模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

AnomalyGPT: Detecting Industrial Anomalies using Large Vision-Language Models

Zhaopeng Gu[1,2] Bingke Zhu[1,3,4] Guibo Zhu[1,4]
Yingying Chen[1,3,4] Ming Tang[1,2] Jinqiao Wang[1,2,3,4]

[1] Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Objecteye Inc., Beijing, China
[4] Wuhan AI Research, Wuhan, China

guzhaopeng2023@ia.ac.cn
{bingke.zhu,gbzhu,yingying.chen,tangm,jqwang}@nlpr.ia.ac.cn

Anomaly Detection

Anomaly Detection is a binary classification identifying unusual or unexpected patterns in a dataset, which deviate significantly from the majority of the data. The goal of anomaly detection is to identify such anomalies, which could represent errors, fraud, or other types of unusual events, and flag them for further investigation.

1. **hard to gain a large amount of defective images.**
2. **various defect types.**

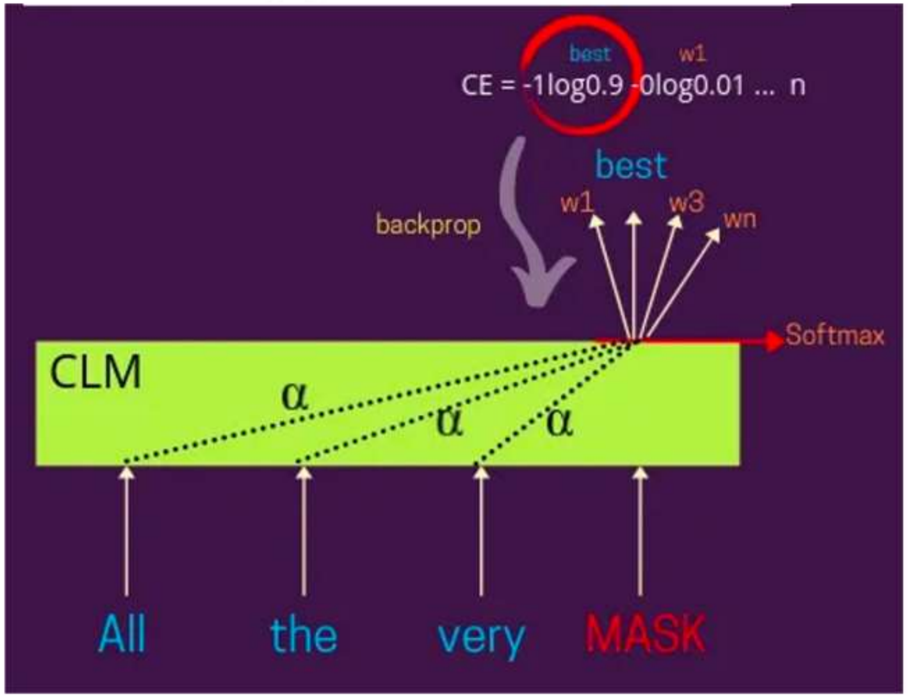
Unsupervised Algorithms

- Reconstruction-based Algorithms: AE, VAE, GAN, etc.
- Normalizing Flow-based Algorithms: CFlow, FastFlow, etc.
- Representation-based Algorithms: SPADE, PatchCore, etc.
- Data augmentation-based Algorithms: DRAEM, CutPaste, etc.

Language Models

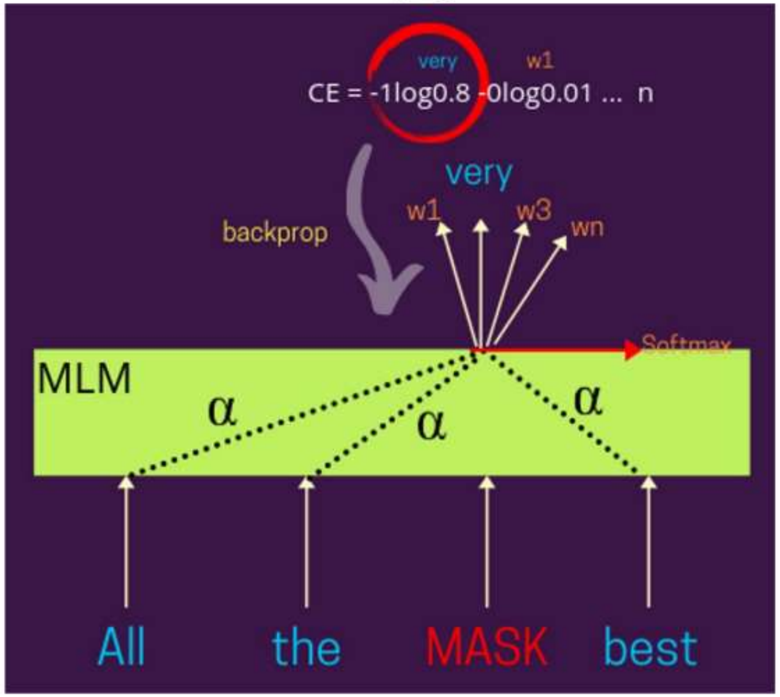
- Causal Language Model (CLM)

$$L_{\mathrm{alm}}(X) = \sum_{n=1}^{N} \log p(x_n | x_1, ..., x_{n-1}; \theta),$$



- Masked Language Model (MLM)

$$L_{\mathrm{mlm}}(X_{\Pi} | X_{-\Pi}) = \frac{1}{K} \sum_{k=1}^{K} \log p(x_{\pi_k} | X_{-\Pi}; \theta).$$



Motivation:

- Large Vision-Language Models (LVLMs) have strong abilities of understanding images, but they lack specific domain knowledge and have a weaker understanding of localized details within objects.
- Most existing IAD methods only provide anomaly scores and necessitate the manual setting of thresholds to distinguish between normal and abnormal samples.

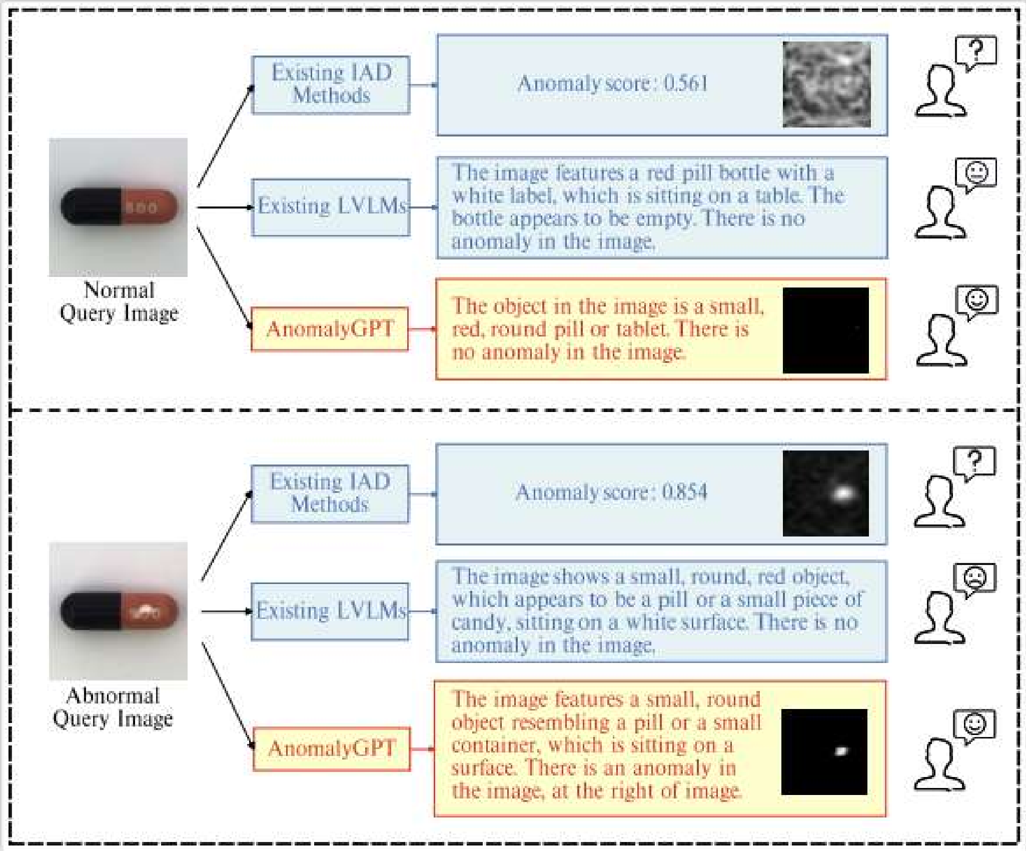
Methods	Few-shot learning	Anomaly score	Anomaly localization	Anomaly judgement	Multi-turn dialogue
Traditional IAD methods		✓	✓		
Few-shot IAD methods	✓	✓	✓		
LVLMs	✓				✓
AnomalyGPT (ours)	✓	✓	✓	✓	✓

Table 1. Comparison between our AnomalyGPT and existing methods across various functionalities. The "Traditional IAD methods" in the table refers to "one-class-one-model" methods such as PatchCore [23], InTra [21], and PyramidFlow [13]. "Few-shot IAD methods" refers to methods that can perform few-shot learning like RegAD [10], Graphcore [29], and WinCLIP [27]. "LVLMs" represents general large vision-language models like MiniGPT-4 [36], LLaVA [17], and PandaGPT [25]. "Anomaly score" in the table represents just providing scores for anomaly detection, while "Anomaly judgement" indicates directly assessing the presence of anomaly.

Introduction

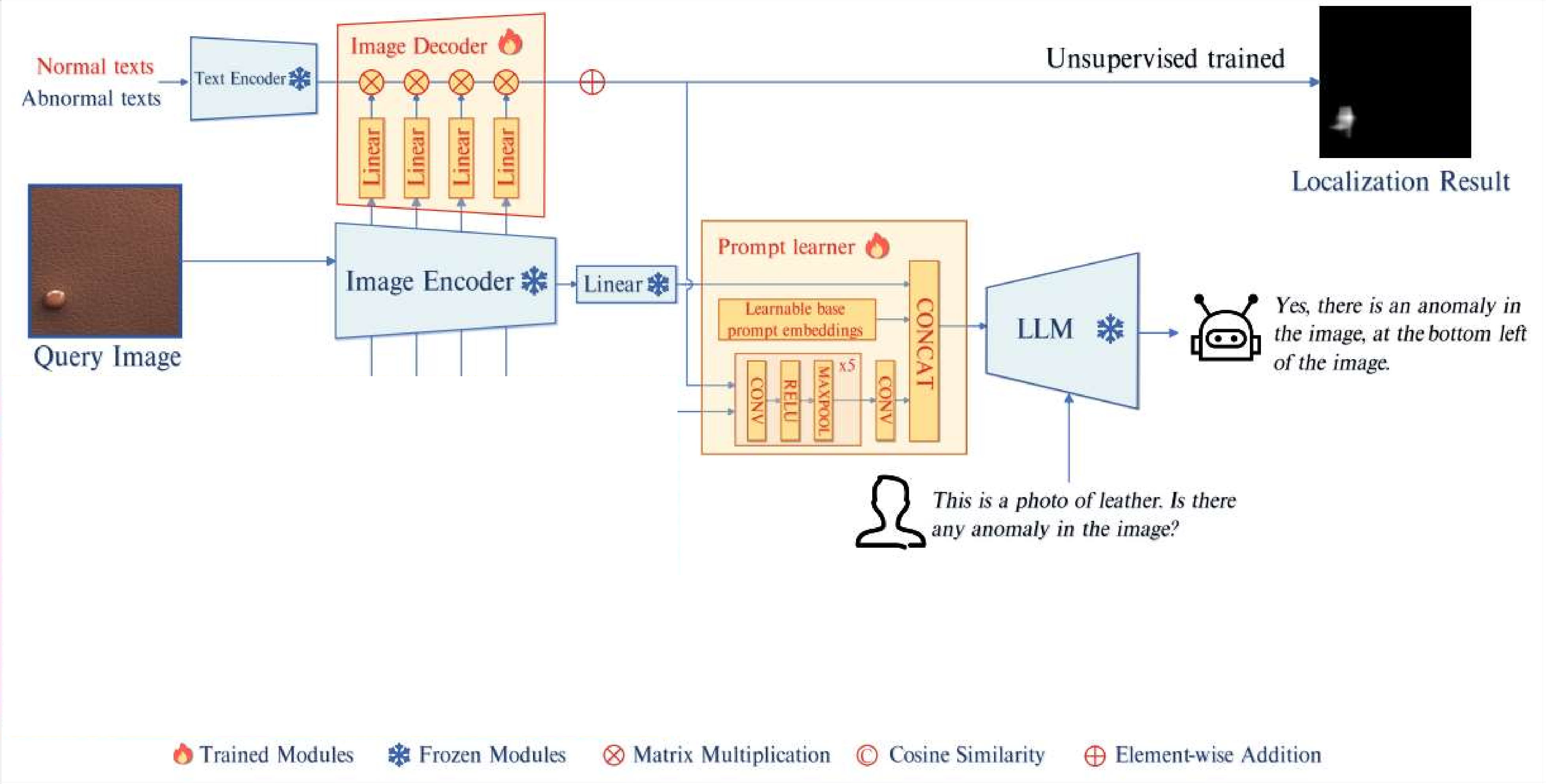
Contributions:

- Sucessfully apply LVLM to the domain of industrial anomaly detection without manually threshold adjustments.
- Use a visual-textual feature-matching-based decoder to address the limitation of the LLM's weaker discernment of fine-grained semantic and alleviate the constrains of LLM's restricted ability to solely generate text outputs.
- Employ prompt embeddings for fine-tuning.
- Be capable of engaging in in-context few-shot learning on new datasets.



Method

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing



Anomaly Simulation

The Poisson editing method [20] has been developed to seamlessly clone an object from one image into another image by solving the Poisson partial differential equations.

[20] Patrick Perez, Michel Gangnet, and Andrew Blake. Poisson image editing. In ACM SIGGRAPH 2003 Papers, pages 313– 318. 2003.

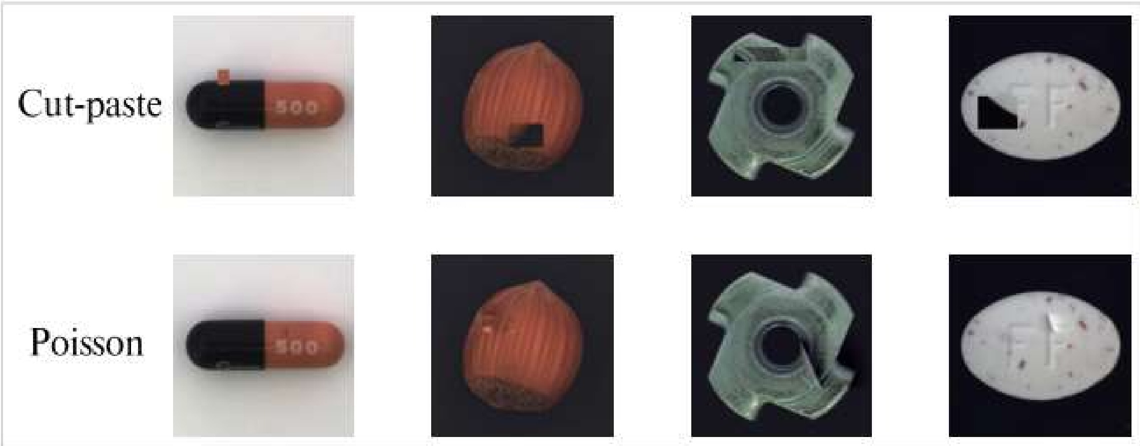
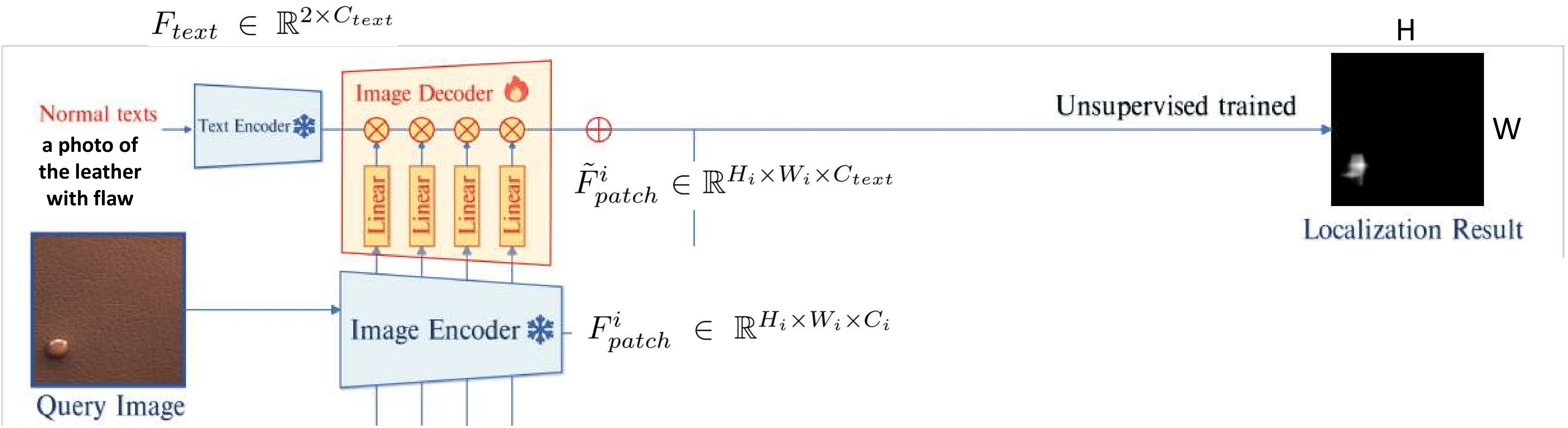


Figure 3. Illustration of the comparison between cut-paste and poisson image editing. The results of cut-paste exhibit evident discontinuities and the results of poisson image editing are more natural.

Image Decoder



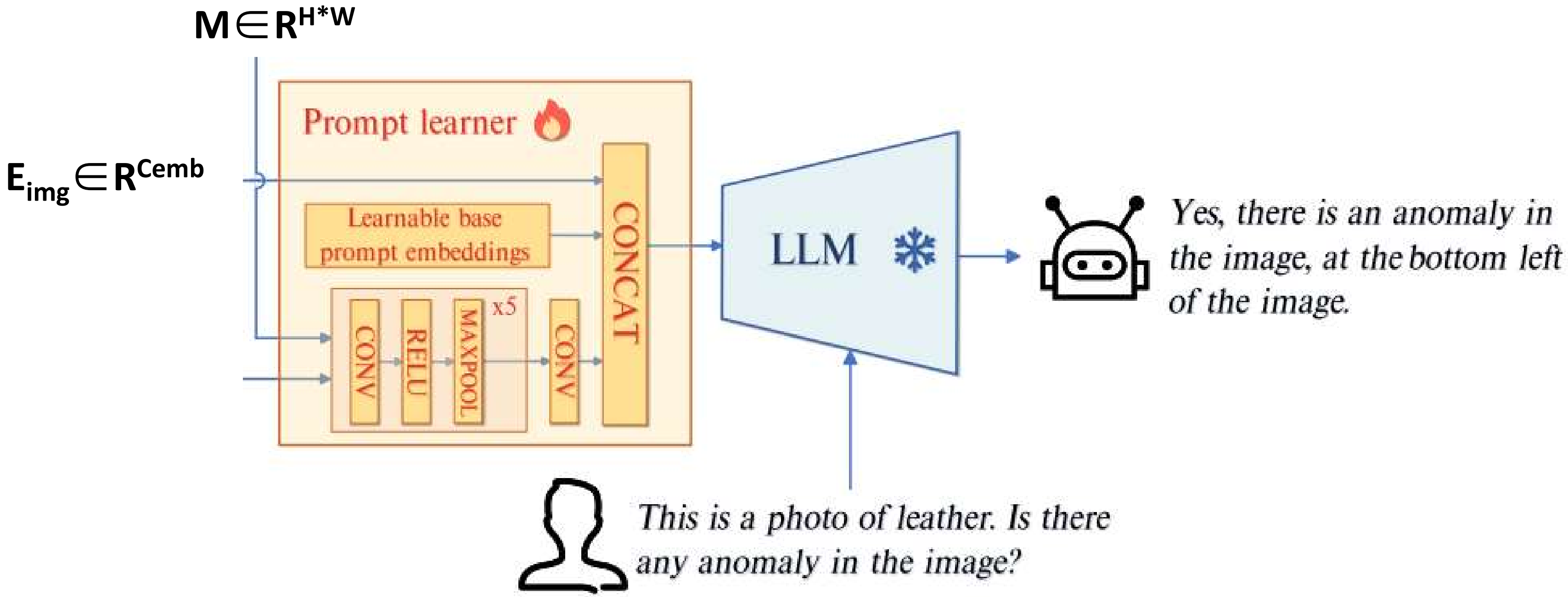
The localization result M can be obtained from the Eq. (1).

$$M = Upsample\left(\sum_{i=1}^{4} softmax(\tilde{F}^i_{patch}F^T_{text})\right). \quad (1)$$

Focal Loss: $L_{focal} = -\frac{1}{n}\sum_{i=1}^{n}(1-p_i)^{\gamma}log(p_i),$

Dice Loss: $L_{dice} = -\frac{\sum_{i=1}^{n} y_i\hat{y}_i}{\sum_{i=1}^{n} y_i^2 + \sum_{i=1}^{n}\hat{y}_i^2},$

Prompt learner



Learnable base prompt embeddings $E_{base} \in \mathbb{R}^{n_1 \times C_{emb}}$

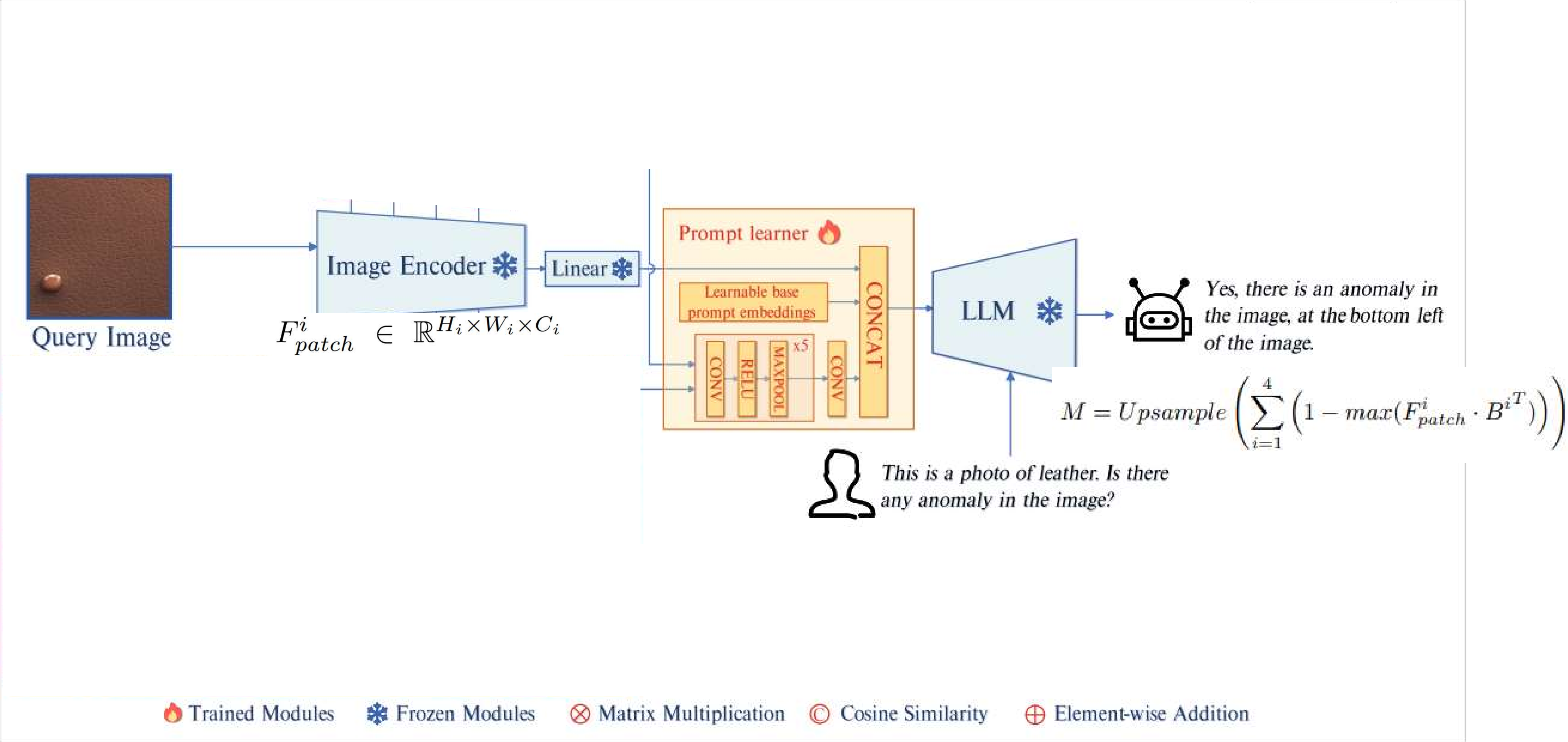
The network converts M into $E_{dec} \in \mathbb{R}^{n_2 \times C_{emb}}$

$$E_{prompt} \in \mathbb{R}^{(n_1+n_2) \times C_{emb}}$$

Prompts fed to the LLM typically follow the format: *### Human: <Img>E_{img}</Img>E_{prompt}[Image Description]Is there any anomaly in the image?###Assistant:*

$$L_{ce} = -\sum_{i=1}^{n} y_i log(p_i),$$

Method


Query Image
Image Encoder
Linear
$F^i_{patch} \in \mathbb{R}^{H_i \times W_i \times C_i}$
Prompt learner
Learnable base prompt embeddings
CONV
RELU
MAXPOOL
x5
CONV
CONCAT
LLM
Yes, there is an anomaly in the image, at the bottom left of the image.
$M = Upsample\left(\sum_{i=1}^{4}\left(1 - max(F^i_{patch} \cdot B^{i^T})\right)\right)$
This is a photo of leather. Is there any anomaly in the image?
Trained Modules
Frozen Modules
Matrix Multiplication
Cosine Similarity
Element-wise Addition

Experiments

Setup	Method	MVTec-AD			VisA		
		Image-AUC	Pixel-AUC	Accuracy	Image-AUC	Pixel-AUC	Accuracy
1-shot	SPADE	81.0 ± 2.0	91.2 ± 0.4	-	79.5 ± 4.0	95.6 ± 0.4	-
	PaDiM	76.6 ± 3.1	89.3 ± 0.9	-	62.8 ± 5.4	89.9 ± 0.8	-
	PatchCore	83.4 ± 3.0	92.0 ± 1.0	-	79.9 ± 2.9	95.4 ± 0.6	-
	WinCLIP	93.1 ± 2.0	95.2 ± 0.5	-	83.8 ± 4.0	**96.4 ± 0.4**	-
	AnomalyGPT (ours)	**94.1 ± 1.1**	**95.3 ± 0.1**	**86.1 ± 1.1**	**87.4 ± 0.8**	96.2 ± 0.1	**77.4 ± 1.0**
2-shot	SPADE	82.9 ± 2.6	92.0 ± 0.3	-	80.7 ± 5.0	96.2 ± 0.4	-
	PaDiM	78.9 ± 3.1	91.3 ± 0.7	-	67.4 ± 5.1	92.0 ± 0.7	-
	PatchCore	86.3 ± 3.3	93.3 ± 0.6	-	81.6 ± 4.0	96.1 ± 0.5	-
	WinCLIP	94.4 ± 1.3	**96.0 ± 0.3**	-	84.6 ± 2.4	**96.8 ± 0.3**	-
	AnomalyGPT (ours)	**95.5 ± 0.8**	95.6 ± 0.2	**84.8 ± 0.8**	**88.6 ± 0.7**	96.4 ± 0.1	**77.5 ± 0.3**
4-shot	SPADE	84.8 ± 2.5	92.7 ± 0.3	-	81.7 ± 3.4	96.6 ± 0.3	-
	PaDiM	80.4 ± 2.5	92.6 ± 0.7	-	72.8 ± 2.9	93.2 ± 0.5	-
	PatchCore	88.8 ± 2.6	94.3 ± 0.5	-	85.3 ± 2.1	96.8 ± 0.3	-
	WinCLIP	95.2 ± 1.3	96.2 ± 0.3	-	87.3 ± 1.8	**97.2 ± 0.2**	-
	AnomalyGPT (ours)	**96.3 ± 0.3**	**96.2 ± 0.1**	**85.0 ± 0.3**	**90.6 ± 0.7**	96.7 ± 0.1	**77.7 ± 0.4**

Table 2. Few-shot IAD results on MVTec-AD and VisA datasets. Results are listed as the average of 5 runs and the best-performing method is in **bold**. The results for SPADE, PaDiM, PatchCore and WinCLIP are reported from [11].

Method	Image-AUC	Pixel-AUC	Accuracy
PaDiM (Unified)	84.2	89.5	-
JNLD (Unified)	91.3	88.6	-
UniAD	96.5	**96.8**	-
AnomalyGPT (ours)	**97.4**	93.1	**93.3**

Table 3. Unsupervised anomaly detection results on MVTec-AD dataset. The best-performing method is in **bold** and the results for PaDiM and JNLD are reported from [35].

Experiments

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

Decoder	Prompt learner	LLM	LoRA	MVTec-AD (unsupervised)			VisA (1-shot)		
				Image-AUC	Pixel-AUC	Accuracy	Image-AUC	Pixel-AUC	Accuracy
		✓		-	-	72.2	-	-	56.5
	✓	✓		-	-	73.4	-	-	56.6
		✓	✓	-	-	79.8	-	-	63.4
✓		✓		97.1	90.9	72.2	85.8	96.2	56.5
✓		✓	✓	97.1	90.9	84.2	85.8	96.2	64.7
✓	✓	✓	✓	96.0	88.1	83.9	85.8	**96.5**	72.7
✓				97.1	90.9	90.3	85.8	96.2	75.4
✓	✓	✓		**97.4**	**93.1**	**93.3**	**87.4**	96.2	**77.4**

Table 4. Results of ablation studies. The ✓ in "Decoder" and "Prompt learner" columns indicate module inclusion. The ✓ in "LLM" column denotes whether use LLM for inference and the ✓ in "LoRA" column denotes whether use LoRA to fine-tune LLM. In settings without LLM, the maximum anomaly score from normal samples is used as the classification threshold. In settings without decoder, due to the sole textual output from the LLM, we cannot compute image-level and pixel-level AUC.

Qualitative Examples

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

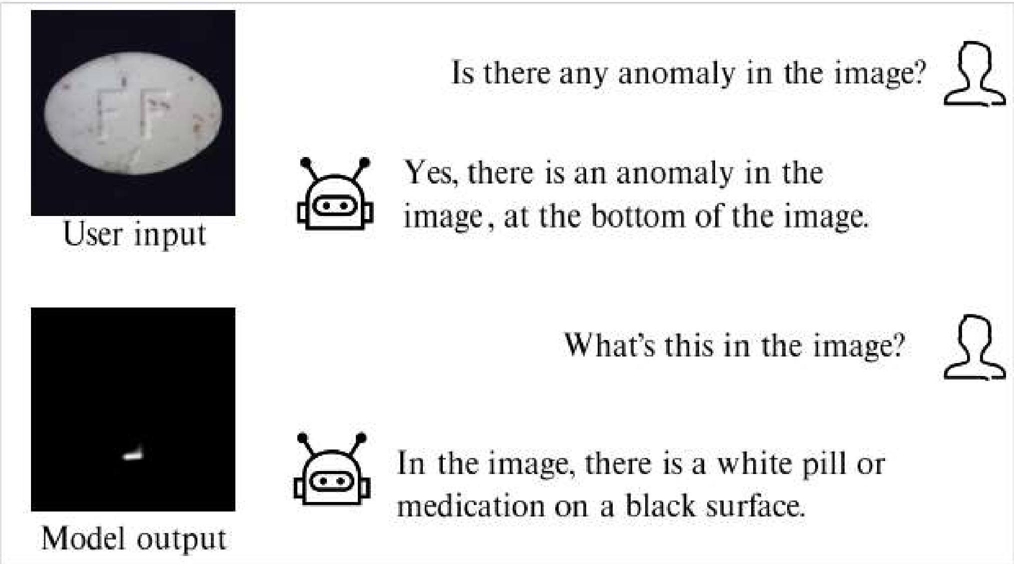


Figure 5. Qualitative example of AnomalyGPT in the unsupervised setting. AnomalyGPT is capable of detecting anomaly, pinpointing its location, providing pixel-level localization results and answering questions about the image.

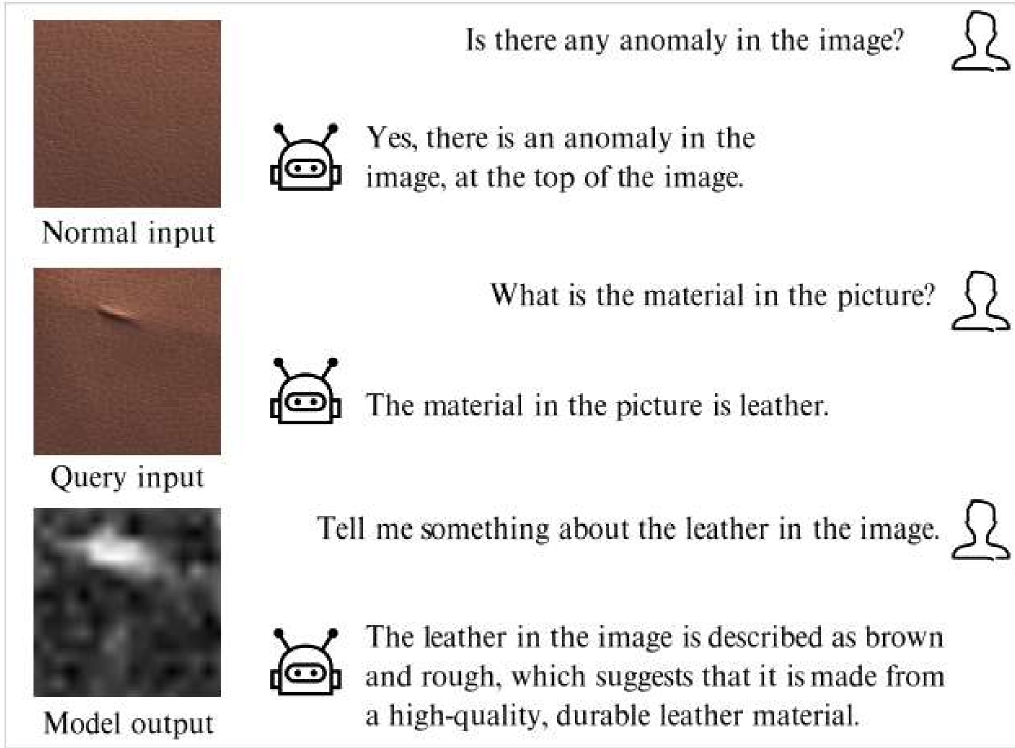


Figure 6. Qualitative example of AnomalyGPT in the **one-normal-shot** setting. The localization performance is slightly lower compared to the unsupervised setting due to the absence of parameter training.

Thanks