Exploring CLIP's Dense Knowledge for Weakly Supervised Semantic Segmentation

Zhiwei Yang[1,2] Yucong Meng[2,3]
Kexue Fu[4] Feilong Tang[1] Shuo Wang[2,3*] Zhijian Song[1,2,3*]
[1]Academy for Engineering and Technology, Fudan University, Shanghai 200433, China
[2]Shanghai Key Laboratory of Medical Image Computing and Computer Assisted Intervention
[3]Digital Medical Research Center, School of Basic Medical Sciences, Fudan University, China
[4]Shandong Computer Science Center (National Supercomputer Center in Jinan)

- Problem / objective
  - Weakly Supervised Semantic Segmentation (WSSS) via CLIP
    - Image-Text Alignment
- Contribution / Key idea
  - Weakly Supervised Semantic Segmentation (WSSS) via CLIP
    - Patch-Text Alignment

- **Weakly Supervised Semantic Segmentation (WSSS)**

❑ **Definition**

- Generate pixel-level predictions using weak annotations like points, scribbles, bounding boxes, or **image-level labels**

Ours

❑ **WSSS 3-stage Pipeline**

1. Generate Class Activation Maps (**CAMs**) by training a classification network
2. Refine CAMs into pseudo labels (**PL**)
3. Use these labels to **train** a segmentation model

➢ Limitation: CAMs intend to highlight the most distinctive object parts, due to the minimal semantic information from image-level labels, significantly limiting WSSS performance.

❑ **WSSS via CLIP**

➢ Limitation: Current methods primarily focus on CLIP's global **image-text alignment**, as shown in Fig. 1 (a). CLIP's dense knowledge with **patch-text alignment** still remains under-explored in WSSS.

Motivation

Motivation

- **ExCEL**: Explore CLIP's dense knowledge via a **patch-text alignment** paradigm for WSSS, i.e., **generating CAMs by calculating patch-wise similarity between text and individual patch tokens**, as shown in Fig. 1 (b).
- **Two key challenges:**
  1. Semantic sparsity in **textual prompts**
     : The template 'a photo of [CLASS]' only indicates object presence but lacks knowledge for localization.
  2. Fine-grained insufficiency in **visual features**
     : CLIP prioritizes global representation due to its image-text pairing nature.
- **Our proposed solution:**
  1. Text Semantic Enrichment (**TSE**) module
  2. Visual Calibration (**VC**) module

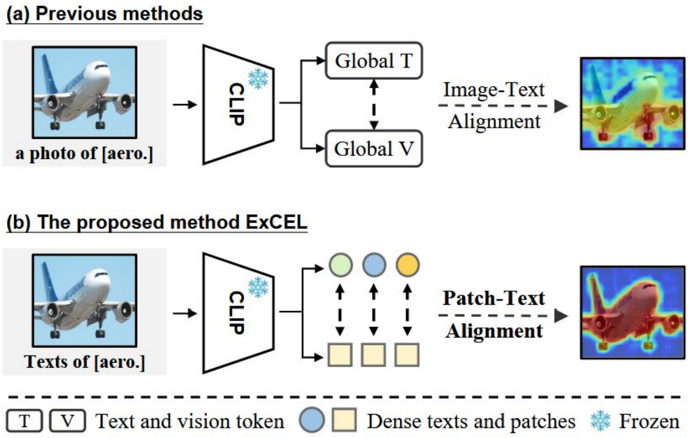


Figure 1. Our motivation. (a) Previous methods leverage CLIP to generate CAMs with global image-text alignment, leaving CLIP's dense knowledge unexplored. (b) The proposed ExCEL explores CLIP's dense knowledge via a novel patch-text alignment paradigm, which generates better CAMs with less training cost.

전유진

YANG, Zhiwei, et al. Exploring CLIP's Dense Knowledge for Weakly Supervised Semantic Segmentation. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025. p. 20223-20232.

CVPR 2025

- **Overview**

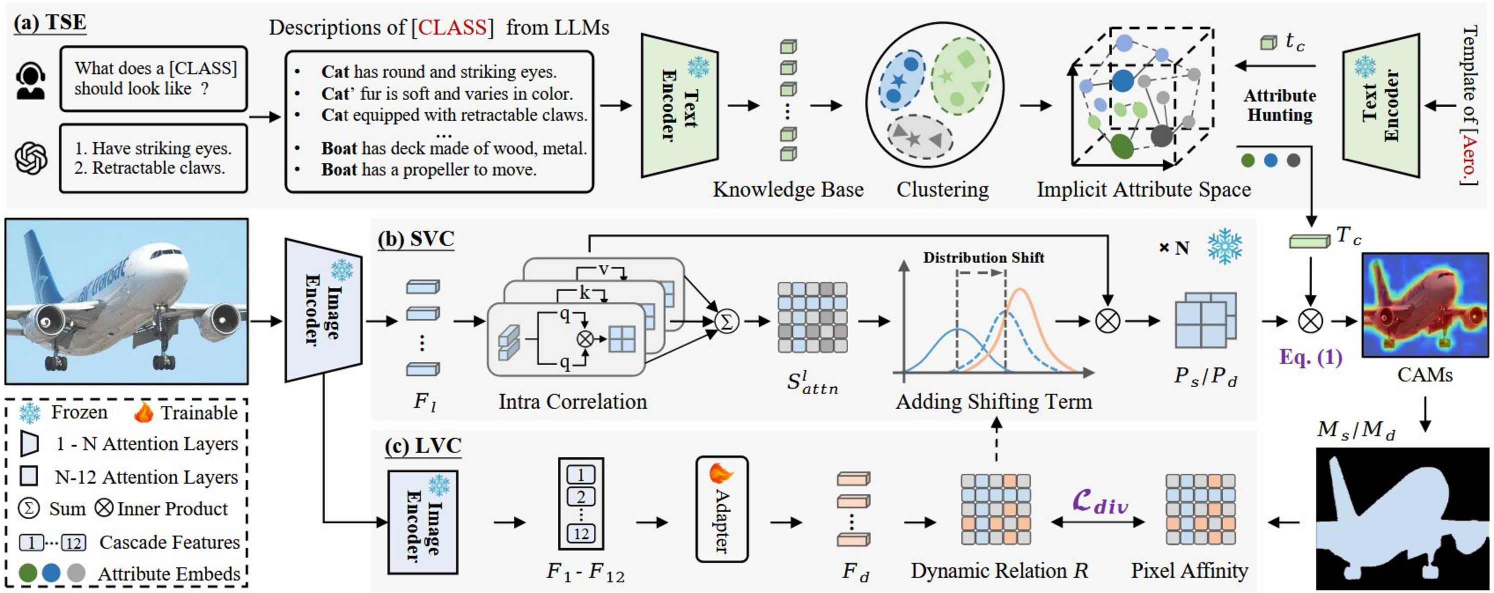


Figure 2. ExCEL Architecture. We explore CLIP's dense knowledge with Text Semantic Enrichment (TSE) and Visual Calibration (VC). (a) TSE uses LLMs to build a knowledge base and clusters it into an implicit attribute space. The final text representation T_c is enhanced by hunting for relevant attributes. For vision modality, (b) we introduce Static Visual Calibration (SVC) to calibrate visual features using the Inter-correlation operation across N intermediate layers. It generates static CAMs with T_c and calibrated features P_s. (c) Learnable Visual Calibration (LVC) designs a learnable adapter to add a dynamic shift R to SVC. It generates optimized features P_d based on static CAMs guidance, creating dynamic CAMs from P_d and T_c. Dynamic CAMs are refined for segmentation supervision. Details are in Sec. 3.1.

전유진

Preliminaries

Patch-text CAM Generation

- Visual features, Text embeddings: $P \in \mathbb{R}^{h \times w \times D}$ $T \in \mathbb{R}^{D \times C}$
- CAM: generated by calculating the patch-wise similarities between text and visual features

$$\mathrm{CAM} = \mathrm{Norm}\left(\cos\left(P, T\right),\right. \quad (1)$$

Framework Overview

1. Enrich textual semantics via TSE.

- Use GPT-4 to generate descriptions for each class, which are encoded into a dataset-wide knowledge base with CLIP's text encoder.
- Cluster this knowledge into class-agnostic attributes
- Use the global text prompt to hunt for its most relevant ones
- They are then aggregated into the final text representation

2. Static CAM generation via SVC SVC module: Intra-correlation operation을 통해, extract **fine-grained details** from intermediate layers.

- Replace CLIP's q-k self-attention with our Intra-correlation operation from intermediate layers
- The calibrated visual features and enhanced text embeddings are used for static CAMs via Eq. (1)

3. Dynamic CAM generation via LVC LVC module: Lightweight adapter를 통해, extract **spatial correlations** from SVC's static CAMs.

- A lightweight adapter is designed to learn dynamic token relations from static CAMs
- The relations are added to SVC and serve as a distribution shift to make the visual features more diverse
- The dynamic CAMs are generated with the enhanced text embeddings and LVC features via Eq. (1)

4. Segmentation training

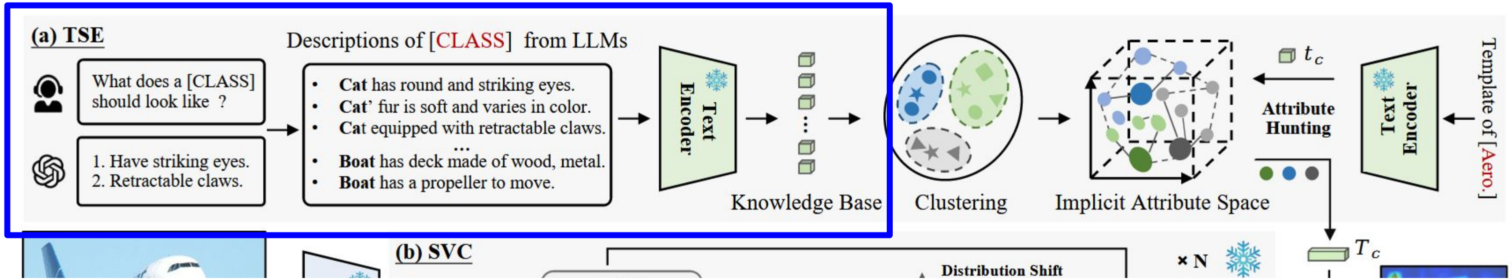
- Dynamic CAMs are refined to pseudo labels for segmentation supervision

Text Semantic Enrichment

Knowledge Base Construction

- Global text template E_c: *'a clean origami of [CLASS]'*
- Instructions for GPT: *"List n descriptions with key properties to describe the [CLASS] in terms of appearance, color, shape, size, or material, etc. These descriptions will help visually distinguish the [CLASS] from other classes in the dataset. Each description should follow the format: 'a clean origami [CLASS]. it + descriptive contexts.'"*
- GPT generate n detailed descriptions for each class, which are subsequently encoded into a dataset-wide knowledge base with CLIP's text encoder.
- **Knowledge base**: $\mathcal{T} = \{\Phi(e_i)\}_{i=1}^{n\times C}$

Knowledge Base Construction



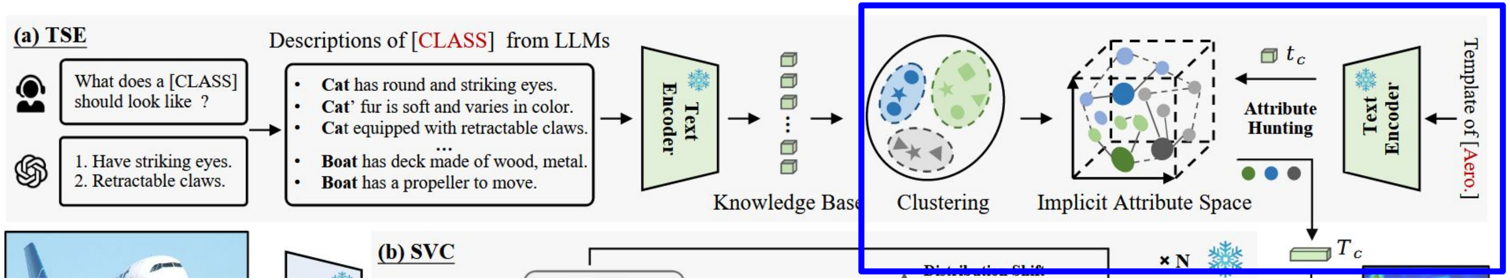
Text Semantic Enrichment

Implicit Attribute Hunting

- Cluster this knowledge into generalized attributes and treat text prompting as an implicit attribute-hunting process
- Each cluster centroid is viewed as the implicit attribute that represents a group of descriptions sharing similar properties
- Attribute feature space: $A = \text{Kmeans}(\mathcal{T}, B) = \{a_i\}_{i=1}^{B}$, (2)
- Global text template, Global text embedding: E_c, $t_c \in \mathbb{R}^{D\times 1}$
- Top-K Attribute neighbors: $A_c = \{a_j : j \in \text{argmax}_{\text{TOPK}} \{t_c^T a_j\}_{j=1}^{B}\}$. (3)
- **Final text representation**:

$$T_c = t_c + \lambda \sum_{j=1}^{K} \text{softmax}\left(t_c^T\ A_c\right) a_j, \quad (4)$$

Text Semantic Enrichment



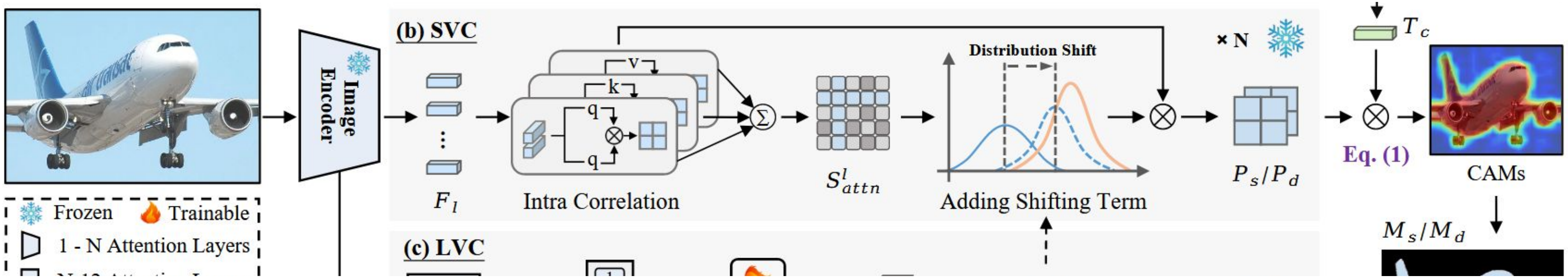
YANG, Zhiwei, et al. Exploring CLIP's Dense Knowledge for Weakly Supervised Semantic Segmentation. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025. p. 20223-20232.
CVPR 2025

Visual Calibrations

Static Visual Calibration

- Input image, features from l-th layer of CLIP: $X \in \mathbb{R}^{3\times\mathcal{H}\times\mathcal{W}}, \quad F_l \in \mathbb{R}^{D_s\times hw}$
- Original attention map: $\mathrm{SA}(q,k) = \mathrm{sofmax}\left(q^T k/\sqrt{D_s}\right), \quad (5)$
- Limitation: The original q-k attention produces overly uniform attention maps, homogenizing diverse tokens from v to capture broad semantics for global image representation, due to the inherent image-text alignment of CLIP.
- Ours: **Intra-correlation** calculates the **attention within each space of {q, k, v} across intermediate layers**, instead of generating q-k correlation
- Attention map from l-th SVC layer: $S^l_{attn} = \sum w_i\, \mathrm{SA}\left(O^l_i, O^l_i\right), O^l_i \in \{q^l, k^l, v^l\}, \quad (6) \quad S^l_{attn} \in \mathbb{R}^{hw\times hw} \quad l \in \{12-N, \ldots, 12\}$
- Calibrated features from the last layer of SVC: $P_s \in \mathbb{R}^{D\times h\times w}$
- **Static CAM** is generated by calibrated visual features from the last layer P_s and text embedding T_c: CAM_s

$$\mathrm{CAM} = \mathrm{Norm}\left(\cos\left(P, T\right),\right. \quad (1)$$



유진

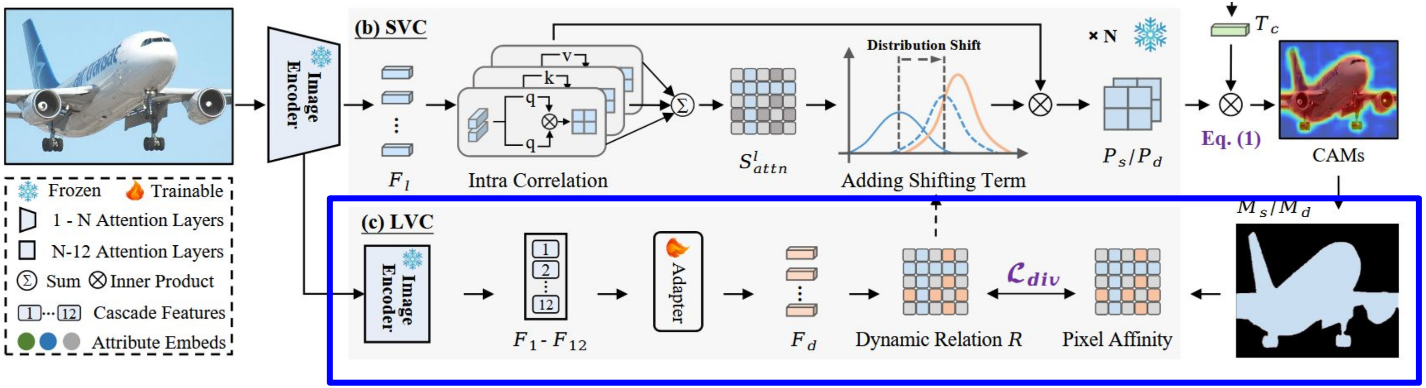
• Visual Calibrations

❑ Learnable Visual Calibration

- Limitation: Although ExCEL generates comparable CAMs without training, its performance is still limited by the fixed features in CLIP.
- Ours: We design a lightweight adapter, which only incorporates a distribution shift to calibrate the fixed features, to dynamically calibrate the visual features with diverse details.
- Frozen features from 1-12th layer of CLIP: $F_l \in \mathbb{R}^{D_s \times hw}$
- Dynamic feature: $F_d \in \mathbb{R}^{D_d \times hw}$ $\quad F_d = \text{Conv}(\text{Concate}\,[\delta_l\,(F_l)]_{l=1}^{12}),$ (7)
- Dynamic token relations: $r \in \mathbb{R}^{hw \times hw}$ $\quad r = \alpha(\cos{(F_d, F_d)} - \beta\overline{\cos{(F_d, F_d)}}),$ (8)
- Dynamic relations: $R \in \mathbb{R}^{hw \times hw}$

$$R_{ij} = \begin{cases} r_{ij}, & \text{if } r_{ij} \geq 0 \\ -inf, & \text{else} \end{cases}. \quad (9)$$

- Optimized attention map: $L^l_{attn} \in \mathbb{R}^{hw \times hw}$ $\quad L^l_{attn} = S^l_{attn} + \text{softmax}(R).$ (10)
- Dynamically calibrated features from the last layer of LVC: $P_d \in \mathbb{R}^{D \times h \times w}$
- **Dynamic CAM**: $\text{CAM} = \text{Norm}\,(\cos{(P, T)}\,,$ (1)



- **Training Objectives**

Diversity Loss

- Objective: To supervise the learning of F_d in LVC module
- Token correlations of F_d : $\hat{\mathcal{R}} \in \mathbb{R}^{hw \times hw}$ $\hat{\mathcal{R}} = \text{sigmoid}(\cos(F_d, F_d))$
- Static pseudo-labels: M_s
- **Diversity loss**:

$$\mathcal{L}_{\text{div}} = \frac{1}{N^+} \sum_{u^+ \in \hat{\mathcal{R}}^+} (1 - u^+) + \frac{1}{N^-} \sum_{u^- \in \hat{\mathcal{R}}^-} u^-, \quad (11)$$

Cross-Entropy Loss

- Objective: To supervise lightweight transformer-based segmentation head from WeCLIP [1]
- Dynamic pseudo-labels:
- **Cross-entropy loss**: $\mathcal{L}_{seg}$

Final Loss

- Adapter + Segmentation Head 학습

$$\mathcal{L}_{\text{ExCEL}} = \mathcal{L}_{seg} + \gamma \mathcal{L}_{\text{div}}, \quad (12)$$