SAIGE

Language-Assisted Feature Transformation for Anomaly Detection

EungGu Yun[†] SAIGE

Heonjin Ha LG Uplus

Yeongwoo Nam Alsemy Inc.

Bryan Dongik Lee Independent

[†]Corresponding author

*The authors conducted this work at SAIGE

Abstract

- Defining normality is key to anomaly detection but challenging with limited data or nuisance attributes.
- Unsupervised methods lack user guidance, often missing anomalies of specific interest.
- **LAFT** integrates user knowledge via natural language for anomaly detection.
- Utilizing shared image-text embedding space of CLIP, it aligns visual features with user requirements.
- LAFT aligns visual features with user preferences, allowing anomalies of interest to be detected.

Anomaly Detection

Anomaly Detection (AD) is the task of identifying abnormal data that deviates from the norm.

- Trained primarily on normal data.
- Useful when abnormal data is rare or unavailable.
- Learns the boundary of normality based on user-provided data.

Challenges of AD:

- Sensitive to biased or non-diverse training data.
- Needs to prioritize relevant attributes (e.g., shape) and ignore others (e.g., color, lighting).
- Struggles with entangled attributes (e.g., Waterbirds dataset).
- Data augmentation and generation help but may fail to generalize or reflect user intent.

Vision-Language Models

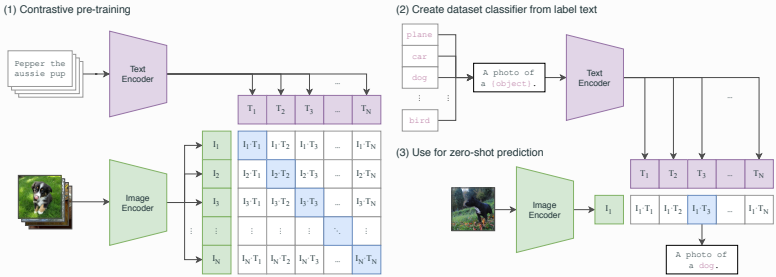

Figure 1: CLIP constructs shared image-text embedding space by jointly training image and text encoders. The two encoders are trained to align the features of image-text pairs. (Figure from Radford et al. [2021])

We can leverage the shared image-text embedding space of CLIP to align visual features through user-provided text prompts.

Motivation

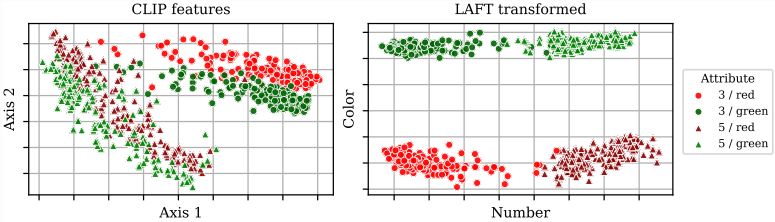

Figure 2: Projection of image features from CLIP's image encoder (**left**) and transformed image features using LAFT (**right**). Without guidance, the image features may not align with the intended attributes. After applying LAFT, the features become more aligned with the desired attributes.

Motivation (Cont'd)

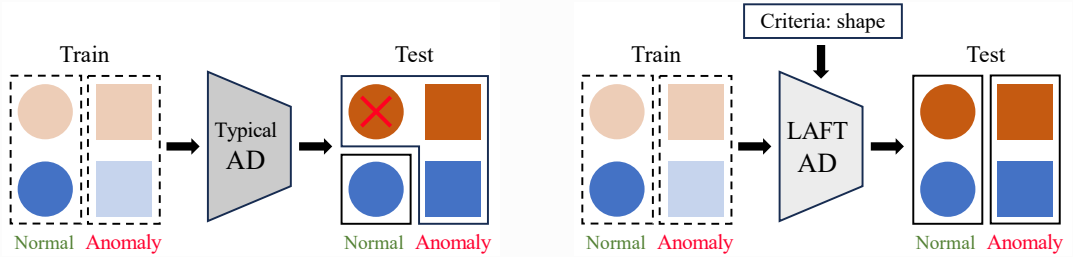

Figure 3: High-level motivation of our method: (**left**) typical image anomaly detection methods treat all test data that differs from the training data as anomalies, while (**right**) our method, LAFT AD, incorporates user preferences into the anomaly detection.

Preliminaries

For a two-stage image anomaly detection model:

- Feature extractor: $v = f(x)$ where x is the input image and v is the feature,
- Anomaly classifier: $s = g(v)$ where s is the anomaly score.

The anomaly score s_i is used to determine the prediction of the anomaly label $\hat{y}_i$.

For an image x and the corresponding anomaly label y:

- Extracted attributes of an image: $a = \{a^1, \cdots, a^m\}$

where each attribute a^j $(j = 1, \cdots, m)$ denotes any characteristics within the image (e.g., shape, color).

The m attributes can be divided into:

- **Relevant attributes**: $a^{\text{rel}} = \{a^j\}_{1 \le j \le n}$
- **Irrelevant attributes**: $a^{\text{irr}} = \{a^j\}_{n < j \le m}$

For example, when detecting anomalies in object shapes, shape matters, but color does not.

Preliminaries (Cont'd)

To properly detect anomalies, the prediction of the model should be invariant to the irrelevant attributes.

This can be achieved by two approaches:

Guide T_{guide} includes only the relevant attributes $a^j \in a^{\text{rel}}$. Some attributes in a^{rel} may be correlated, so the transformed feature may not include all relevant attributes.

Ignore T_{ignore} excludes all irrelevant attributes $\forall a^j \in a^{\text{irr}}$. In many cases this is harder to achieve than the above approach, because the transformation should be able to remove all irrelevant attributes.

We want the transformed feature $v' = T(v)$ to be informative, containing enough information about relevant attributes. Here, $I(;)$ represents the mutual information between the two arguments:

$$I((a^1, \cdots, a^n); v) \sim I((a^1, \cdots, a^n); T(v)). \tag{1}$$

Method - Overview

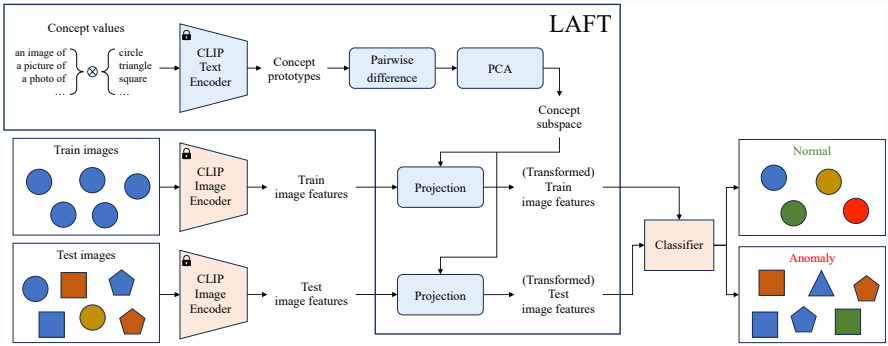

Figure 4: Overview of our method, LAFT, a transformation module, and LAFT AD, combining LAFT with a kNN classifier. Our approach uses CLIP's text and image encoders without any additional training. The key idea is to use text prompts containing concept values to construct a concept subspace for the target attribute. This process involves computing pairwise differences of concept prototypes and extracting robust concept axes via PCA.

Method - Text Prompt

Following Ming et al. [2022], we assume that the text contains **concept prototypes** representing the attributes. We provide the method with a list of prompts composed of templates and values. We use the actual values of the desired attribute (e.g., “circle”, “square”) rather than the attribute name (e.g., “shape”).

For instance, to capture the concept of hair color, we can construct the prompt as:

“a photo of a person with brown hair”
“a potrait of a man with black hair”
“an image of a blond child”

By using the actual values of the desired attribute in the prompts, we aim for the method to capture the difference between the concept prototypes of the attribute.

Method - Concept Subspace

Mikolov [2013] showed that simple arithmetic operations between text embeddings can capture meaningful relationships (e.g., vec(`biggest`) − vec(`big`) ≈ vec(`smallest`) − vec(`small`)).

For prompts t_i and t_j, where $1 \leq i < j \leq n$, we compute the pairwise differences of the text features:

$$\Delta v_{ij} := E_{\text{text}}(t_i) - E_{\text{text}}(t_j) \tag{2}$$

where n represents the number of prompts, and E_{text} denotes CLIP's text encoder. We apply PCA to extract the principal axes from these vectors:

$$\{c_k\}_{1 \leq k \leq d} := \text{PCA}(\{\Delta v_{ij}\}_{1 \leq i < j \leq n}, d) \tag{3}$$

where d is the number of components, and $\{c_k\}$ represents the d principal axes, collectively referred to as the **concept axes**.

Method - Feature Transformation

For each image feature $v_i = f(x_i)$ encoded by CLIP's image encoder, we project the features onto the concept subspace:

$$v_i' = T_{\text{guide}}(v_i) := \sum_{k=1}^{d} \frac{\langle v_i, c_k \rangle}{\langle c_k, c_k \rangle} c_k, \tag{4}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

Conversely, to remove the irrelevant attributes, we orthogonally project the features onto the concept subspace:

$$\bar{v}_i' = T_{\text{ignore}}(v_i) := v_i - \sum_{k=1}^{d} \frac{\langle v_i, \bar{c}_k \rangle}{\langle \bar{c}_k, \bar{c}_k \rangle} \bar{c}_k, \tag{5}$$

where $\bar{c}_k$ represents the concept axes associated with the irrelevant attributes.

Experiments

Table 1: Anomaly detection performance (%) on Colored MNIST and Waterbirds datasets. Standard deviations are computed over five different seeds, with results for deterministic cases omitted. The best values are shown in **bold**, and the second-best values are underlined.

		Colored MNIST: Number			Waterbirds: Bird		
Guidance	Method	AUROC ↑	AUPRC ↑	FPR95 ↓	AUROC ↑	AUPRC ↑	FPR95 ↓
Baseline							
Subset of normal images	kNN	92.4 ± 0.2	91.8 ± 0.2	31.9 ± 0.6	82.3	91.2	48.1
+ All normal images	kNN	98.0 ± 0.0	97.1 ± 0.0	7.5 ± 0.2	83.0	91.5	44.7
+ Anomalous images	LinearProbe	99.8 ± 0.0	99.8 ± 0.0	0.5 ± 0.1	91.0 ± 0.0	96.7 ± 0.0	34.2 ± 0.0
Guide							
Language	MCM	62.9	52.5	60.8	88.8	95.4	40.0
	ZOE	91.2	92.4	47.3	<u>92.2</u>	<u>97.1</u>	32.8
	CLIPN-C	73.0 ± 2.5	61.7 ± 3.0	51.0 ± 0.9	71.2 ± 2.8	86.5 ± 1.1	100.0 ± 0.0
	CLIPN-A	73.2 ± 2.2	61.6 ± 2.7	50.0 ± 0.6	82.3 ± 0.8	91.9 ± 0.3	55.6 ± 1.2
	WinCLIP	91.1	92.4	48.0	<u>92.2</u>	97.0	<u>32.6</u>
Image + Language	WinCLIP+	92.6 ± 1.3	91.3 ± 2.0	38.8 ± 1.5	91.8 ± 0.2	96.9 ± 0.1	33.4 ± 1.5
	InCTRL	94.0 ± 1.3	92.4 ± 2.6	25.5 ± 4.1	83.6 ± 1.0	92.0 ± 0.7	63.5 ± 3.1
	LAFT AD (ours)	**98.5** ± 0.0	**98.4** ± 0.0	**6.9** ± 0.1	**95.6**	**98.4**	**20.6**
Ignore							
Image + Language	LAFT AD (ours)	<u>97.4</u> ± 0.1	<u>96.9</u> ± 0.2	<u>10.4</u> ± 0.4	84.8	92.2	38.6

Experiments (Cont'd)

Table 2: Anomaly detection performance (%) on CelebA dataset. Standard deviations are computed over five different seeds, with results for deterministic cases omitted. The best values are shown in **bold**, and the second-best values are underlined.

		Hair color			Eyeglasses		
Guidance	Method	AUROC ↑	AUPRC ↑	FPR95 ↓	AUROC ↑	AUPRC ↑	FPR95 ↓
Baseline							
Subset of normal images	kNN	83.3	96.6	62.4	83.0	21.6	47.7
+ All normal images	kNN	83.4	96.6	62.4	85.3	22.0	43.2
+ Anomalous images	LinearProbe	98.2 $\pm$ 0.0	99.7 $\pm$ 0.0	9.6 $\pm$ 0.0	99.7 $\pm$ 0.0	98.4 $\pm$ 0.0	0.1 $\pm$ 0.0
Guide							
Language	MCM	84.5	97.2	68.4	5.7	3.3	100.0
	ZOE	93.9	99.0	35.7	82.6	31.5	67.3
	CLIPN-C	82.8 $\pm$ 1.7	96.8 $\pm$ 0.4	72.0 $\pm$ 2.4	1.4 $\pm$ 0.1	3.7 $\pm$ 0.0	100.0 $\pm$ 0.0
	CLIPN-A	84.7 $\pm$ 1.2	97.2 $\pm$ 0.3	70.2 $\pm$ 2.1	1.2 $\pm$ 0.1	3.4 $\pm$ 0.0	100.0 $\pm$ 0.0
	WinCLIP	93.7	98.9	37.7	83.6	34.6	66.2
Image + Language	WinCLIP+	92.8 $\pm$ 0.3	98.8 $\pm$ 0.1	41.2 $\pm$ 1.4	85.0 $\pm$ 2.4	26.8 $\pm$ 3.0	47.8 $\pm$ 6.9
	InCTRL	85.7 $\pm$ 0.9	96.9 $\pm$ 0.3	67.8 $\pm$ 1.6	87.8 $\pm$ 1.6	30.4 $\pm$ 2.9	29.6 $\pm$ 4.4
	LAFT AD (ours)	**95.0**	**99.2**	**29.8**	**98.1**	**80.7**	**5.9**

Conclusion

We introduced **LAFT (Language-Assisted Feature Transformation)**, a novel, training-free method to integrate user guidance into anomaly detection using natural language.

LAFT leverages the shared embedding space of vision-language models (CLIP) to:

- Construct **concept subspaces** from user-provided text prompts representing desired attributes.
- **Transform image features** by projecting them relative to these concept axes, effectively emphasizing or suppressing specific visual attributes based on user intent.

Our framework allows users to flexibly adjust the normality boundary via language, aligning the anomaly detection process with specific requirements and prior knowledge.

References

T. Mikolov. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li. Delving into out-of-distribution detection with vision-language representations. Advances in Neural Information Processing Systems (NeurIPS), 2022.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning (ICML), 2021.