를 학습**

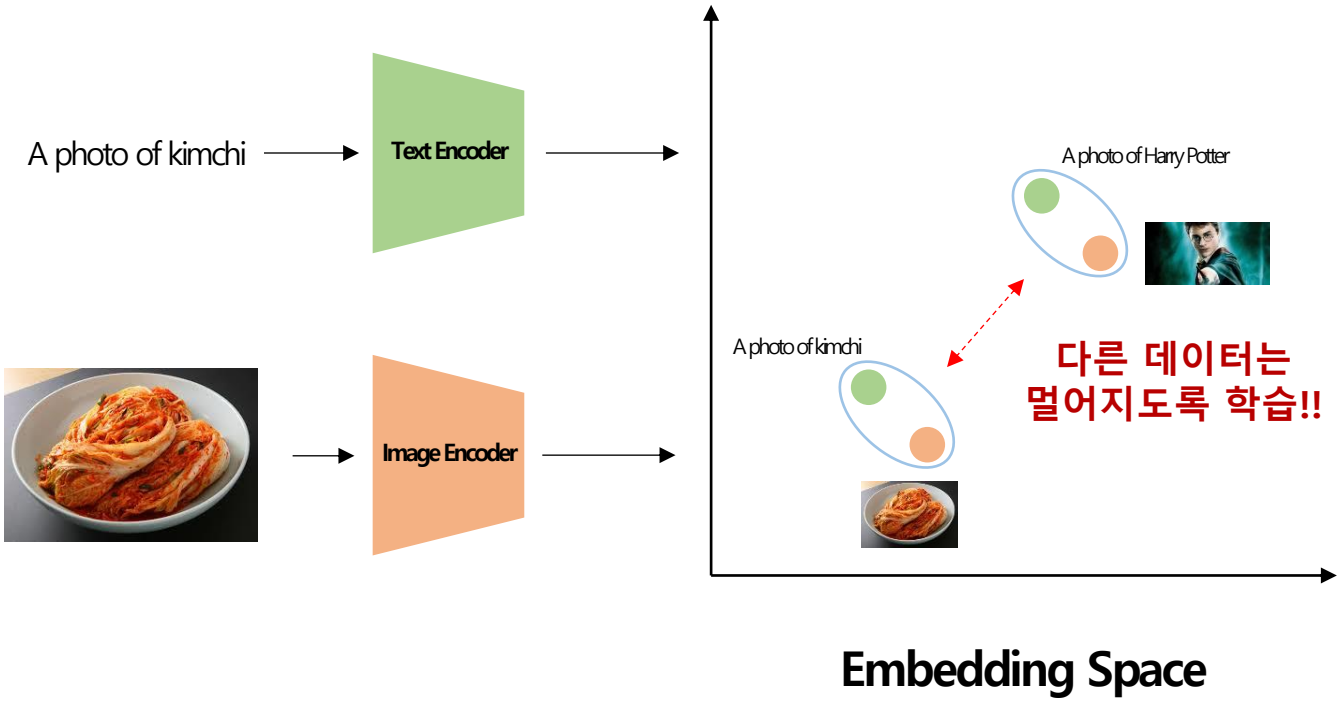


WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

- Zero-shot Classification: 어떤 레이블이 들어와도 분류가 가능한 모델
- 기존 분류 모델: 이미지를 고정된 레이블에 대응시켜 학습
- **CLIP: 이미지 4억 장과 이에 대한 설명 (자연어) 간 Contrastive Learning 수행**

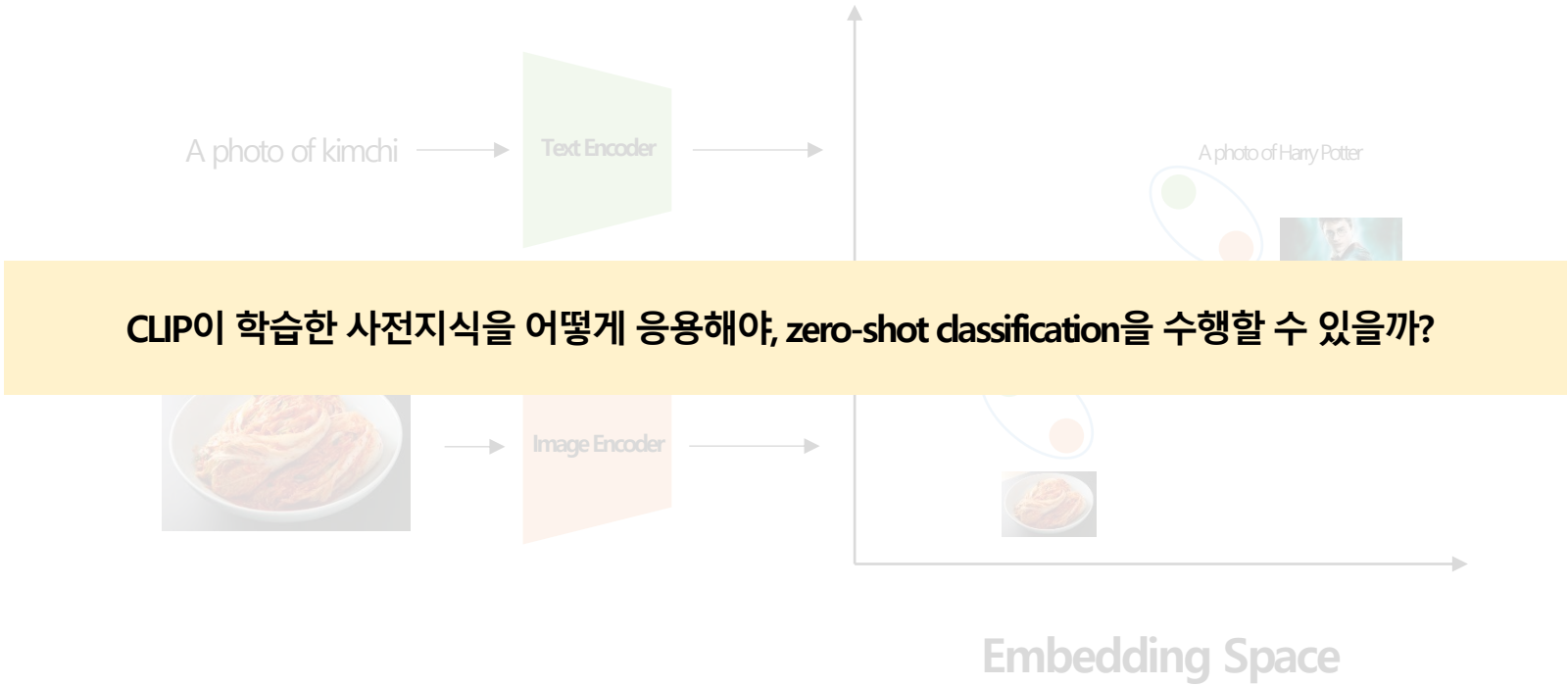
  → 이미지와 자연어 간 **의미 관계를 학습**



WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

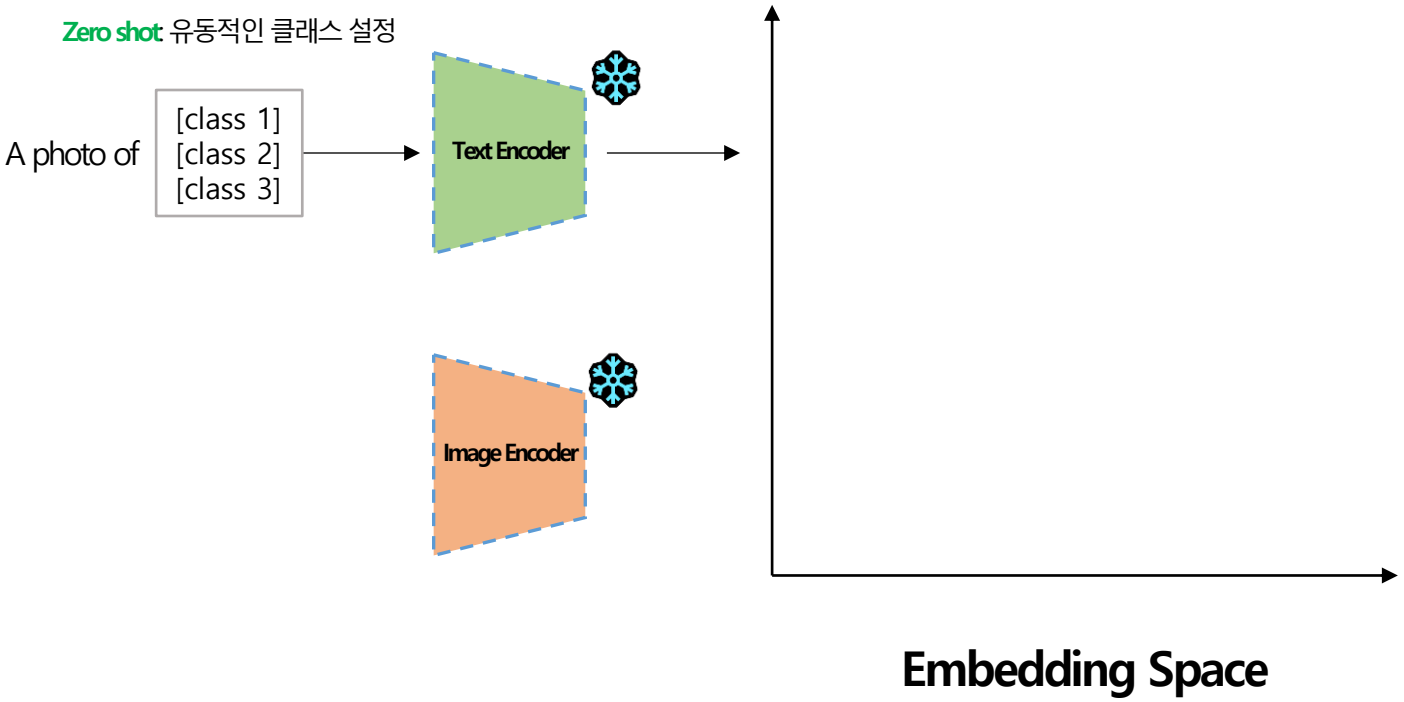
- Zero-shot Classification: 어떤 레이블이 들어와도 분류가 가능한 모델
- 기존 분류 모델: 이미지를 고정된 레이블에 대응시켜 학습
- **CLIP: 이미지 4억 장과 이에 대한 설명 (자연어) 간 Contrastive Learning 수행**
  → 이미지와 자연어 간 **의미 관계를 학습**



WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

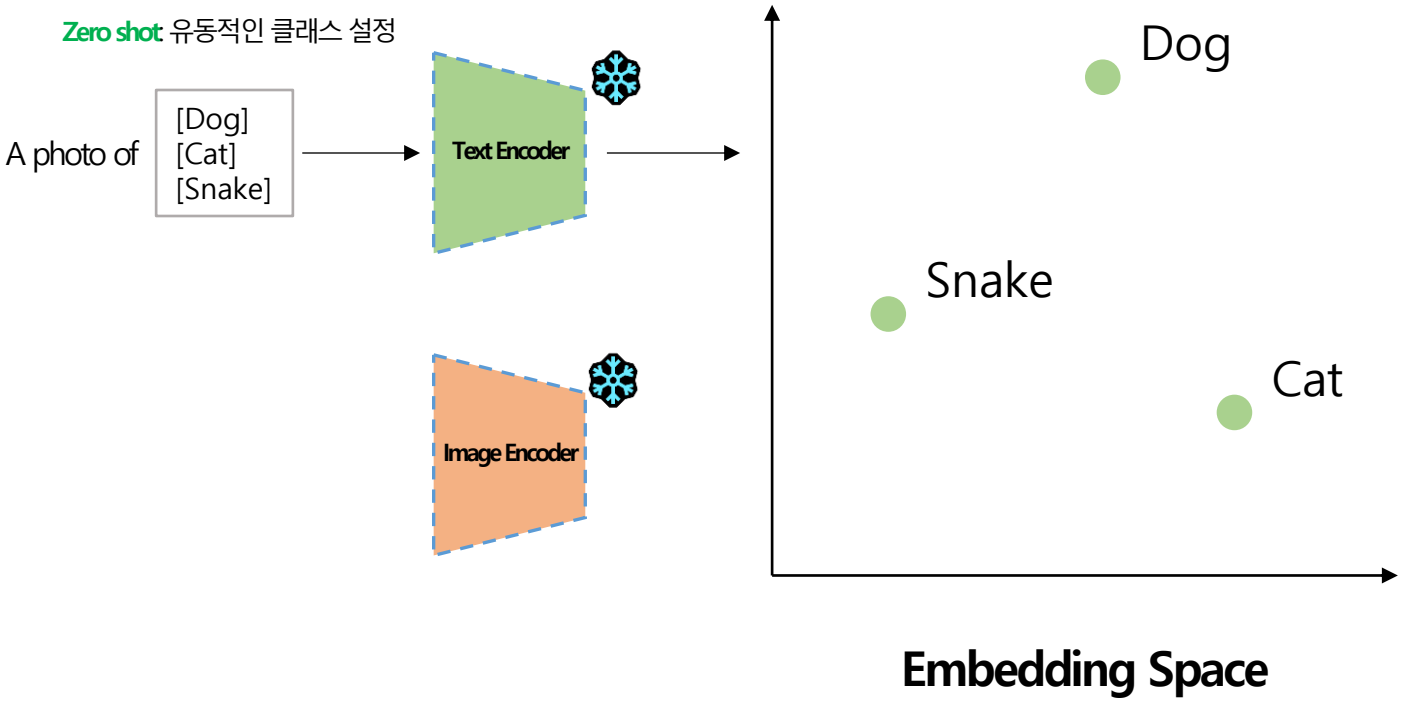
- Text Encoder: 자연어로 주어진 클래스 정보를 인코딩
- Image Encoder: 분류하고자 하는 이미지 정보를 인코딩
- 유사도 기반으로 분류 수행



WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

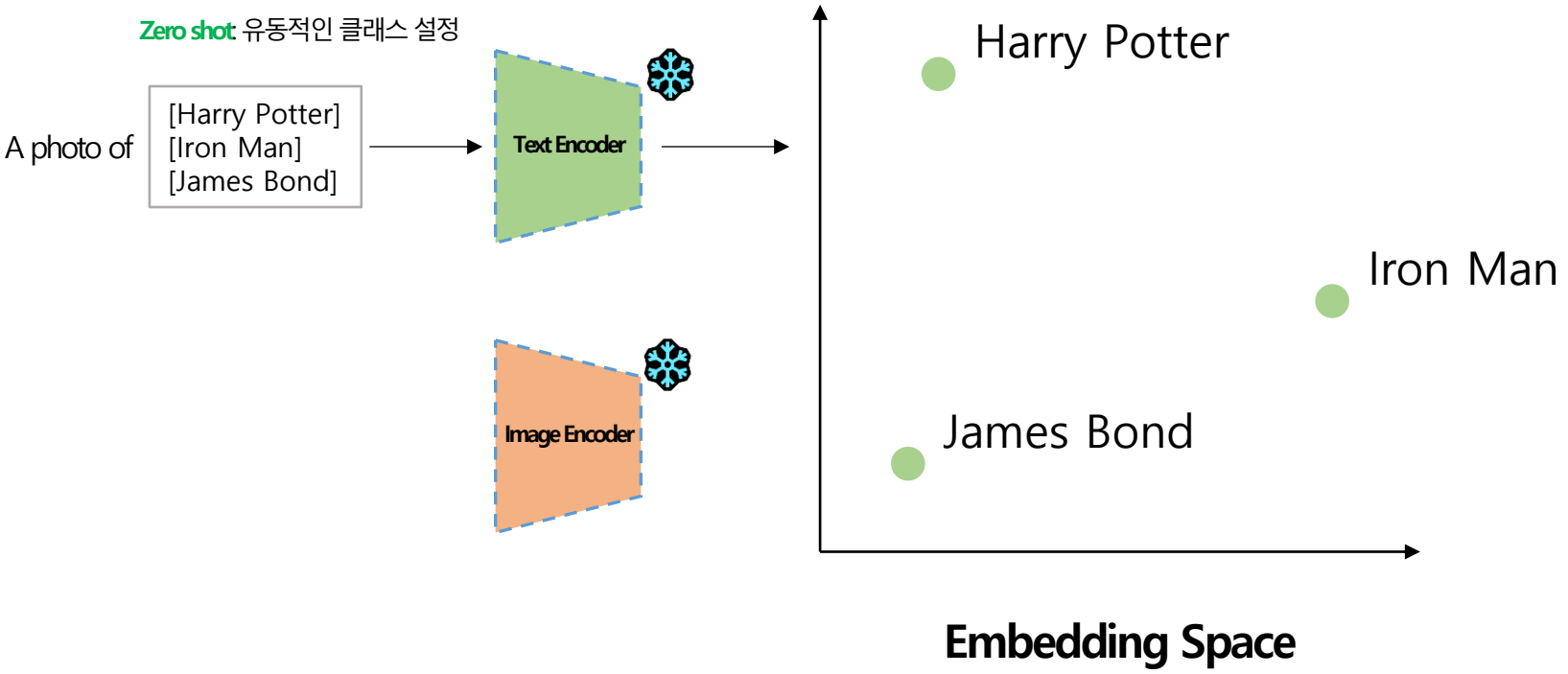
- Text Encoder: 자연어로 주어진 클래스 정보를 인코딩
- Image Encoder: 분류하고자 하는 이미지 정보를 인코딩
- 유사도 기반으로 분류 수행



WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

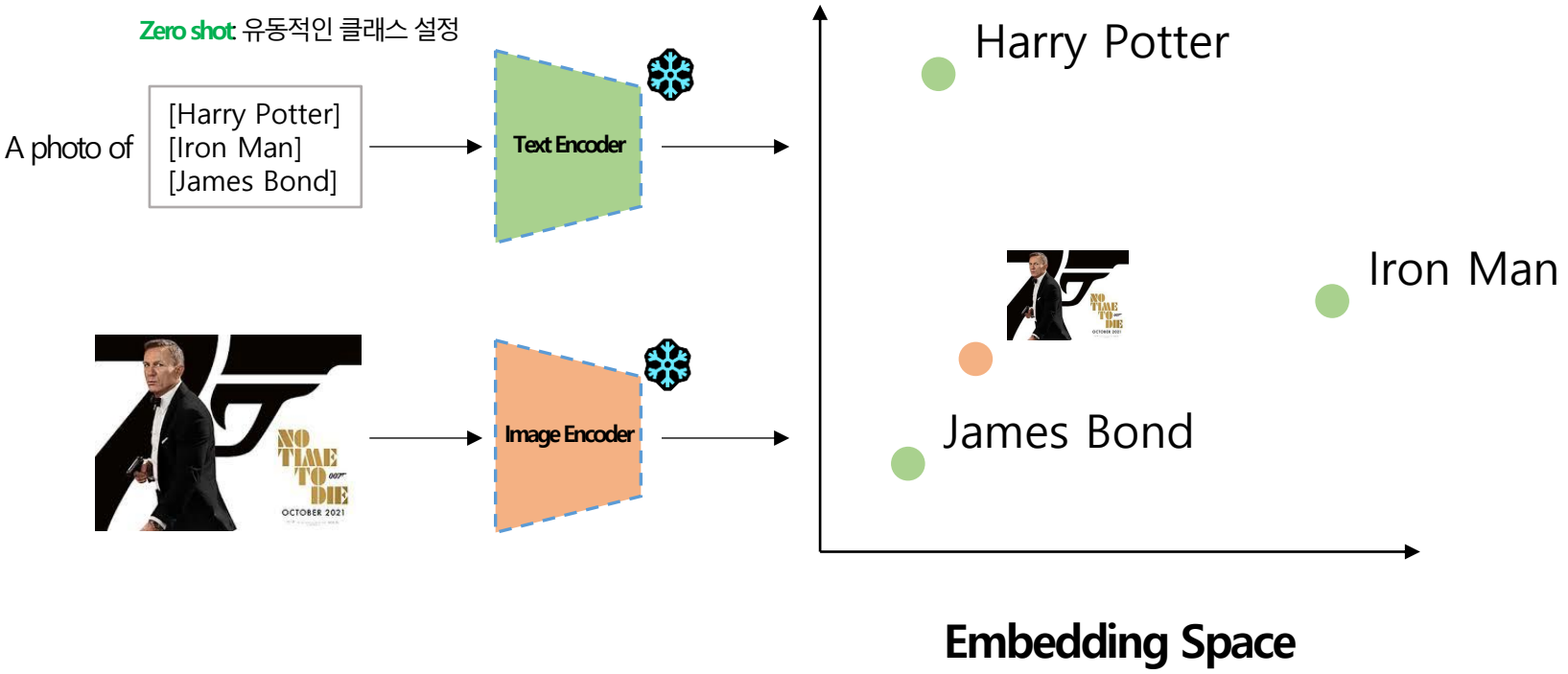
- Text Encoder: 자연어로 주어진 클래스 정보를 인코딩
- Image Encoder: 분류하고자 하는 이미지 정보를 인코딩
- 유사도 기반으로 분류 수행



WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

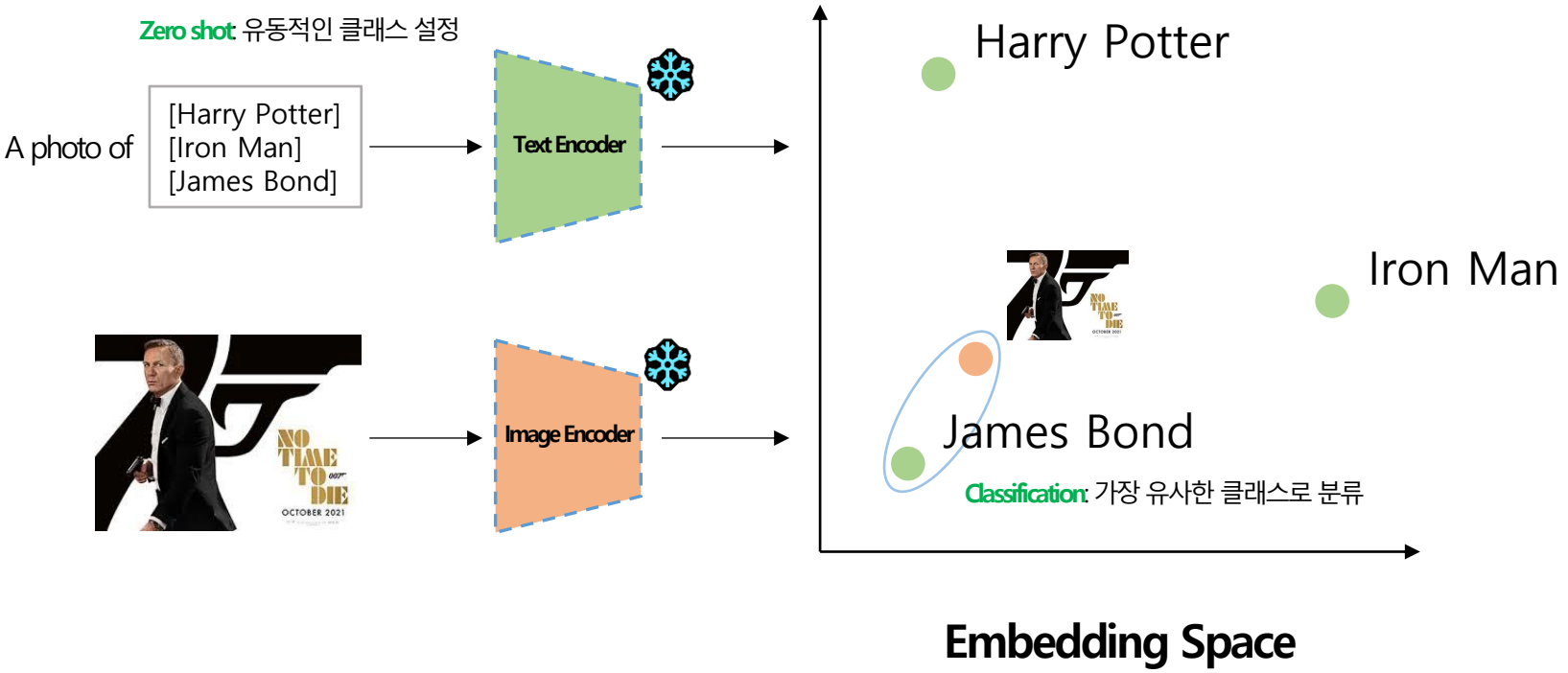
- Text Encoder: 자연어로 주어진 클래스 정보를 인코딩
- Image Encoder: 분류하고자 하는 이미지 정보를 인코딩
- 유사도 기반으로 분류 수행



WinCLIP

❖ **CLIP: Contrastive Language-Image Pre-Training Model**

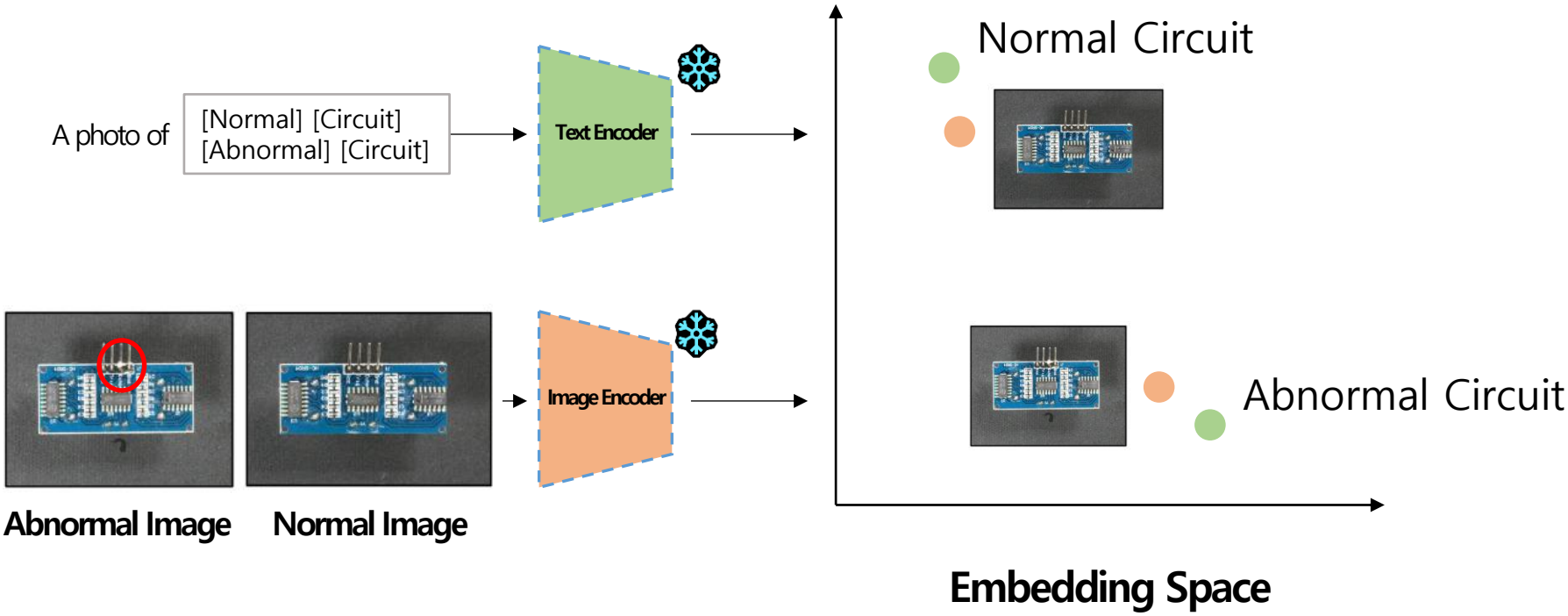
- Text Encoder: 자연어로 주어진 클래스 정보를 인코딩
- Image Encoder: 분류하고자 하는 이미지 정보를 인코딩
- 유사도 기반으로 분류 수행



WinCLIP

❖ Problem Definition ①

- 이상치 뿐 아니라, 학습을 위한 **정상 데이터도 없는 상황**
- **Question: CLIP이 가지고 있는 사전지식을 이상치 탐지에도 응용해볼 수는 없을까?**
- Baseline: CLIP에게 '정상'과 '비정상'을 클래스로 주고 이진 분류를 시켜보자!



WinCLIP

❖ **Problem Definition ①**

- **무엇이 정상이고, 비정상인가? → Task 맥락에 따라 다르다!**
- 정확한 이상치 탐지를 위해서는 **맥락에 대한 정보**를 제공할 필요가 있음

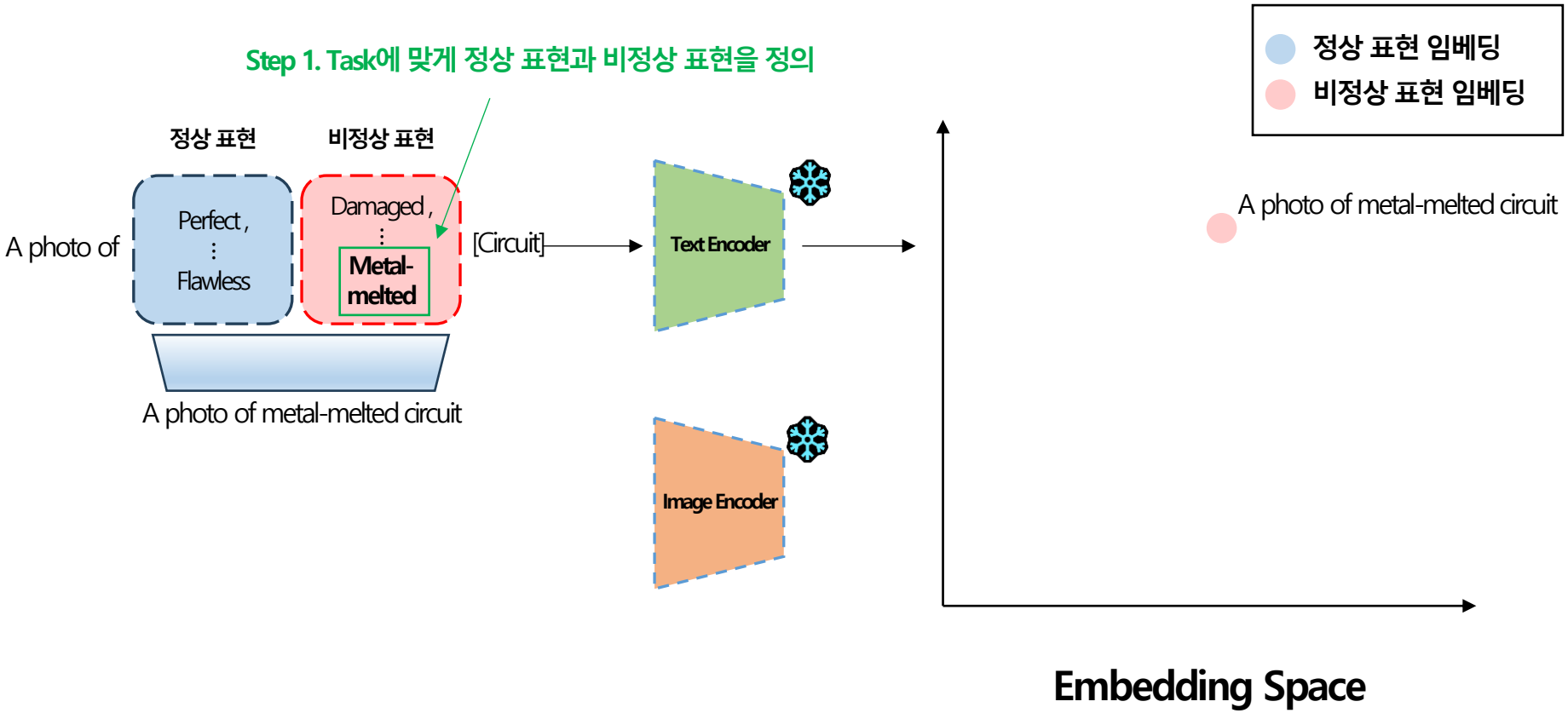
구멍 난 바지. 정상 제품인가, 불량품인가?

클래스를 단순히 [정상], [비정상]로 설정하는 것은 이러한 맥락을 무시

WinCLIP

❖ Proposed Method ①: Compositional Prompt Ensemble

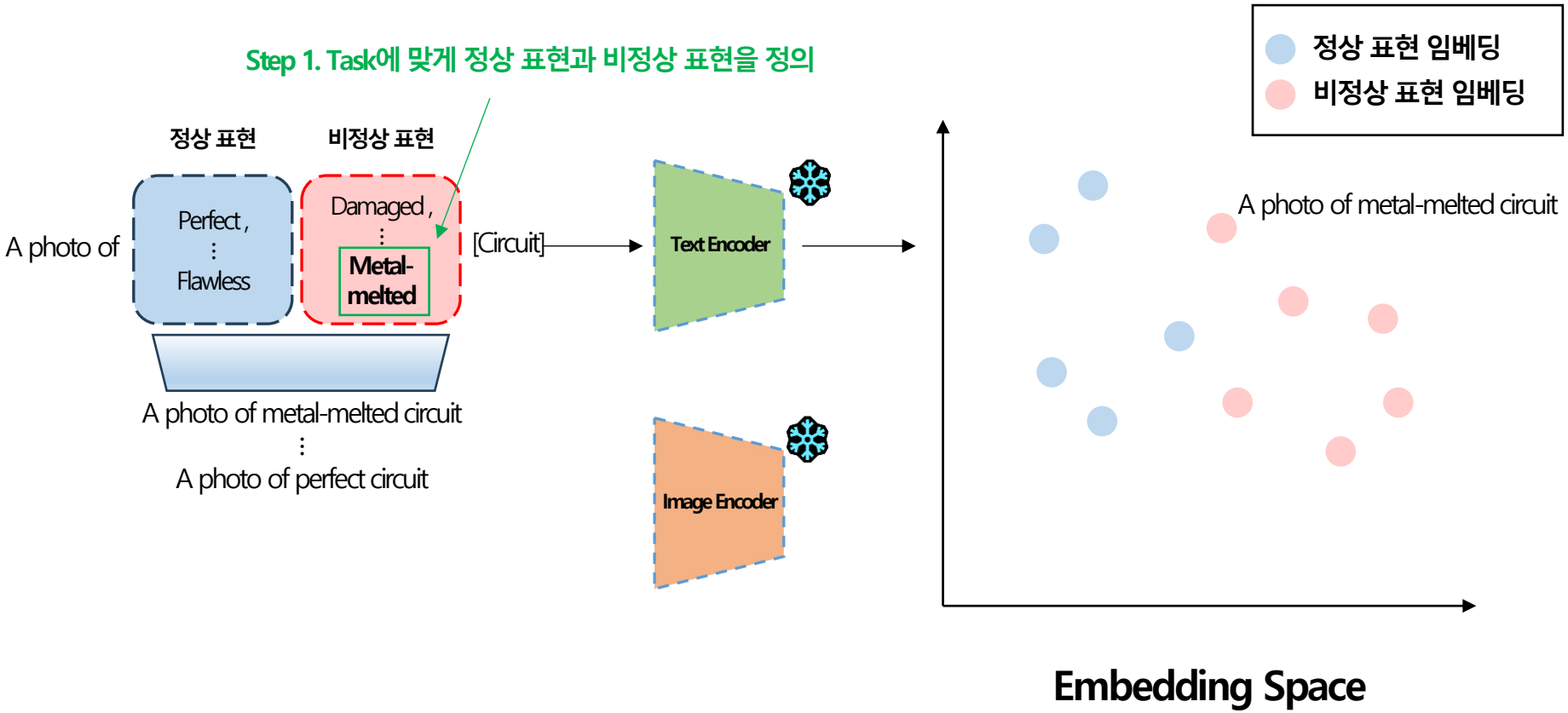
- 단순 [정상], [비정상] 클래스 확장 → 맥락 정보를 반영한 클래스 표현 획득
- 이에 기반한 이상치 탐지 수행



WinCLIP

❖ Proposed Method ①: Compositional Prompt Ensemble

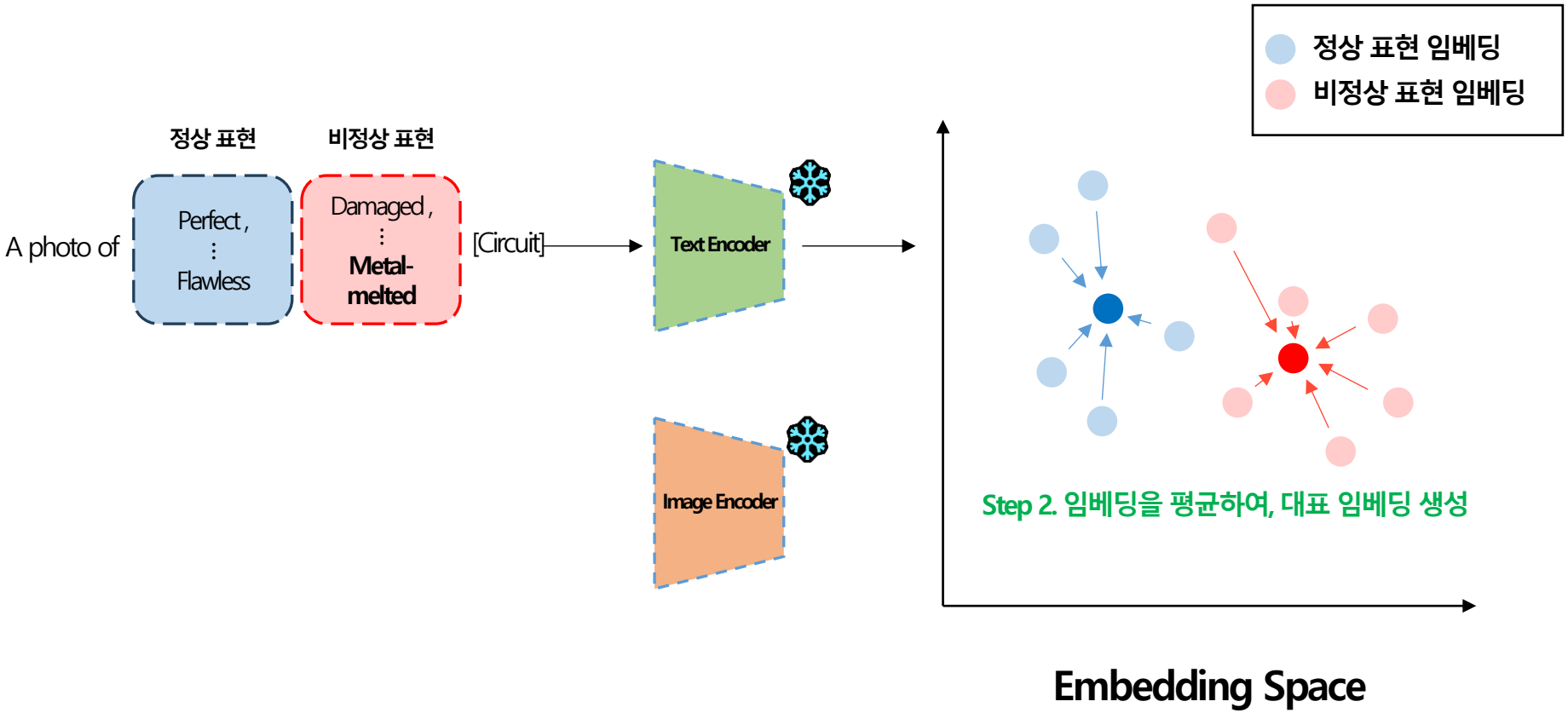
- 단순 [정상], [비정상] 클래스 확장 → 맥락 정보를 반영한 클래스 표현 획득
- 이에 기반한 이상치 탐지 수행



WinCLIP

❖ Proposed Method ①: Compositional Prompt Ensemble

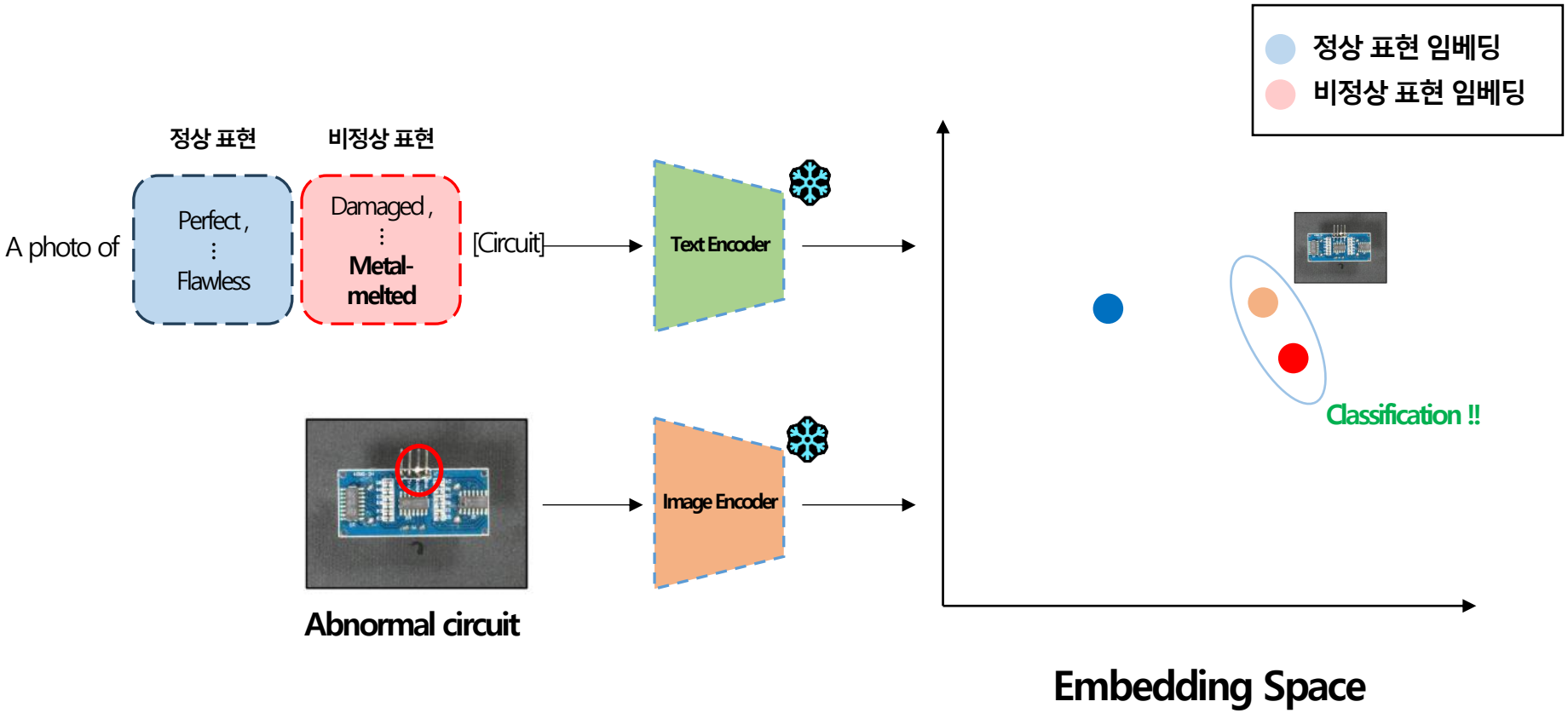
- 단순 [정상], [비정상] 클래스 확장 → 맥락 정보를 반영한 클래스 표현 획득
- 이에 기반한 이상치 탐지 수행



WinCLIP

❖ Proposed Method ①: Compositional Prompt Ensemble

- 단순 [정상], [비정상] 클래스 확장 → 맥락 정보를 반영한 클래스 표현 획득
- 이에 기반한 이상치 탐지 수행



WinCLIP

❖ **Zero-shot Anomaly Classification 성능 비교**

* a photo of a [normal] [object]
an image of a [normal] [object]
a close-up of a [normal] [object]

- CLIP-AC: 단순 [정상], [비정상] 클래스로 이진 분류 수행
- Prompt ensemble: 정상/비정상 표현은 고정 + 문장 표현*을 여럿 적용 후 ensemble
- **WinCLIP**: Compositional Prompt Ensemble 적용
- 성능 지표 (모두 100에 가까울수록 좋음):
  1. AUROC: 모델에 전체적인 분류 성능 평가
  2. AUPR: AUROC와 유사하나, 이상치 탐지 능력에 더 큰 비중을 둠
  3. F1-max: Threshold에 따른 F1-Score 중 가장 높은 값

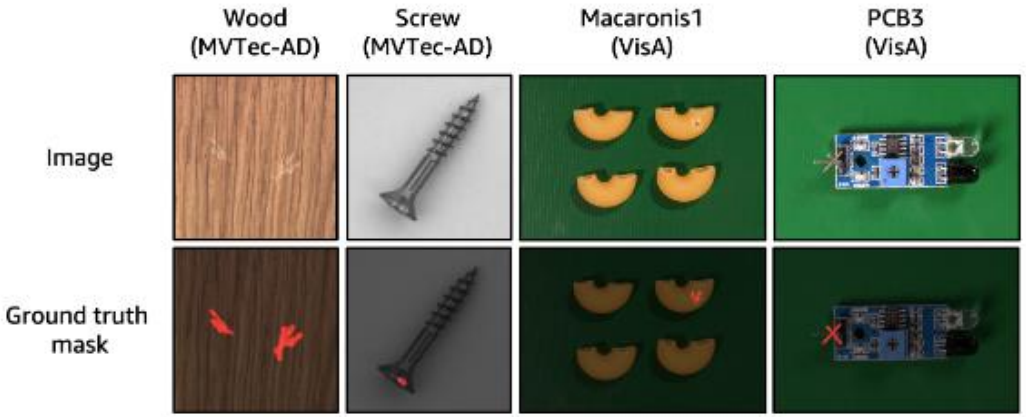
Anomaly Classification		MVTec-AD			VisA		
Setup	Method	AUROC	AUPR	F_1-max	AUROC	AUPR	F_1-max
0-shot	CLIP-AC [27]	74.0±0.0	89.1±0.0	88.5±0.0	59.3±0.0	67.0±0.0	74.4±0.0
	+ Prompt ens. [27]	74.1±0.0	89.5±0.0	87.8±0.0	58.2±0.0	66.4±0.0	74.0±0.0
	WinCLIP (ours)	**91.8**±0.0	**96.5**±0.0	**92.9**±0.0	**78.1**±0.0	**81.2**±0.0	**79.0**±0.0

WinCLIP

❖ Zero-Shot Anomaly Segmentation

- **Anomaly segmentation(AS):** 이상치에 해당하는 영역을 분할 (Pixel-wise classification)
- 구체적이고 지역적인 정보 요구 (Local information)
- CLIP은 전역적 정보에 집중 → Segmentation에 부적합

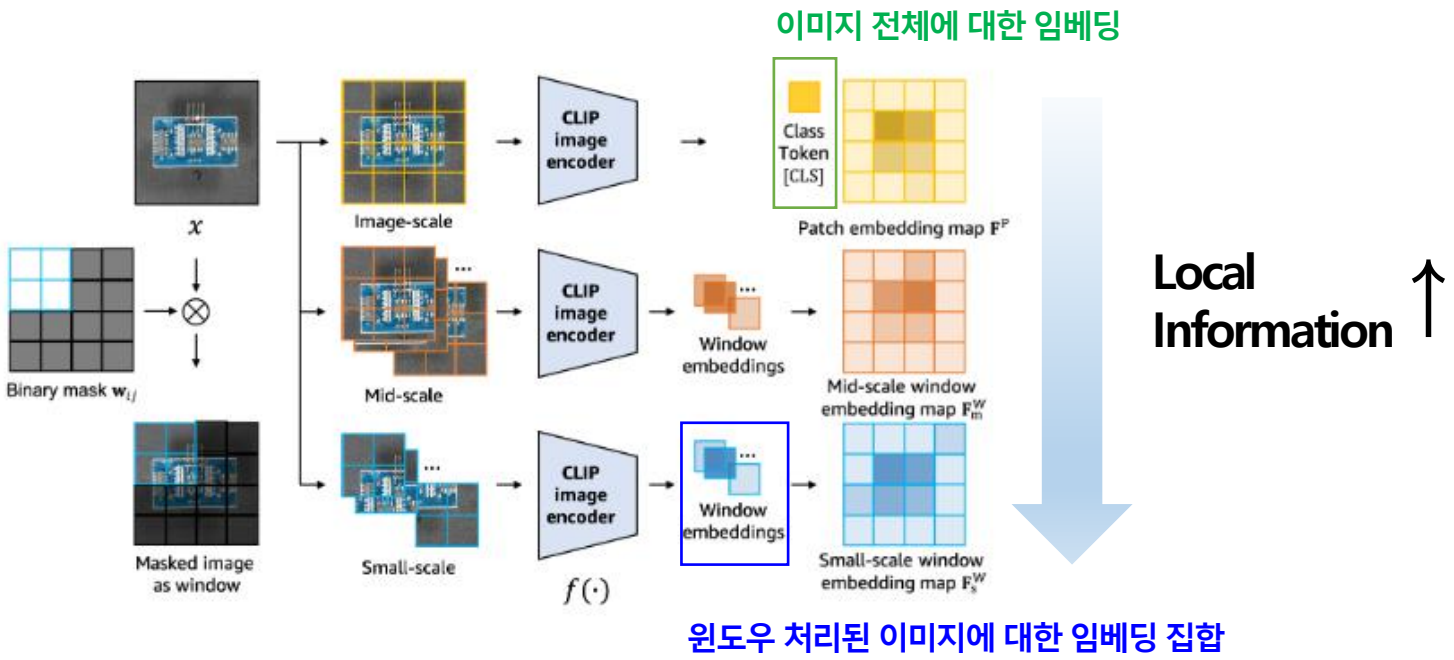
Anomaly Segmentation



WinCLIP

❖ **Window-based CLIP(WinCLIP) for zero-shot anomaly segmentation**

- Window 기반 이미지 cropping 적용 후, 이미지 임베딩 수행
- Crop 범위 내에서만 특징 추출이 가능하므로, local information 취득 가능
- Window size가 작을수록 더 조밀한 특징 추출 가능



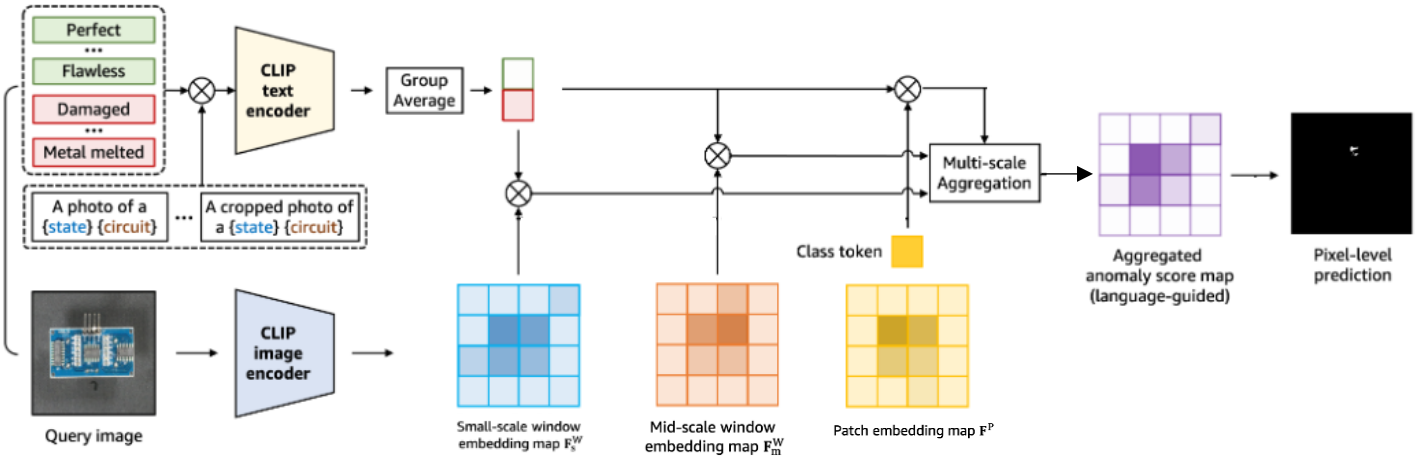
Zero-shot AD

❖ **Window-based CLIP(WinCLIP) for zero-shot anomaly segmentation**

① 윈도우 임베딩 별 유사도 점수 계산

② 윈도우 별 유사도 점수 종합 → Pixel level 유사도 점수 계산

③ ①, ②번을 각 스케일에서 수행 후 종합 → 최종 pixel level 유사도 점수 계산



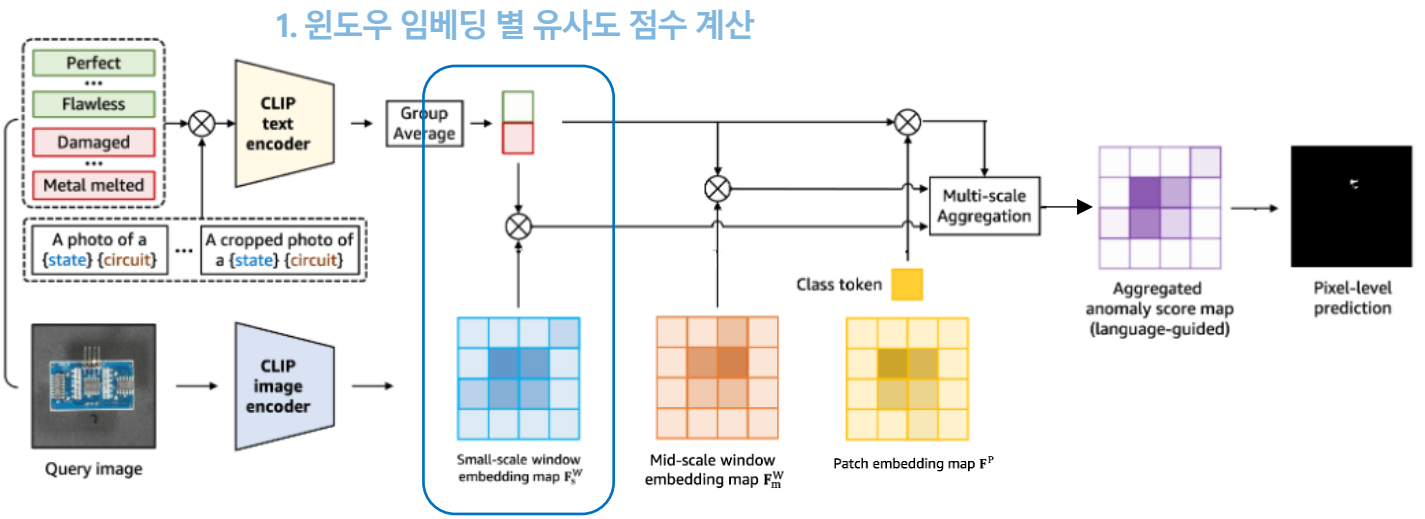
Zero-shot AD

❖ **Window-based CLIP(WinCLIP) for zero-shot anomaly segmentation**

① 윈도우 임베딩 별 유사도 점수 계산

② 윈도우 별 유사도 점수 종합 → Pixel level 유사도 점수 계산

③ ①, ②번을 각 스케일에서 수행 후 종합 → 최종 pixel level 유사도 점수 계산



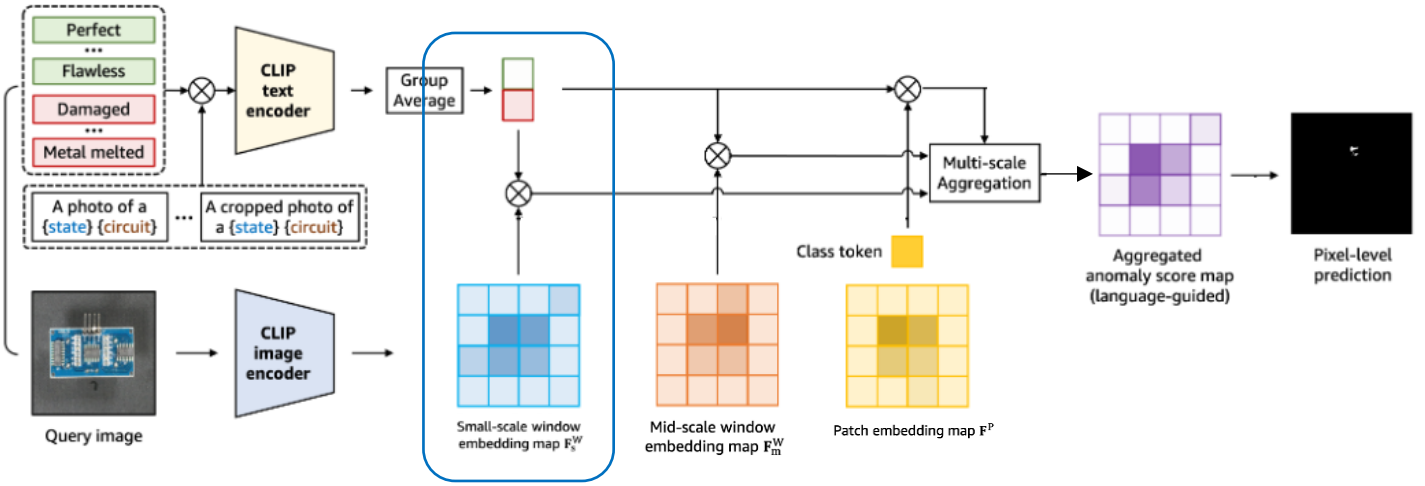
Zero-shot AD

❖ **Window-based CLIP(WinCLIP) for zero-shot anomaly segmentation**

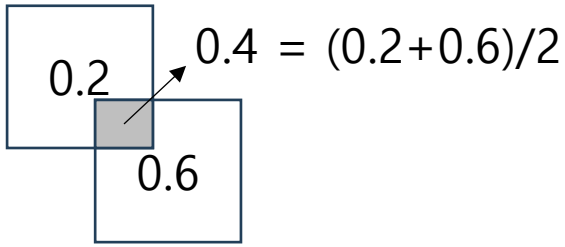
① 윈도우 임베딩 별 유사도 점수 계산

② 윈도우 별 유사도 점수 종합 → Pixel level 유사도 점수 계산

③ ①, ②번을 각 스케일에서 수행 후 종합 → 최종 pixel level 유사도 점수 계산



2. 윈도우 별 점수를, 픽셀 별로 세세하게 종합



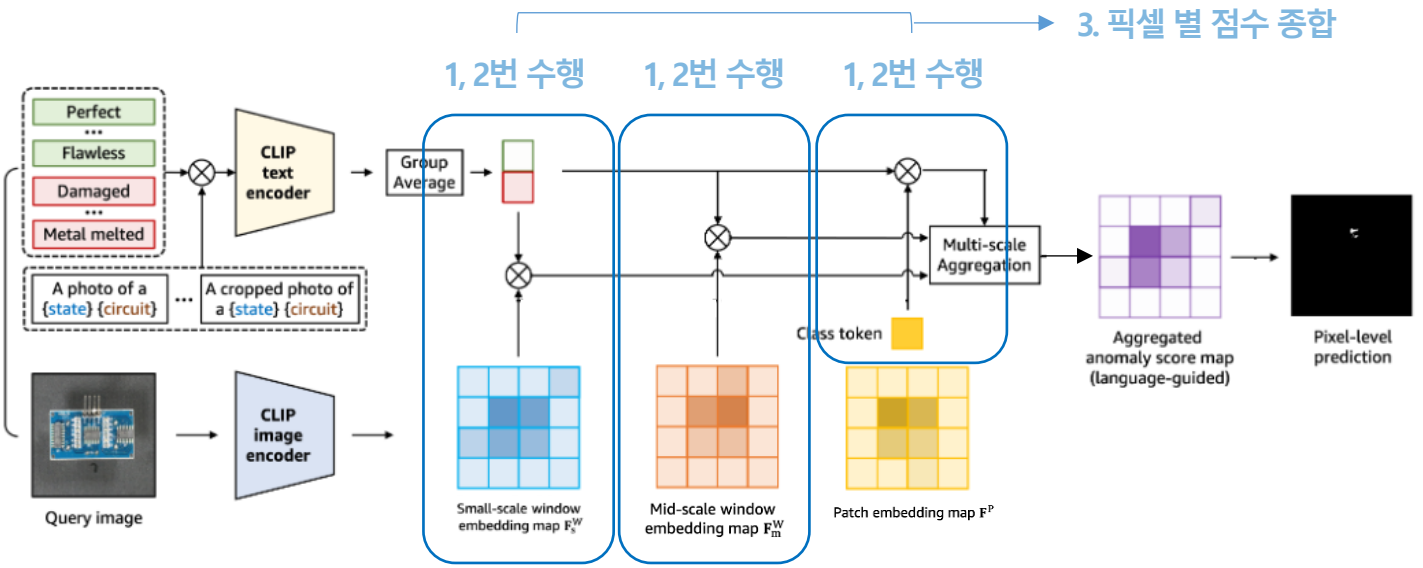
Zero-shot AD

❖ **Window-based CLIP(WinCLIP) for zero-shot anomaly segmentation**

① 윈도우 별 유사도 점수 계산

② 윈도우 별 유사도 점수 종합 → Pixel level 유사도 점수 계산

③ ①, ②번을 각 스케일에서 수행 후 종합 → 최종 pixel level 유사도 점수 계산



Zero-shot AD

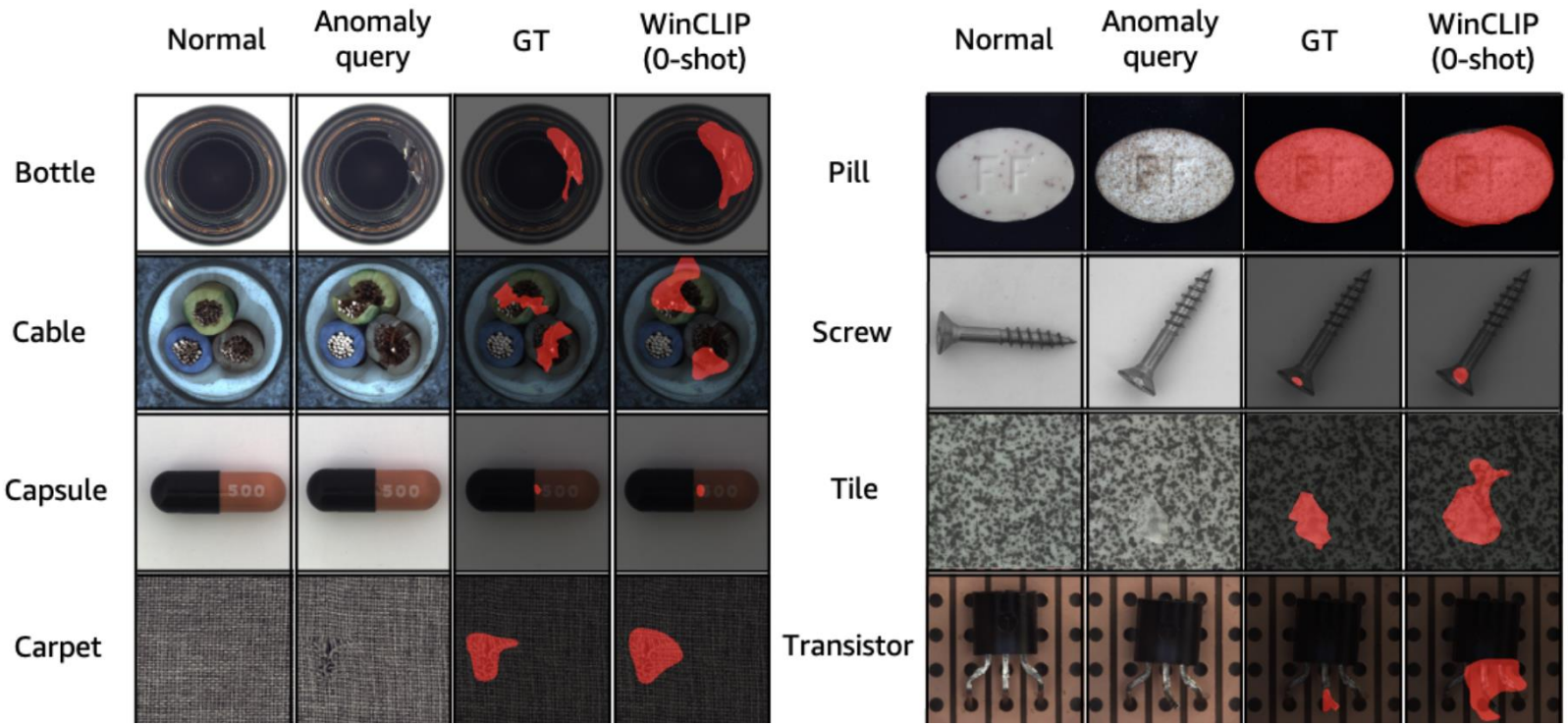
❖ Zero-shot AS 성능 비교

- Trans-MM: & MaskCLIP: Patch-wise feature에 의존 (Local information ↓)
- **WinCLIP**: Compositional Prompt Ensemble + Window Mechanism 적용
- 성능 지표 (모두 1에 가까울수록 좋음):
  1. pAUROC: pixel level AUROC (이상 객체가 작을 경우, 과대 평가 위험)
  2. PRO: pAUROC에 비해, 작은 이상 객체에도 강건한 성능 지표
  3. F1-max: Threshold에 따른 F1-Score 중 가장 높은 값

Anomaly Segmentation		MVTec-AD			VisA		
Setup	Method	pAUROC	PRO	F_1-max	pAUROC	PRO	F_1-max
0-shot	Trans-MM [5]	57.5±0.0	21.9±0.0	12.1±0.0	49.4±0.0	10.2±0.0	3.1±0.0
	MaskCLIP [57]	63.7±0.0	40.5±0.0	18.5±0.0	60.9±0.0	27.3±0.0	7.3±0.0
	WinCLIP (ours)	**85.1±0.0**	**64.6±0.0**	**31.7±0.0**	**79.6±0.0**	**56.8±0.0**	**14.8±0.0**

Zero-shot AD

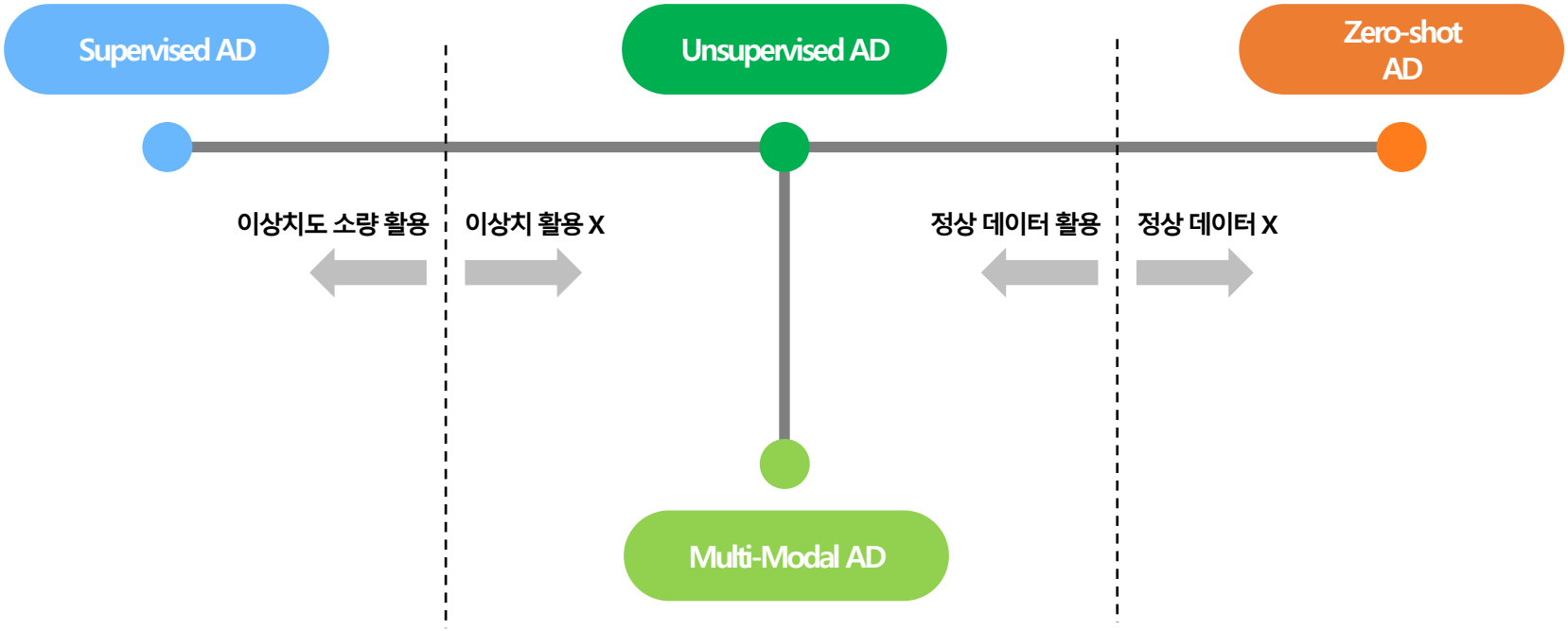
❖ **Zero-shot AS Segmentation Mask**



Multi-Modal AD

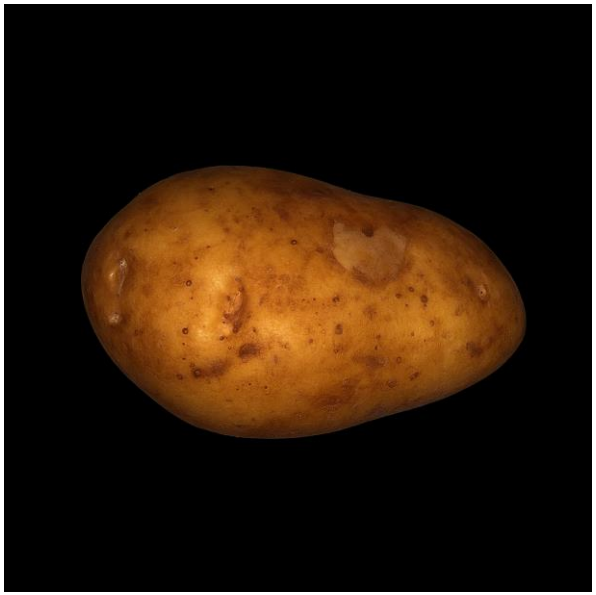
❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

- Topic 1: 무엇보다 AD 정확도가 중요 → Supervised AD
- Topic 2: 정상 데이터도 없음 → Zero-shot AD
- **Topic 3 : 이미지 정보만으로는 탐지할 수 없는 이상치 → Multi-Modal AD**



Multi-Modal AD

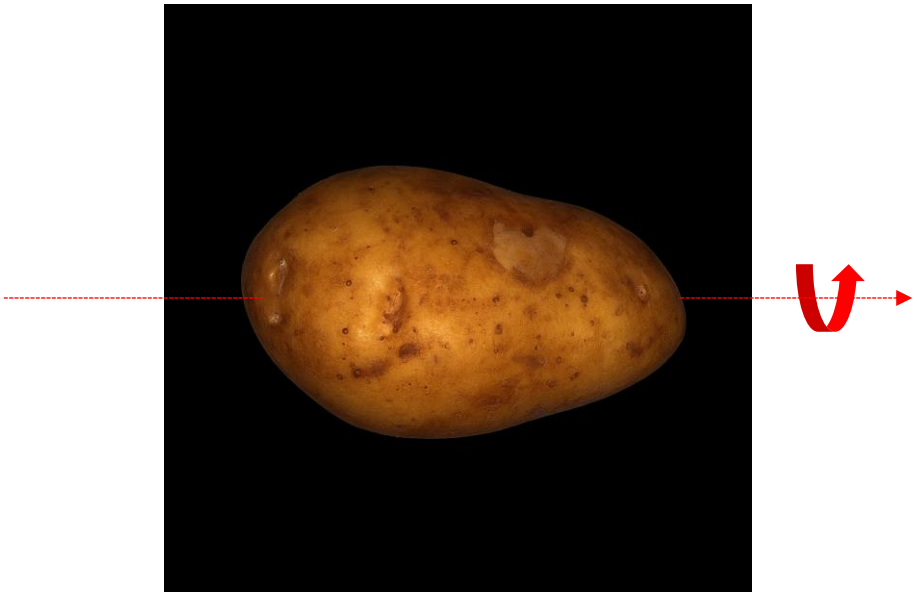
Quiz: 아래 감자는 정상 감자일까요?



Top view

Multi-Modal AD

Quiz: 아래 감자는 정상 감자일까요?



Top view

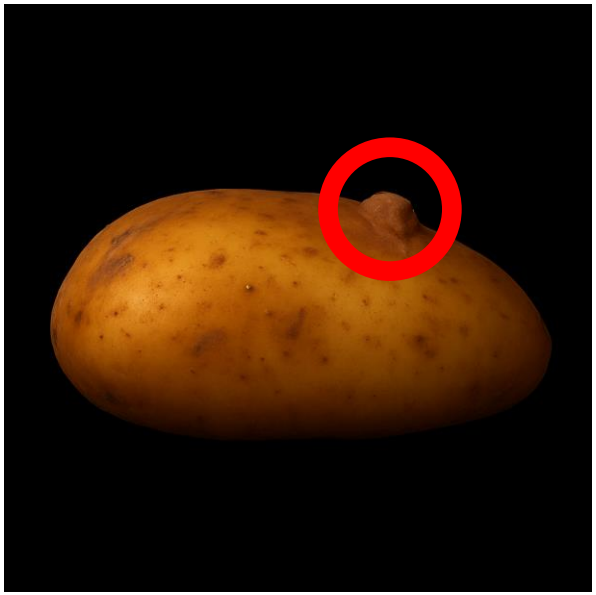
Multi-Modal AD

Quiz: 아래 감자는 정상 감자일까요?

Frontal view

Multi-Modal AD

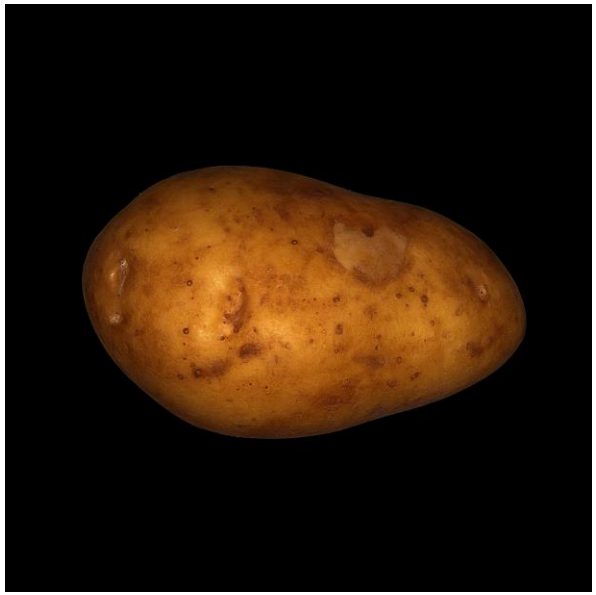
Quiz: 아래 감자는 정상 감자일까요?



Frontal view

Multi-Modal AD

2D 이미지 한계: 구조적 정보 손실



Top view

Frontal view

Multi-Modal AD

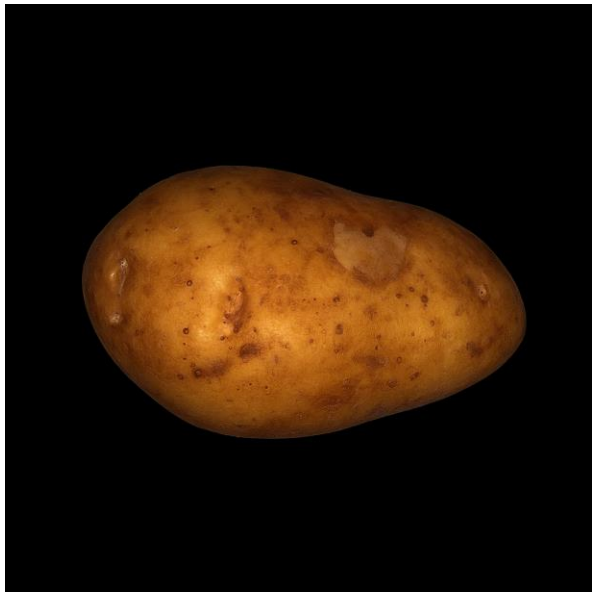
2D 이미지 한계: 구조적 정보 손실

Top view

Frontal view

Multi-Modal AD

2D 이미지 한계: 구조적 정보 손실



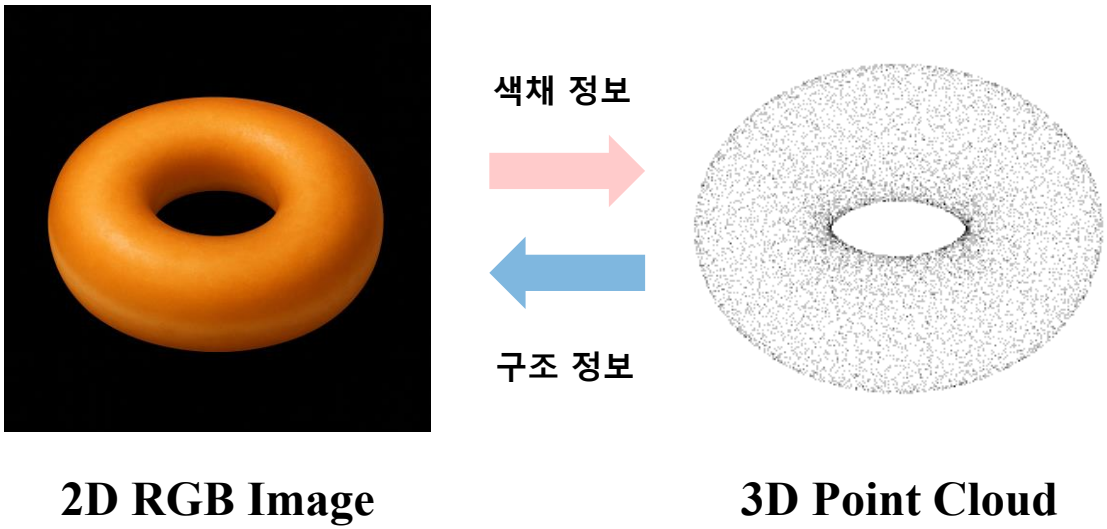
Top view

Frontal view

Multi-Modal AD

❖ **Multimodality in Anomaly Detection(AD)**

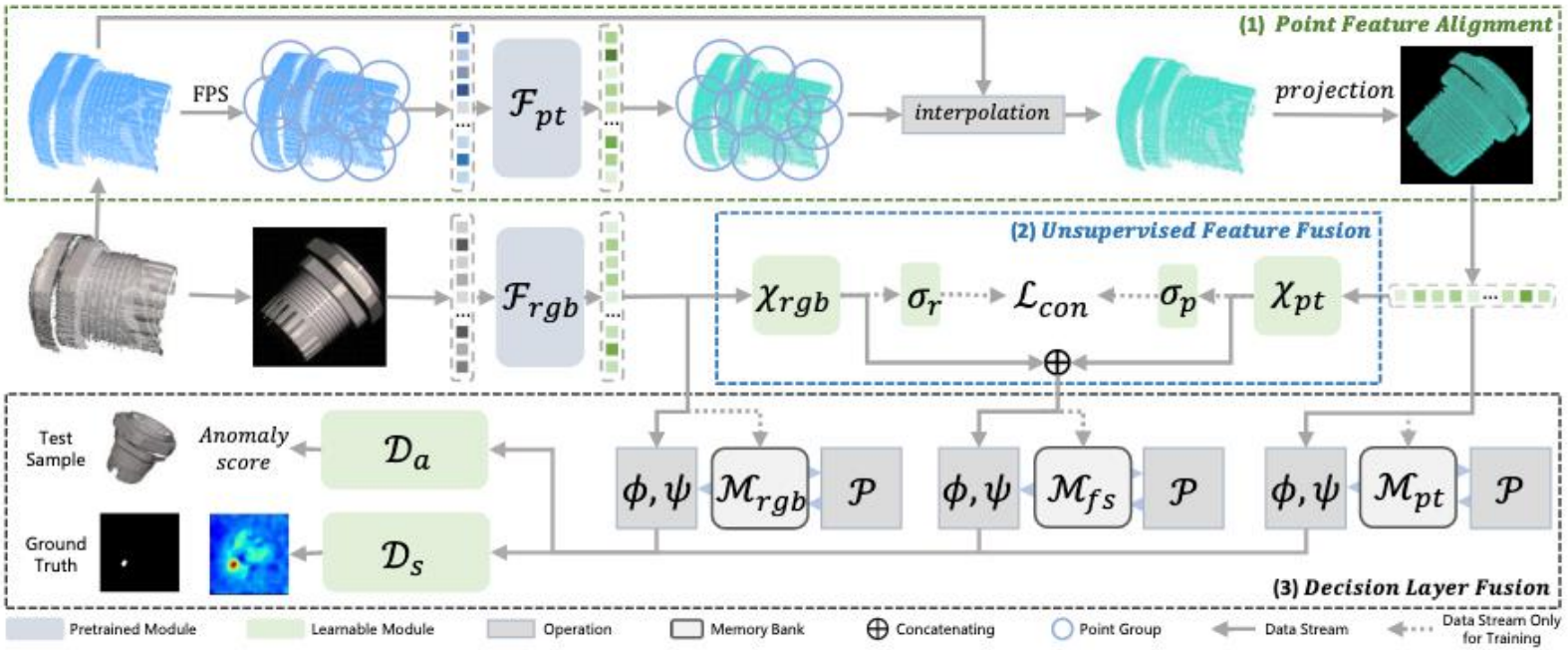
- 2D RGB Image: 색깔이 다른 이상치 탐지에 용이
- 3D Point Cloud: 구조가 다른 이상치 탐지에 용이
- **2D 색채 정보와 3D 형태 정보를 함께 활용하는 것이 좋다!!**



Multi-Modal AD

❖ **Multimodal Industrial Anomaly Detection via Hybrid Fusion (CVPR'23)**

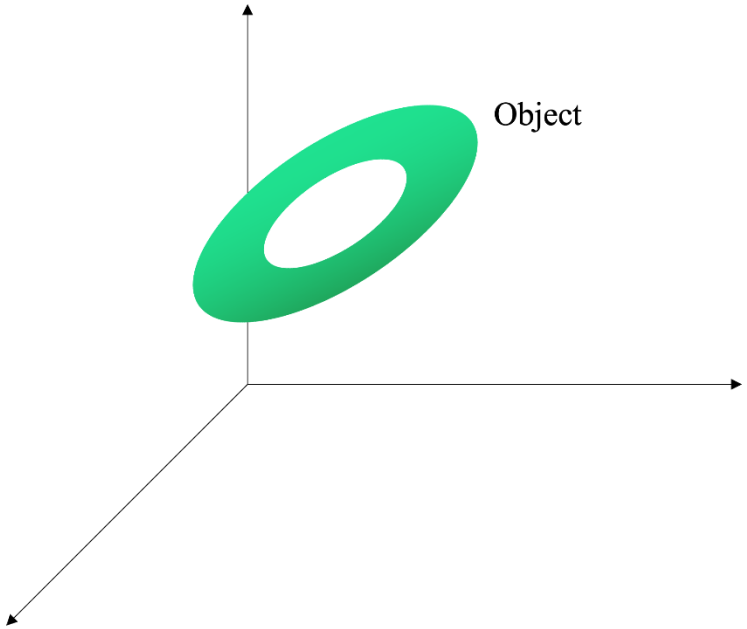
- Unsupervised Anomaly Detection 방법론: 훈련 시 정상 데이터만 활용
- **핵심**: 2D 특징과 3D 특징을 ①융합 & ②이상치 탐지에 활용



① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

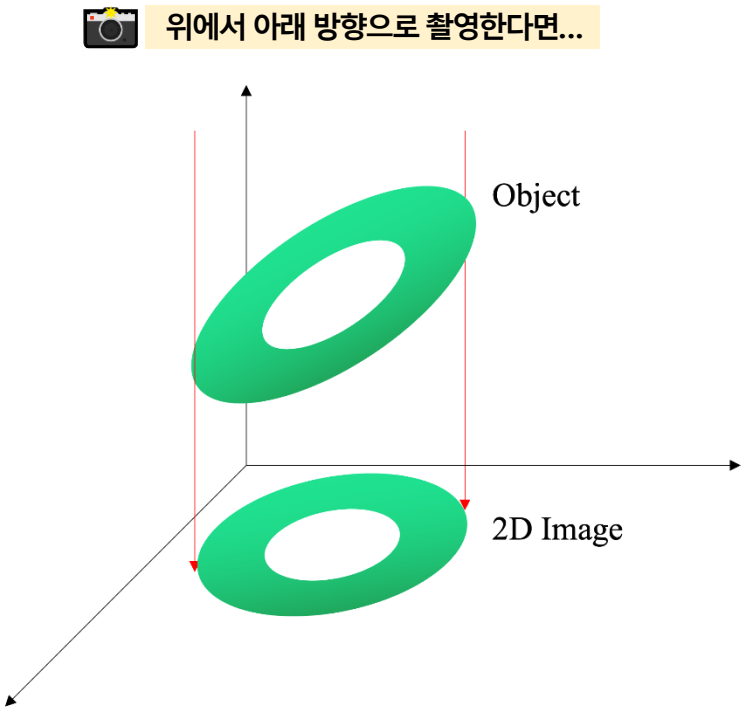
- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ 3D Point Cloud: Point Transformer



① 2D & 3D 특징 융합

❖ Step 1. 개별 특징 추출

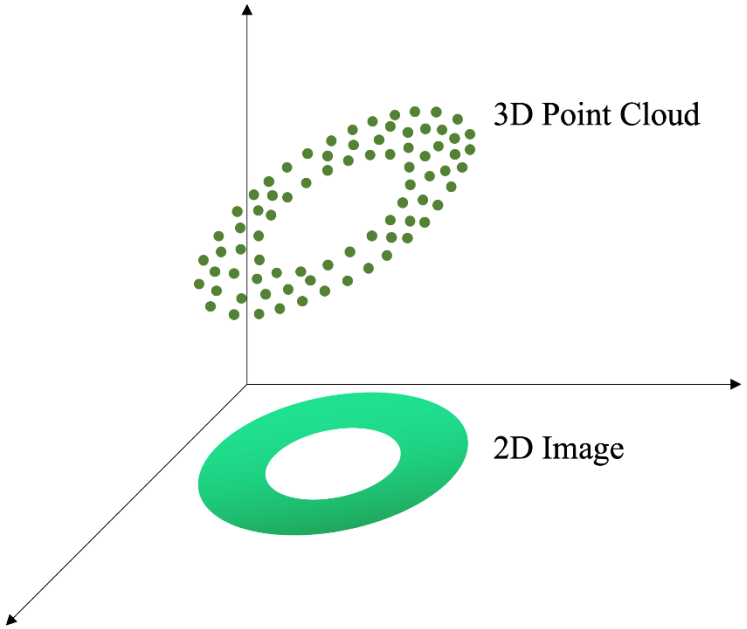
- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ 3D Point Cloud: Point Transformer



① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ 3D Point Cloud: Point Transformer



① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

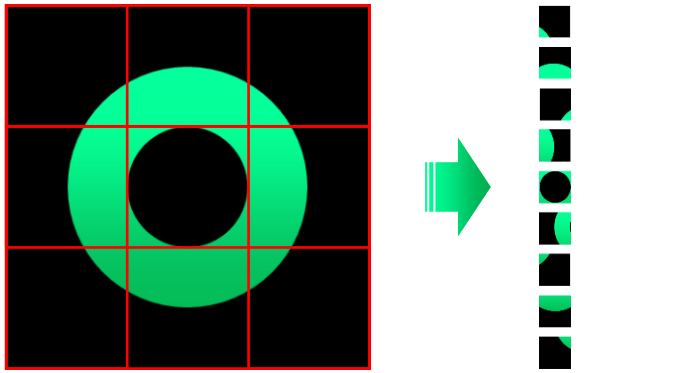
- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ **2D RGB Image: Vision Transformer**
  - ➢ 3D Point Cloud: Point Transformer

2D Image

① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ **2D RGB Image: Vision Transformer**
  - ➢ 3D Point Cloud: Point Transformer

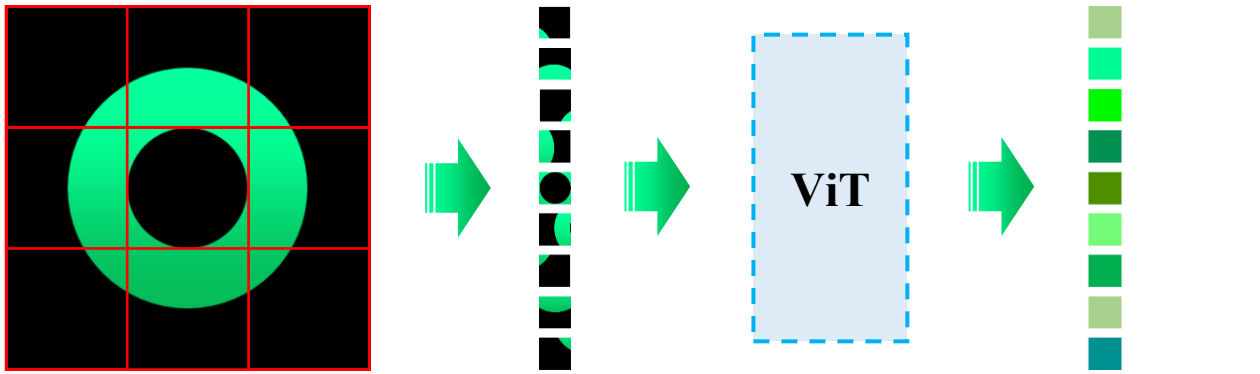


2D Image Tokenization

① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

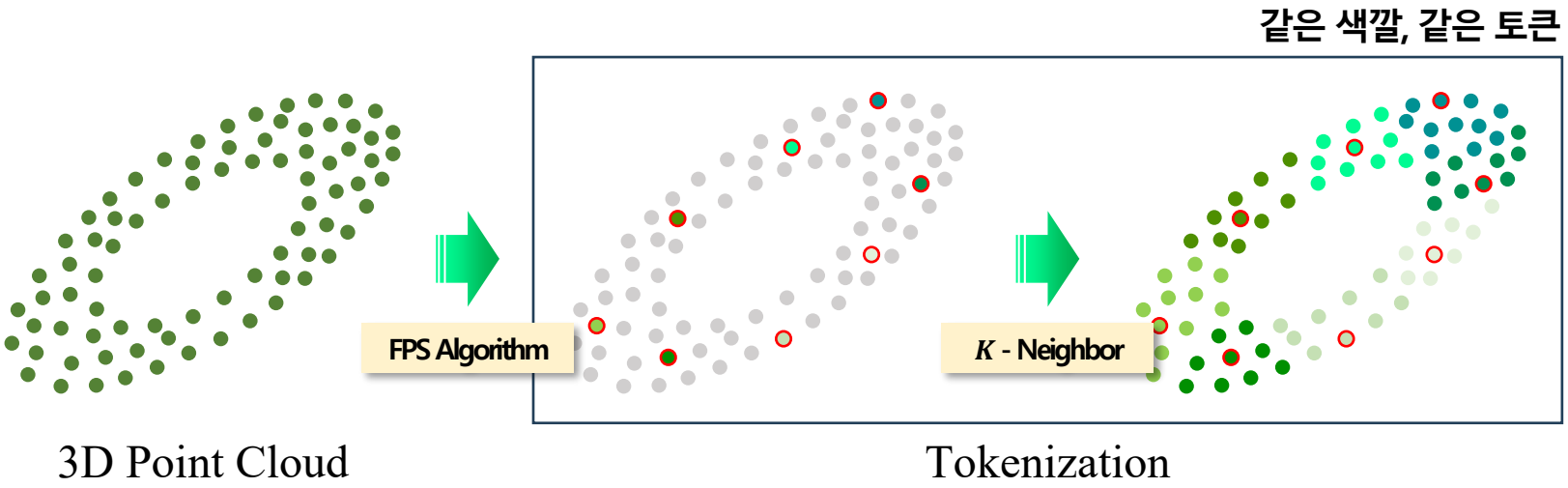
- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ **2D RGB Image: Vision Transformer**
  - ➢ 3D Point Cloud: Point Transformer



① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

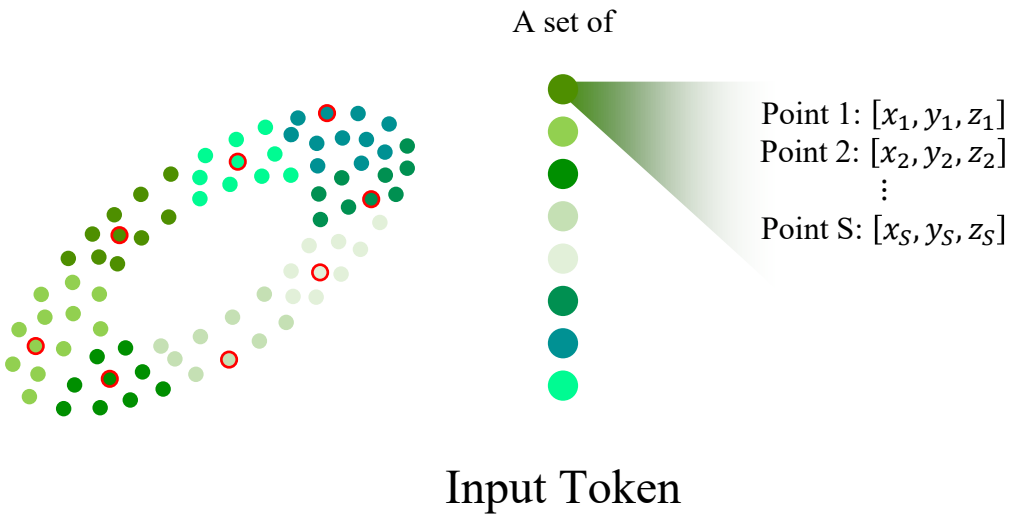
- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ **3D Point Cloud: Point Transformer**



① 2D & 3D 특징 융합

❖ Step 1. 개별 특징 추출

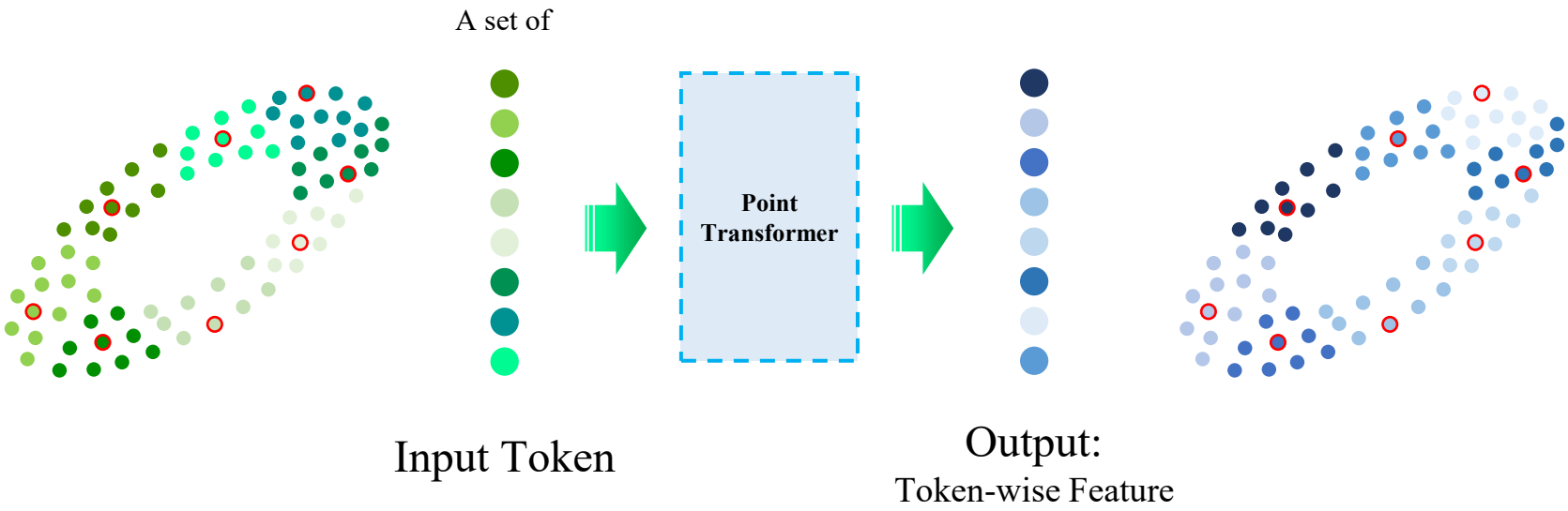
- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ **3D Point Cloud: Point Transformer**



① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

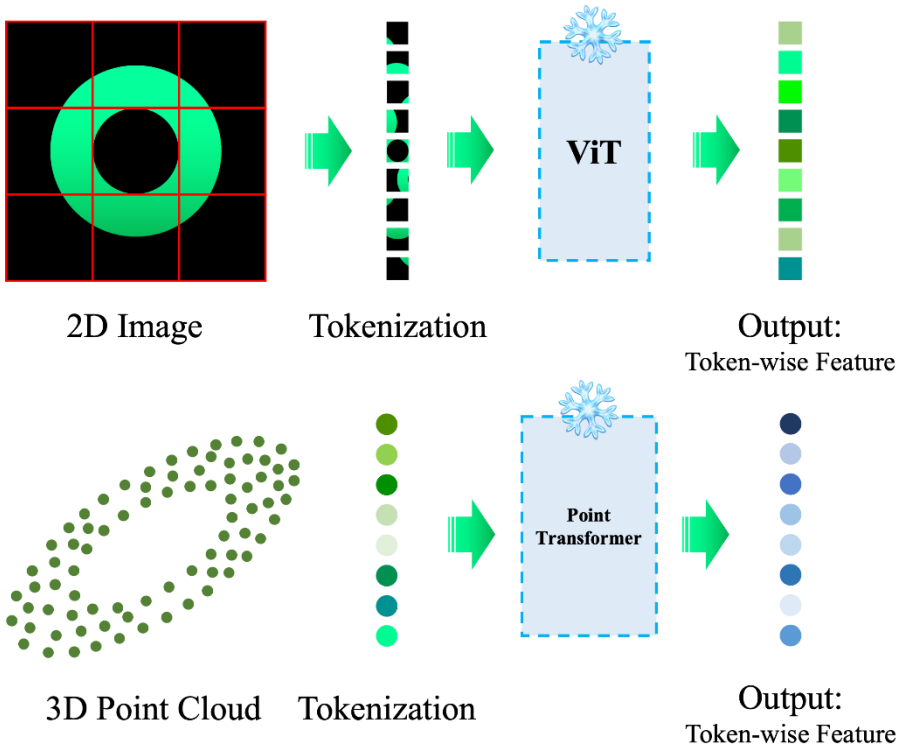
- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ **3D Point Cloud: Point Transformer**



① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ 3D Point Cloud: Point Transformer

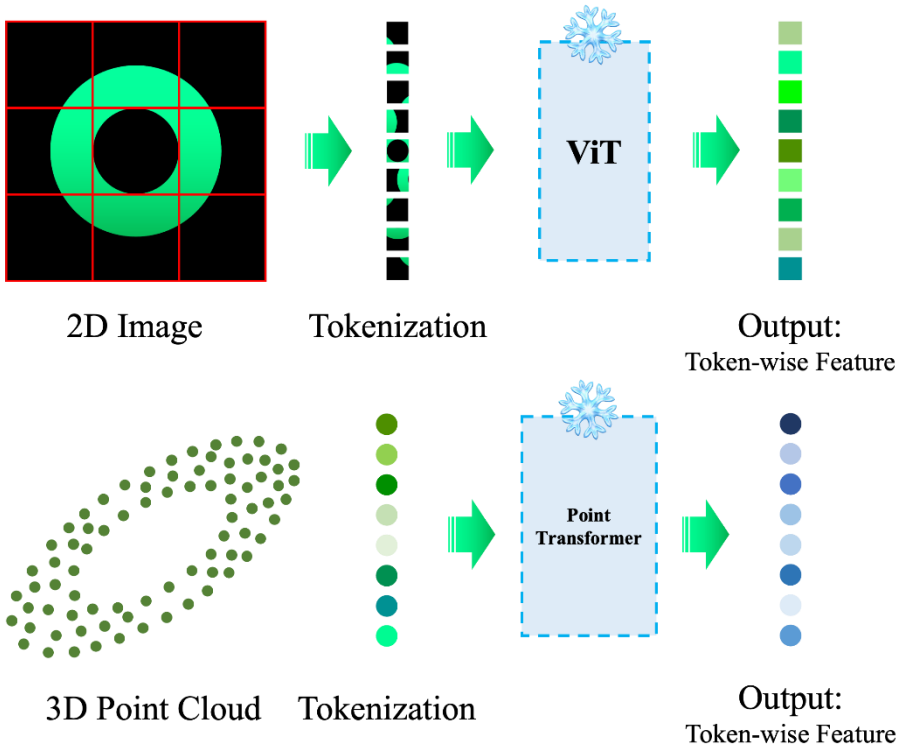


바로 합칠 수 있을까?

① 2D & 3D 특징 융합

❖ **Step 1. 개별 특징 추출**

- 2D & 3D 모두 사전 학습된 transformer 구조 사용
  - ➢ 2D RGB Image: Vision Transformer
  - ➢ 3D Point Cloud: Point Transformer

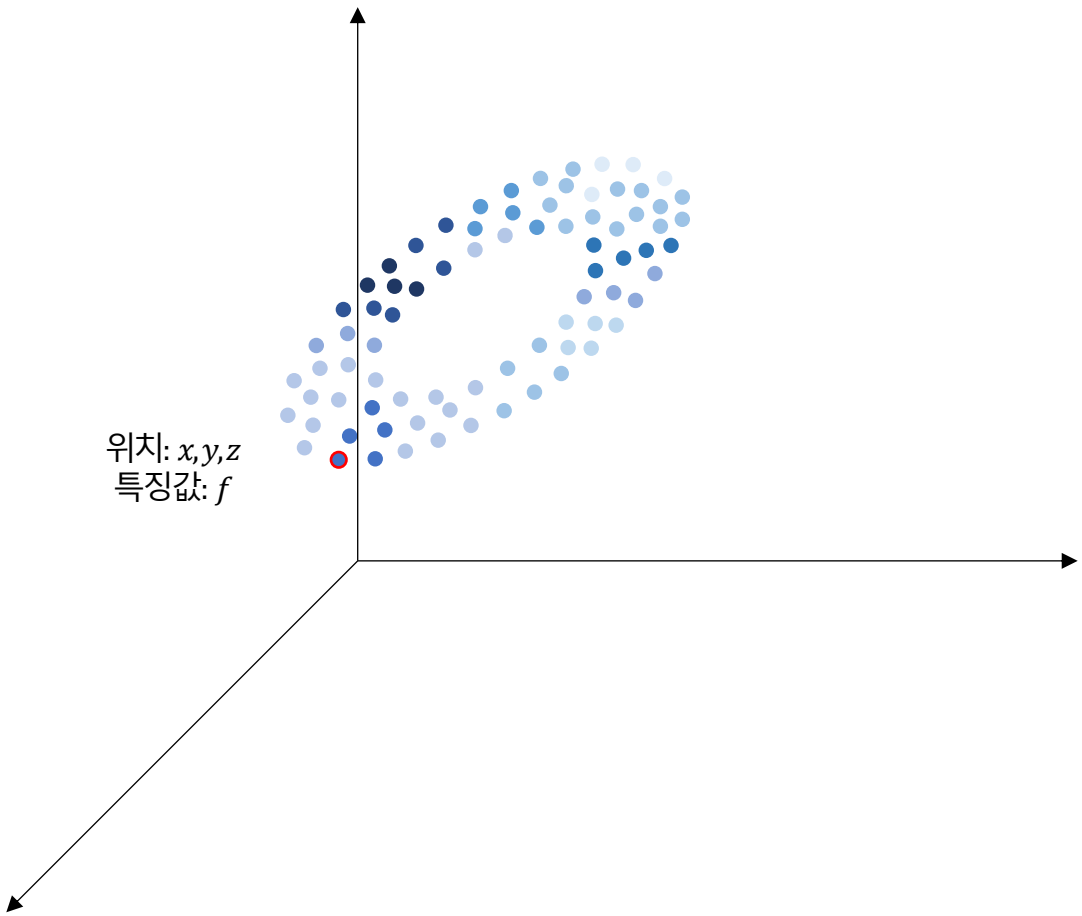


2D & 3D Mismatch !!

① 2D & 3D 특징 융합

❖ **Step 2. Point Feature Alignment**

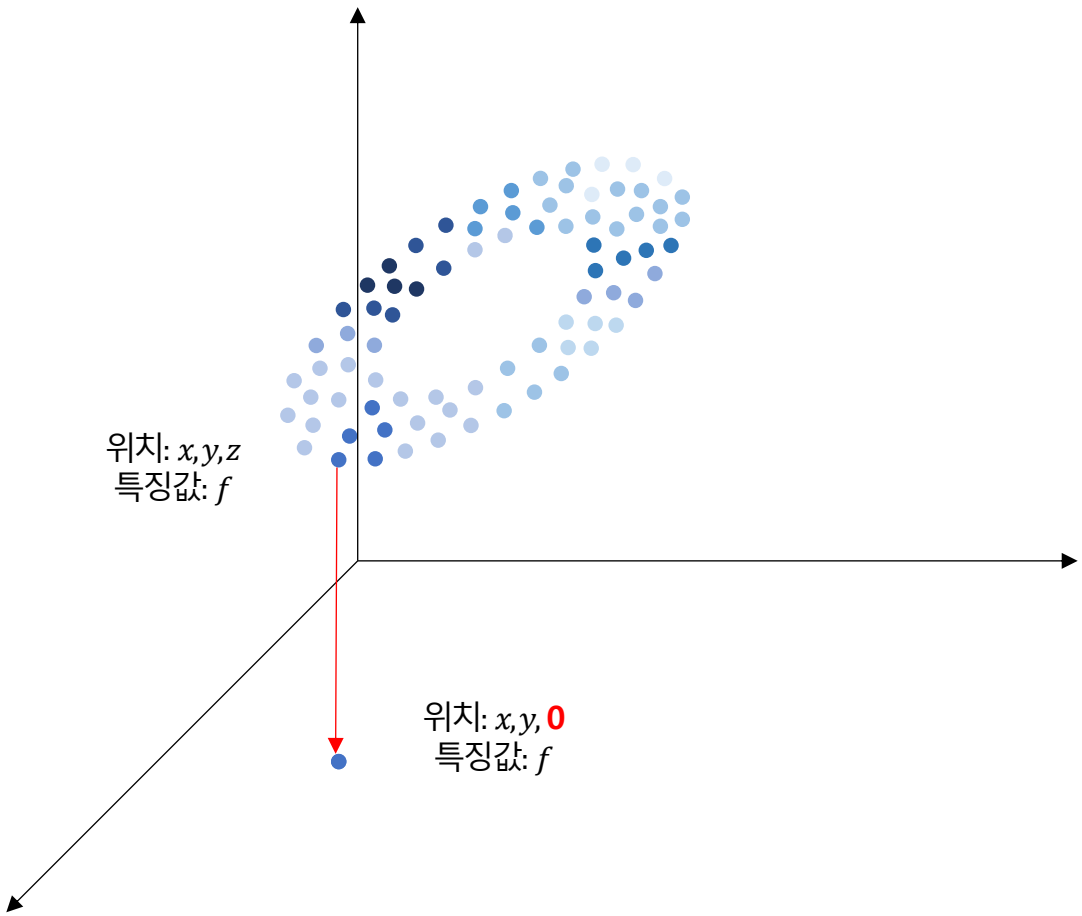
- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!



① 2D & 3D 특징 융합

❖ **Step 2. Point Feature Alignment**

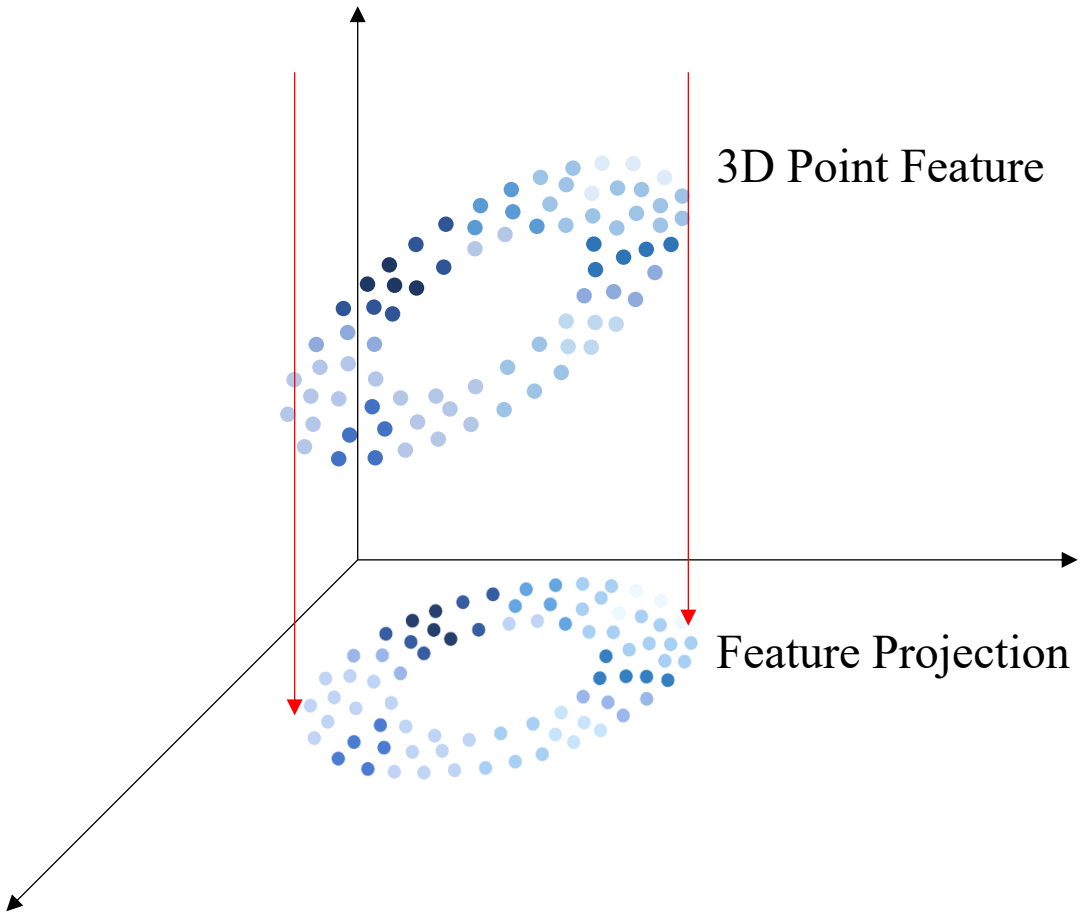
- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!



① 2D & 3D 특징 융합

❖ **Step 2. Point Feature Alignment**

- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!



① 2D & 3D 특징 융합

❖ **Step 2. Point Feature Alignment**

- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!

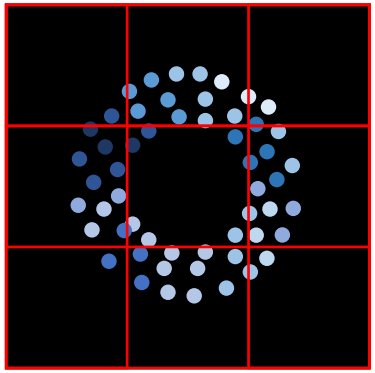
3D 특징을 2D Image에 정렬

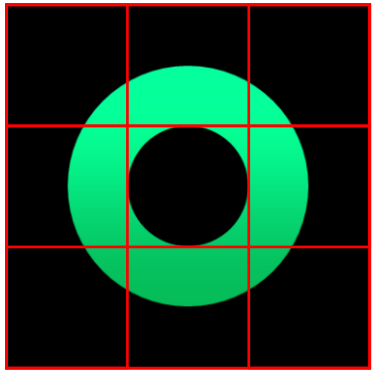
① 2D & 3D 특징 융합

❖ **Step 2. Point Feature Alignment**

- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!

3D 특징을 2D Image에 정렬

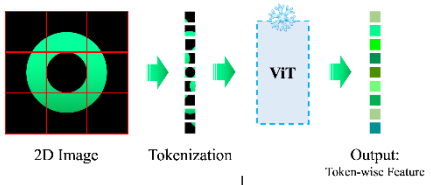




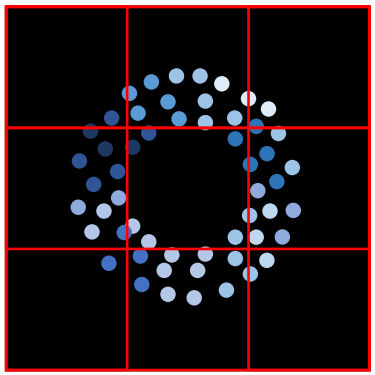
① 2D & 3D 특징 융합

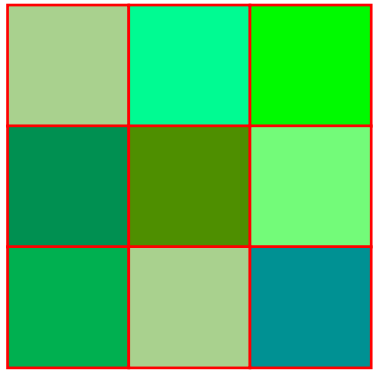
❖ **Step 2. Point Feature Alignment**

- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!



정렬한 특징을 패치 별 특징으로





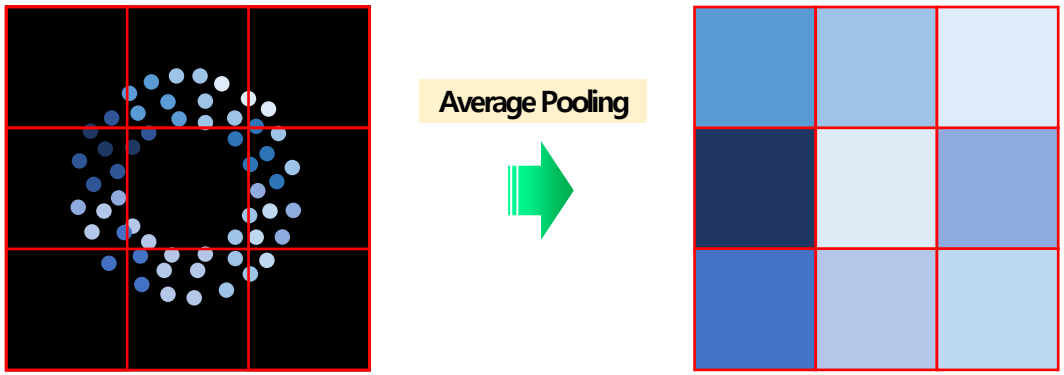
2D Feature

① 2D & 3D 특징 융합

❖ **Step 2. Point Feature Alignment**

- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!

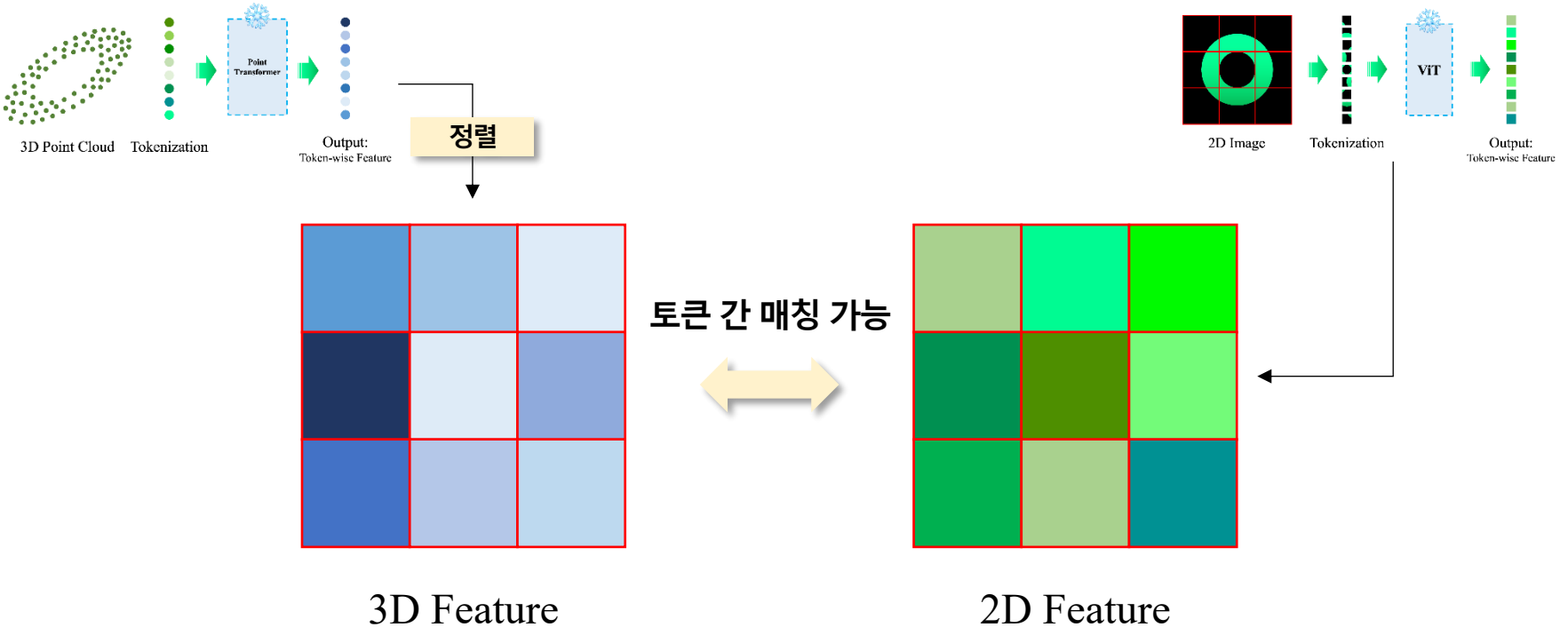
정렬한 특징을 패치 별 특징으로



① 2D & 3D 특징 융합

❖ Step 2. Point Feature Alignment

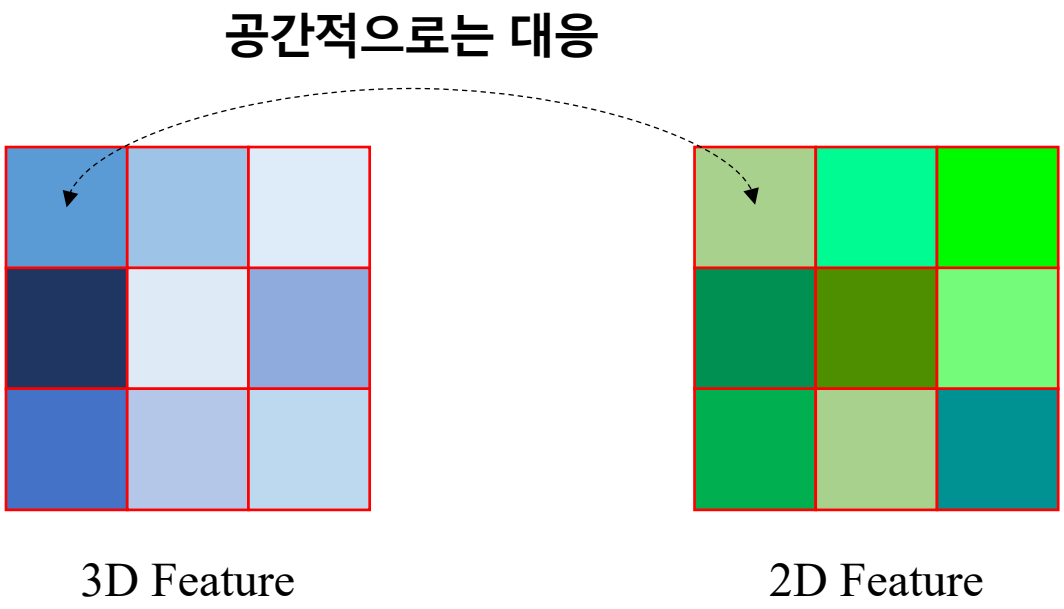
- 두 feature를 융합할 수 있도록, 3D Point feature를 2D로 정렬하자!!



① 2D & 3D 특징 융합

❖ **Step 3. Unsupervised Feature Fusion**

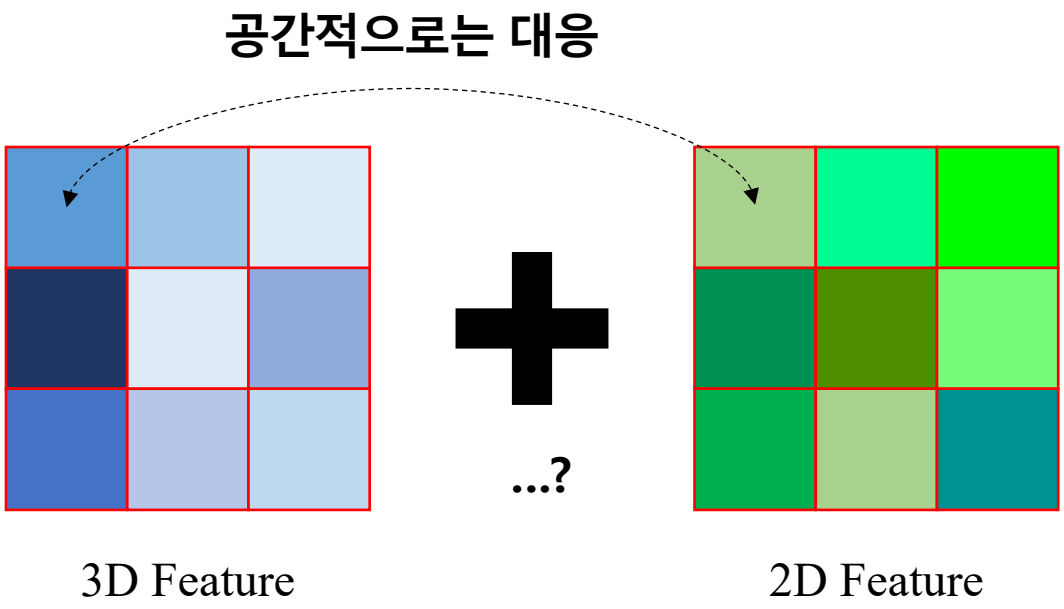
- 3D Feature와 2D Feature는 독립적으로 추출 → 상호작용 고려 X
- 따라서, 패치 별 단순 덧셈은 X



① 2D & 3D 특징 융합

❖ **Step 3. Unsupervised Feature Fusion**

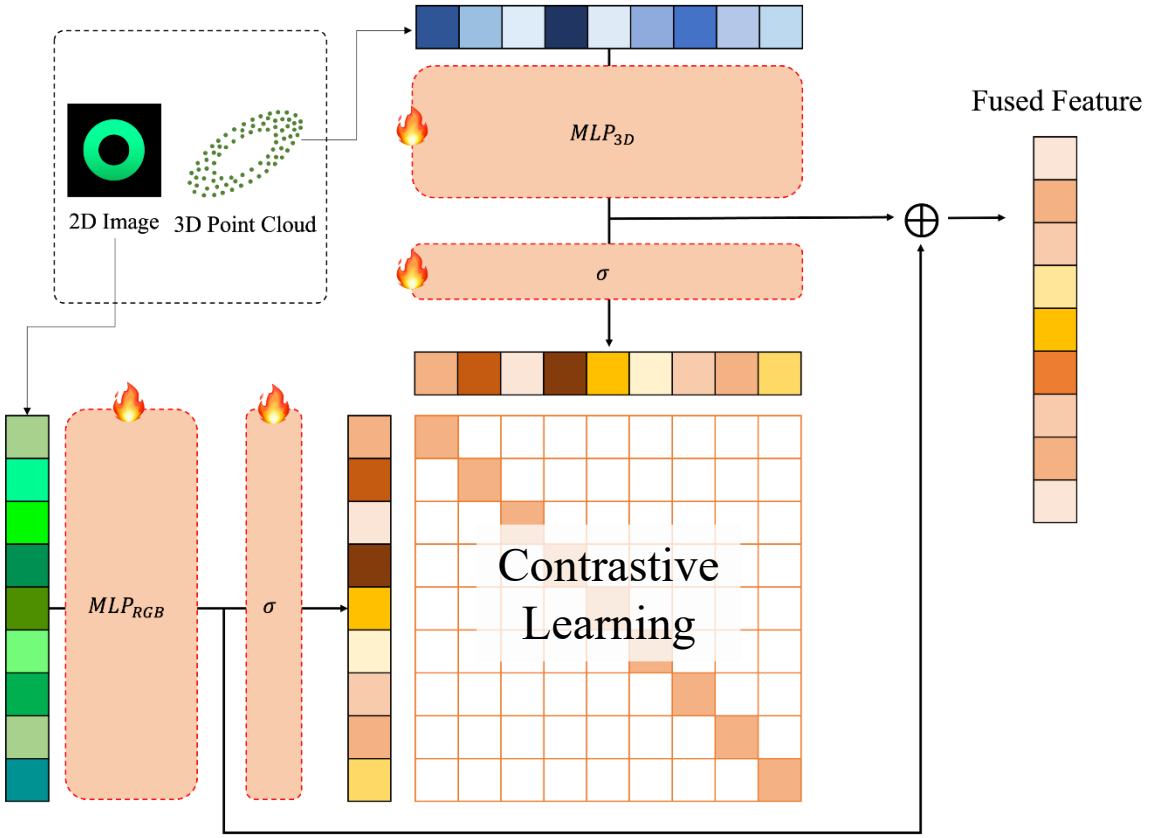
- 3D Feature와 2D Feature는 독립적으로 추출 → 상호작용 고려 X
- 따라서, 패치 별 단순 덧셈은 X



① 2D & 3D 특징 융합

❖ **Step 3. Unsupervised Feature Fusion**

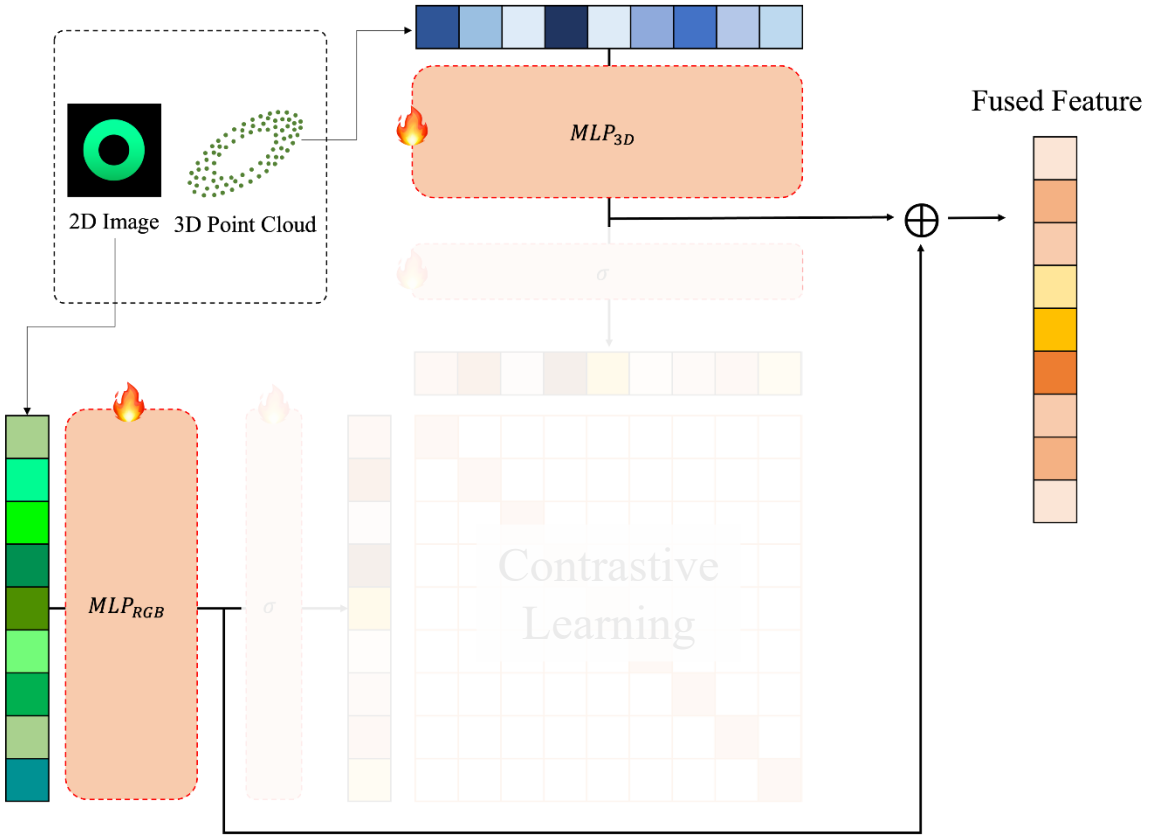
- 각 특징에 대해 MLP와 FC Layer (σ) 적용
- 동일한 패치에 해당하는 3D 특징과 2D 특징이 유사해지도록 학습 → Contrastive Learning 활용
- 융합 시, 최종 레이어 이전 특징을 사용



① 2D & 3D 특징 융합

❖ **Step 3. Unsupervised Feature Fusion**

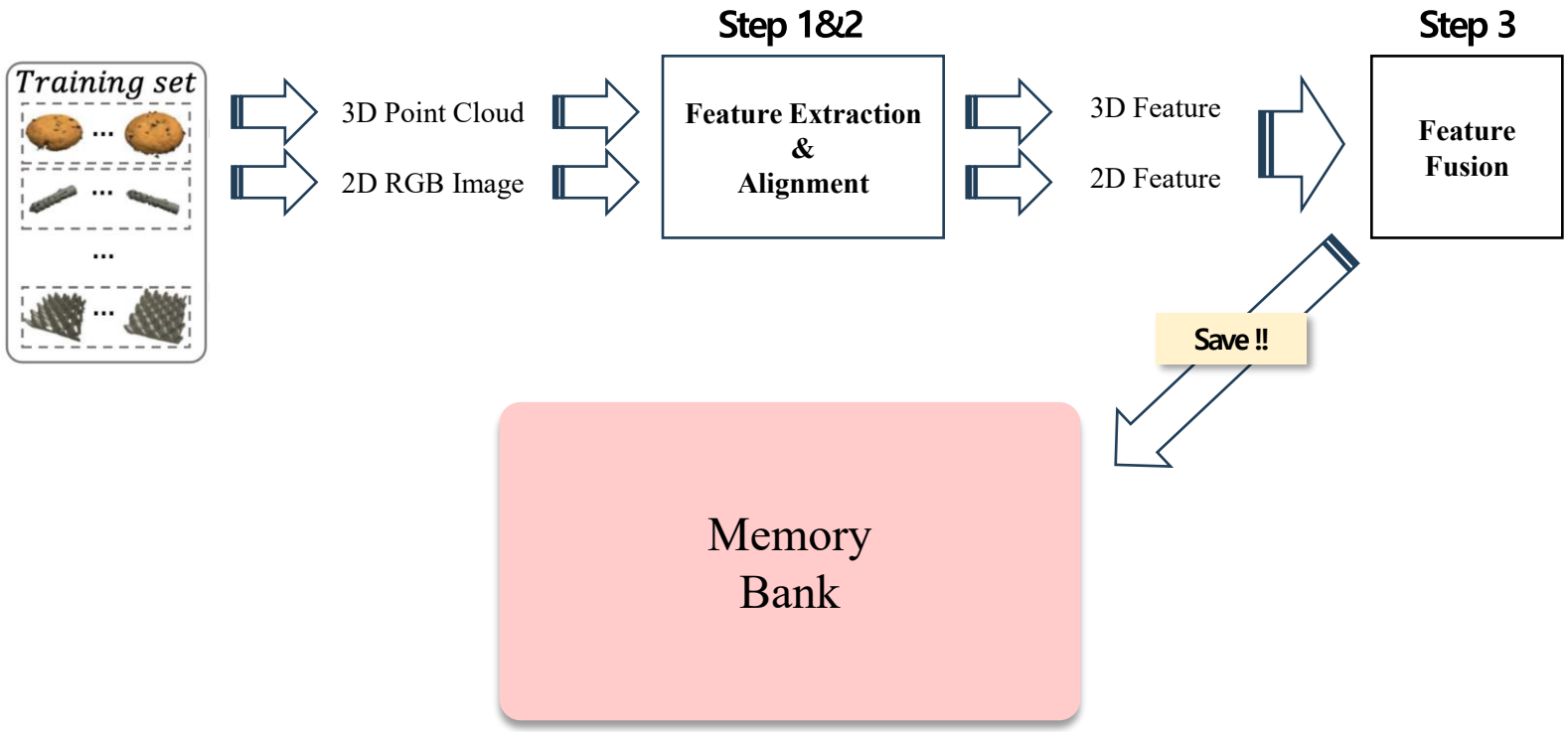
- 각 특징에 대해 MLP와 FC Layer (σ) 적용
- 동일한 패치에 해당하는 3D 특징과 2D 특징이 유사해지도록 학습 → Contrastive Learning 활용
- 융합 시, 최종 레이어 이전 특징을 사용



② 이상치 탐지에 활용

❖ **Memory Bank**

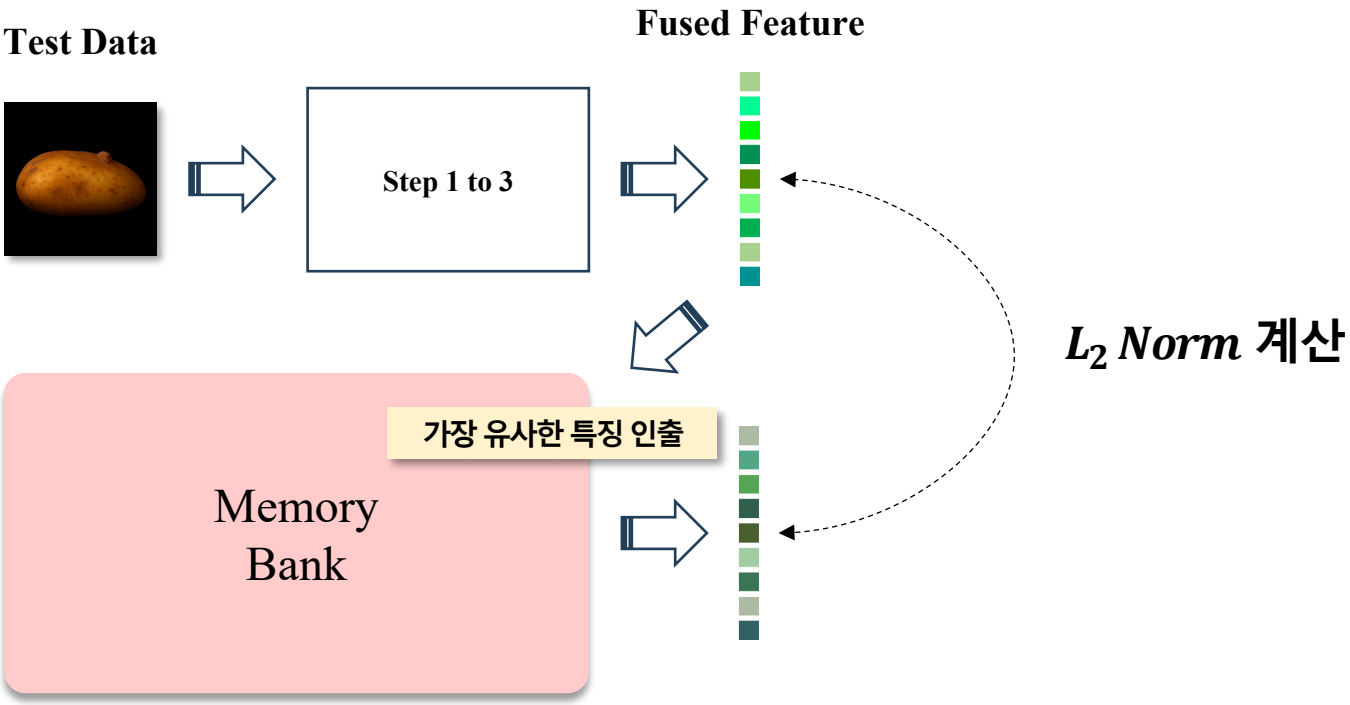
- **학습 시, 정상 데이터로부터 추출된 정상 특징을 모아두자 (Unsupervised AD)**
- 추론 시, Memory bank에서 가장 유사한 정상 특징 인출
  - ➢ 해당 특징과의 거리로 AD 수행



② 이상치 탐지에 활용

❖ **Memory Bank**

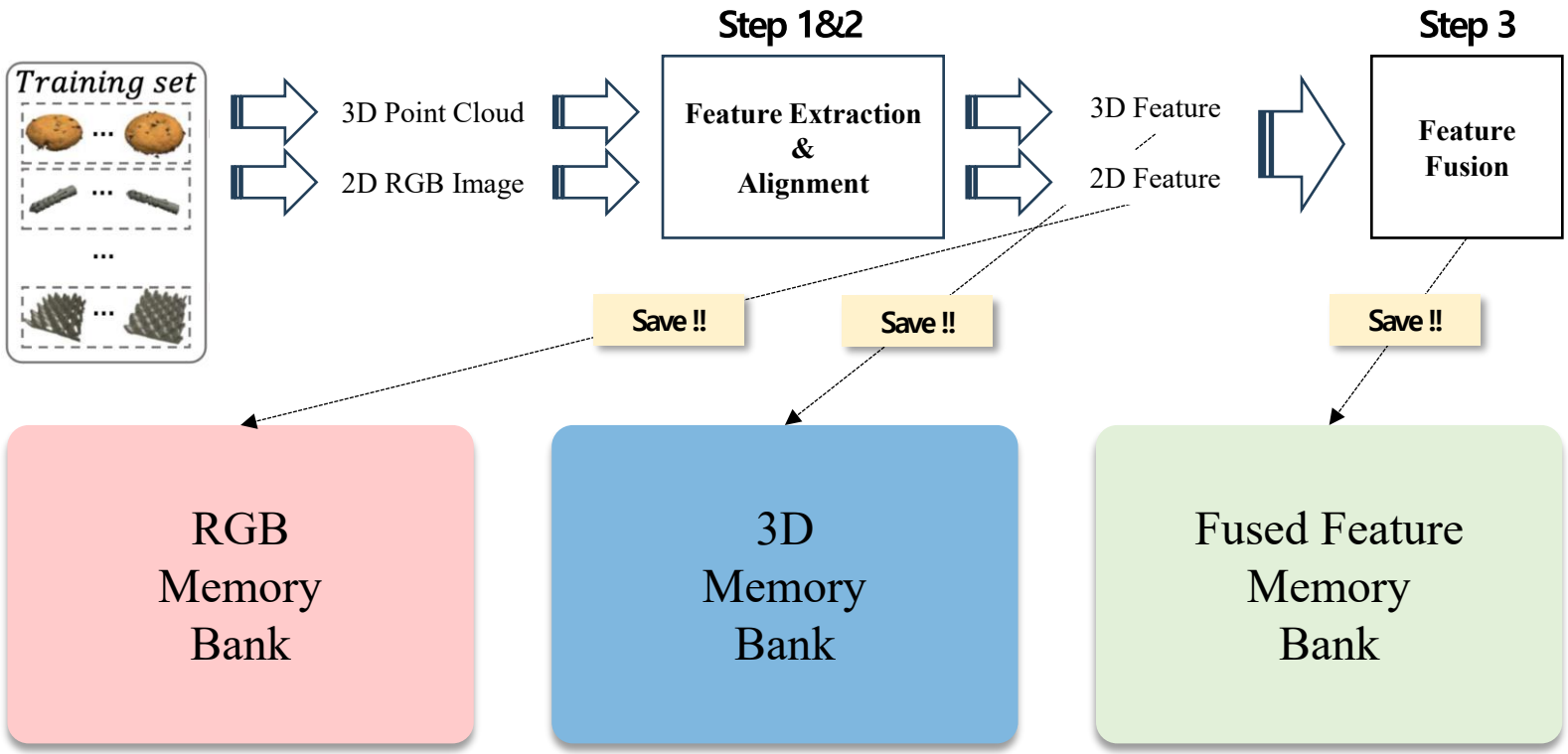
- 학습 시, 정상 데이터로부터 추출된 정상 특징을 모아두자 (Unsupervised AD)
- **추론 시, Memory bank에서 가장 유사한 정상 특징 인출**
  - ➢ **해당 특징과의 거리로 AD 수행**



② 이상치 탐지에 활용

❖ **Multi Memory Bank**

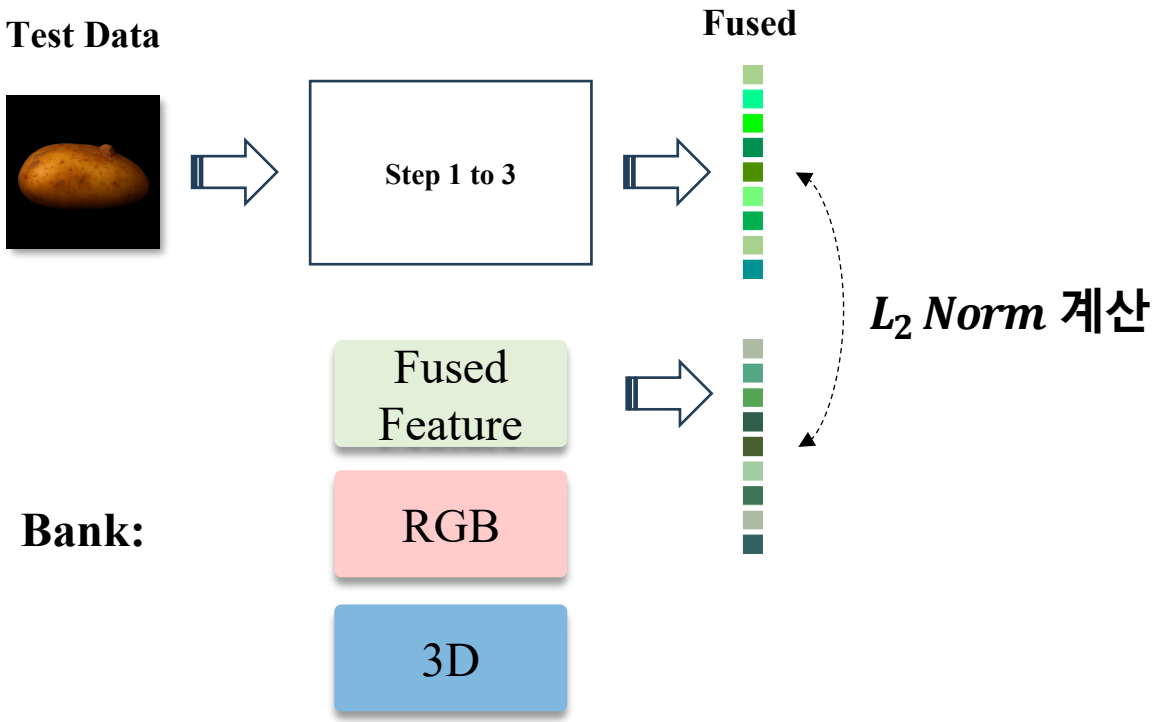
- 경우에 따라, 이상치 탐지에 중요한 정보가 다를 수 있음
  - ➢ 색채 정보 / 형태 정보 / 색채&형태 융합 정보
- 따라서, 각 정보에 대해 Memory Bank를 만들자!!



② 이상치 탐지에 활용

❖ **Multi Memory Bank**

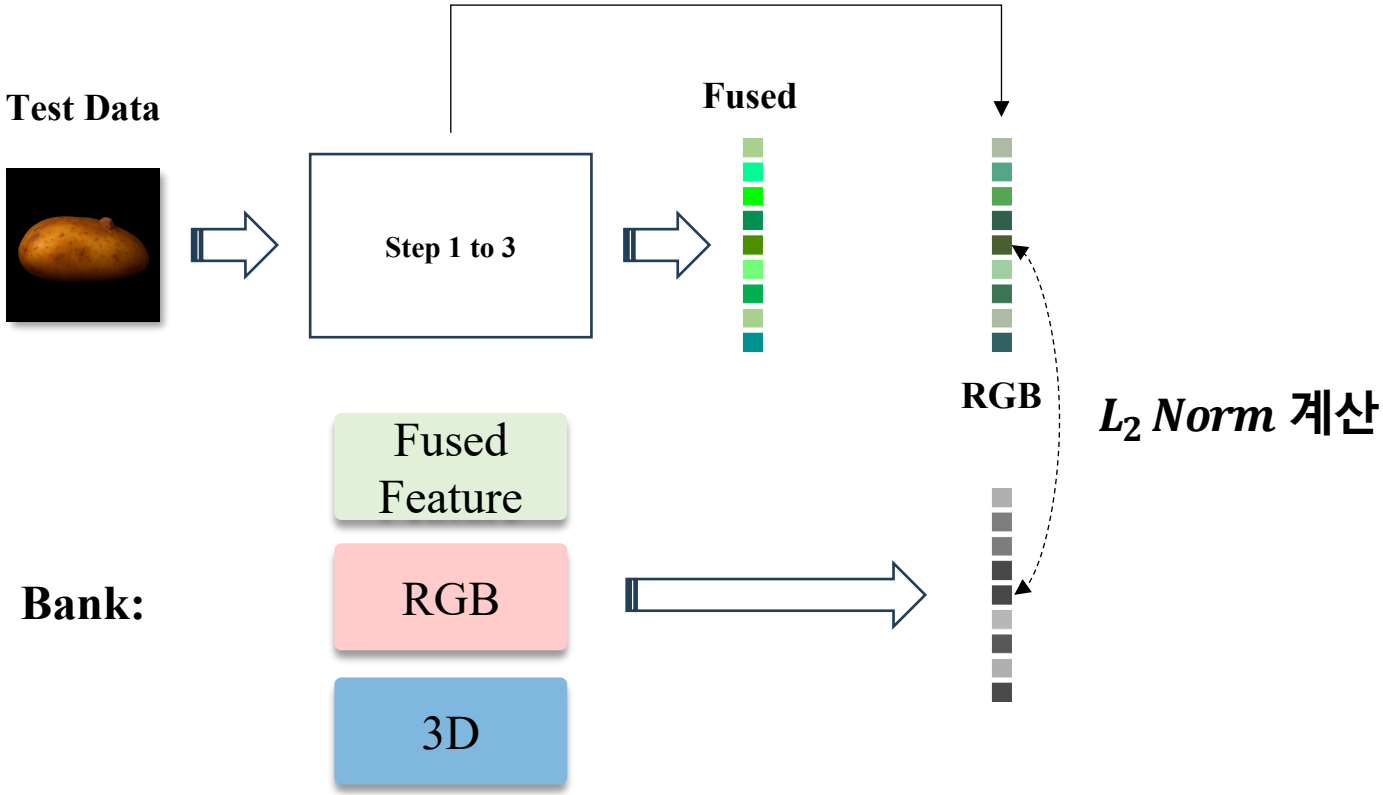
- 경우에 따라, 이상치 탐지에 중요한 정보가 다를 수 있음
  - ➢ 색채 정보 / 형태 정보 / 색채&형태 융합 정보
- 따라서, 각 정보에 대해 Memory Bank를 만들자!!



② 이상치 탐지에 활용

❖ **Multi Memory Bank**

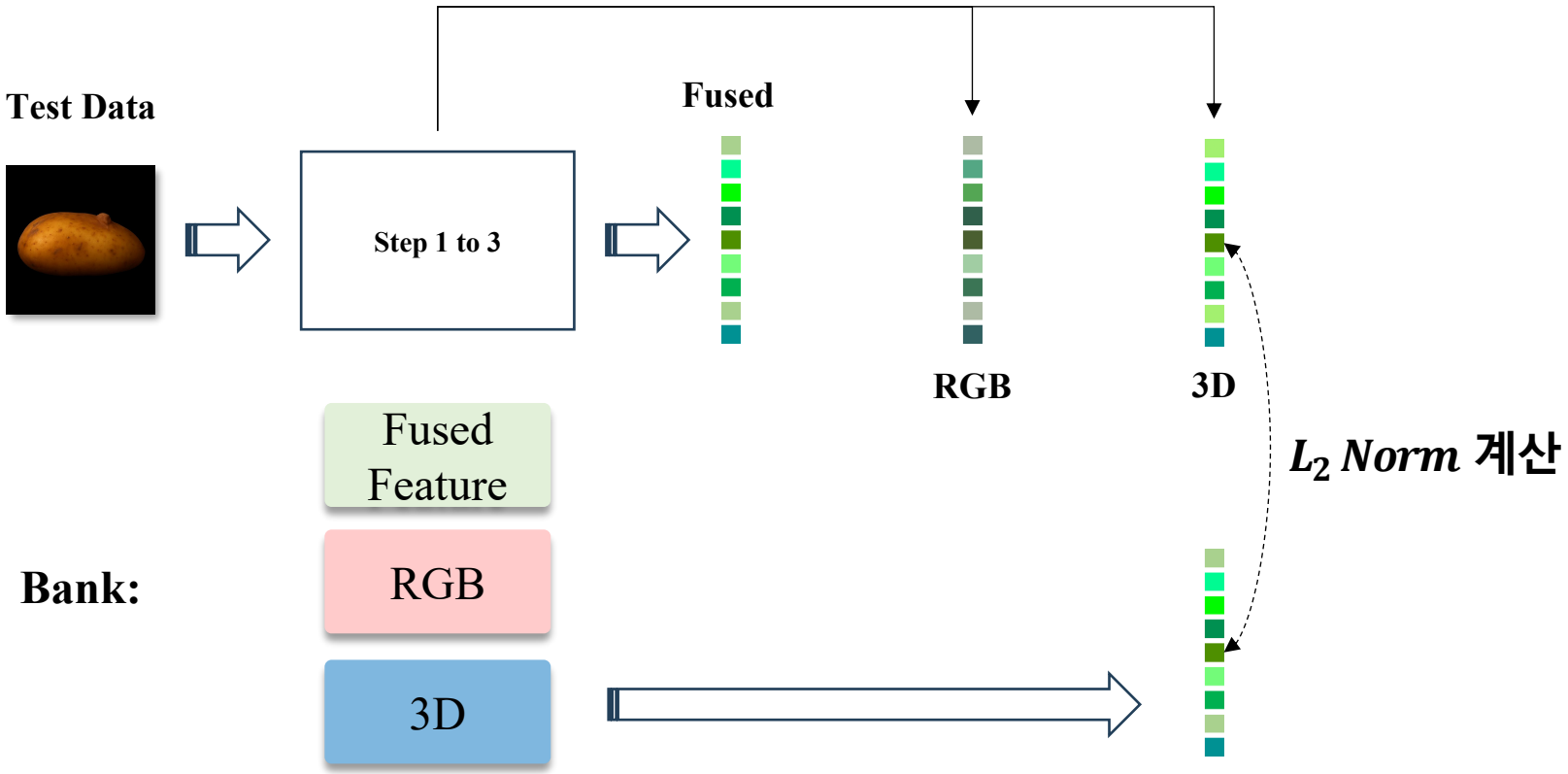
- 경우에 따라, 이상치 탐지에 중요한 정보가 다를 수 있음
  - ➢ 색채 정보 / 형태 정보 / 색채&형태 융합 정보
- 따라서, 각 정보에 대해 Memory Bank를 만들자!!



② 이상치 탐지에 활용

❖ **Multi Memory Bank**

- 경우에 따라, 이상치 탐지에 중요한 정보가 다를 수 있음
  - ➢ 색채 정보 / 형태 정보 / 색채&형태 융합 정보
- 따라서, 각 정보에 대해 Memory Bank를 만들자!!



Experiments

❖ Anomaly Classification 실험

- MVTec-3D 데이터 사용
  - ➢ (Image, Point Cloud, Label) 쌍으로 구성
- 지표: AUROC – 전체적인 분류 성능 확인, **1에 가까울수록 성능 ↑**

	Method	Bagel	Cable Gland	Carrot	Cookie	Dowel	Foam	Peach	Potato	Rope	Tire	Mean
3D	Depth GAN [3]	0.530	0.376	0.607	0.603	0.497	0.484	0.595	0.489	0.536	0.521	0.523
	Depth AE [3]	0.468	0.731	0.497	0.673	0.534	0.417	0.485	0.549	0.564	0.546	0.546
	Depth VM [3]	0.510	0.542	0.469	0.576	0.609	0.699	0.450	0.419	0.668	0.520	0.546
	Voxel GAN [3]	0.383	0.623	0.474	0.639	0.564	0.409	0.617	0.427	0.663	0.577	0.537
	Voxel AE [3]	0.693	0.425	0.515	0.790	0.494	0.558	0.537	0.484	0.639	0.583	0.571
	Voxel VM [3]	0.750	**0.747**	0.613	0.738	0.823	0.693	0.679	0.652	0.609	0.690	0.699
	3D-ST [4]	0.862	0.484	0.832	0.894	0.848	0.663	0.763	0.687	**0.958**	0.486	0.748
	FPFH [17]	0.825	0.551	0.952	0.797	0.883	0.582	0.758	0.889	0.929	0.653	0.782
	AST [28]	0.881	0.576	**0.965**	0.957	0.679	**0.797**	**0.990**	0.915	0.956	0.611	0.833
	Ours	**0.941**	0.651	**0.965**	**0.969**	**0.905**	0.760	0.880	**0.974**	0.926	**0.765**	**0.874**
RGB	DifferNet [27]	0.859	0.703	0.643	0.435	0.797	0.790	0.787	0.643	0.715	0.590	0.696
	PADiM [9]	**0.975**	0.775	0.698	0.582	0.959	0.663	0.858	0.535	0.832	0.760	0.764
	PatchCore [26]	0.876	0.880	0.791	0.682	0.912	0.701	0.695	0.618	0.841	0.702	0.770
	STFPM [32]	0.930	0.847	0.890	0.575	0.947	0.766	0.710	0.598	0.965	0.701	0.793
	CS-Flow [16]	0.941	**0.930**	0.827	0.795	**0.990**	0.886	0.731	0.471	0.986	0.745	0.830
	AST [28]	0.947	0.928	0.851	**0.825**	0.981	**0.951**	0.895	0.613	**0.992**	**0.821**	**0.880**
	Ours	0.944	0.918	**0.896**	0.749	0.959	0.767	**0.919**	**0.648**	0.938	0.767	0.850
RGB + 3D	Depth GAN [3]	0.538	0.372	0.580	0.603	0.430	0.534	0.642	0.601	0.443	0.577	0.532
	Depth AE [3]	0.648	0.502	0.650	0.488	0.805	0.522	0.712	0.529	0.540	0.552	0.595
	Depth VM [3]	0.513	0.551	0.477	0.581	0.617	0.716	0.450	0.421	0.598	0.623	0.555
	Voxel GAN [3]	0.680	0.324	0.565	0.399	0.497	0.482	0.566	0.579	0.601	0.482	0.517
	Voxel AE [3]	0.510	0.540	0.384	0.693	0.446	0.632	0.550	0.494	0.721	0.413	0.538
	Voxel VM [3]	0.553	0.772	0.484	0.701	0.751	0.578	0.480	0.466	0.689	0.611	0.609
	PatchCore + FPFH [17]	0.918	0.748	0.967	0.883	0.932	0.582	0.896	0.912	0.921	**0.886**	0.865
	AST [28]	0.983	0.873	**0.976**	0.971	0.932	0.885	**0.974**	**0.981**	**1.000**	0.797	0.937
	Ours	**0.994**	**0.909**	0.972	**0.976**	**0.960**	**0.942**	0.973	0.899	0.972	0.850	**0.945**

Conclusion

❖ **Seminar Goal: AD 전략 선택의 폭을 넓혀보자!**

- Supervised AD: 이상치를 소량 활용하여 성능 향상
- Zero-shot AD: 학습에 활용 가능한 데이터가 없어도, 이상치 탐지 가능!
- Multi-Modal AD: 이미지 정보만으로는 확인할 수 없는 이상치를 탐지하는 것이 가능!