The 29th Meeting on Image Recognition and Understanding

SPG: Sparse-Projected Guides with Sparse Autoencoders for Zero-Shot Anomaly Detection

Tomoyasu NANAUMI[1,a)] Yukino TSUZUKI[1] Junichi OKUBO[1]
Junichiro FUJII[1] Takayoshi YAMASHITA[2]

Abstract

We propose Sparse-Projected Guides (SPG), a prompt-free framework for cross-dataset zero-shot anomaly detection and segmentation using frozen foundation-model features. SPG learns sparse guide coefficients on a labeled auxiliary dataset and generates normal/anomaly guide vectors through a sparse autoencoder dictionary, enabling transfer to unseen target categories without using target-dataset samples for training or fine-tuning. In cross-dataset evaluation on MVTec AD and VisA, SPG achieves reasonable image-level detection and strong pixel-level segmentation; with DINOv3, it attains the highest pixel-level AUROC among compared methods. The learned guide coefficients also allow atom-level inspection via high-coefficient anomaly-guide atoms.

1. Introduction

Industrial visual inspection requires both anomaly detection and anomaly segmentation, namely deciding whether an image is defective and localizing defective regions. Many recent methods achieve strong performance when training data are available for each target category [1], [15], but this assumption is often impractical in deployment. We therefore study cross-dataset zero-shot anomaly detection and segmentation, where all learnable parameters are trained only on a labeled auxiliary dataset and the resulting model is deployed on unseen target categories without using any target-dataset samples for training or fine-tuning. Foundation models such as CLIP [14] and DINO [16] have recently become common backbones for anomaly detection and segmentation, enabling decision criteria to be defined in frozen representation spaces. Most prior work, especially in CLIP-based settings, defines normal/anomaly references through text prompts, using either prompt engineering or auxiliary-data-driven prompt learning/adaptation [4], [6], [8], [10], [13], [17]. Beyond CLIP-style prompting, recent studies suggest that strong frozen visual backbones can also support anomaly localization with prompt-free criteria operating directly on visual patch tokens [5]. These observations motivate a prompt-free criterion defined directly on frozen visual features.

To this end, we propose Sparse-Projected Guides (SPG), a prompt-free framework for zero-shot anomaly detection and segmentation that uses guide vectors as an alternative to prompts. Our design is inspired by recent progress on Sparse Autoencoders (SAEs) [11], which map representations inside foundation models into a sparse, structured latent space over a learned dictionary [3], [7]. Building on this insight, we use the SAE dictionary to parameterize guide vectors as sparse, inspectable combinations of atoms, rather than as a single opaque reference embedding.

We summarize our main contributions as follows: (1) we introduce SPG, a prompt-free framework for cross-dataset zero-shot anomaly detection and segmentation that constructs normal/anomaly guide vectors via an SAE dictionary and its sparse latent space; because SPG uses only frozen patch tokens, it is compatible with both vision-language and vision-only backbones; (2) in cross-dataset evaluation on MVTec AD and VisA, SPG achieves strong pixel-level segmentation and reasonable image-level detection, and with DINOv3 attains the highest pixel-level AUROC in our comparisons; and (3) post hoc inspection of guide-emphasized SAE atoms suggests that the anomaly guide emphasizes both category-general and category-specific visual cues.

2. Method

2.1 Overview

We propose a two-stage learning framework on the frozen feature space of a pretrained visual encoder, as shown in Fig. 1. We parameterize normal/anomaly guide vectors by sparse coefficients in the SAE latent space. Stage 1 learns an SAE on patch tokens to obtain a dictionary that enables sparse decomposition of token representations. Stage 2 then freezes the SAE and learns the guide coefficients in the SAE latent space. The learned guide coefficients are converted into guide vectors through the frozen SAE dictionary and applied to a target dataset without further training.

2.1.1 Problem setting.

We study cross-dataset zero-shot anomaly detection and segmentation: all learnable parameters are optimized using a labeled auxiliary dataset, and the learned guides are deployed on a disjoint target dataset without any training

[1] Yachiyo Engineering Co., Ltd.
[2] Chubu University
[a)] tm-nanaumi@yachiyo-eng.co.jp

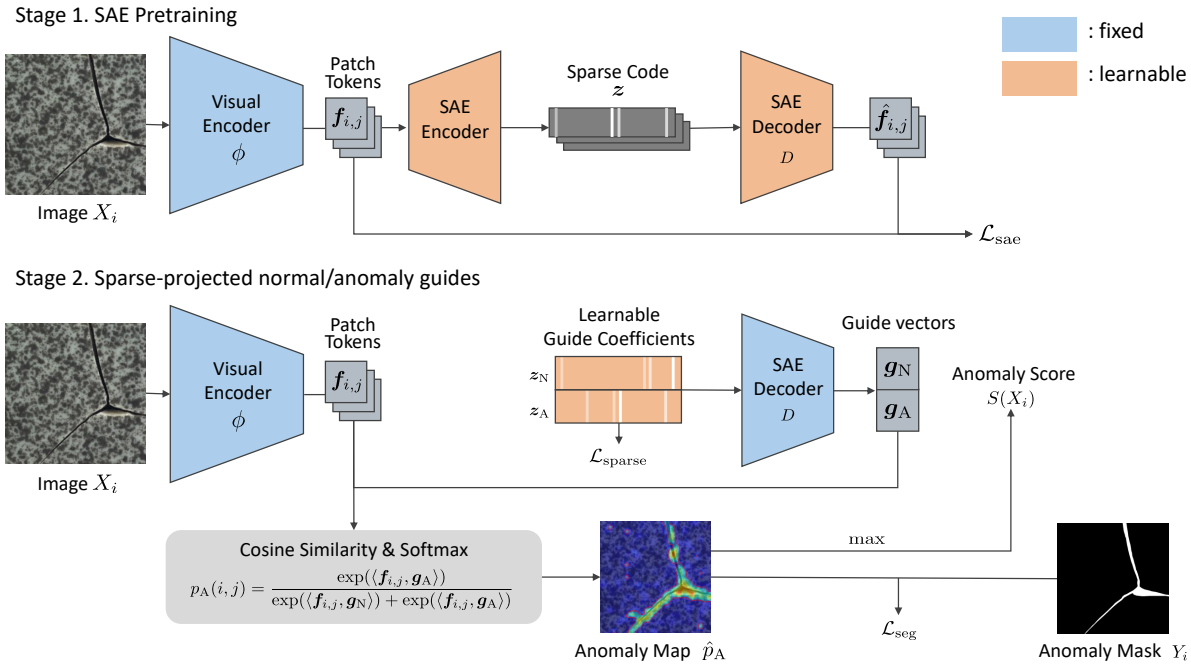


Fig. 1: Overview of SPG (Sparse-Projected Guides). Stage 1 trains a Sparse Autoencoder (SAE) on patch tokens extracted by a frozen visual encoder to learn a dictionary in which normal/anomaly guide vectors can be parameterized as sparse guide coefficients. Stage 2 freezes both the encoder and SAE, and learns only non-negative, sparsity-regularized guide coefficients that generate normal/anomaly guide vectors via the SAE dictionary. At inference, cosine similarities between patch tokens and the two guides are converted into an anomaly probability map via a two-class softmax, which is upsampled to the image resolution. Image-level anomaly scores are computed by aggregating the anomaly map (default: max pooling; see Eq. (8) for the log-sum-exp generalization with temperature τ).

or fine-tuning on target-dataset samples.

We denote a labeled auxiliary (training) dataset and a target dataset as $\mathcal{D}_{\text{train}} = \{(X_i, Y_i, y_i)\}_{i=1}^{N_{\text{train}}}$, $\mathcal{D}_{\text{target}} = \{(X_i, Y_i, y_i)\}_{i=1}^{N_{\text{target}}}$. Here, $X_i \in \mathbb{R}^{H\times W\times 3}$ is an image, $Y_i \in \{0,1\}^{H\times W}$ is a pixel-level anomaly mask, and $y_i \in \{0,1\}$ is an image-level anomaly label. The categories of $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{target}}$ are disjoint. All learnable parameters are trained only on $\mathcal{D}_{\text{train}}$ and applied to $\mathcal{D}_{\text{target}}$ without using any samples from $\mathcal{D}_{\text{target}}$ for training or fine-tuning; at inference, Y_i and y_i are used only for evaluation.

2.1.2 Frozen visual encoder and token features.

We use a pretrained visual encoder ϕ (e.g., the visual backbone of VLMs or self-supervised ViTs) and keep it frozen throughout all stages. Given an image X_i, the encoder outputs a sequence of patch tokens:

$$\phi(X_i) = \{\boldsymbol{f}_{i,j}\}_{j=1}^{m}, \quad \boldsymbol{f}_{i,j} \in \mathbb{R}^d, \tag{1}$$

where $m = H_p W_p$ is the number of patches.

2.1.3 Sparse autoencoder (SAE) as a sparse dictionary representation.

For each token feature $\boldsymbol{f} \in \mathbb{R}^d$, the SAE maps it to a higher-dimensional intermediate representation $\bar{\boldsymbol{z}} \in \mathbb{R}^C (C \gg d)$, applies a sparsification operator $\mathcal{S}(\cdot)$ to obtain a sparse code $\tilde{\boldsymbol{z}} \in \mathbb{R}^C$, and reconstructs the original feature using a linear dictionary $D = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_C] \in \mathbb{R}^{d\times C}$:

$$\bar{\boldsymbol{z}} = \text{Enc}(\boldsymbol{f}), \quad \tilde{\boldsymbol{z}} = \mathcal{S}(\bar{\boldsymbol{z}}), \quad \hat{\boldsymbol{f}} = \text{Dec}(\tilde{\boldsymbol{z}}) = D\tilde{\boldsymbol{z}}, \tag{2}$$

where $\text{Enc}(\cdot)$ is a single linear layer that maps from $\mathbb{R}^d$ to $\mathbb{R}^C$. We use TopK sparsification for $\mathcal{S}(\cdot)$ in Stage 1.

2.2 Stage 1: SAE pretraining on patch tokens.

In Stage 1, we keep the visual encoder ϕ fixed and train the SAE on patch tokens from $\mathcal{D}_{\text{train}}$. The goal is to represent token features as sparse linear combinations of an overcomplete dictionary.

For each patch token $\boldsymbol{f}_{i,j}$, we compute an intermediate code and apply TopK sparsification:

$$\bar{\boldsymbol{z}}_{i,j} = \text{Enc}(\boldsymbol{f}_{i,j}), \quad \tilde{\boldsymbol{z}}_{i,j} = \text{TopK}(\bar{\boldsymbol{z}}_{i,j}; k) \tag{3}$$

where $\text{TopK}(\cdot; k)$ keeps only the top-k entries and sets the others to zero. Reconstruction is given by $\hat{\boldsymbol{f}} = D\tilde{\boldsymbol{z}}_{i,j}$. We train the SAE by minimizing the reconstruction loss:

$$\mathcal{L}_{\text{sae}} = \frac{1}{N_{\text{train}} m} \sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{m} \|\boldsymbol{f}_{i,j} - D\tilde{\boldsymbol{z}}_{i,j}\|_2^2. \tag{4}$$

TopK-based sparsification explicitly controls the sparsity level compared to ℓ_1-regularized alternatives. After Stage 1, the learned dictionary D and encoder Enc are frozen and are not updated in later stages.

2.3 Stage 2: Sparse-projected normal/anomaly guide vectors in dictionary coordinates.

In Stage 2, we learn guide coefficients in the SAE latent space, which define normal/anomaly guide vectors via the SAE dictionary. While prior methods often use handcrafted or learned prompt embeddings as reference vectors,

our guides are expressed as sparse linear combinations of SAE dictionary atoms, thereby exposing a small set of active atoms for inspection.

Parameterization of guide coefficients. We introduce learnable guide coefficient parameters $\bar{\boldsymbol{z}}_N, \bar{\boldsymbol{z}}_A \in \mathbb{R}^C$ for the normal and anomaly guide vectors and apply an element-wise non-negativity operator $\rho(\cdot)$ to obtain non-negative coefficients:

$$\tilde{\boldsymbol{z}}_N = \rho(\bar{\boldsymbol{z}}_N), \quad \tilde{\boldsymbol{z}}_A = \rho(\bar{\boldsymbol{z}}_A). \tag{5}$$

The corresponding guide vectors in the d-dimensional token feature space are defined by the SAE dictionary:

$$\boldsymbol{g}_N = D\tilde{\boldsymbol{z}}_N, \quad \boldsymbol{g}_A = D\tilde{\boldsymbol{z}}_A. \tag{6}$$

Anomaly Scoring. For each patch token $\boldsymbol{f}_{i,j}$, we obtain the anomaly probability from its similarities to the normal and anomaly guides via a temperature-scaled two-class softmax:

$$p_A(i,j) = \frac{\exp(\langle \boldsymbol{f}_{i,j}, \boldsymbol{g}_A \rangle / T)}{\exp(\langle \boldsymbol{f}_{i,j}, \boldsymbol{g}_N \rangle / T) + \exp(\langle \boldsymbol{f}_{i,j}, \boldsymbol{g}_A \rangle / T)}, \tag{7}$$

where $\langle \cdot, \cdot \rangle$ denotes cosine similarity between ℓ_2-normalized vectors. Unless otherwise stated, $T = 0.07$. We then upsample p_A to the input resolution by bilinear interpolation to obtain the pixel-level anomaly map $\hat{p}_A \in [0,1]^{H \times W}$.

To obtain an image-level anomaly score, we use max pooling over $\hat{p}_A$ by default. More generally, max pooling is the $\tau \to 0$ limit of temperature-scaled log-sum-exp:

$$S_\tau(X_i) = \tau \log \left(\frac{1}{HW} \sum_{u,v} \exp\left(\frac{\hat{p}_A(u,v)}{\tau} \right) \right). \tag{8}$$

Smaller τ makes the aggregation max-like ($\tau \to 0$), whereas larger τ aggregates evidence more uniformly (mean-like); we analyze τ in the appendix.

Training. In Stage 2, we learn the guide coefficients from pixel-level anomaly masks Y in $\mathcal{D}_{\text{train}}$ while encouraging sparsity in the non-negative guide coefficients. Specifically, we optimize only the guide coefficient parameters, while freezing the visual encoder and SAE:

$$\min_{\bar{\boldsymbol{z}}_N, \bar{\boldsymbol{z}}_A} \mathcal{L}_{\text{seg}}(\hat{p}_A, Y) + \mathcal{L}_{\text{sparse}}(\bar{\boldsymbol{z}}_N, \bar{\boldsymbol{z}}_A), \tag{9}$$

where $\mathcal{L}_{\text{sparse}}(\bar{\boldsymbol{z}}_N, \bar{\boldsymbol{z}}_A) = \beta \left(\|\rho(\bar{\boldsymbol{z}}_N)\|_1 + \|\rho(\bar{\boldsymbol{z}}_A)\|_1 \right)$ encourages the guides to concentrate on a small set of dictionary atoms, and $\mathcal{L}_{\text{seg}}$ combines focal loss [9] and Dice loss [12]:

$$\mathcal{L}_{\text{seg}}(\hat{p}_A, Y) = \mathcal{L}_{\text{focal}}(\hat{p}_A, Y) + \lambda_{\text{dice}} \mathcal{L}_{\text{dice}}(\hat{p}_A, Y). \tag{10}$$

This design enables cross-dataset transfer: the learned guides can be applied to the target dataset without additional training.

3. Experiments

3.1 Experimental Setup

Datasets and protocol. We use MVTec AD [2] and VisA [18], two industrial anomaly benchmarks that provide image-level labels and pixel-level masks but differ in object categories and visual appearance. Following prior work, we evaluate in a cross-dataset zero-shot setting: one dataset is treated as an auxiliary dataset for training, and the other as a target dataset.

Implementation details. We use DINOv3 [16] (ViT-L)[*1] as the default image encoder. We also report results with OpenCLIP (ViT-L/14@336px)[*2] to compare with prior CLIP-based methods. All images are resized to 448 × 448 for DINOv3 and 518 × 518 for OpenCLIP, and patch tokens are extracted from the final (24th) transformer block, excluding the CLS token. In Stage 1, we train an SAE on auxiliary token features for 50 epochs with dictionary width $C = 4096$ and TopK sparsification $k = 32$. Appendix ablations show that the best (C, k) depends on the metric and transfer direction, so we use a single default setting in the main paper. In Stage 2, we learn normal/anomaly guide coefficients in the SAE latent space, and construct the corresponding guide vectors via the frozen SAE dictionary from Stage 1. We use ReLU for $\rho(\cdot)$ in Eq. (5) and the ℓ_1 penalty in Stage 2 as the default guide parameterization. Additional implementation details are provided in the appendix.

Metrics. We evaluate image-level detection with Area Under the Receiver Operating Characteristic curve (AUROC) and Average Precision (AP), where AUROC measures threshold-independent ranking and AP reflects precision–recall performance under class imbalance. For pixel-level segmentation, we report pixel-level AUROC and AUPRO [2], which evaluate pixel-wise ranking and region-level localization, respectively.

3.2 Quantitative Evaluation

We compare SPG with recent zero-shot anomaly detection and segmentation methods on MVTec AD and VisA under the cross-dataset setting. Table 1 reports image-level AUROC/AP and pixel-level AUROC/AUPRO. We report SPG (OpenCLIP) for a backbone-matched comparison with CLIP-based baselines and SPG (DINOv3) to evaluate the same prompt-free guide design with a vision-only backbone.

Under the backbone-matched OpenCLIP setting, SPG shows mixed performance across transfer directions: it is broadly comparable to several CLIP-based baselines on MVTec→VisA, but less competitive on VisA→MVTec. This suggests that SPG should be interpreted as a prompt-free alternative, not as a uniform improvement over prompt-based CLIP baselines under the same visual backbone.

With a vision-only DINOv3 backbone, SPG yields strong pixel-level segmentation and reasonable image-level detection. It achieves the highest pixel-level AUROC in both transfer directions among the compared methods, while its AUPRO on MVTec→VisA remains slightly below that of VCP-CLIP. Since SPG scores anomalies from frozen patch-token similarities to learned guides, the results suggest that

[*1] https://huggingface.co/facebook/dinov3-vitl16-pretrain-lvd1689m

[*2] https://github.com/mlfoundations/open_clip

Table 1: Cross-dataset zero-shot performance on MVTec AD and VisA (%). We report image-level AUROC / AP and pixel-level AUROC / AUPRO (higher is better). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Image-level (Detection)				
Method	VisA→MVTec		MVTec→VisA	
	AUROC	AP	AUROC	AP
WinCLIP	91.8	96.5	78.1	81.2
AnomalyCLIP	91.5	96.2	82.1	85.4
AdaCLIP	89.2	–	85.8	–
VCP-CLIP	92.1	96.9	83.8	87.6
AA-CLIP	90.5	–	84.6	–
AdaptCLIP	93.5	96.7	84.8	87.6
SPG (OpenCLIP)	79.3	89.6	82.6	85.7
SPG (DINOv3)	91.4	95.7	80.2	84.3
Pixel-level (Segmentation)				
Method	VisA→MVTec		MVTec→VisA	
	AUROC	AUPRO	AUROC	AUPRO
WinCLIP	85.1	64.6	79.6	56.8
AnomalyCLIP	91.1	81.4	95.5	87.0
AdaCLIP	88.7	–	95.5	–
VCP-CLIP	92.0	87.3	95.7	90.7
AA-CLIP	91.9	–	95.5	–
AdaptCLIP	90.9	–	95.7	–
SPG (OpenCLIP)	89.9	71.9	94.7	87.1
SPG (DINOv3)	92.3	87.7	96.0	89.1

localization is sensitive to the transferability of the underlying patch-token representation.

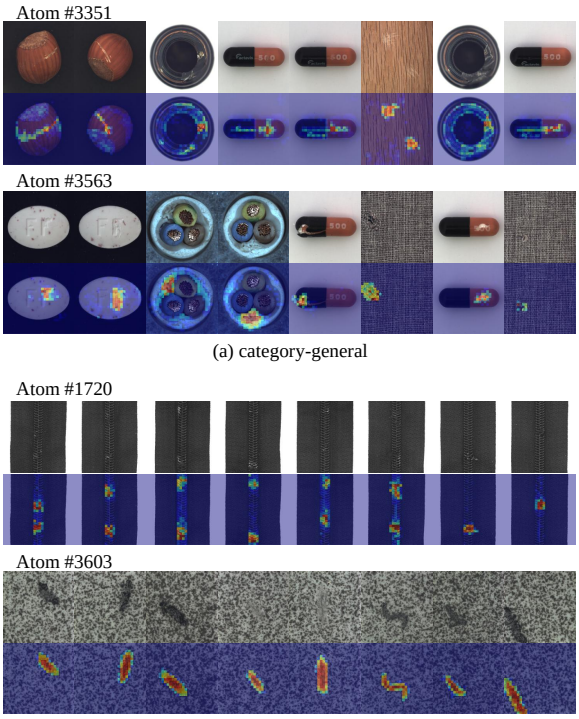
3.3 Qualitative analysis of SAE atoms

To interpret what the learned anomaly guide captures, we visualize representative SAE atoms from the anomaly guide.

Selecting atoms emphasized by the anomaly guide. Recall that the anomaly guide is defined by the SAE dictionary D and the non-negative guide coefficients $\tilde{\boldsymbol{z}}_A$ (Eq. (6)). Let $\mathcal{I}_A = \mathrm{supp}(\tilde{\boldsymbol{z}}_A)$ denote its active atom set. We select a small number of atoms in $\mathcal{I}_A$ with the largest coefficients $\tilde{\boldsymbol{z}}_A[a]$ as representative atoms emphasized by the learned anomaly guide.

Patch retrieval on the auxiliary dataset. To interpret each selected atom, we retrieve patches that strongly activate it on the auxiliary dataset used to train the SAE. For each auxiliary image X_i, we extract patch tokens $\boldsymbol{f}_{i,j}$ with the frozen encoder and compute SAE sparse codes $\tilde{\boldsymbol{z}}_{i,j}$ (Eqs. (1) to (3)). We define the activation of the atom a at patch (i,j) as $\tilde{\boldsymbol{z}}_{i,j}[a]$ and retrieve the top-K patches with the largest activation across the auxiliary dataset ($K = 8$ in our visualizations). For context, we also visualize an atom-activation heatmap by placing $\tilde{\boldsymbol{z}}_{i,j}[a]$ on the patch grid and upsampling it to the input resolution.

Observation. Figure 2 shows representative SAE atoms selected from the anomaly guide's active set. The atoms in Fig. 2(a) (e.g., #3351 and #3563) appear broadly activated by anomalous regions across multiple auxiliary categories, including cracks, missing parts, and other localized abnormal structures, suggesting category-general anomaly cues in a shared SAE latent space. In contrast, the atoms in Fig. 2(b) (e.g., #1720 and #3603) show more concentrated activations associated with category-specific local structures, such as object parts or characteristic surface patterns. This contrast supports our motivation that sparse guide coefficients can be inspected to reveal which components the anomaly guide relies on, rather than relying on a single opaque prompt embedding as the reference vector. We treat these visualizations as atom-level interpretability evidence, without claiming semantic disentanglement of the entire SAE dictionary.



(a) category-general

(b) category-specific

Fig. 2: Qualitative interpretation of SAE atoms emphasized by the anomaly guide. We visualize representative atoms from the anomaly guide's active set using top-activating patches from the auxiliary dataset and upsampled activation heatmaps. Atoms in (a) appear broadly activated across categories, whereas those in (b) are more category-specific.

4. Conclusion

We proposed SPG for cross-dataset zero-shot anomaly detection and segmentation, which constructs normal/anomaly guide vectors by projecting sparse coefficients in an SAE latent space back to the frozen token feature space. SPG follows a two-stage procedure on an auxiliary dataset: it first trains an SAE on patch tokens, then freezes it and optimizes only sparse guide coefficients. In cross-dataset evaluations on MVTec AD and VisA, SPG achieves strong pixel-level segmentation and reasonable image-level detection; with a DINOv3 backbone, it attains the best pixel-level AUROC among the compared methods in our study. Moreover, inspection of the dictionary atoms with large anomaly-guide coefficients suggests that a sparse subset of SAE atoms emphasizes both category-general and category-specific visual cues within a shared latent space.

References

[1] Batzner, K., Heckler, L. and König, R.: EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies, *WACV* (2024).

[2] Bergmann, P., Fauser, M., Sattlegger, D. and Steger, C.: MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection, *CVPR* (2019).

[3] Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T. and Olah, C.: Towards Monosemanticity: Decomposing Language Models With Dictionary Learning, *Transformer Circuits Thread* (2023). `https://transformer-circuits.pub/2023/monosemantic-features/index.html`.

[4] Cao, Y., Zhang, J., Frittoli, L., Cheng, Y., Shen, W. and Boracchi, G.: AdaCLIP: Adapting CLIP with Hybrid Learnable Prompts for Zero-Shot Anomaly Detection, *ECCV* (2024).

[5] Damm, S., Laszkiewicz, M., Lederer, J. and Fischer, A.: AnomalyDINO: Boosting Patch-based Few-shot Anomaly Detection with DINOv2, *WACV* (2025).

[6] Gao, B.-B., Zhu, Y., Yan, J., Cai, Y., Zhang, W., Wang, M., Liu, J., Liu, Y., Wang, L. and Wang, C.: AdaptCLIP: Adapting CLIP for Universal Visual Anomaly Detection, *AAAI* (2026).

[7] Gao, L., Dupré la Tour, T., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J. and Wu, J.: Scaling and Evaluating Sparse Autoencoders, *ICLR* (2025).

[8] Jeong, J., Zou, Y., Kim, T., Zhang, D., Ravichandran, A. and Dabeer, O.: WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation, *CVPR* (2023).

[9] Lin, T., Goyal, P., Girshick, R. B., He, K. and Dollár, P.: Focal Loss for Dense Object Detection, *ICCV* (2017).

[10] Ma, W., Zhang, X., Yao, Q., Tang, F., Wu, C., Li, Y., Yan, R., Jiang, Z. and Zhou, S. K.: AA-CLIP: Enhancing Zero-Shot Anomaly Detection via Anomaly-Aware CLIP, *CVPR* (2025).

[11] Makhzani, A. and Frey, B.: k-Sparse Autoencoders, *arXiv preprint arXiv:1312.5663* (2013).

[12] Milletari, F., Navab, N. and Ahmadi, S.-A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, *3DV* (2016).

[13] Qu, Z., Tao, X., Prasad, M., Shen, F., Zhang, Z., Gong, X. and Ding, G.: VCP-CLIP: A Visual Context Prompting Model for Zero-Shot Anomaly Segmentation, *ECCV* (2024).

[14] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. and Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision, *ICML* (2021).

[15] Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T. and Gehler, P.: Towards Total Recall in Industrial Anomaly Detection, *CVPR* (2022).

[16] Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P. and Bojanowski, P.: DINOv3, *arXiv preprint arXiv:2508.10104* (2025).

[17] Zhou, Q., Pang, G., Tian, Y., He, S. and Chen, J.: AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection, *ICLR* (2024).

[18] Zou, Y., Jeong, J., Pemula, L., Zhang, D. and Dabeer, O.: SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation, *ECCV* (2022).

SPG: Sparse-Projected Guides with Sparse Autoencoders for Zero-Shot Anomaly Detection

Supplementary Material

1. Implementation Details

We use DINOv3 [8] (ViT-L)[*1] as the default image encoder. We also report results with OpenCLIP (ViT-L/14@336px)[*2] to compare with prior CLIP-based methods. All images are resized to 448×448 for DINOv3 and 518×518 for OpenCLIP, and patch-token features are extracted from the final (24th) transformer block, excluding the CLS token. For both Stage 1 and Stage 2, we use the Adam [4] optimizer with learning rate 0.001 and batch size 16.

Stage 1: SAE pretraining. In Stage 1, we train an SAE on token features from the auxiliary dataset to obtain the dictionary vectors. We train for 50 epochs on the auxiliary dataset. The width of dictionary is set to $C = 4096$. We use a TopK sparsification operator $\mathcal{S} = \text{TopK}$ with $k = 32$.

Stage 2: Sparse-projected normal/anomaly guide vectors. In Stage 2, we learn normal/anomaly guide coefficients in the SAE latent space, and construct the corresponding guide vectors via the frozen SAE dictionary from Stage 1. To stabilize training, we apply exponential moving average (EMA) to the guide coefficient parameters $(\bar{z}_N, \bar{z}_A)$ with a decay coefficient of 0.999. We use ReLU as the non-negativity operator $\rho(\cdot)$ in Eq. (5) of the main paper. We set $\beta = 0.01$ in Eq. (9) for the sparsity regularizer and $\lambda_{\text{dice}} = 1.0$ in Eq. (10) of the main paper.

2. Additional Quantitative Analysis

To better understand which components drive these results and how robust the framework is in practice, we next analyze its sensitivity to several design choices. Specifically, we study: (i) SAE hyperparameters that control the dictionary capacity and sparsity of the codes, (ii) the choice of frozen visual backbone ϕ used to extract patch tokens, and (iii) the aggregation function used to convert the patch-level anomaly map into an image-level anomaly score. Unless otherwise stated, we keep all training schedules and losses fixed and change only the factor under study.

SAE hyperparameter selection. We evaluate the effect of the SAE dictionary width C and the TopK sparsity level k on cross-dataset zero-shot performance. Following Sec. 1, we sweep $k \in \{8, 16, 32, 64, 128\}$ and $C \in \{2048, 3072, 4096, 6144, 8192\}$, while keeping all other components fixed. Figure S1 summarizes the results for both transfer directions and metrics. Overall, the best (C, k) is both metric- and direction-dependent. For VisA→MVTec, image-level AUROC/AP peaks at $(C, k) = (2048, 16)$, while the pixel-level optimum is less consistent across metrics: pixel-level AUROC slightly increases with a wider dictionary around $C = 6144$ and $k = 32$–64, whereas pixel-level AUPRO attains its maximum at $(2048, 16)$. For MVTec→VisA, the same $(2048, 16)$ setting is not optimal, and pixel-level performance stays close across a broad range of (C, k). Overall, these results suggest that tuning (C, k) for a single metric or transfer direction may not generalize.

Aggregation for image-level anomaly scoring. Our default image-level score uses max pooling over the upsampled anomaly map, which corresponds to the $\tau \to 0$ limit of temperature-controlled log-sum-exp pooling. Figure S2 compares max/mean pooling with log-sum-exp pooling across temperatures τ. In both auxiliary→target directions, smaller τ (i.e., a more "max-like" aggregation) consistently improves image-level AUROC/AP, while larger τ approaches mean pooling and degrades performance. This suggests that image-level decisions benefit from preserving localized high-confidence responses in the anomaly map; in contrast, uniform averaging can dilute small or spatially sparse anomalies.

Effect of backbone choice. We analyze how the choice of visual backbone affects SPG under the same training and evaluation protocol. Specifically, we compare ViT-L backbones from OpenCLIP, SigLIP[*3] [9], DINOv2[*4] [6], and DINOv3 by rerunning Stage 1 (SAE training on auxiliary patch tokens) and Stage 2 (guide coefficient learning) for each backbone. Figure S3 summarizes the results for two cross-dataset settings. For VisA→MVTec, DINOv3 yields the best performance consistently across both image-level and pixel-level metrics (image AUROC/AP 91.4/95.7, pixel AUROC/AUPRO 92.3/87.7), followed by DINOv2. For MVTec→VisA, image-level detection is relatively close across backbones (AUROC ≈ 80–83; AP ≈ 84–87), while pixel-level segmentation shows clearer differences, with DINOv3 achieving the highest pixel AU-

[*1] https://huggingface.co/facebook/dinov3-vitl16-pretrain-lvd1689m
[*2] https://github.com/mlfoundations/open_clip
[*3] https://huggingface.co/google/siglip-large-patch16-384
[*4] https://huggingface.co/facebook/dinov2-large

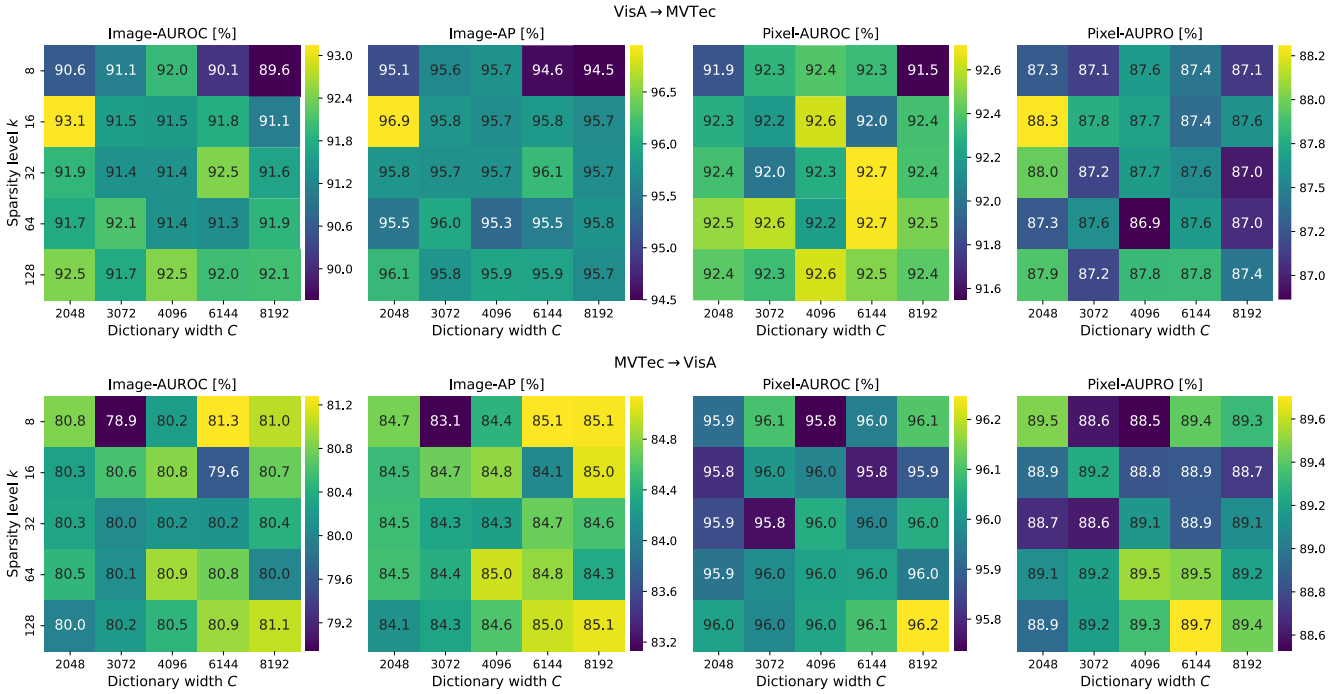


Fig. S1: Sensitivity to SAE hyperparameters. The Heatmaps summarize cross-dataset performance when sweeping the SAE dictionary width C and the TopK sparsity level k. The top row reports VisA → MVTec AD transfer and the bottom row reports MVTec → VisA transfer. Each column corresponds to image-level AUROC / AP and pixel-level AUROC / AUPRO, respectively, highlighting that the optimal (C, k) can depend on both the transfer direction and the evaluation metric.

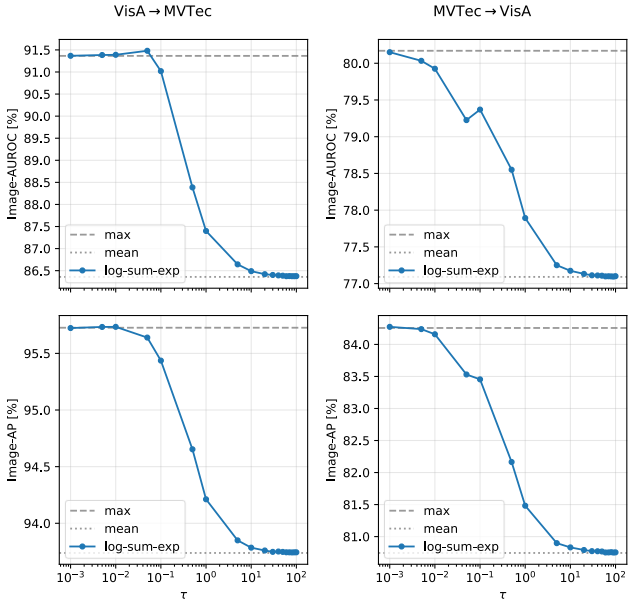


Fig. S2: Ablation of image-level anomaly score aggregation via temperature-controlled log-sum-exp pooling. We vary the temperature τ in log-sum-exp pooling applied to the upsampled anomaly map to produce an image-level anomaly score. Curves report image-level AUROC and AP for both transfer directions. The figure compares the $\tau \to 0$ (max-like) regime against larger τ (mean-like) aggregation, showing that more max-like pooling better preserves localized high-confidence anomaly responses for detection.

ROC/AUPRO (96.0/89.1) and SigLIP-L being notably lower in AUPRO (81.8). Overall, SPG is compatible with both self-supervised ViTs and VLM-based visual encoders, while the choice of backbone can materially influence segmentation quality.

3. TopK Projection for Stage 2 Guide Learning

The main paper parameterizes Stage 2 guide coefficients using ReLU together with an ℓ_1 sparsity regularizer. Here, we study an alternative based on explicit TopK sparsification to assess how the choice of sparsity mechanism affects guide learning.

Let $\bar{\boldsymbol{z}}_N, \bar{\boldsymbol{z}}_A \in \mathbb{R}^C$ denote learnable coefficient parameters for the normal/anomaly guides, respectively. We obtain k_g-sparse coefficients by

$$\tilde{\boldsymbol{z}}_N = \mathrm{TopK}(\bar{\boldsymbol{z}}_N, k_g), \quad \tilde{\boldsymbol{z}}_A = \mathrm{TopK}(\bar{\boldsymbol{z}}_A, k_g), \tag{1}$$

where $\mathrm{TopK}(\cdot; k_g)$ keeps the k_g largest entries and sets the remaining entries to zero. Using the frozen SAE dictionary $\boldsymbol{D}$, the resulting guides are computed as $\boldsymbol{g}_N = \boldsymbol{D}\tilde{\boldsymbol{z}}_N$ and $\boldsymbol{g}_A = \boldsymbol{D}\tilde{\boldsymbol{z}}_A$.

We sweep $k_g \in \{2^3, 2^4, \ldots, 2^{10}\}$ under the same cross-dataset protocol as in the main paper, namely VisA→MVTec and MVTec→VisA. For fair comparison with the ReLU+ℓ_1 formulation, we plot performance against the total guide sparsity $\|\tilde{\boldsymbol{z}}_N\|_0 + \|\tilde{\boldsymbol{z}}_A\|_0$. For the TopK formulation, this corresponds to $2k_g$. We evaluate performance using Image AUROC, Image AP, Pixel AUROC, and Pixel

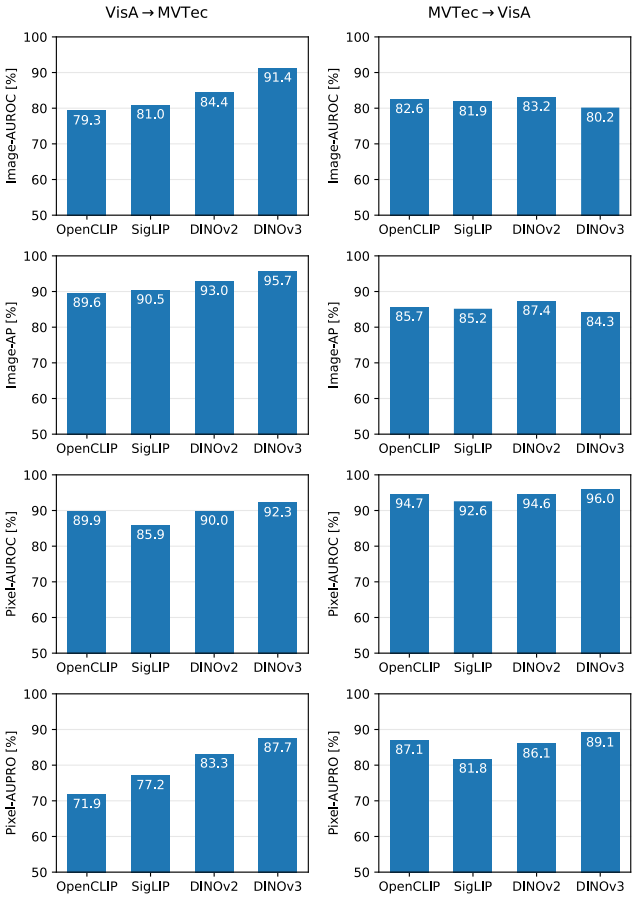


Fig. S3: Effect of the visual backbone on SPG under cross-dataset transfer. We instantiate SPG with different frozen encoders and rerun Stage 1 (SAE training on auxiliary patch tokens) and Stage 2 (guide-coefficient learning) for each backbone under the same protocol. Results are reported for two transfer directions (left: VisA → MVTec; right: MVTec → VisA). Each plot reports image-level AUROC/AP and pixel-level AUROC/AUPRO, isolating how backbone choice influences both detection and segmentation quality within the same guide-based scoring rule.

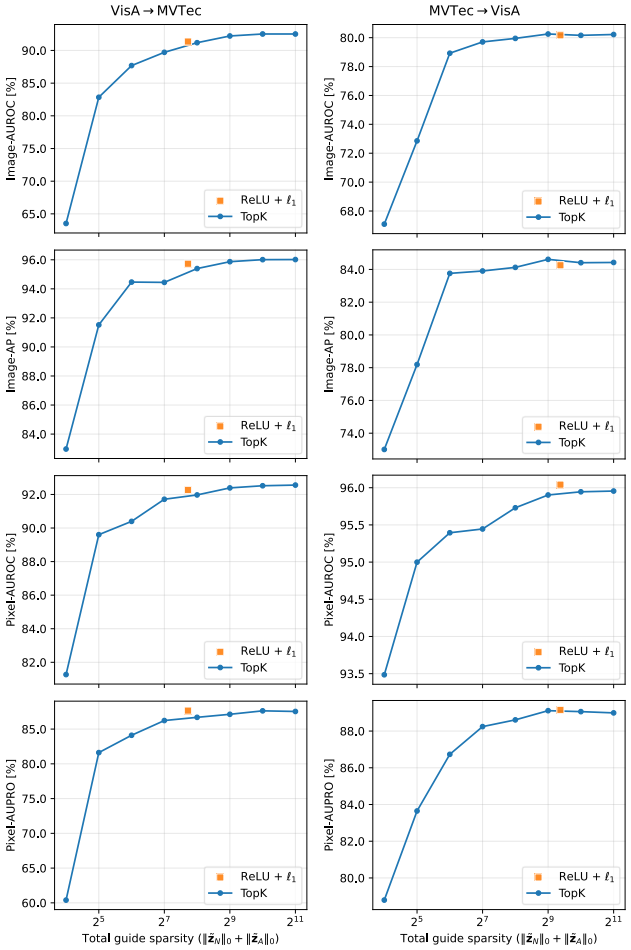


Fig. S4: Stage 2 guide learning with explicit TopK sparsification. We replace the ReLU+ℓ_1 parameterization in the main paper with k_g-sparse guide coefficients obtained by TopK projection, and plot performance against the total guide sparsity $\|\tilde{\boldsymbol{z}}_N\|_0 + \|\tilde{\boldsymbol{z}}_A\|_0$. Orange markers denote the effective sparsity obtained under the ReLU+ℓ_1 formulation in the main paper. In both transfer directions, performance improves as the sparsity budget increases, indicating that overly small TopK budgets over-constrain the guides.

AUPRO in Fig. S4.

Performance improves consistently as the total TopK sparsity increases, and sufficiently large sparsity can match or slightly surpass the ReLU+ℓ_1 formulation used in the main paper. At the same time, the effective sparsity obtained under the ReLU+ℓ_1 formulation differs substantially across transfer directions: the total number of non-zero normal/anomaly guide coefficients is 211 for VisA→MVTec and 644 for MVTec→VisA, respectively, for a dictionary width of 4096. This observation suggests that different transfer directions may favor different effective sparsity levels. In this sense, a fixed TopK constraint may be less flexible than the ReLU+ℓ_1 formulation, which provides a softer and more continuous sparsity control.

4. Per-Category Performance

This section reports category-level results under cross-dataset transfer. Following the main paper, we compare against WinCLIP [3], AnomalyCLIP [10], AdaCLIP [1], VCP-CLIP [7], AA-CLIP [5], and AdaptCLIP [2].

Tables S1 to S4 summarize VisA→MVTec AD results, including Image AUROC, Image AP, Pixel AUROC, and Pixel AUPRO, respectively. Tables S5 to S8 report the corresponding results for MVTec AD→VisA. Entries marked with "–" indicate that category-wise values were not reported in the corresponding original paper. When only dataset-level means were reported, we retain those mean values in the last row for reference.

References

[1] Cao, Y., Zhang, J., Frittoli, L., Cheng, Y., Shen, W. and Boracchi, G.: AdaCLIP: Adapting CLIP with Hybrid Learnable Prompts for Zero-Shot Anomaly Detection, *ECCV* (2024).

[2] Gao, B.-B., Zhu, Y., Yan, J., Cai, Y., Zhang, W., Wang, M., Liu, J., Liu, Y., Wang, L. and Wang, C.: AdaptCLIP: Adapting CLIP for Universal Visual Anomaly Detection, *AAAI* (2026).

Table S1: Category-wise image-level AUROC (%) under the cross-dataset zero-shot setting (VisA → MVTec AD). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Bottle	99.2	89.3	94.4	–	–	90.7	78.7	99.3
Cable	86.5	69.8	90.6	–	–	83.6	57.3	86.4
Capsule	72.9	89.9	91.5	–	–	95.2	90.0	89.9
Carpet	100.0	100.0	82.1	–	–	100.0	92.3	97.8
Grid	98.8	97.0	90.0	–	–	98.9	76.2	99.4
Hazelnut	93.9	97.2	80.2	–	–	94.8	84.7	93.5
Leather	100.0	99.8	99.8	–	–	100.0	97.2	100.0
Metal nut	97.1	93.6	83.5	–	–	93.7	69.8	97.5
Pill	79.1	81.8	82.9	–	–	86.8	87.9	74.5
Screw	83.3	81.1	87.0	–	–	85.9	69.4	86.9
Tile	100.0	100.0	90.5	–	–	99.6	87.6	95.1
Toothbrush	87.5	84.7	93.6	–	–	86.4	62.2	84.7
Transistor	88.0	92.8	82.1	–	–	88.7	72.6	66.5
Wood	99.4	96.8	98.3	–	–	98.7	73.5	99.1
Zipper	91.5	98.5	91.5	–	–	99.3	90.8	99.7
Mean	91.8	91.5	89.2	92.1	90.5	93.5	79.3	91.4

Table S2: Category-wise image-level AP (%) under the cross-dataset zero-shot setting (VisA → MVTec AD). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Bottle	99.8	97.0	–	–	–	97.1	92.6	99.8
Cable	91.2	81.4	–	–	–	89.9	67.8	92.4
Capsule	91.5	97.9	–	–	–	99.0	97.8	98.0
Carpet	100.0	100.0	–	–	–	99.9	97.2	99.2
Grid	99.6	99.1	–	–	–	99.5	88.9	99.8
Hazelnut	96.9	98.6	–	–	–	97.0	90.5	97.1
Leather	100.0	99.9	–	–	–	99.9	99.0	100.0
Metal nut	99.3	98.5	–	–	–	98.3	91.9	99.4
Pill	95.7	95.4	–	–	–	97.0	97.6	94.8
Screw	93.1	92.5	–	–	–	93.8	88.0	95.7
Tile	100.0	100.0	–	–	–	99.7	90.7	98.4
Toothbrush	95.6	93.7	–	–	–	92.5	84.7	94.7
Transistor	87.1	90.6	–	–	–	87.1	69.2	67.0
Wood	99.8	99.2	–	–	–	99.5	91.0	99.7
Zipper	97.5	99.6	–	–	–	99.8	97.7	99.9
Mean	96.5	96.2	–	96.9	–	96.7	89.6	95.7

[3] Jeong, J., Zou, Y., Kim, T., Zhang, D., Ravichandran, A. and Dabeer, O.: WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation, *CVPR* (2023).

[4] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *ICLR* (2015).

[5] Ma, W., Zhang, X., Yao, Q., Tang, F., Wu, C., Li, Y., Yan, R., Jiang, Z. and Zhou, S. K.: AA-CLIP: Enhancing Zero-Shot Anomaly Detection via Anomaly-Aware CLIP, *CVPR* (2025).

[6] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A. and Bojanowski, P.: DINOv2: Learning Robust Visual Features without Supervision, *Transactions on Machine Learning Research* (2024).

[7] Qu, Z., Tao, X., Prasad, M., Shen, F., Zhang, Z., Gong, X. and Ding, G.: VCP-CLIP: A Visual Context Prompting Model for Zero-Shot Anomaly Segmentation, *ECCV* (2024).

[8] Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P. and Bojanowski, P.: DINOv3, *arXiv preprint arXiv:2508.10104* (2025).

[9] Zhai, X., Mustafa, B., Kolesnikov, A. and Beyer, L.: Sigmoid Loss for Language Image Pre-Training, *ICCV* (2023).

[10] Zhou, Q., Pang, G., Tian, Y., He, S. and Chen, J.: AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection, *ICLR* (2024).

Table S3: Category-wise pixel-level AUROC (%) under the cross-dataset zero-shot setting (VisA → MVTec AD). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Bottle	89.5	90.4	90.4	94.1	–	92.4	89.5	96.6
Cable	77.0	78.9	79.8	78.6	–	76.6	74.2	80.4
Capsule	86.9	95.8	82.3	96.6	–	95.8	93.7	95.6
Carpet	95.4	98.8	97.3	99.6	–	99.2	98.6	99.5
Grid	82.2	97.3	96.9	97.9	–	97.4	96.9	99.3
Hazelnut	94.3	97.1	97.8	97.7	–	97.6	94.6	97.7
Leather	96.7	98.6	99.2	99.6	–	99.1	99.1	99.7
Metal nut	61.0	74.4	74.3	74.0	–	76.5	77.0	87.0
Pill	80.0	92.0	86.4	90.1	–	89.9	87.0	85.9
Screw	89.6	97.5	98.4	98.5	–	97.9	98.2	98.0
Tile	77.6	94.6	88.5	93.5	–	95.8	91.8	89.2
Toothbrush	86.9	91.9	94.9	94.7	–	87.7	93.2	94.2
Transistor	74.7	71.0	63.2	69.8	–	69.3	66.1	65.9
Wood	93.4	96.5	87.9	97.6	–	96.9	93.4	95.9
Zipper	91.6	91.4	93.8	98.1	–	91.8	95.2	99.1
Mean	85.1	91.1	88.7	92.0	91.9	90.9	89.9	92.3

Table S4: Category-wise pixel-level AUPRO (%) under the cross-dataset zero-shot setting (VisA → MVTec AD). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Bottle	76.4	80.9	–	88.5	–	–	72.4	91.4
Cable	42.9	64.4	–	67.7	–	–	39.4	73.7
Capsule	62.1	87.2	–	94.0	–	–	70.5	86.0
Carpet	84.1	90.1	–	98.4	–	–	85.7	98.2
Grid	57.0	75.6	–	92.4	–	–	75.6	94.7
Hazelnut	81.6	92.4	–	84.9	–	–	78.1	86.4
Leather	91.1	92.2	–	99.2	–	–	91.2	98.5
Metal nut	31.8	71.0	–	73.4	–	–	67.4	84.1
Pill	65.0	88.2	–	94.9	–	–	80.3	89.6
Screw	68.5	88.0	–	93.2	–	–	81.6	86.3
Tile	51.2	87.6	–	90.6	–	–	69.3	85.2
Toothbrush	67.7	88.5	–	87.8	–	–	83.6	88.6
Transistor	43.4	58.1	–	56.1	–	–	45.0	60.8
Wood	74.1	91.2	–	95.7	–	–	82.5	95.4
Zipper	71.7	65.3	–	92.2	–	–	55.6	96.0
Mean	64.6	81.4	–	87.3	–	–	71.9	87.7

Table S5: Category-wise image-level AUROC (%) under the cross-dataset zero-shot setting (MVTec AD → VisA). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Candle	95.4	79.3	96.0	–	–	87.4	84.5	74.7
Capsules	85.0	81.5	85.1	–	–	93.9	92.4	95.8
Cashew	92.1	76.3	91.8	–	–	86.1	84.3	70.2
Chewinggum	96.5	97.4	96.4	–	–	96.7	96.2	97.7
Fryum	80.3	93.0	93.0	–	–	91.5	91.9	94.9
Macaroni1	76.2	87.2	91.6	–	–	83.3	77.3	72.7
Macaroni2	63.7	73.4	64.1	–	–	69.9	61.2	64.2
PCB1	73.6	85.4	81.1	–	–	84.1	75.6	65.9
PCB2	51.2	62.2	75.3	–	–	64.6	69.9	81.8
PCB3	73.4	62.7	64.7	–	–	65.3	66.4	53.3
PCB4	79.6	93.9	93.4	–	–	97.7	93.1	91.1
Pipe fryum	69.7	92.4	96.6	–	–	96.7	98.0	99.6
Mean	78.1	82.1	85.8	84.6	82.6	84.8	82.6	80.2

Table S6: Category-wise image-level AP (%) under the cross-dataset zero-shot setting (MVTec AD → VisA). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Candle	95.8	81.1	–	–	–	90.5	88.0	81.2
Capsules	90.9	88.7	–	–	–	96.8	96.3	98.1
Cashew	96.4	89.4	–	–	–	94.0	93.0	84.0
Chewinggum	98.6	98.9	–	–	–	98.7	98.1	99.1
Fryum	90.1	96.8	–	–	–	95.9	96.4	97.7
Macaroni1	75.8	86.0	–	–	–	85.1	82.5	77.8
Macaroni2	60.3	72.1	–	–	–	69.2	61.2	63.8
PCB1	78.4	87.0	–	–	–	86.1	80.7	73.8
PCB2	49.2	64.3	–	–	–	67.8	70.7	85.6
PCB3	76.5	70.0	–	–	–	72.0	69.7	58.3
PCB4	77.7	94.4	–	–	–	97.5	93.0	92.0
Pipe fryum	82.3	96.3	–	–	–	98.0	99.0	99.8
Mean	81.2	85.4	–	87.6	–	87.6	85.7	84.3

Table S7: Category-wise pixel-level AUROC (%) under the cross-dataset zero-shot setting (MVTec AD → VisA). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Candle	88.9	98.8	98.9	99.2	–	98.7	98.8	98.9
Capsules	81.6	95.0	98.6	98.7	–	94.2	95.6	99.4
Cashew	84.7	93.8	95.9	93.1	–	93.4	91.7	96.2
Chewinggum	93.3	99.3	99.6	99.5	–	99.5	99.5	99.5
Fryum	88.5	94.6	94.4	94.6	–	95.3	92.0	94.7
Macaroni1	70.9	98.3	99.5	99.6	–	98.1	98.4	98.4
Macaroni2	59.3	97.6	98.8	98.7	–	98.2	98.1	98.1
PCB1	61.2	94.1	93.7	92.1	–	96.2	92.4	91.3
PCB2	71.6	92.4	84.3	92.1	–	93.0	89.5	94.6
PCB3	85.3	88.4	91.8	89.2	–	87.9	90.8	89.1
PCB4	94.4	95.7	96.1	95.6	–	95.3	95.3	95.1
Pipe fryum	75.4	98.2	94.6	96.4	–	98.4	94.0	97.2
Mean	79.6	95.5	95.5	95.7	95.5	95.7	94.7	96.0

Table S8: Category-wise pixel-level AUPRO (%) under the cross-dataset zero-shot setting (MVTec AD → VisA). Baseline results are taken from the original papers, and "–" denotes metrics not reported.

Category	WinCLIP	AnomalyCLIP	AdaCLIP	VCP-CLIP	AA-CLIP	AdaptCLIP	SPG (OpenCLIP)	SPG (DINOv3)
Candle	83.5	96.2	–	96.2	–	–	96.5	89.9
Capsules	35.3	78.5	–	91.2	–	–	81.8	98.6
Cashew	76.4	91.6	–	95.6	–	–	93.8	88.7
Chewinggum	70.4	91.2	–	92.2	–	–	94.8	92.8
Fryum	77.4	86.8	–	92.2	–	–	90.6	92.9
Macaroni1	34.3	89.8	–	97.5	–	–	90.2	91.7
Macaroni2	21.4	84.2	–	90.5	–	–	83.5	91.1
PCB1	26.3	81.7	–	88.1	–	–	83.7	86.4
PCB2	37.2	78.9	–	79.1	–	–	72.3	80.2
PCB3	56.1	77.1	–	78.9	–	–	78.2	73.1
PCB4	80.4	91.3	–	89.7	–	–	84.7	86.6
Pipe fryum	82.3	96.8	–	96.5	–	–	95.2	97.7
Mean	56.8	87.0	–	90.7	–	–	87.1	89.1