SHAPE: STRUCTURAL HIERARCHY OF AFFINITIES FOR PATCH ENHANCEMENT IN TRAINING-FREE OPEN-VOCABULARY SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-free Open-Vocabulary Semantic Segmentation (OVSS) assigns text-specified class labels to image pixels without additional training. In this setting, frozen CLIP provides visual–text alignment, while a vision foundation model (VFM) supplies structural cues for localization. Despite this, predictions still suffer from patch noise, where isolated patches disagree with surrounding context, and region noise, where coherent object regions fragment. To address these issues, we formulate structural refinement as support selection: the VFM determines which patches contribute to each update, while CLIP remains the sole source of class evidence. We propose Structural Hierarchy of Affinities for Patch Enhancement (SHAPE), which introduces three complementary experts at different structural levels—Edge, Node, and Region—corresponding to relation, patch, and region support, respectively. The Edge Expert leverages VFM-derived patch affinities as relation-level support to aggregate CLIP visual features. The Node Expert reduces patch noise by refining each patch's class scores using its patch-specific VFM neighbors. Finally, the Region Expert reduces residual region noise by sharing class scores within coherent regions among patches whose predictions agree. Across eight benchmarks with a ViT-B/16 backbone, SHAPE achieves the best mIoU among weakly supervised and training-free methods on every dataset, reaching average scores of 44.0 and 47.9 without and with background classes, respectively. Code will be made publicly available upon publication.

1 INTRODUCTION

Open-vocabulary semantic segmentation (OVSS) assigns every pixel a class from a text vocabulary provided at test time. CLIP enables this without segmentation-specific training by embedding images and class names in a shared space (Radford et al., 2021). Training-free methods turn its visual tokens into dense class scores and improve localization by adapting attention or features at inference (Zhou et al., 2022; Li et al., 2025; Wang et al., 2024; Lan et al., 2024a). Yet CLIP's image-level objective does not explicitly enforce consistency among patches within an object, leaving errors that better features alone may not resolve.

These errors manifest in two forms. *Patch-wise inconsistency* occurs when an isolated patch disagrees with surrounding patches of the same class. *Region-level fragmentation* occurs when different parts of a coherent region receive different labels. We refer to these as patch noise and region noise, respectively. Isolated errors require support from similar patches, while fragmented regions call for shared regional cues. Fig. 1 highlights these two recurring errors in existing methods and the more coherent predictions produced by SHAPE.

Sharing information among patches can improve consistency, but its effectiveness depends on the *support*, i.e., the patches contributing to each update. Support within the same semantic region can correct local errors, whereas support crossing class boundaries can propagate them. Self-supervised vision foundation models (VFMs) provide structural cues for identifying related patches, including distant parts of the same object (Caron et al., 2021; Oquab et al., 2024). This motivates a clear division of roles: CLIP provides class evidence, while the VFM determines where that evidence should be aggregated or shared.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

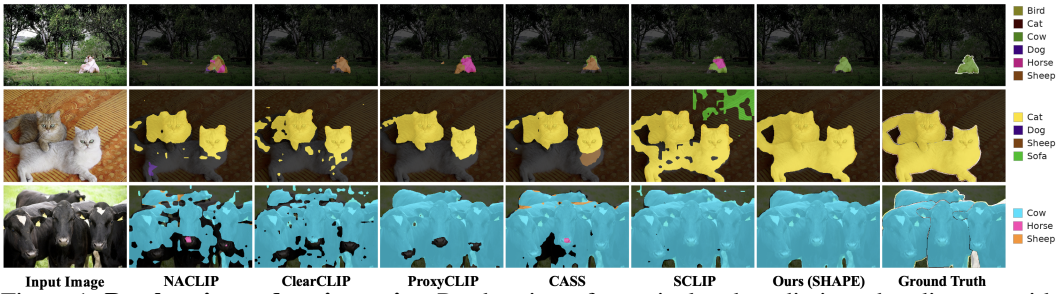

Figure 1: **Patch noise and region noise.** Patch noise refers to isolated predictions that disagree with surrounding patches, as seen in the cows (bottom), whereas region noise occurs when a semantic region is split into inconsistent labels, as shown in the cow (top) and cats (middle). SHAPE reduces both types of noise by selectively exploiting structural support through its Edge, Node, and Region Experts at relation, patch, and region levels, respectively.

Existing structural transfer methods use VFM relations primarily to improve feature aggregation before scoring (Kim et al., 2025). SHAPE extends this structural guidance beyond feature aggregation by organizing VFM structure into complementary supports at different scales. Relation-level support guides feature aggregation, patch-specific support provides evidence for correcting isolated errors, and regional support provides a shared reference for reducing fragmentation. Because regional groups may mix semantic classes, regional sharing is further conditioned on prediction agreement.

We propose Structural Hierarchy of Affinities for Patch Enhancement (SHAPE), a training-free framework that instantiates these three support types with complementary experts. The *Edge Expert* uses VFM relations to guide CLIP feature aggregation and produce the initial class scores. The *Node Expert* reduces patch noise by refining each patch's class scores using a patch-specific neighborhood in VFM feature space. The *Region Expert* reduces residual region noise by sharing a region-level score only when member predictions sufficiently agree. Across all three experts, both encoders stay frozen: the VFM determines structural support, while CLIP remains the sole source of class evidence. In summary, our contributions are:

- We formulate structural refinement for training-free OVSS as support selection, where VFM structure determines which patches contribute to each update while CLIP remains the sole source of class evidence.
- From this formulation, we propose SHAPE, instantiating three complementary support types in sequentially applied experts: relation-level support for aggregation in the Edge Expert, patch-specific neighborhoods in the Node Expert, and agreement-gated regional support in the Region Expert.
- Across eight benchmarks, SHAPE achieves the best mIoU among weakly supervised and training-free methods on every dataset, while GT-defined diagnostics confirm reductions in both patch noise and region noise.

2 Related Work

Open-vocabulary semantic segmentation. Open-vocabulary semantic segmentation (OVSS) assigns pixels to classes specified by a text vocabulary at inference time. CLIP provides a practical foundation by embedding images and class names in a shared representation space (Radford et al., 2021). Training-based approaches, including LSeg, OpenSeg, and SegCLIP, learn dense visual–text alignment from additional data (Li et al., 2022; Ghiasi et al., 2022; Luo et al., 2023). We study training-free OVSS, without task-specific training and with the pretrained encoders kept frozen.

Training-free dense prediction with CLIP. MaskCLIP converts frozen CLIP tokens into per-patch class scores (Zhou et al., 2022). SCLIP, ClearCLIP, NACLIP, and GEM improve dense representations through attention modification, representation decomposition, or spatial aggregation (Wang et al., 2024; Lan et al., 2024a; Hajimiri et al., 2025; Bousselham et al., 2024). SC-CLIP corrects anomalous tokens, while CLIP-DIY combines predictions across patch scales (Bai et al., 2025; Wysoczańska et al., 2024a). GLA-CLIP extends attention across windows and adjusts their interactions to improve global–local consistency (Lee et al., 2026). These approaches improve the

features or contextual information used for dense prediction. SHAPE addresses isolated patch errors and fragmented predictions that can remain after feature-level refinement.

Structural guidance from vision foundation models. CLIP is trained for image–text alignment, whereas self-supervised VFMs such as DINO and DINOv2 encode object-structural cues in their patch features (Caron et al., 2021; Oquab et al., 2024). This property supports object discovery and unsupervised segmentation (Siméoni et al., 2021; Wang et al., 2022b; Melas-Kyriazi et al., 2022; Wang et al., 2022a; 2023; Hamilton et al., 2022). In training-free OVSS, ProxyCLIP derives proxy attention from VFM similarity, CASS transfers spectrally refined affinity graphs into CLIP, and Trident combines foundation models to improve dense prediction (Lan et al., 2024b; Kim et al., 2025; Shi et al., 2025). These methods establish VFM structure as a useful source of guidance for constructing dense visual representations. SHAPE builds on structural transfer in its Edge Expert and reuses the same VFM features for patch-specific neighbor correction and agreement-gated regional score sharing.

Graph and region refinement. Graph propagation provides a framework for sharing label evidence between related samples (Zhu et al., 2003; Zhou et al., 2003). In training-free OVSS, PEARL refines pixel logits through text-aware Laplacian propagation (Pei et al., 2026). Region adjacency graphs have also been used to rectify CLIP features using image-derived structural priors (Huang et al., 2026). For score refinement, SHAPE uses two complementary supports: Node selects a separate feature neighborhood for each patch, while Region forms spectral groups that provide a common class-score reference. Node blends each patch's score with its neighbors' mean; Region shares a group's mean only when member predictions sufficiently agree with the mean's predicted class. Both stages apply a single update, combining patch-specific correction with selective regional consistency (Sec. 3).

3 METHOD

SHAPE combines the Edge, Node, and Region experts in Fig. 2 to control which patches contribute to an update and how strongly they share information. VFM relations define these supports, while CLIP's visual–text space provides class evidence; Appendix A.3 formalizes the underlying principle. Given an image with N patches and C class names, the experts produce score maps $\mathbf{S}^{\mathrm{E}}, \mathbf{S}^{\mathrm{N}}, \mathbf{S}^{\mathrm{R}} \in \mathbb{R}^{N\times C}$, with both encoders frozen. Edge aggregates visual features before scoring, while Node and Region refine scores through patch-specific neighbor correction and agreement-gated regional sharing, respectively.

3.1 PRELIMINARIES

Patch-affinity graphs and spectral representations. We represent an image as a weighted graph, where each node is a patch and each edge weight indicates how similar two patches are. A symmetric patch-affinity matrix A can be decomposed as $A = U\Lambda U^{\top}$, where the columns of U are eigenvectors and the diagonal entries of Λ are their corresponding eigenvalues. Each eigenvector assigns one value to each patch, forming a pattern across the image. The eigenvalues form the graph's spectrum, which SHAPE uses to compare the structural properties of attention-head affinity graphs. To group patches, spectral methods use eigenvectors of a graph Laplacian or a degree-normalized affinity matrix. With self-supervised visual features, these eigenvectors can reveal meaningful regions by assigning similar values to strongly connected patches, without dense labels (Melas-Kyriazi et al., 2022). Collecting each patch's values from several selected eigenvectors forms a vector called its spectral embedding. Grouping patches with similar embeddings then uses connections across the graph, rather than only the similarity between two patches.

3.2 EDGE EXPERT: STRUCTURE-GUIDED FEATURE AGGREGATION

The Edge Expert provides the first support in SHAPE: a relation-level support that decides, before any class score is computed, which patches contribute to each CLIP feature. CLIP is trained to align whole images with text, so self-attention in its visual encoder may mix information from unrelated patches, whereas self-supervised VFM features provide cues about object structure (Caron et al., 2021; Oquab et al., 2024). We therefore let VFM relations guide CLIP's attention while CLIP value

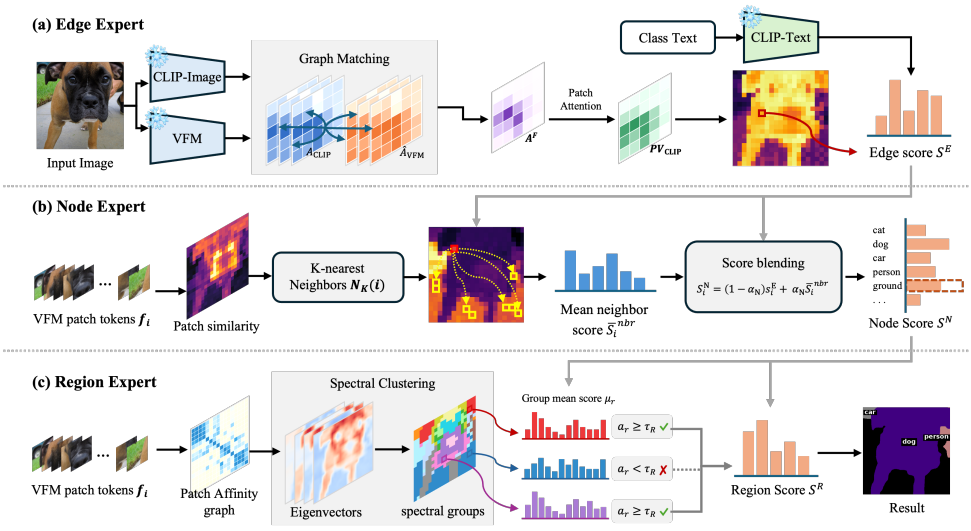

Figure 2: **Overview of SHAPE.** (a) The Edge Expert uses relation-level VFM support to guide CLIP feature aggregation. (b) The Node Expert uses a patch-specific VFM neighborhood to refine each score. (c) The Region Expert uses an agreement-gated regional support to reduce residual fragmentation. Together, the three experts organize VFM structure into complementary supports for feature aggregation, patch-specific correction, and regional consistency.

vectors remain the sole source of class evidence. To transfer VFM relations into CLIP attention, we build on the spectral VFM-to-CLIP attention distillation of CASS (Kim et al., 2025), adopting its head matching, low-rank eigenvalue scaling, and object-presence text refinement. The resulting class scores serve as the initial scores on which the Node and Region Experts operate.

Relation-level support. The Edge Expert processes overlapping windows of N_ω patches. For encoder $m \in \{\text{CLIP}, \text{VFM}\}$ and head $h \in \{1, \dots, H\}$, let $A_{m,h} \in \mathbb{R}^{N_\omega \times N_\omega}$ be the head's patch affinity, built from the query, key, and value self-similarities (Eq. 7), and $V_{m,h}$ its value projection. Each CLIP head h may be paired with a VFM head $\pi(h)$ by spectral head matching, and the VFM affinity is spectrally refined into $\widehat{A}_{\text{VFM},\pi(h)}$. With transfer weight $\gamma_h \geq 0$ ($\gamma_h = 0$ if unpaired), range-matching operator $\text{Match}(X; Y)$, and spatial preference G (Hajimiri et al., 2025), each head computes

$$A_h^{\text{F}} = \frac{A_{\text{CLIP},h} + \gamma_h \ \text{Match}(\widehat{A}_{\text{VFM},\pi(h)}; A_{\text{CLIP},h})}{1 + \gamma_h}, \qquad P_h = \text{softmax}_{\text{row}}\left(A_h^{\text{F}} + G\right), \qquad Z_h = P_h V_{\text{CLIP},h}. \tag{1}$$

Row i of P_h is the relation-level support of patch i: a soft weighting over patches whose affinity logits retain CLIP's relations (weight $1/(1+\gamma_h)$) while adding VFM structure (weight $\gamma_h/(1+\gamma_h)$).

What the support controls. Fix a head, write each CLIP value as $V_j = \mathbf{u}_{c(j)} + \mathbf{e}_j$, where $c(i)$ and $c(j)$ denote the semantic segments containing patches i and j, respectively, $\mathbf{u}_{c(j)}$ is the corresponding segment reference feature, and $\mathbf{e}_j$ is its residual. Let $\eta_i = \sum_{j:c(j)\neq c(i)} P_{ij}$ be the attention mass that patch i places outside its segment. Then

$$\|Z_i - \mathbf{u}_{c(i)}\| \leq \eta_i \, \Delta_i^{\text{feat}} + \Big\| \sum_j P_{ij} \mathbf{e}_j \Big\|, \tag{2}$$

where Δ_i^{feat} is the largest reference gap to another segment (Appendix A.1.1). The relation-level support targets the first term: VFM structure is meant to lower the cross-segment leakage η_i. The bound concerns features only and does not constrain how they are scored, which is the gap addressed by the Node and Region Experts.

Class scoring. Let $\mathbf{z}_i^\omega = \Phi([Z_1 \cdots Z_H]i)$ be the CLIP-projected patch feature in window ω. Let $\Omega(i)$ be the windows containing i, and let $\mathcal{T}_c$ denote the set of CLIP text embeddings for class c and

its synonyms. For each $\mathbf{t} \in \mathcal{T}_c$, let $\widetilde{\mathbf{t}}$ be its image-adjusted version obtained by the object-presence text refinement, and let $g^\omega(\mathbf{t})$ denote the global CLIP image–text similarity between window ω and the original text embedding $\mathbf{t}$. The Edge score $\mathbf{S}^{\mathrm{E}} \in \mathbb{R}^{N \times C}$ passed to Node Expert is

$$s^{\mathrm{E}}_{i,c} = \max_{\mathbf{t} \in \mathcal{T}_c} \frac{1}{|\Omega(i)|} \sum_{\omega \in \Omega(i)} \ell^\omega_i(\mathbf{t}), \qquad \ell^\omega_i(\mathbf{t}) = (1 - \lambda_{\mathrm{global}}) \cos(\mathbf{z}^\omega_i, \widetilde{\mathbf{t}}) + \lambda_{\mathrm{global}} g^\omega(\mathbf{t}). \quad (3)$$

3.3 Node Expert: Patch-Specific Evidence Sharing

The Edge Expert guides CLIP feature aggregation but does not enforce agreement among the resulting class scores, so visually similar patches may still receive different predictions. The Node Expert reduces this patch noise by combining each patch's class scores with those of its VFM neighbors. Unlike spatial neighborhoods, which can miss useful matches elsewhere and repeatedly draw on the same local errors, VFM neighborhoods let distant patches with similar semantic content contribute (Caron et al., 2021; Melas-Kyriazi et al., 2022). In Fig. 2(b), a ground patch near the dog's boundary has dog parts among its spatial neighbors, whereas its VFM neighbors are distant ground patches whose scores reinforce ground. Because neighbors are selected by visual features rather than current predictions, an incorrect CLIP prediction does not determine which patches correct it.

Neighbor selection and score blending. After merging the Edge windows, we align the VFM's final normalized patch features to the full CLIP grid. For each patch, we select its K nearest neighbors by cosine similarity, excluding the patch itself. This gives each patch its own neighbor set $\mathcal{N}_K(i)$. We average the neighbors' Edge scores and combine this average with the patch's own Edge score using $\alpha_{\mathrm{N}} \in [0, 1]$:

$$\bar{\mathbf{s}}^{\mathrm{nbr}}_i = \frac{1}{K} \sum_{j \in \mathcal{N}_K(i)} \mathbf{s}^{\mathrm{E}}_j, \qquad \mathbf{s}^{\mathrm{N}}_i = (1 - \alpha_{\mathrm{N}}) \mathbf{s}^{\mathrm{E}}_i + \alpha_{\mathrm{N}} \bar{\mathbf{s}}^{\mathrm{nbr}}_i. \quad (4)$$

All patches are updated once using the same Edge scores $\mathbf{S}^{\mathrm{E}}$, so no patch uses another patch's updated Node score. Each selected neighbor contributes equally to the average. The blending weight α_{N} controls how much the patch relies on this average while retaining part of its original score. VFM features determine which neighbors contribute, while the shared scores still cover the full CLIP class vocabulary.

Error analysis of score blending. Whether this update helps depends on which neighbors are selected. Same-segment neighbors share the underlying class, so averaging their scores reduces noise, more effectively when their errors are less correlated with each other and with the patch's own error. Neighbors from other segments instead introduce bias toward an incorrect class. Blending the neighbor average with the original score thus trades off variance reduction against this potential bias. Appendix A.4 derives the error of this update and shows why partial blending ($0 < \alpha_{\mathrm{N}} < 1$) can help even when full replacement ($\alpha_{\mathrm{N}} = 1$) is too aggressive.

3.4 Region Expert: Agreement-Gated Regional Consistency

Region noise occurs when patches within the same semantic region receive different labels, even after individual patch errors have been reduced. The Node Expert updates each patch using its own neighbors, so patches in the same semantic region may still rely on different sources of evidence. The Region Expert addresses this by forming spectral groups from the VFM graph and computing a shared class-score reference for each group. It then checks whether the members' predictions agree enough to use that reference.

Spectral group construction. We build a graph whose edges represent positive cosine similarities between VFM patch features. We normalize the graph by node degree and use its leading eigenvectors to represent each patch. These representations describe how patches are connected through the graph, allowing patches with similar connection patterns to be grouped together. We normalize each patch representation to unit length and apply k-means to form spectral groups. Unlike Node's patch-specific supports, the Region Expert constructs a shared support for each spectral group. Its mean score provides a common reference for reducing score differences among the group's members.

Agreement-gated score sharing. Patches with similar features do not always belong to the same class. Sharing a group's mean score without checking its members' predictions could therefore spread errors. For each group r, we compute the mean Node score $\boldsymbol{\mu}_r$ and identify its highest-scoring class. We then calculate a_r, the fraction of patches whose predicted class matches that class. A high a_r means that many members already support the class suggested by the group mean. Feature similarity determines which patches form a group, while prediction agreement determines whether they share their scores. The update is

$$\mathbf{s}_i^{\mathrm{R}} = \begin{cases} (1-\alpha_{\mathrm{R}})\mathbf{s}_i^{\mathrm{N}} + \alpha_{\mathrm{R}}\boldsymbol{\mu}_{r(i)}, & a_{r(i)} \geq \tau_{\mathrm{R}}, \\ \mathbf{s}_i^{\mathrm{N}}, & a_{r(i)} < \tau_{\mathrm{R}}. \end{cases} \tag{5}$$

Here, $r(i)$ is the group containing patch i, τ_{R} is the agreement threshold, and α_{R} controls how strongly the group mean influences each patch. Appendix A.2 details the graph construction and agreement calculation. When a group passes the agreement check, each patch's score moves toward the group mean. The mean itself stays unchanged, while each patch's difference from it is multiplied by $1 - \alpha_{\mathrm{R}}$. The update therefore makes scores within the group more similar. A patch adopts the group mean's predicted class when its remaining deviation from the mean is sufficiently small relative to the margin between the mean's highest-scoring class and the alternatives. Groups below the agreement threshold keep their original Node scores. Agreement does not guarantee correctness, because group members can agree on a wrong class.

Inference. We apply all three experts at each image scale $\xi \in \mathcal{S}$, the set of image scales. At each scale, Node and Region refinement operate on pre-softmax scores over the full patch grid. We align the final Region outputs to a common pixel grid by bilinear interpolation and denote them by $\widetilde{\mathbf{S}}_{\xi}^{\mathrm{R}}$. The multi-scale prediction is

$$\bar{\mathbf{S}} = \frac{1}{|\mathcal{S}|}\sum_{\xi\in\mathcal{S}} \widetilde{\mathbf{S}}_{\xi}^{\mathrm{R}}, \qquad \hat{y}_x = \arg\max_c \bar{s}_{x,c}. \tag{6}$$

Here $\mathcal{S}$ is the set of image scales. We average only the final Region outputs across scales. On benchmarks that include a background class—VOC21 (Everingham et al., 2010), Context60 (Mottaghi et al., 2014), and COCO-Object (Lin et al., 2014)—we instead assign a pixel to background when $\max_c[\mathrm{softmax}(\kappa\bar{\mathbf{s}}_x)]_c < \theta_{\mathrm{bg}}$. Here κ scales the scores before softmax, and θ_{bg} is the background threshold. All other pixels retain the highest-scoring class.

4 Experiments

4.1 Experimental Setup

Benchmarks and metric. We evaluate mIoU on eight OVSS benchmarks: VOC20/21 (Everingham et al., 2010), Pascal Context59/60 (Mottaghi et al., 2014), COCO-Object (Lin et al., 2014), Cityscapes (Cordts et al., 2016), ADE20K (Zhou et al., 2017), and COCO-Stuff164K (Caesar et al., 2018). VOC21, Context60, and COCO-Object include an explicit background class; the remaining protocols evaluate only their specified foreground classes. Background is scored as a valid class, whereas pixels marked as ignored in the annotations are excluded from the mIoU calculation.

Implementation and evaluation. All main experiments use frozen CLIP ViT-B/16, with DINO ViT-B/8 as the VFM. We process 224×224 crops with stride 112 at scales $\{1.0, 1.5\}$, applying all three experts at each scale. The Inference paragraph of Sec. 3.4 describes the procedure, and Tables 5 and 6 (Appendix A.7) list the hyperparameters. All refinement operations are performed at inference time without additional training or gradient updates. Background thresholding is used for datasets with an explicit background class. The main experiments use a single NVIDIA RTX 4090 GPU.

4.2 Comparison with Prior Methods

We compare SHAPE with three methods that use additional training data and eleven training-free baselines, reporting protocols with and without an evaluated background class separately.

Benchmarks without background. Table 1 compares SHAPE with recent training-free baselines. SHAPE achieves the best mIoU on all five benchmarks, outperforming the strongest respective baselines by 0.1 points on VOC20, 1.1 on Context59, 1.3 on Cityscapes, and 0.8 each on ADE20K and COCO-Stuff. It also achieves the highest mean mIoU of 44.0. Cityscapes and Context59 evaluate extended regions such as roads and sky, where reducing within-region prediction inconsistencies may contribute to the larger gains. In contrast, VOC20 evaluates only foreground object classes and shows a smaller improvement.

Table 1: Comparison without background (mIoU). Bold and underlined values denote the best and second-best results; parentheses show the gain over the best baseline.

Method	Extra Training	VOC20	PC59	City.	ADE.	C-Stf	Avg.
weakly-supervised methods with additional training data							
TCL (Cha et al., 2023)	✓	83.2	33.7	24.0	17.1	22.4	36.1
CoDe (Wu et al., 2024)	✓	81.1	37.4	28.9	17.7	23.9	37.8
CLIP-DINOiser (Wysoczańska et al., 2024b)	✓	80.8	35.8	31.1	20.1	24.6	38.5
training-free methods (no extra training data)							
CLIP (Radford et al., 2021)	✗	81.1	24.2	20.1	11.5	16.3	30.6
MaskCLIP (Zhou et al., 2022)	✗	64.1	28.5	28.3	13.7	18.8	30.7
NACLIP (Hajimiri et al., 2025)	✗	77.1	35.0	35.5	17.4	23.3	37.7
GEM (Bousselham et al., 2024)	✗	81.5	35.5	32.3	16.9	23.3	37.9
ClearCLIP (Lan et al., 2024a)	✗	82.3	35.7	32.8	17.3	24.0	38.4
SCLIP (Wang et al., 2024)	✗	80.8	35.3	34.2	17.5	23.6	38.3
ProxyCLIP (Lan et al., 2024b)	✗	78.9	38.7	39.7	19.7	26.2	40.6
LaVG (Kang & Cho, 2024)	✗	82.0	33.8	26.2	15.6	23.2	36.2
PEARL (Pei et al., 2026)	✗	86.9	38.5	37.6	19.4	26.3	41.7
GLA-CLIP (Lee et al., 2026)	✗	84.2	39.8	<u>40.8</u>	20.0	<u>26.9</u>	42.3
CASS (Kim et al., 2025)	✗	<u>87.8</u>	<u>40.1</u>	39.6	<u>20.3</u>	26.7	<u>42.9</u>
SHAPE (Ours)	✗	**87.9** (+0.1)	**41.2** (+1.1)	**42.1** (+1.3)	**21.1** (+0.8)	**27.7** (+0.8)	**44.0**

Benchmarks with background. Table 2 shows that SHAPE achieves the best mIoU on all three benchmarks, with a mean of 47.9 compared with GLA-CLIP's 47.1. Refinement improves over the Edge output by 2.3 points on VOC21, substantially more than on VOC20 (Table 7), indicating a larger gain under the background-inclusive evaluation protocol.

Table 2: Background-inclusive comparison (mIoU). Notation follows Table 1.

Method	Extra Training	VOC21	PC60	C-Obj	Avg.
weakly-supervised methods with additional training data					
TCL (Cha et al., 2023)	✓	55.0	31.1	31.6	39.2
CoDe (Wu et al., 2024)	✓	57.7	30.8	32.2	40.2
CLIP-DINOiser (Wysoczańska et al., 2024b)	✓	62.2	33.7	34.7	43.5
training-free methods (no extra training data)					
CLIP (Radford et al., 2021)	✗	34.8	22.4	22.5	26.6
MaskCLIP (Zhou et al., 2022)	✗	48.3	26.1	26.9	33.8
NACLIP (Hajimiri et al., 2025)	✗	58.9	33.1	33.1	41.7
GEM (Bousselham et al., 2024)	✗	57.0	33.5	34.1	41.5
ClearCLIP (Lan et al., 2024a)	✗	57.8	33.5	33.3	41.5
SCLIP (Wang et al., 2024)	✗	58.9	33.5	33.2	41.9
ProxyCLIP (Lan et al., 2024b)	✗	59.1	36.5	29.9	41.8
LaVG (Kang & Cho, 2024)	✗	62.0	33.8	34.2	43.3
PEARL (Pei et al., 2026)	✗	64.1	36.2	37.3	45.9
GLA-CLIP (Lee et al., 2026)	✗	<u>66.3</u>	37.4	<u>37.7</u>	<u>47.1</u>
CASS (Kim et al., 2025)	✗	64.4	<u>37.9</u>	36.6	46.3
SHAPE (Ours)	✗	**66.9** (+0.6)	**38.8** (+0.9)	**37.9** (+0.2)	**47.9**

Different CLIP backbones. Table 3 reports results with ViT-B/32, ViT-B/16, and ViT-L/14. SHAPE is best in six of nine backbone–dataset settings and within 0.9 points of the best in the rest. It improves the average mIoU of vanilla CLIP by 11.9, 20.2, and 16.5 points, respectively, showing that its benefits are not tied to a specific patch size. SHAPE is least effective on ViT-B/32, where it trails CASS on PC59 and COCO-Stuff, possibly because its coarser patches often span multiple semantic regions.

4.3 Ablation Study: Each Expert Reduces Noise

GT-defined noise metrics. To measure noise directly, we evaluate score maps on a common patch grid using GT-defined neighborhoods and regions independent of the experts' internal supports.

Table 3: Backbone comparison (mIoU) across CLIP visual encoders of different capacities and patch resolutions. V21: VOC21; PC59: Context59; C-Stf: COCO-Stuff. Formatting follows Table 2.

	ViT-B/32			ViT-B/16			ViT-L/14		
Method	V21	PC59	C-Stf	V21	PC59	C-Stf	V21	PC59	C-Stf
CLIP	39.1	25.4	17.3	34.8	24.2	16.3	35.7	23.7	16.6
MaskCLIP	48.2	28.5	18.3	48.3	28.5	18.8	45.6	26.2	17.6
NACLIP	53.3	32.5	21.2	58.9	35.0	23.3	52.9	31.7	21.4
GEM	50.1	28.4	16.7	57.0	35.5	23.3	45.1	25.4	17.9
ClearCLIP	50.1	31.7	21.0	57.8	35.7	24.0	49.6	31.1	21.2
SCLIP	52.3	32.6	21.3	58.9	35.3	23.6	52.1	31.7	21.6
ProxyCLIP	57.9	35.4	23.6	59.1	38.7	26.2	59.8	37.8	**26.2**
CASS	57.6	**35.9**	**24.6**	64.4	40.1	26.7	58.5	36.4	24.7
SHAPE (Ours)	**58.2** (+0.3)	35.0	24.4	**66.9** (+2.5)	**41.2** (+1.1)	**27.7** (+1.0)	**61.4** (+1.6)	**37.9** (+0.1)	26.1

At the patch level, Agree measures whether a patch prediction is consistent with the dominant prediction in its local semantic neighborhood, while Dev measures the distance between its class-score vector and the neighborhood mean. Thus, higher Agree and lower Dev indicate fewer local inconsistencies in both predicted labels and score distributions. At the region level, Frag measures how strongly a connected semantic region is split into disconnected predicted pieces, while Disp measures the variation of class scores within that region. Lower Frag and Disp therefore indicate less fragmentation and greater region-level score consistency. Appendix A.6 provides the formal definitions and evaluation protocol for all four metrics.

Table 4: GT-defined noise metrics. Cityscapes uses the full validation set ($n = 500$); Context59 and ADE20K use 300-image subsets, with the same images evaluated across all configurations. mIoU is computed on the patch grid and differs from the main benchmark mIoU.

	Patch-level		Region-level		
Method	Agree ↑	Dev ↓	Frag ↓	Disp ↓	mIoU ↑
Cityscapes ($n = 500$, full)					
Edge	0.877	0.032	5.86	0.058	37.0
Edge+Node	0.893	0.026	5.02	0.053	38.4
Edge+Region	0.884	0.028	5.65	0.051	38.1
Edge+Node+Region (Ours)	**0.897**	**0.024**	**4.98**	**0.047**	**39.1**
Context59 ($n = 300$ subset)					
Edge	0.845	0.038	7.12	0.067	34.8
Edge+Node	0.872	0.029	5.87	0.060	35.4
Edge+Region	0.858	0.033	6.60	0.061	35.2
Edge+Node+Region (Ours)	**0.879**	**0.027**	**5.57**	**0.056**	**35.7**
ADE20K ($n = 300$ subset)					
Edge	0.791	0.063	6.86	0.105	14.3
Edge+Node	0.825	0.049	5.59	0.093	14.8
Edge+Region	0.807	0.056	6.32	0.096	14.5
Edge+Node+Region (Ours)	**0.835**	**0.046**	**5.25**	**0.086**	**15.0**

Effects of the experts on noise. As shown in Table 4, the full sequence consistently reduces noise across all three datasets, raising Agree by 2.0–4.4 points and lowering Dev, Frag, and Disp by 15–29% over Edge. These reductions accompany patch-grid mIoU gains of 0.7–2.1 points, and the reductions in patch-level disagreement ($1 - \text{Agree}$, 16–22%) and Frag (15–23%) closely track each other, indicating that correcting local inconsistencies yields more coherent regions. Node and Region each improve all four metrics, and combining them improves them further, suggesting that they correct partly complementary errors; accordingly, their composition yields a larger eight-benchmark gain (1.0 vs. 0.6 points; Table 7). Fig. 3 illustrates this progression.

4.4 Qualitative Results

Fig. 4 compares SHAPE with prior methods. Most prior methods leave holes and scattered errors inside the laptop and refrigerators (patch noise) and label the toucan's head and beak differently from its body (region noise), whereas SHAPE predicts each of these objects as a single coherent region. Fig. 3 further traces these improvements through the individual experts. In the top row, Node removes most of the scattered false predictions around the dog and Region removes the remaining one, reducing patch noise. In the bottom row, the fragment on the car shrinks after Node and is relabeled as car after Region, leaving a single coherent car region and reducing region noise.

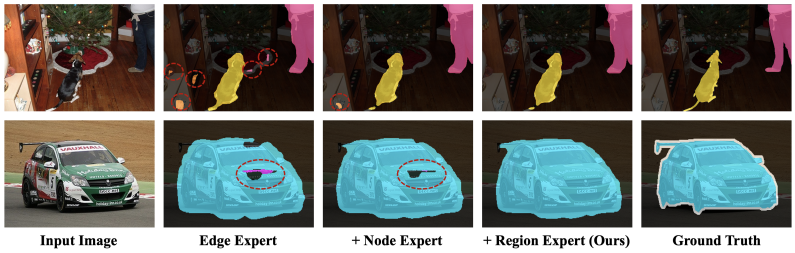

Figure 3: **Progressive refinement through the three experts.** Each row shows the input image, predictions from $\mathbf{S}^{\mathrm{E}}$, $\mathbf{S}^{\mathrm{N}}$, and $\mathbf{S}^{\mathrm{R}}$, and ground truth. The examples illustrate the progressive reduction of patch noise and region noise.

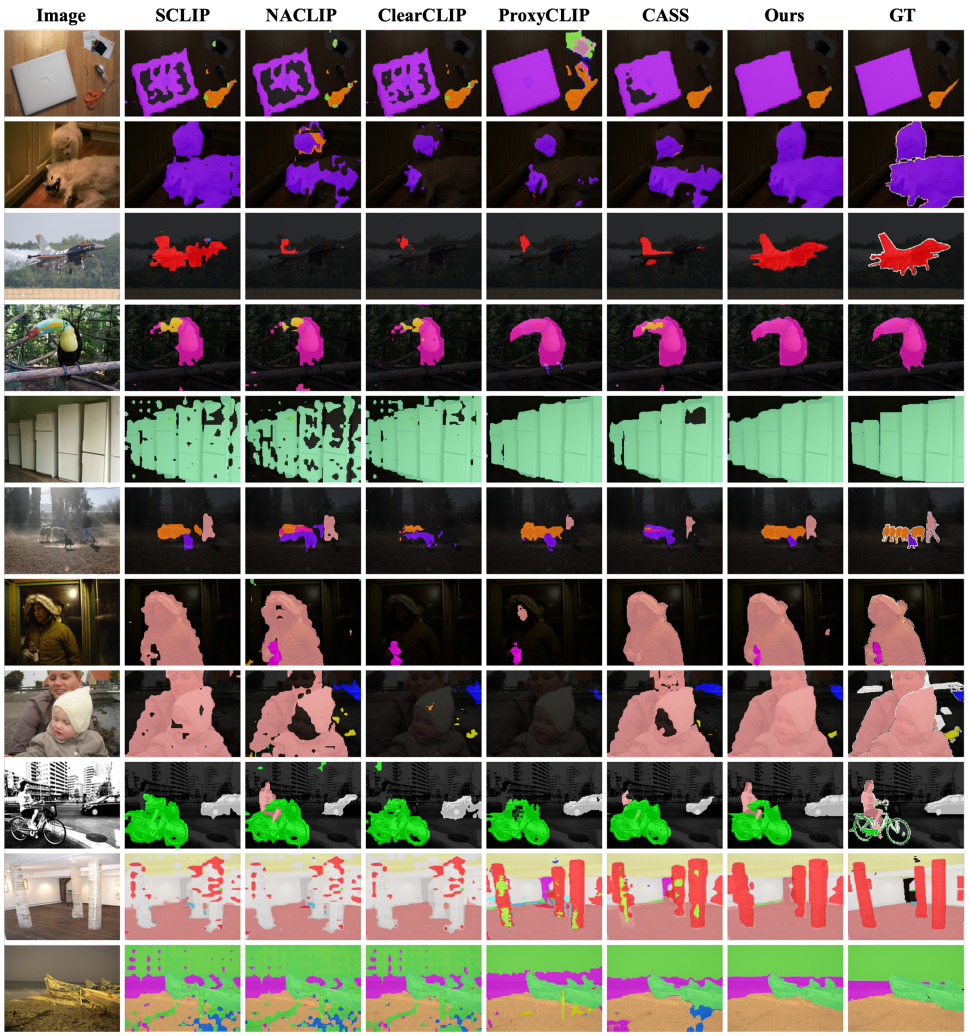

Figure 4: **Qualitative comparison with prior methods.** We compare segmentation predictions from SCLIP, NACLIP, ClearCLIP, ProxyCLIP, CASS, and SHAPE with the ground truth. SHAPE produces more complete object regions with fewer scattered prediction errors.

5 Conclusion

We propose SHAPE, a training-free framework for open-vocabulary semantic segmentation that refines CLIP predictions using structural guidance from a frozen VFM. SHAPE organizes VFM structure into relation-level, patch-specific, and regional supports to reduce patch noise and region noise. Experiments on eight benchmarks show consistent improvements in mIoU and noise metrics, demonstrating the effectiveness of hierarchical structural refinement.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting and editing, code development and debugging, and the presentation of mathematical and experimental analyses. They were also used for wording, formatting, and reference-related checks. All AI-assisted code, derivations, experimental results, citations, and claims were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Appendix A.1.2 details the head-assignment convention, Appendix A.6 specifies the noise-metric protocol used for Table 4, and Appendix A.7 lists the hyperparameters and inference settings. Appendices A.9 and A.10 provide additional ablations and sensitivity analyses. The released code will include the evaluated image lists and measurement scripts.

REFERENCES

Sule Bai, Yong Liu, Yifei Han, Haoji Zhang, Yansong Tang, Jie Zhou, and Jiwen Lu. Self-calibrated CLIP for training-free open-vocabulary segmentation. *IEEE Transactions on Image Processing*, 34:8271–8284, 2025. URL https://ieeexplore.ieee.org/document/11291123.

Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. In *CVPR*, 2024.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *CVPR*, 2023.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/5240_ECCV_2022_paper.php.

Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *WACV*, pp. 5061–5071, 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Hajimiri_Pay_Attention_to_Your_Neighbours_Training-Free_Open-Vocabulary_Semantic_Segmentation_WACV_2025_paper.html.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011. doi: 10.1137/090771806. URL https://doi.org/10.1137/090771806.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022. URL https://www.iclr.cc/virtual/2022/poster/6068.

Qiming Huang, Hao Ai, and Jianbo Jiao. Structure-aware feature rectification with region adjacency graphs for training-free open-vocabulary semantic segmentation. In *WACV*, pp. 4118–4127, 2026.

Dahyun Kang and Minsu Cho. In defense of lazy visual grounding for open-vocabulary semantic segmentation. In *ECCV*, 2024.

Chanyoung Kim, Dayun Ju, Woojung Han, Ming-Hsuan Yang, and Seong Jae Hwang. Distilling spectral graph for object-context aware open-vocabulary semantic segmentation. In *CVPR*, 2025.

Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. ClearCLIP: Decomposing CLIP representations for dense vision-language inference. In *ECCV*, 2024a.

Mengcheng Lan, Xinjiang Wang, Yiping Ke, Jiaxing Xu, Litong Feng, and Wayne Zhang. ProxyCLIP: Proxy attention improves CLIP for open-vocabulary segmentation. In *ECCV*, 2024b.

ByeongCheol Lee, Hyun Seok Seong, Sangeek Hyun, Gilhan Park, WonJun Moon, and Jae-Pil Heo. Looking beyond the window: Global-local aligned CLIP for training-free open-vocabulary semantic segmentation. In *CVPR*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Lee_Looking_Beyond_the_Window_Global-Local_Aligned_CLIP_for_Training-free_Open-Vocabulary_CVPR_2026_paper.pdf.

Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. URL https://iclr.cc/virtual/2022/poster/6809.

Yi Li, Hualiang Wang, Yiqun Duan, Jiheng Zhang, and Xiaomeng Li. A closer look at the explainability of contrastive language-image pre-training. *Pattern Recognition*, 162:111409, 2025. doi: 10.1016/j.patcog.2025.111409. URL https://doi.org/10.1016/j.patcog.2025.111409.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *ICML*, pp. 23033–23044, 2023. URL https://proceedings.mlr.press/v202/luo23a.html.

Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Gensheng Pei, Xiruo Jiang, Xinhao Cai, Tao Chen, Yazhou Yao, and Byeungwoo Jeon. PEARL: Geometry aligns semantics for training-free open-vocabulary semantic segmentation. In *CVPR*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Yuheng Shi, Minjing Dong, and Chang Xu. Harnessing vision foundation models for high-performance, training-free open vocabulary segmentation. In *ICCV*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Shi_Harnessing_Vision_Foundation_Models_for_High-Performance_Training-Free_Open_Vocabulary_Segmentation_ICCV_2025_paper.html.

Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021. URL https://bmva-archive.org.uk/bmvc/2021/conference/papers/paper_1339.html.

Feng Wang, Jieru Mei, and Alan Yuille. SCLIP: Rethinking self-attention for dense vision-language inference. In *ECCV*, 2024.

Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M. Alvarez. FreeSOLO: Learning to segment objects without annotations. In *CVPR*, pp. 14176–14186, 2022a. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_FreeSOLO_Learning_To_Segment_Objects_Without_Annotations_CVPR_2022_paper.html.

Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *CVPR*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wang_Cut_and_Learn_for_Unsupervised_Object_Detection_and_Instance_Segmentation_CVPR_2023_paper.html.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *CVPR*, 2022b.

Ji-Jia Wu, Andy Chia-Hao Chang, Chieh-Yu Chuang, Chun-Pei Chen, Yu-Lun Liu, Min-Hung Chen, Hou-Ning Hu, Yung-Yu Chuang, and Yen-Yu Lin. Image-text co-decomposition for text-supervised semantic segmentation. In *CVPR*, 2024.

Monika Wysoczańska, Michaël Ramamonjisoa, Tomasz Trzciński, and Oriane Siméoni. CLIP-DIY: CLIP dense inference yields open-vocabulary semantic segmentation for-free. In *WACV*, pp. 1403–1413, 2024a. URL https://openaccess.thecvf.com/content/WACV2024/html/Wysoczanska_CLIP-DIY_CLIP_Dense_Inference_Yields_Open-Vocabulary_Semantic_Segmentation_For-Free_WACV_2024_paper.html.

Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. CLIP-DINOiser: Teaching CLIP a few DINO tricks for open-vocabulary semantic segmentation. In *ECCV*, pp. 320–337, 2024b.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.

Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *ECCV*, 2022.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NeurIPS*, volume 16, 2003. URL https://papers.nips.cc/paper_files/paper/2003/hash/87682805257e619d49b8e0dfdc14affa-Abstract.html.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, pp. 912–919, 2003. URL https://mlg.eng.cam.ac.uk/pub/pdf/ZhuGhaLaf03a.pdf.

A Appendix

A.1 Edge Expert details

The spectral head matching, low-rank eigenvalue scaling, and object-presence text refinement follow CASS (Kim et al., 2025); the multi-view affinity below is our addition. This section records the implemented procedure.

Multi-view affinity. For encoder m and head h, let $Q_{m,h}$, $K_{m,h}$, and $V_{m,h} \in \mathbb{R}^{N_\omega \times d}$ be the final-block projections, excluding special tokens, with head dimension d. After aligning the VFM projections to the CLIP grid and matching their numerical ranges, we use

$$A_{m,h} = \frac{Q_{m,h}Q_{m,h}^\top + K_{m,h}K_{m,h}^\top + V_{m,h}V_{m,h}^\top}{3\sqrt{d}}. \tag{7}$$

Spectral refinement. For each VFM head h (the head subscript is omitted below; the result is $\widehat{A}_{\mathrm{VFM},h}$ in Eq. 1), randomized rank-r_{E} SVD (Halko et al., 2011) and eigenvalue filtering give

$$A_{\mathrm{VFM}} \approx U\,\mathrm{diag}(\lambda)U^\top, \qquad \widetilde{\lambda}_\ell = \lambda_\ell \exp\left[-\tau_{\mathrm{E}}\left(1 - \frac{\lambda_\ell}{\lambda_{\max}}\right)\right]. \tag{8}$$

Here N_ω is the number of window patches, r_{E} is the retained rank, and the columns of $U \in \mathbb{R}^{N_\omega} \times r_{\mathrm{E}}$ are eigenvectors of the positive-semidefinite affinity. The vector λ contains their nonnegative eigenvalues, $\lambda_{\max}$ is the largest, and τ_{E} controls spectral decay. Reconstruction uses

$$\widehat{A}_{\mathrm{VFM}} = U\,\mathrm{diag}\Big(g_{\mathcal{D}}(\widetilde{\lambda})\Big)\,U^\top, \qquad \mathrm{diag}(\widehat{A}_{\mathrm{VFM}}) := 0. \tag{9}$$

The function $g_{\mathcal{D}}$ rescales the filtered eigenvalues before reconstructing the VFM affinity, after which the diagonal of the reconstructed affinity is set to zero.

Affinity fusion and attention. The fused affinity and attention are given in Eq. 1. The transfer weight is $\gamma_h = \lambda_{\mathrm{fuse}}\,\delta_{\pi(h),h}$ for paired heads and $\gamma_h = 0$ for unpaired CLIP heads, which therefore retain their original affinity. The global strength λ_{fuse} is a hyperparameter (Table 6), and $\delta_{\pi(h),h}$ is the spectral distance of the matched pair (Appendix A.1.2). Only the product enters Eq. 1, so a single per-head weight suffices. The spatial preference G follows NACLIP (Hajimiri et al., 2025).

Class scoring. The dense branch omits CLIP's final residual and feed-forward updates, so Φ in Eq. 3 is applied directly to the concatenated head outputs. Text embeddings combine multiple prompts with image-conditioned adjustment, whose strength is set by λ_{text}; h_{text} is the threshold of the hierarchical clustering applied to the class text embeddings. We fuse local and global scores in ℓ_i^ω with weight $\lambda_{\mathrm{global}}$, average overlapping windows, and take the maximum over synonyms to obtain $\mathbf{S}^{\mathrm{E}}$ (Eq. 3).

A.1.1 Rationale and feature-level analysis

Multi-view affinity. For one head, concatenate the aligned projections as $F = [Q\ K\ V]$. The multi-view affinity of Eq. 7 is exactly

$$A = \frac{FF^\top}{3\sqrt{d}}, \qquad x^\top Ax = \frac{\|F^\top x\|^2}{3\sqrt{d}} \geq 0. \tag{10}$$

Equation 10 establishes positive semidefiniteness without assuming independence between projection views.

Relative spectral attenuation. Before the rescaling $g_{\mathcal{D}}$ and diagonal removal, the spectral filter obeys, for retained eigenvalues $\lambda_a \geq \lambda_b > 0$ and $\tau_{\mathrm{E}} \geq 0$,

$$\frac{\widetilde{\lambda}_a}{\widetilde{\lambda}_b} = \frac{\lambda_a}{\lambda_b}\exp\left[\frac{\tau_{\mathrm{E}}(\lambda_a - \lambda_b)}{\lambda_{\max}}\right] \geq \frac{\lambda_a}{\lambda_b}. \tag{11}$$

Equation 11 follows by dividing the filter responses. Subsequent rescaling and diagonal removal need not preserve eigenvalue ratios or positive semidefiniteness.

How attention leakage affects a feature. Fix a query patch i and a nonnegative attention row $P_{i,:}$ summing to one. For each patch index j, write its CLIP value as $V_j = u_{c(j)} + e_j$, where $c(j)$ denotes the semantic segment containing patch j, $u_{c(j)}$ is the corresponding segment reference feature, and e_j is its residual. Define the cross-segment attention mass and maximum segment-reference gap as

$$\eta_i = \sum_{j:c(j)\neq c(i)} P_{ij}, \qquad \Delta_i^{\text{feat}} = \max_{j:c(j)\neq c(i)} |u_{c(j)} - u_{c(i)}|. \tag{12}$$

Here $\Delta_i^{\text{feat}} = 0$ if no other segment occurs. Using the quantities in Eq. 12, the aggregated feature satisfies Eq. 2 in Sec. 3.2.

Proof. Since the attention weights sum to one, $Z_i - u_{c(i)} = \sum_j P_{ij}(u_{c(j)} - u_{c(i)}) + \sum_j P_{ij} e_j$. Same-segment reference differences vanish. Applying the triangle inequality to the remaining differences bounds their contribution by $\eta_i \Delta_i^{\text{feat}}$.

The bound separates cross-segment mixing from aggregated residuals before class-score projection; it does not guarantee improved segmentation.

A.1.2 Head assignment convention

Head pairs are compared by the spectra of the affinities in Eq. 7. Each head signature uses the top 20 eigenvalues normalized by its ℓ_1 norm, following the spectral head-matching convention of CASS (Kim et al., 2025). Let $u_{h'}$ and v_h denote the VFM and CLIP head signatures, respectively, and let H be the common head count. With $\pi(h)$ denoting the VFM head assigned to CLIP head h, the distance and assignment are

$$\delta_{h',h} = \| \operatorname{sort}(u_{h'}) - \operatorname{sort}(v_h) \|_1, \qquad \pi = \arg\max_{\pi \in \mathfrak{S}_H} \sum_h \delta_{\pi(h),h}. \tag{13}$$

Here $\mathfrak{S}_H$ is the set of head permutations. CLIP head h receives VFM head $\pi(h)$ with transfer weight $\gamma_h = \lambda_{\text{fuse}}\, \delta_{\pi(h),h}$ in Eq. 1. The first index of $\delta_{h',h}$ denotes a VFM head and the second denotes a CLIP head; the assignment maximizes the summed distance. Selecting pairs by maximal spectral distance follows the complementary matching of CASS (Kim et al., 2025), which pairs each CLIP head with the VFM head whose spectral structure differs most from its own.

A.2 Node features and regional grouping

Let $\mathbf{f}_i$ be the ℓ_2-normalized VFM feature of patch i, taken after the final LayerNorm. The features are resized to the CLIP grid and averaged over overlapping windows. They are the VFM block outputs, distinct from the query, key, and value projections used in the Edge Expert. Node neighbors are selected as $\mathcal{N}_K(i) = \operatorname{TopK}_{j\neq i}(\mathbf{f}_i^\top \mathbf{f}_j)$.

Spectral groups. For the spectral clustering introduced in Sec. 3.1, these features define the patch affinity and its degree-normalized form:

$$A_{ij} = \begin{cases} \max(\mathbf{f}_i^\top \mathbf{f}_j, 0), & i \neq j, \\ 0, & i = j, \end{cases} \quad D_{ii} = \max\left\{ \sum_j A_{ij}, 10^{-8} \right\}, \quad \widetilde{A} = D^{-1/2} A D^{-1/2}. \tag{14}$$

We retain the eigenvectors of $\widetilde{A}$ associated with its r_{eig} largest eigenvalues, normalize the resulting patch representations, and apply k-means to obtain M spectral groups. The members of group r are denoted by $\mathcal{I}_r$. A spectral group is a set of patches grouped by VFM structure and need not be spatially connected. Throughout the appendix, *group* refers to this Region-Expert support, whereas *region* refers to a spatially connected semantic region used only in the GT-defined diagnostics.

Group score and agreement. Both quantities are computed from the Node output:

$$\boldsymbol{\mu}_r = \frac{1}{|\mathcal{I}_r|} \sum_{i\in\mathcal{I}_r} \mathbf{s}_i^{\text{N}}, \quad a_r = \frac{1}{|\mathcal{I}_r|} \sum_{i\in\mathcal{I}_r} \mathbf{1}\big[\arg\max_c s_{i,c}^{\text{N}} = \arg\max_c \mu_{r,c} \big]. \tag{15}$$

The agreement test compares each member's argmax with the argmax of the group's mean score, rather than with the majority vote of the member argmax labels.

A.3 SUPPORT SELECTION AND POOLING ERROR

Model and conditioning. Write $\mathbf{s}_j = \boldsymbol{\nu}_{c(j)} + \boldsymbol{\epsilon}_j$, where $c(j)$ is the semantic segment, $\boldsymbol{\nu}_{c(j)}$ its reference score, and $\boldsymbol{\epsilon}_j$ the residual. All expectations condition on fixed supports and reference scores. For data-dependent supports, the assumptions below must hold conditionally; VFM selection alone does not establish them.

For the uniform-pooling special case, assume (A1) a constant reference score within each segment; (A2) zero-mean residuals with $\mathbb{E}\|\boldsymbol{\epsilon}_j\|^2 = \sigma^2$ and $\mathbb{E}\langle\boldsymbol{\epsilon}_j, \boldsymbol{\epsilon}_k\rangle = \rho\sigma^2$ for distinct support members, with $0 \le \rho < 1$; and (A3) a support Ω of m patches has a fraction p in the query's segment, while the others have mean reference score $\boldsymbol{\nu}'$ at distance $\Delta = \|\boldsymbol{\nu}' - \boldsymbol{\nu}_{c(i)}\|$. For $p = 1$, the bias is zero and the choice of $\boldsymbol{\nu}'$ is immaterial.

Proposition 1 (Uniform pooling error)**.** *Under (A1)–(A3), the pooled score $\bar{\mathbf{s}}_\Omega = m^{-1}\sum_{j\in\Omega}\mathbf{s}_j$ satisfies*

$$\mathbb{E}\|\bar{\mathbf{s}}_\Omega - \boldsymbol{\nu}_{c(i)}\|^2 = \underbrace{\sigma^2\left(\rho + \frac{1-\rho}{m}\right)}_{\text{variance}} + \underbrace{(1-p)^2\Delta^2}_{\text{squared bias}}. \tag{16}$$

If the query also has error σ^2, pooling has smaller expected squared error exactly when $(1-p)^2\Delta^2 < (1-\rho)(1-1/m)\sigma^2$.

Proof. The pooled reference differs from the query reference by $(1-p)(\boldsymbol{\nu}' - \boldsymbol{\nu}_{c(i)})$. The mean residual is zero, so the cross term between that bias and the residual vanishes. Its variance is

$$\mathbb{E}\left\|\frac{1}{m}\sum_{j\in\Omega}\boldsymbol{\epsilon}_j\right\|^2 = \frac{m\sigma^2 + m(m-1)\rho\sigma^2}{m^2}.$$

Adding the squared bias gives the stated pooling error; subtracting the query error gives the stated condition.

Support quality remains an empirical question: GT-class purity is only a proxy for segment purity.

A.4 NODE EXPERT: SCORE BLENDING AND ITS ERROR

In row i of P_K, the K selected neighbors each receive weight $1/K$, and all other entries are zero. Thus, $P_K\mathbf{S}^{\mathrm{E}}$ gives each patch's mean neighbor score, and Eq. 4 becomes

$$\mathbf{S}^{\mathrm{N}} = \big((1-\alpha_{\mathrm{N}})I + \alpha_{\mathrm{N}}P_K\big)\mathbf{S}^{\mathrm{E}}. \tag{17}$$

For $0 \le \alpha_{\mathrm{N}} \le 1$, this operator is row-stochastic even if P_K is asymmetric. It preserves segment-constant scores when every selected neighbor belongs to the query segment.

Proposition 2 (Error of the Node score update)**.** *Fix a query i, its selected neighbors, and their reference scores. All expectations below are conditional on these fixed quantities, and all residuals are assumed to have zero mean. Let $\mathbf{b}_i$ be the neighbors' mean reference score minus the query's reference $\boldsymbol{\nu}_{c(i)}$, and let $\bar{\boldsymbol{\epsilon}}_i$ be the neighbors' mean residual. Define*

$$q_i = \mathbb{E}\|\boldsymbol{\epsilon}_i\|^2, \quad v_i = \mathbb{E}\|\bar{\boldsymbol{\epsilon}}_i\|^2, \quad \chi_i = \mathbb{E}\langle\boldsymbol{\epsilon}_i, \bar{\boldsymbol{\epsilon}}_i\rangle.$$

Here q_i is query residual variance, v_i is neighbor-mean residual variance, and χ_i is their cross moment.

For a blend weight $\alpha \in [0,1]$, the expected squared error from the query's reference is

$$R_i(\alpha) = (1-\alpha)^2 q_i + \alpha^2\big(v_i + \|\mathbf{b}_i\|^2\big) + 2\alpha(1-\alpha)\chi_i. \tag{18}$$

Let $d_i = q_i + v_i - 2\chi_i + \|\mathbf{b}_i\|^2$. If $d_i > 0$, the weight that minimizes this error on $[0,1]$ is

$$\alpha_i^* = \mathrm{clip}_{[0,1]}\left(\frac{q_i - \chi_i}{d_i}\right).$$

If $q_i > \chi_i$, any $0 < \alpha \le 1$ with $\alpha < 2(q_i - \chi_i)/d_i$ gives lower expected squared error than keeping the original query score.

Proof. Subtracting the query's reference from the blended score gives $\alpha \mathbf{b}_i + (1-\alpha)\boldsymbol{\epsilon}_i + \alpha\bar{\boldsymbol{\epsilon}}_i$. Since $\mathbf{b}_i$ is fixed and the residuals have zero mean, their cross terms with $\mathbf{b}_i$ have zero expectation. Expanding the squared norm and taking expectations gives the stated expected error, or equivalently,

$$R_i(\alpha) - q_i = -2\alpha(q_i - \chi_i) + \alpha^2 d_i.$$

Because

$$d_i = \mathbb{E}\|\boldsymbol{\epsilon}_i - \bar{\boldsymbol{\epsilon}}_i\|^2 + \|\mathbf{b}_i\|^2 \geq 0,$$

minimizing this quadratic gives the optimal weight and the error-reduction condition. If $d_i = 0$, both nonnegative terms are zero, so $q_i = \chi_i$ and the error is constant.

The optimum α_i^* is analytical only. The implementation uses the shared $\alpha_{\mathrm{N}} = 0.5$ for the benchmark results in Tables 1 and 2.

If the query and all K neighbors have equal variance and pairwise correlation ρ, the expression simplifies to

$$R_i(\alpha) = \rho\sigma^2 + (1-\rho)\sigma^2\left((1-\alpha)^2 + \frac{\alpha^2}{K}\right) + \alpha^2\|\mathbf{b}_i\|^2. \tag{19}$$

The query-specific term distinguishes blending from pure pooling; no independence between CLIP and VFM is assumed.

A.5 Region Expert: spectral supports and exact contraction

Group projection. For a fixed partition into nonempty groups, let $\mathbf{B}_{ir} = 1$ if patch i belongs to group r and zero otherwise. The group-mean operator is

$$\Pi = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top.$$

$\mathbf{B}^\top\mathbf{B}$ contains group sizes. Thus $\Pi\mathbf{S}$ repeats each group mean at its members, with $\Pi^\top = \Pi$ and $\Pi^2 = \Pi$.

Proposition 3 (Mean preservation, contraction, and class margin)**.** *Fix a group with input score vectors x_i, mean μ, and a common blend $\beta \in [0,1]$. For $x_i' = (1-\beta)x_i + \beta\mu$, the mean is unchanged and*

$$x_i' - \mu = (1-\beta)(x_i - \mu), \qquad \sum_i \|x_i' - \mu\|^2 = (1-\beta)^2 \sum_i \|x_i - \mu\|^2. \tag{20}$$

Pairwise differences contract by the same factor $1-\beta$. If the mean has unique winning class y with margin $\delta_{\mathrm{cls}} = \min_{c\neq y}(\mu_y - \mu_c) > 0$, every member satisfying

$$2(1-\beta)\|x_i - \mu\|_\infty < \delta_{\mathrm{cls}} \tag{21}$$

predicts y after the update.

Proof. Averaging x_i' gives $(1-\beta)\mu + \beta\mu = \mu$. Subtracting this mean gives the first contraction identity; squaring and summing proves the second. Subtracting two updated members proves the pairwise statement. For any $c \neq y$,

$$x_{i,y}' - x_{i,c}' = (\mu_y - \mu_c) + (1-\beta)\big((x_i - \mu)_y - (x_i - \mu)_c\big) \geq \delta_{\mathrm{cls}} - 2(1-\beta)\|x_i - \mu\|_\infty.$$

The strict margin condition makes all these differences positive.

The gated update uses $\beta = \alpha_{\mathrm{R}}$ for accepted groups and $\beta = 0$ otherwise. For a pure group with common target $\boldsymbol{\nu}$,

$$\frac{1}{|\mathcal{I}_r|}\sum_i \|x_i' - \boldsymbol{\nu}\|^2 = \|\mu - \boldsymbol{\nu}\|^2 + \frac{(1-\beta)^2}{|\mathcal{I}_r|}\sum_i \|x_i - \mu\|^2.$$

Thus dispersion decreases while mean bias is unchanged. This target-error identity need not hold for mixed groups, and contraction on feature groups does not guarantee improvement on GT-defined supports.

A.6 GT-DEFINED NOISE METRICS

Scope. Table 4 uses Cityscapes (500 images), Context59 (300-image subset), and ADE20K (300-image subset). All configurations share the evaluated images and GT supports.

Ground-truth supports. For a patch i with GT class y_i, define the same-class neighbors

$$\mathcal{N}_{\mathrm{GT}}(i) = \{j : j \text{ is 8-adjacent to } i,\ y_j = y_i\}. \tag{22}$$

Let $\mathcal{R}_{\mathrm{GT}}$ be the 8-connected components of valid same-class patches, $\mathbf{s}_i$ the stage score vector, and $\hat{y}_i = \arg\max_c s_{i,c}$. These GT supports are independent of the experts' feature neighbors and spectral groups; they are semantic regions, not instance masks.

Shared evaluation supports. All configurations share Edge scores, VFM features, patch grid, class vocabulary, and GT map. Node-only and Region-only take $\mathbf{S}^{\mathrm{E}}$ as input; the sequence passes Node scores to Region. Diagnostics use one base-scale patch grid and pre-softmax scores before κ, pixel interpolation, or multi-scale averaging.

Patch GT is the modal valid pixel label in each footprint. Patches with no valid labels or tied modes are discarded and act as connectivity barriers. Both GT regions and predicted pieces use 8-connectivity.

Patch-level metrics. Let $\mathcal{V}_+$ contain valid patches with at least one same-class GT neighbor, and let

$$\bar{\mathbf{s}}_{\mathcal{N}}(i) = \frac{1}{|\mathcal{N}_{\mathrm{GT}}(i)|} \sum_{j \in \mathcal{N}_{\mathrm{GT}}(i)} \mathbf{s}_j. \tag{23}$$

With mode breaking prediction ties by the smallest class index, define

$$\mathrm{Agree}(\mathbf{S}) = \frac{1}{|\mathcal{V}_+|} \sum_{i \in \mathcal{V}_+} \mathbf{1}[\hat{y}_i = \mathrm{mode}\{\hat{y}_j : j \in \mathcal{N}_{\mathrm{GT}}(i)\}], \tag{24}$$

$$\mathrm{Dev}(\mathbf{S}) = \frac{1}{|\mathcal{V}_+|} \sum_{i \in \mathcal{V}_+} \|\mathbf{s}_i - \bar{\mathbf{s}}_{\mathcal{N}}(i)\|_2. \tag{25}$$

The query is excluded from its neighbor set. Patches without valid neighbors are omitted; an image with no eligible patches is excluded only from the means of these two metrics.

Region-level metrics. For a GT region $R \in \mathcal{R}_{\mathrm{GT}}$, let $\bar{\mathbf{s}}_R = |R|^{-1} \sum_{i \in R} \mathbf{s}_i$ and let $\mathrm{CC}_8(B)$ count the 8-connected components in patch set B, with an empty set contributing zero. Define

$$\mathrm{Frag}(\mathbf{S}) = \frac{1}{|\mathcal{R}_{\mathrm{GT}}|} \sum_{R \in \mathcal{R}_{\mathrm{GT}}} \sum_{c=1}^{C} \mathrm{CC}_8(\{i \in R : \hat{y}_i = c\}), \tag{26}$$

$$\mathrm{Disp}(\mathbf{S}) = \frac{1}{|\mathcal{R}_{\mathrm{GT}}|} \sum_{R \in \mathcal{R}_{\mathrm{GT}}} \frac{1}{|R|} \sum_{i \in R} \|\mathbf{s}_i - \bar{\mathbf{s}}_R\|_2. \tag{27}$$

Connectivity is restricted to R, so paths outside it do not connect fragments. Nonempty regions receive equal weight; singletons contribute one fragment and zero dispersion.

Aggregation and accuracy. Noise metrics are averaged per image and then across the evaluated images, subject to eligibility. Patch-grid mIoU uses one dataset-level confusion matrix over valid patches and averages class IoUs with nonzero union. All configurations share the same valid GT masks.

Because Dev and Disp depend on score magnitude, comparisons use the same score scale and patch grid. They are interpreted with mIoU and the discrete metrics: a consistently wrong region can have $\mathrm{Frag} = 1$ and $\mathrm{Disp} = 0$.

A.7 HYPERPARAMETERS AND IMPLEMENTATION SETTINGS

The following settings are used for the SHAPE results in Tables 1 and 2.

Table 5: Dataset-specific hyperparameters used in SHAPE.

Parameter	V21	PC60	C-Obj	V20	PC59	C-Stf	City	ADE
λ_{text}	0.03	0.04	0.01	0.02	0.04	0.02	0.03	0.05
λ_{global}	0.10	0.25	0.25	0.40	0.25	0.20	0.10	0.30
h_{text}	0.10	0.09	0.10	0.10	0.09	0.02	0.14	0.06

V21: VOC21; PC60: Context60; C-Obj: COCO-Object; V20: VOC20; PC59: Context59; C-Stf: COCO-Stuff; City: Cityscapes; ADE: ADE20K.

Table 5 summarizes the dataset-specific text and fusion settings: λ_{text}, λ_{global}, and h_{text} control text adjustment, global–local score fusion, and hierarchical text clustering, respectively (Appendix A.1).

Table 6: Shared expert and inference hyperparameters.

Component	Parameter	Description	Value
Edge	r_{E}	Retained spectral components	30
	τ_{E}	Spectral decay coefficient	0.5
	λ_{fuse}	Cross-model affinity fusion strength	10
Node	K	Feature-space neighbors	10
	α_{N}	Neighborhood score blending weight	0.5
Region	r_{eig}	Spectral embedding dimensions	8
	M	Spectral groups	16
	α_{R}	Group-mean score blending weight	0.25
	τ_{R}	Prediction agreement threshold	0.5
Inference	$\mathcal{S}$	Input scales	$\{1.0, 1.5\}$
	κ	Final score multiplier	40
	θ_{bg}	Background probability threshold (when applicable)	0.1

For Edge spectral rescaling, the default $g_{\mathcal{D}}$ maps the input range $[a, b]$ of the filtered eigenvalues to $[0.5a,\ 1.5b]$; it is the identity mapping on VOC20 and Cityscapes and the default on the remaining datasets.

A.8 COMPOSITION OF THE NODE AND REGION EXPERTS

This eight-benchmark comparison complements the noise measurements in Table 4. All variants share the same Edge output, with Node and Region evaluated individually and in combination.

Table 7: Ablation of Node and Region Expert composition across eight benchmarks (mIoU).

Method	VOC20	VOC21	PC59	PC60	City.	ADE.	C-Obj	C-Stf	Avg.
Edge	87.7	64.6	40.4	38.1	40.9	20.7	36.8	26.9	44.5
Edge+Node	87.9	65.7	40.9	38.5	41.6	21.0	37.5	27.4	45.1
Edge+Region	87.9	66.0	40.9	38.5	41.5	20.9	37.4	27.3	45.1
Edge+Node+Region (Ours)	**87.9**	**66.9**	**41.2**	**38.8**	**42.1**	**21.1**	**37.9**	**27.7**	**45.5**

Adding either Node or Region to the common Edge output improves mIoU on all eight benchmarks. Their combination achieves the highest average mIoU, supporting the use of both refinement stages in the full three-expert pipeline.

A.9 SUPPORT ABLATION FOR THE NODE EXPERT

We compare three support-selection rules for the Node Expert (Table 8): spatial neighbors, CLIP-feature neighbors, and VFM-feature neighbors.

Table 8: Node support selection on Cityscapes ($n = 500$) and Context59 ($n = 2000$; hence its mIoU differs from the 300-image subset in Table 4). Purity is the fraction of support patches sharing the query GT class; mIoU is measured on the patch grid.

	Cityscapes		Context59	
Support / operator	Purity ↑	mIoU ↑	Purity ↑	mIoU ↑
K nearest grid patches	0.859	37.56	0.838	35.82
kNN in CLIP features	0.917	37.47	0.917	35.82
kNN in VFM features (ours)	**0.927**	**38.40**	**0.924**	**36.39**

VFM-feature neighbors achieve the highest purity and mIoU on both datasets. CLIP-feature neighbors substantially improve purity over spatial neighbors but do not improve mIoU, indicating that support purity alone does not fully explain the benefit of VFM-based selection.

A.10 SENSITIVITY TO NEIGHBORHOOD AND REGION PARAMETERS

Setup. We vary one parameter at a time under single-scale inference (Table 9), using the same qkv affinity construction, spectral filter, and head assignment as the main method. For Node, we vary K with $\alpha_{\mathrm{N}} = 0.5$. For Region, we vary M with $\tau_{\mathrm{R}} = 0.5$ or τ_{R} with $M = 16$, fixing $\alpha_{\mathrm{R}} = 0.25$. Experiments use Cityscapes ($n = 500$), Context59 ($n = 2000$), VOC21 ($n = 1449$), and ADE20K ($n = 2000$). Shared defaults are $K = 10$, $M = 16$, and $\tau_{\mathrm{R}} = 0.5$.

Table 9: Sensitivity of the Node and Region Experts to shared hyperparameters (mIoU). Bold indicates the shared default used in the full SHAPE configuration.

Parameter	Cityscapes	Context59	VOC21	ADE20K
Node only: neighborhood size				
$K = 5$	38.81	36.92	63.87	20.37
$\mathbf{K = 10}$	**39.11**	**37.10**	**64.35**	**20.44**
$K = 20$	39.32	37.30	65.00	20.47
$K = 30$	39.43	37.49	65.41	20.41
Region only: cluster count ($\tau_{\mathrm{R}} = 0.5$)				
$M = 8$	38.68	37.16	65.23	20.28
$\mathbf{M = 16}$	**38.73**	**37.18**	**65.07**	**20.33**
$M = 32$	38.90	37.02	64.59	20.34
Region only: agreement threshold ($M = 16$)				
$\tau_{\mathrm{R}} = 0.0$	37.09	37.21	65.41	20.29
$\tau_{\mathrm{R}} = 0.3$	38.20	37.23	65.34	20.29
$\tau_{\mathrm{R}} = \mathbf{0.5}$	**38.74**	**37.18**	**65.07**	**20.32**
$\tau_{\mathrm{R}} = 0.7$	38.63	36.94	64.29	20.24

Neighborhood size. Larger K improves performance on Cityscapes, Context59, and VOC21, while ADE20K is relatively insensitive. We nevertheless retain $K = 10$ as a fixed shared setting rather than tuning the neighborhood size to individual datasets. This choice keeps the support relatively selective while using the same configuration across benchmarks.

Region granularity and selective sharing. Performance is only mildly sensitive to M, with no single value best across all datasets. The effect of τ_{R} is more dataset dependent, indicating that agreement gating can both suppress unreliable sharing and withhold useful sharing. We use the shared defaults $M = 16$ and $\tau_{\mathrm{R}} = 0.5$ across datasets.

Numerical precision. Repeated default Region-only runs differed by 0.01 mIoU on Cityscapes and ADE20K. We therefore treat differences of approximately 0.01–0.02 mIoU as numerical ties.