Statistical Visual Computing Lab

UC San Diego

MITSUBISHI ELECTRIC Changes for the Better

MITSUBISHI ELECTRIC RESEARCH LABORATORIES

Long-Tailed Anomaly Detection with Learnable Class Names

Chih-Hui Ho[1], Kuan-Chuan Peng[2], Nuno Vasconcelos[1]

[1]University of California San Diego , [2] Mitsubishi Electric Research Laboratories (MERL)

CVPR JUNE 17-21, 2024 SEATTLE, WA

Introduction

- Anomaly detection (AD) aims to identify defective images and localize the defects.
- Fig. 1 shows that AD models should be able to detect defects over many image classes,
  (1) without relying on hard-coded class names that can be uninformative.
  (2) learn without anomaly supervision.
  (3) robust to the long-tailed distributions of real-world applications.
- To address these challenges, we formulate the problem of long-tailed AD by introducing several datasets split with different levels of class imbalance.
- A novel method, LTAD, is proposed to detect defects from multiple and long-tailed classes, without relying on dataset class names.

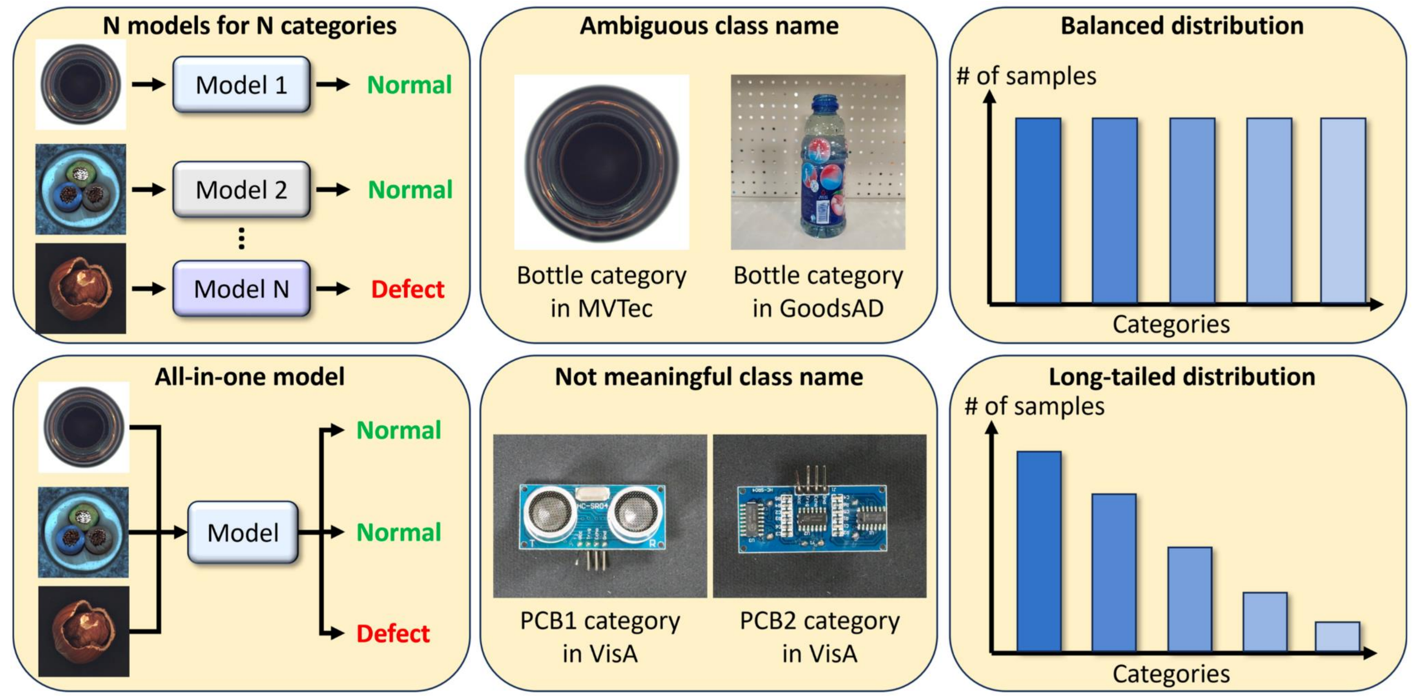

Fig 1. Challenges of long-tailed AD include (Left) designing a single model to detect anomalies over multiple image classes, (Middle) uninformative class names, and (Right) long-tailed data distributions.

Dataset Split & Preliminary Study

- To study how long-tailed distribution affect the performance, we first proposed several new long-tail dataset splits, as shown in left of Fig. 2
  - Imbalance type (e.g. exponential decay and step decay)
  - Class imbalance factor $\beta = \frac{\max\{N_c\}}{\min\{N_c\}}$, where N_c is the sample number of class c
- Performance degrades as the number of sample decreases (See the right of Fig. 2).

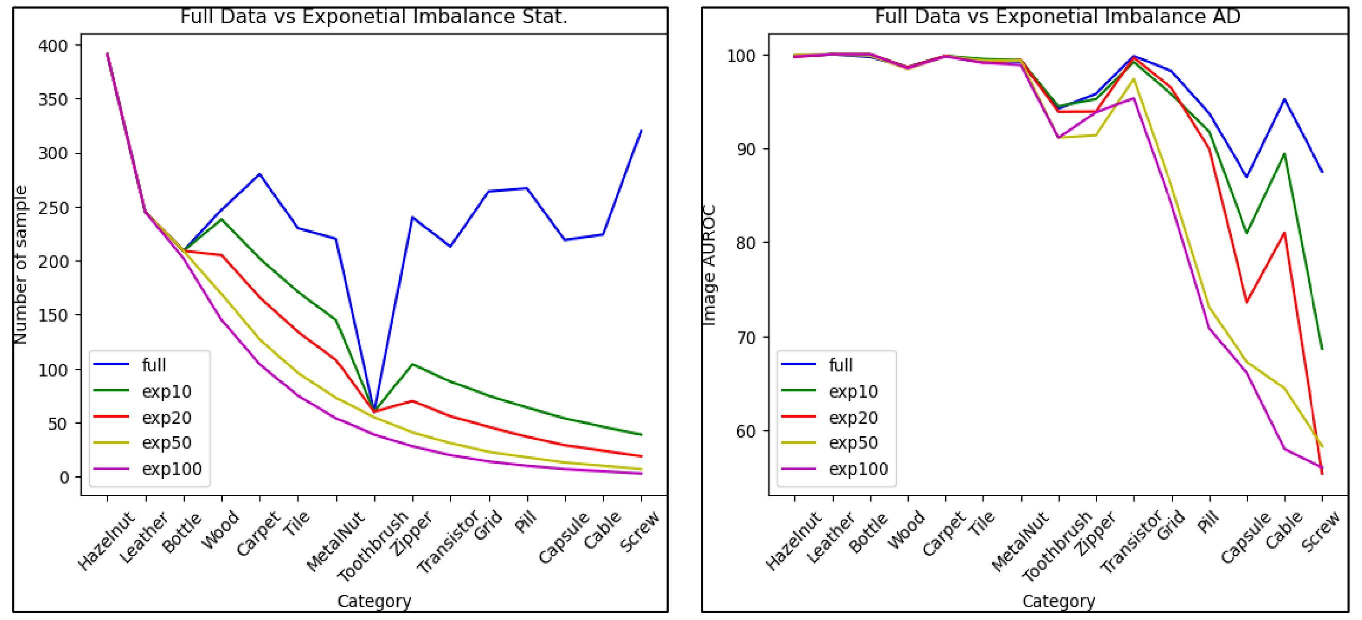

Fig 2. Image classes (x-axis) are sorted by popularity. (Left) Dataset distribution of MVTec [1] vs. long-tailed version. (Right) AD performance of UniAD [2] on the two datasets.

Proposed Method

Training

- The proposed training pipeline, LTAD, contains 2 phases
  - Phase 1: Learn to synthesize feature for tail classes
  - Phase 2: Train to predict the anomaly map using the real/synthesized feature
- For implementation, we use the pretrained visual-language model ALIGN [3], which contains a text encoder and an image encoder that align the image and text to the same feature space.

Phase 1: Class sensitive data augmentation

- Goal : Learn to synthesize feature for tail classes.
- With ALIGN, we proposed a text conditional VAE for synthesizing features (top of Fig. 3).
- Since the class name is unknown, a pseudo class name s_c is learned for each category c.
- MSE loss minimizes reconstruction difference of encoder/decoder feature.
- KL divergence loss regularizes the latent distribution.

Phase 2: Anomaly Detection

- Goal: Train to predict the anomaly map using the real/synthesized feature.
- Phase 2 takes normal feature p_i^n as input (i.e. Real feature or synthesized feature from phase 1).
- Since only normal patch feature p_i^n is available during training, noise is added to the normal feature to create abnormal feature p_i^a.
- Phase 2 contains 2 submodules, including the semantic AD (SAD) module (top of Fig. 3 phase 2) and the reconstruction module (RM) (bottom of Fig. 3 phase 2).
- Reconstruction module (RM)
  - Maps the input feature to normal feature and the MSE loss is used to minimize $||p_i^n - RM(p_i^a)||_2^2$ during training.
- Semantic AD (SAD) module
  - Maps a patch feature p_i to text space and the projected feature is denoted as $\hat{p_i}$.
  - The learned pseudo-class name s_c is concatenated with normal prompt v^n (e.g. a normal s_c) and abnormal prompt v^a (e.g. a broken s_c).
  - The text encoder T outputs the normal text feature $t_{n,c} = T([v^n; s_c])$ and the abnormal text feature $t_{a,c} = T([v^a; s_c])$.
  - The semantic anomaly score of a patch p_i is $S_{sem}(p_i) = \frac{\exp(\hat{p_i} \cdot t_{a,c})}{\exp(\hat{p_i} \cdot t_{n,c}) + \exp(\hat{p_i} \cdot t_{a,c})}$.
  - Ground truth is 1 when $p_i = p_i^a$ and vice versa.
  - Binary cross entropy (BCE) loss is applied on each patch for training.

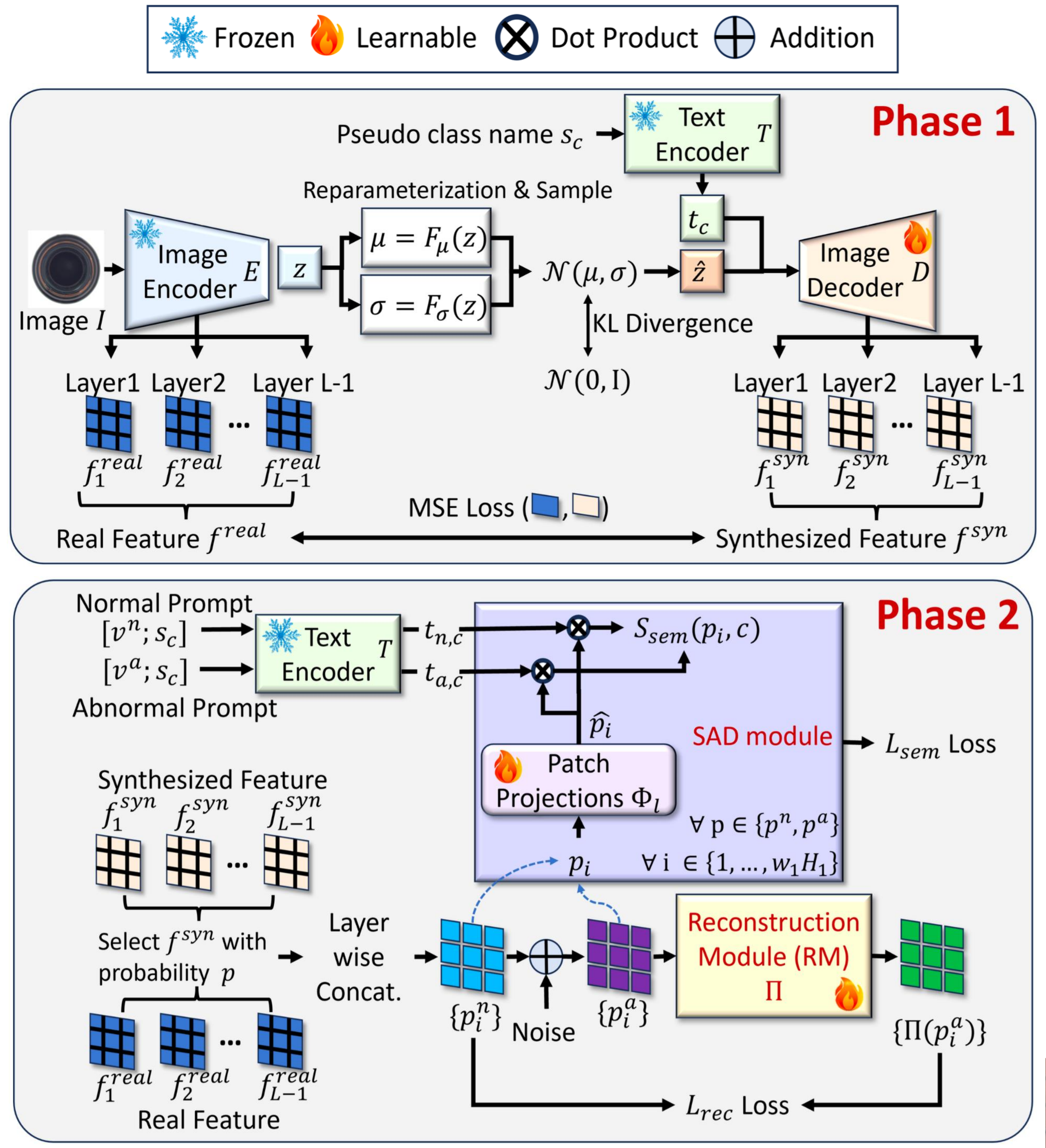

Fig 3. The proposed LTAD training contains Phase 1 (Top) and Phase 2 (Bottom).

Inference

- During testing, RM anomaly score of a patch p_i is $S_{rec}(p_i) = ||p_i - RM(p_i)||_2^2$.
  - When p_i is normal, $S_{RM}(p_i)$ is small
  - When p_i is abnormal, $S_{RM}(p_i)$ is large
- The SAD anomaly score and RM anomaly score are fused with a dataset specific hyperparameter λ.

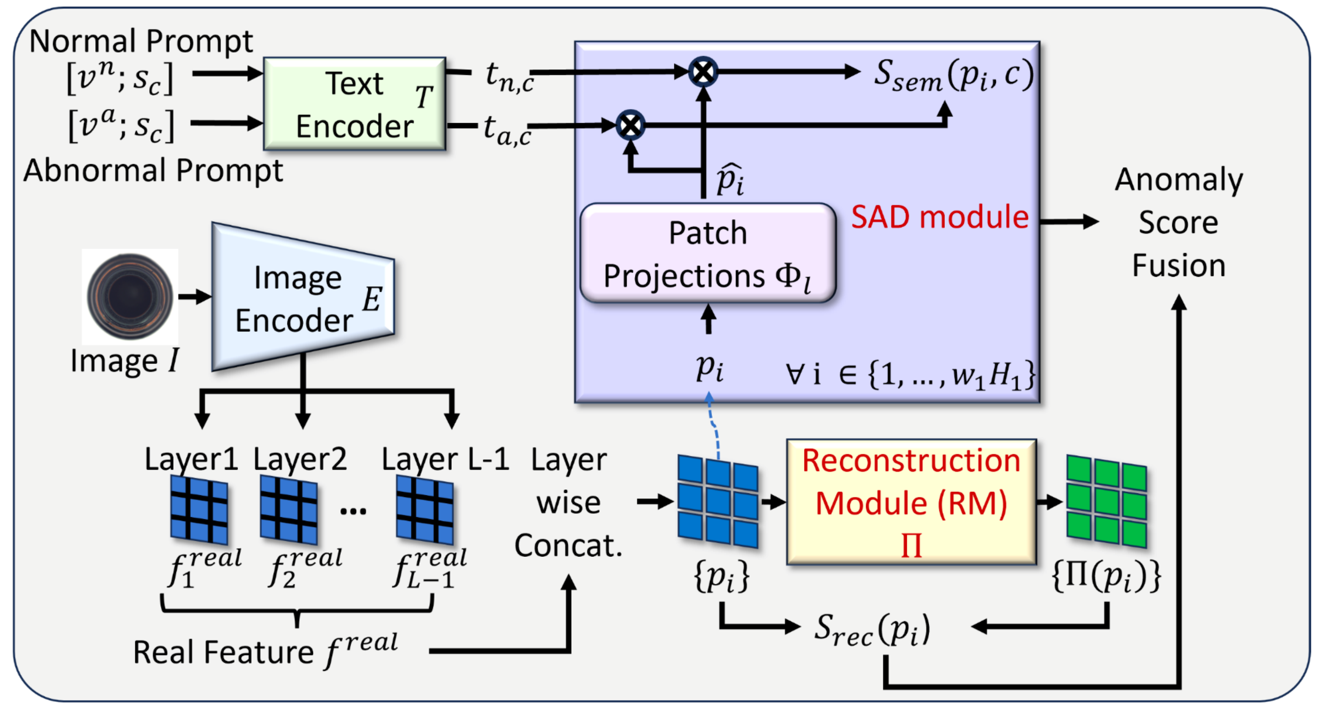

Fig 4. Inference stage of the proposed LTAD.

Experiment

Config.	Task	Cut & Paste	MKD	DRAEM	RegAD	UniAD	AnomalyGPT	LTAD w/o SAD	LTAD
*exp*100	Det.	75.89	78.92	79.57	82.43	87.70	87.44	<u>88.74</u>	**88.86**
	Seg.	N/A	85.95	85.17	**95.20**	93.95	89.68	<u>94.00</u>	<u>94.46</u>
*exp*200	Det.	75.07	79.93	78.82	N/A	<u>86.21</u>	85.80	**86.94**	86.05
	Seg.	N/A	86.01	82.95	N/A	93.26	90.15	<u>93.40</u>	**94.18**
*step*100	Det.	76.57	79.61	69.82	81.54	83.37	85.95	<u>87.05</u>	**87.36**
	Seg.	N/A	85.90	79.65	**95.10**	91.47	89.28	93.13	<u>93.83</u>
*step*200	Det.	76.53	79.31	71.64	N/A	81.32	82.47	<u>85.33</u>	**85.60**
	Seg.	N/A	86.03	76.79	N/A	89.29	89.45	<u>91.78</u>	**92.12**

Table 1. Quantitative result on MVTec [1] dataset.

Config.	Task	RegAD	UniAD	AnomalyGPT	LTAD w/o SAD	LTAD
*exp*100	Det.	71.36	77.31	70.34	<u>79.27</u>	**80.00**
	Seg.	94.40	95.03	80.32	<u>95.07</u>	**95.56**
*exp*200	Det.	72.10	76.87	69.78	<u>78.55</u>	**80.21**
	Seg.	94.69	<u>94.80</u>	79.48	94.51	**95.36**
*exp*500	Det.	N/A	73.67	68.18	<u>77.25</u>	**78.53**
	Seg.	N/A	<u>94.35</u>	78.83	94.04	**94.66**
*step*100	Det.	71.80	78.83	71.98	<u>82.80</u>	**84.80**
	Seg.	94.99	96.04	82.30	<u>96.16</u>	**96.57**
*step*200	Det.	71.65	77.64	69.78	<u>83.79</u>	**84.03**
	Seg.	94.52	95.66	81.97	<u>95.89</u>	**96.27**
*step*500	Det.	N/A	71.84	62.88	<u>82.42</u>	**83.33**
	Seg.	N/A	95.03	81.48	<u>95.50</u>	**96.41**

Table 2. Quantitative result on VisA [4] dataset.

Config.	Task	RegAD	UniAD	AnomalyGPT	LTAD w/o SAD	LTAD
*exp*100	Det.	84.86	84.34	85.31	<u>93.35</u>	**94.40**
	Seg.	90.29	90.13	77.20	<u>96.93</u>	**97.30**
*exp*200	Det.	84.86	83.56	83.29	<u>92.83</u>	**94.29**
	Seg.	90.29	89.73	77.16	<u>96.16</u>	**97.19**
*exp*500	Det.	84.86	81.35	83.47	<u>92.08</u>	**93.54**
	Seg.	90.29	88.63	76.87	<u>95.99</u>	**97.01**
*step*100	Det.	84.86	81.11	86.48	<u>91.94</u>	**93.97**
	Seg.	90.28	89.11	78.76	<u>96.38</u>	**97.07**
*step*200	Det.	84.86	80.33	84.73	<u>91.78</u>	**93.79**
	Seg.	90.29	89.07	78.29	<u>96.04</u>	**96.84**
*step*500	Det.	84.86	80.04	85.08	<u>91.82</u>	**92.78**
	Seg.	90.29	88.53	78.75	<u>95.64</u>	**96.65**

Table 3. Quantitative result on DAGM [5] dataset.

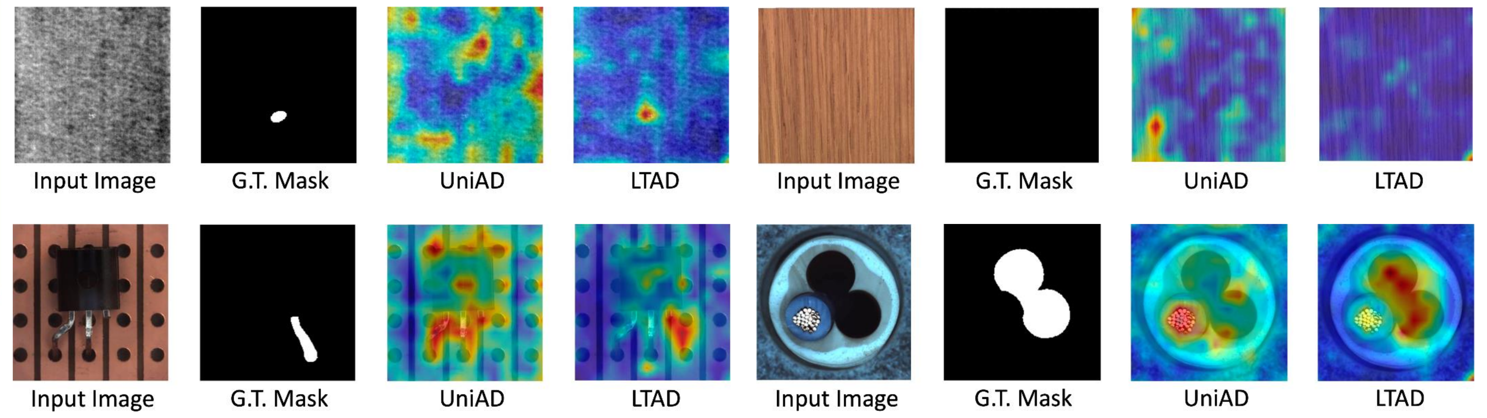

Fig 5. Quantitative result of the proposed LTAD and the baseline.

assign $s_{c=i}$ to class i	use text encoder T	Detection			Segmentation		
		All	High	Low	All	High	Low
✗	✓	72.76	81.06	65.49	63.74	62.09	65.18
✓	✗	59.79	63.34	56.69	69.83	70.95	68.85
✓	✓	**84.12**	**97.02**	**72.84**	**91.36**	**95.13**	**88.07**

Table 4. Importance of pseudo class name s_c on MVTec-*step*100.

v^n	v^a: a broken	a damaged	an abnormal	a defective
a	**84.12** / 91.36	82.95 / 91.70	82.20 / 91.33	83.66 / **91.87**
a normal	83.71 / 91.39	82.74 / 91.21	83.47 / 91.23	82.94 / 91.26
a good	75.68 / 90.75	82.14 / 91.22	81.03 / 91.15	82.09 / 91.23
a flawless	65.63 / 87.61	79.09 / 91.00	76.13 / 90.24	83.89 / 91.42

Table 5. Ablation on different normal/abnormal text prompts (*i.e.*, v^a and v^n) on MVTec *step*100.

Acknowledgement

CH and KCP were supported by Mitsubishi Electric Research Laboratories. CH and NV were partially funded by NSF awards IIS-2303153 and gift from Qualcomm.

References

[1] Bergmann et. al, MVTec AD — A comprehensive real-world dataset for unsupervised anomaly detection. *CVPR* 2019
[2] You et. al. A unified model for multi-class anomaly detection. NeurIPS 2022.
[3] Jia et. al, Scaling up visual and vision-language representation learning with noisy text supervision. ICML, 2021
[4] Zou et al. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. ECCV 2022
[5] Wieler et al., Weakly supervised learning for industrial optical inspection, 2007.