Beyond Pixel-wise Supervision: Local Structure Regularization for Semantic Segmentation

Wei Huang[1,2] Chenying Liu[1,2] Yilei Shi[1]
Zhitong Xiong[1] Xiao Xiang Zhu[1,2]
[1]Chair of Data Science in Earth Observation
Technical University of Munich, 80333 Munich, Germany
[2]Munich Center for Machine Learning, 80333 Munich, Germany
{w2wei.huang, chenying.liu, yilei.shi,
zhitong.xiong, xiaoxiang.zhu}@tum.de

Abstract

Visual foundation models (VFMs) provide strong region-level representations for semantic segmentation, substantially improving high-level semantic understanding. However, accurate boundary recovery remains challenging, especially for fine structures and semantic transitions. A key reason is that conventional pixel-level supervision largely treats pixels independently and lacks explicit modeling of local structure and neighboring-pixel relationships. To address this issue, we propose a **Local Structure** (LS) regularization framework that constrains the second-order local organization of semantic transition regions. Unlike existing boundary losses that mainly supervise contour location, distance, overlap, or topology, LS models boundary neighborhoods through a structure-tensor representation of semantic probability fields. It decomposes local boundary structure into two complementary cues: **Local Structure Magnitude** (LSM), which measures the strength of anisotropic class transitions, and **Local Structure Orientation** (LSO), which describes their dominant geometric orientation. LS is model-agnostic, compatible with different VFM-backbone–decoder combinations, and introduces no extra inference cost. Extensive experiments on four diverse datasets show that LS consistently improves boundary-sensitive performance while also enhancing overall IoU, demonstrating its effectiveness for boundary-structure learning in semantic segmentation. The code of LS is at `https://github.com/zhu-xlab/Local-Structure-Regularization`.

1 Introduction

Semantic segmentation aims to assign a semantic category to each pixel in an image. Recently, large-scale pretrained visual foundation models (VFMs), especially ViT-based ones (Oquab et al., 2024; Strudel et al., 2021; Xie et al., 2021; Cheng et al., 2022), have shown strong performance in semantic segmentation. With rich pretrained priors and transferable representations, these models provide powerful region- and object-level semantic understanding. Consequently, in many downstream segmentation tasks, high-level semantic modeling is no longer the main bottleneck during fine-tuning.

As region-level semantic understanding becomes increasingly reliable, further performance improvement depends more on semantic boundary quality, yet accurate recovery of fine structures remains challenging. Although boundary pixels occupy only a small fraction of an image and are often under-emphasized by standard region-overlap metrics such as overall Intersection over Union (IoU), they play a disproportionately important role in visual quality and boundary-aware evaluation (Cheng et al., 2021; Takikawa et al., 2019). Moreover, semantic boundaries not only determine contour localization accuracy, but also affect the local structure consistency of class-transition regions.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Existing semantic segmentation methods still mainly rely on pixel-wise supervision, such as cross-entropy (CE). These losses are effective for semantic discrimination, but they supervise pixels individually and provide limited constraints on how neighboring pixels are organized. In contrast, semantic boundaries are local transition structures whose quality depends on both accurate contour localization and coherent neighborhood geometry. Boundary-aware losses further improve contour localization through diverse forms of supervision, including boundary alignment, distance consistency, overlap-based optimization, and topology preservation. However, most of them impose constraints on boundary pixels or regions, rather than explicitly modeling the local structure of semantic transitions, such as boundary response strength and dominant orientation. This limitation becomes more important in VFM-based segmentation, where strong region-level semantics have already been learned and further improvement increasingly relies on preserving fine local structures.

To address this issue, we move beyond pixel-wise supervision and explicitly model the **Local Structure** (LS) around semantic boundaries. LS characterizes local semantic transitions from two complementary aspects: transition strength and dominant orientation. This formulation leads to two cues defined in this work, namely **Local Structure Magnitude** (LSM) and **Local Structure Orientation** (LSO), as shown in Fig. 1. LSM captures the strength of local anisotropic transitions, while LSO describes their geometric orientation. Although inspired by the classical structure tensor, LS adapts and extends it into a differentiable and trainable regularization objective on semantic probability fields, enabling LSM and LSO to supervise deep segmentation models end-to-end.

Building on this formulation, we propose LS regularization for semantic segmentation. Rather than replacing conventional pixel-wise supervision, LS augments it with an auxiliary loss imposed on semantic boundary-buffer regions. By jointly regularizing LSM and LSO, LS encourages predictions to preserve not only semantic correctness, but also coherent local boundary structures.

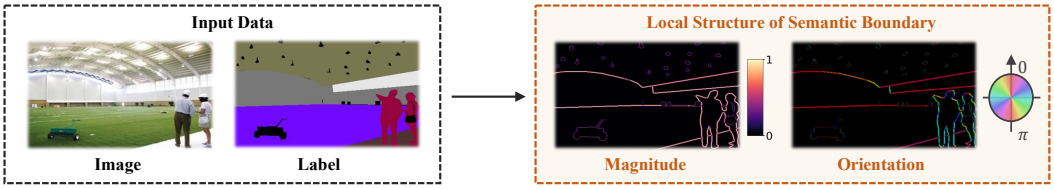


Figure 1: Overview of the proposed LS regularization, which models semantic boundary structure with LSM and LSO and enforces prediction-target consistency within boundary-buffer regions.

Extensive experiments on three generic computer vision datasets, including ADE20K, COCO, and Cityscapes, and a remote sensing semantic segmentation dataset, ISPRS-Potsdam, demonstrate that the proposed method consistently improves boundary quality while maintaining or even improving overall IoU. Moreover, LS introduces only small training-time overhead, incurs no extra inference complexity, and can be readily integrated with different VFM backbones and segmentation decoders, making it a practical plug-in enhancement for semantic segmentation.

Our main contributions are threefold:

- We revisit a remaining limitation of VFM-based semantic segmentation: although pretrained VFMs provide strong region-level semantics, they do not necessarily ensure coherent local boundary structures in dense predictions.
- We adapt and extend classical structure-tensor analysis from scalar image fields to multi-channel semantic probability fields. Based on this formulation, we derive two complementary cues, **Local Structure Magnitude** (LSM) and **Local Structure Orientation** (LSO), to characterize the strength and orientation of semantic transitions.
- We develop a class-aware and boundary-buffer-weighted **LS** regularization loss that enables LSM and LSO to provide end-to-end supervision for local boundary-structure learning. The proposed loss is model-agnostic, introduces no inference overhead, and consistently improves both boundary-sensitive and overall segmentation performance.

2 Related Work

2.1 Semantic segmentation with visual foundation models

Recent semantic segmentation systems increasingly build on large-scale pretrained ViT-based visual backbones. Earlier representative models include ViT itself (Dosovitskiy et al., 2021) and masked-pretraining variants such as MAE (He et al., 2022), while more recent visual foundation models (VFMs), such as DINOv2 (Oquab et al., 2024), DINOv3 (Siméoni et al., 2025), and RADIOv2.5 (Heinrich et al., 2025), further improve the quality and transferability of visual representations through large-scale self-supervised pretraining. These backbones provide strong patch- and object-level semantic features, making them increasingly attractive for semantic segmentation.

To adapt such backbones to dense prediction, existing methods typically combine them with different prediction heads or decoders, ranging from simple linear heads to stronger dense prediction modules such as UPerNet (Xiao et al., 2018) and DPT (Ranftl et al., 2021). In addition, ViT-Adapter explicitly introduces dense-prediction inductive bias to better transfer plain ViT features to segmentation and related dense prediction tasks (Chen et al., 2023). In practice, these backbone–decoder or adapter combinations constitute the dominant technical route of VFM-based segmentation models. Related studies in remote sensing explore multiscale cross-domain fusion for joint hyperspectral and LiDAR classification (Song et al., 2025) and robust image–text retrieval under noisy correspondence (Song et al., 2026).

This paradigm provides a solid foundation for semantic segmentation, with strong region-level semantics and transferability. However, strong semantic representations do not automatically guarantee coherent local boundary structures in dense predictions. Therefore, further improvement still requires explicit regularization of local boundary organization.

2.2 Boundary-aware and structure-aware supervision

The dominant supervision for semantic segmentation remains pixel-wise or region-based losses, such as CE, Dice, and IoU-oriented objectives (Berman et al., 2018). These losses mainly optimize per-pixel category assignment or region overlap, but provide limited constraints on how neighboring pixels should be structurally organized around semantic boundaries.

To move beyond isolated pixel classification, many works introduce explicit boundary-aware or structure-aware supervision. Representative examples include Gated-SCNN (Takikawa et al., 2019), which enhances boundary localization through a dedicated shape stream, Active Boundary Loss (Wang et al., 2022), which encourages boundary alignment, Boundary Loss (Kervadec et al., 2021), which emphasizes contour consistency from a distance-based perspective, and clDice (Shit et al., 2021), which improves topological preservation by constraining soft centerline overlap. More recent studies also explore boundary-enhanced feature bridging (An et al., 2025) and topology-preserving persistent feature matching (Wen et al., 2025).

These methods enhance boundary-aware supervision from different perspectives, but their objectives differ from the proposed LS regularization. Boundary Loss and distance-map losses encode contour distance, ABL focuses on boundary alignment, clDice emphasizes topological connectivity, and Boundary DoU optimizes boundary-region overlap. In contrast, LS explicitly models second-order structural relationships among neighboring pixels in semantic transition regions. By jointly constraining anisotropic magnitude and dominant orientation through differentiable structure-tensor analysis, LS provides a complementary geometric prior beyond contour-distance, alignment-, overlap-, or topology-oriented supervision.

3 Local Structure Regularization for Semantic Segmentation

This section presents the proposed Local Structure (LS) regularization, as illustrated in Fig. 2. We first introduce the generic VFM-based semantic segmentation framework, then construct a boundary-buffer region to localize semantic transitions. Based on this region, we model semantic boundary structure using a structure-tensor-based representation, which consists of Local Structure Magnitude (LSM) and Local Structure Orientation (LSO). Finally, we formulate the LS consistency loss and integrate it into the overall training objective.

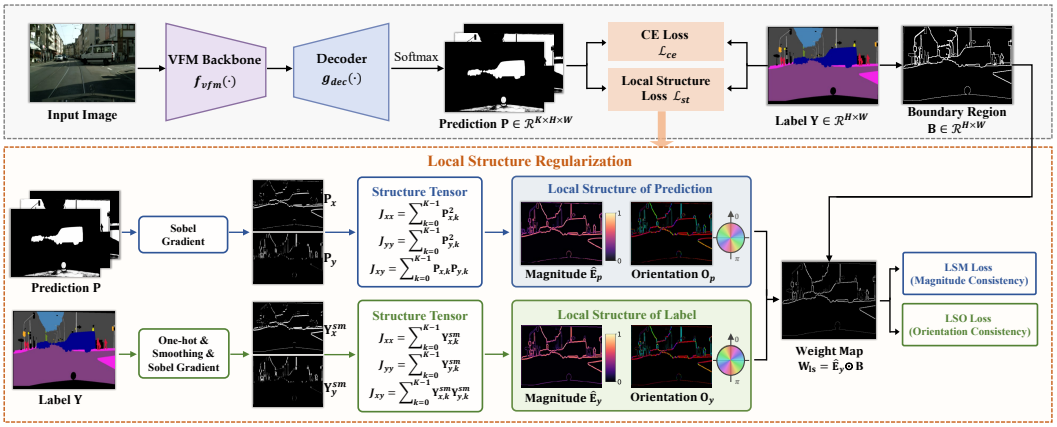


Figure 2: Overview of the proposed LS regularization. Given the prediction and target label maps, LS constructs a boundary-buffer region and computes structure-tensor-based LSM and LSO representations for both branches. It then enforces magnitude and orientation consistency within the boundary-buffer region to regularize local semantic-transition structures.

3.1 Segmentation prediction generation

Our method is built on a generic VFM-based semantic segmentation framework. Given an input RGB image $\mathbf{X} \in \mathbb{R}^{3\times H\times W}$, we feed it into a VFM backbone and a segmentation decoder to obtain the prediction probability map:

$$\mathbf{P} = \mathrm{Softmax}(g_{\mathrm{dec}}(f_{\mathrm{vfm}}(\mathbf{X}))), \qquad \mathbf{P} \in [0,1]^{K\times H\times W}, \tag{1}$$

where $f_{\mathrm{vfm}}(\cdot)$ and $g_{\mathrm{dec}}(\cdot)$ denote the VFM backbone and the segmentation decoder, respectively, and K is the number of semantic classes. The proposed LS regularization is imposed on the output probability field $\mathbf{P}$, and therefore does not depend on a specific backbone or decoder architecture.

3.2 Boundary-buffer region construction

Semantic boundaries are not isolated pixels but local transition regions where neighboring pixels jointly form structured class changes. Therefore, instead of applying LS regularization over the entire image, we first construct a boundary-buffer mask from the target label map.

Given a label map $\mathbf{Y} \in \{0,1,\ldots,K-1\}^{H\times W}$, we convert it into a K-channel one-hot map $\mathbf{Y}^{\mathrm{oh}} \in \{0,1\}^{K\times H\times W}$. Based on $\mathbf{Y}^{\mathrm{oh}}$, we identify semantic transition regions by checking whether a local neighborhood contains both foreground and background responses for each class. Specifically, the boundary-buffer mask is defined as

$$\mathbf{B} = \max_{k\in\{0,\ldots,K-1\}} \left[\mathbb{I}\Big(\mathrm{MaxPool}_{k_b}(\mathbf{Y}_k^{\mathrm{oh}}) - \mathrm{MinPool}_{k_b}(\mathbf{Y}_k^{\mathrm{oh}}) > 0\Big)\right], \tag{2}$$

where $\mathbf{B} \in \{0,1\}^{H\times W}$ denotes the binary boundary-buffer mask, k_b is the neighborhood size, and $\mathbb{I}(\cdot)$ is the indicator function.

Since each channel of $\mathbf{Y}^{\mathrm{oh}}$ is binary-valued, the term

$$\mathrm{MaxPool}_{k_b}(\mathbf{Y}_k^{\mathrm{oh}}) - \mathrm{MinPool}_{k_b}(\mathbf{Y}_k^{\mathrm{oh}}) \tag{3}$$

is positive only when a class transition exists within the local neighborhood. Therefore, $\mathbf{B}$ forms a boundary-buffer band rather than a one-pixel contour. This design is suitable for local structure modeling, because both boundary magnitude and orientation are defined over a local neighborhood. Ignored pixels are further excluded from the supervision region using a valid mask.

3.3 Structure-tensor-based LS representation

Given the boundary-buffer region, local boundary structure is modeled for both the prediction and target branches. Specifically, the classical structure tensor is adapted from scalar image fields to

multi-channel semantic probability fields for differentiable boundary-structure regularization, with additional background provided in Appendix A. The prediction branch directly uses the probability map $\mathbf{P}$, while the target branch uses a channel-wise smoothed one-hot label map:

$$\mathbf{Y}^{\mathrm{sm}} = G_{\mathrm{avg}}^{k_s} * \mathbf{Y}^{\mathrm{oh}}, \tag{4}$$

where k_s is the smoothing kernel size. This step suppresses local label noise and zigzag artifacts around semantic boundaries, providing a more stable target field for subsequent LS estimation.

To unify the formulation, let $\mathbf{U} \in \mathbb{R}^{K\times H\times W}$ denote a generic multi-channel field, which can be either the prediction probability map $\mathbf{P}$ or the smoothed target map $\mathbf{Y}^{\mathrm{sm}}$. We first compute channel-wise spatial gradients using Sobel operators:

$$\mathbf{U}_x = S_x * \mathbf{U}, \qquad \mathbf{U}_y = S_y * \mathbf{U}, \tag{5}$$

where S_x and S_y denote the horizontal and vertical Sobel kernels, respectively. Specifically, for each semantic channel k, the gradients are computed as

$$U_{x,k} = S_x * U_k, \qquad U_{y,k} = S_y * U_k, \qquad k \in \{0, \ldots, K-1\}. \tag{6}$$

These gradients provide first-order local variation cues of the semantic probability or target field.

Based on these gradients, we construct the local structure tensor components as

$$J_{xx}(\mathbf{U}) = \sum_{k=0}^{K-1} U_{x,k}^2, \qquad J_{yy}(\mathbf{U}) = \sum_{k=0}^{K-1} U_{y,k}^2, \qquad J_{xy}(\mathbf{U}) = \sum_{k=0}^{K-1} U_{x,k}U_{y,k}. \tag{7}$$

Here, J_{xx} and J_{yy} measure local gradient energy along the horizontal and vertical axes, while J_{xy} captures their correlation. Compared with directly using raw gradients, the structure tensor provides a more stable second-order representation of local anisotropic boundary patterns.

Two complementary LS components are then derived from the structure tensor: Local Structure Magnitude and Local Structure Orientation.

Local Structure Magnitude. LSM measures the strength and reliability of local anisotropic transition patterns. It is defined as

$$E(\mathbf{U}) = \sqrt{\left(J_{xx}(\mathbf{U}) - J_{yy}(\mathbf{U})\right)^2 + \left(2J_{xy}(\mathbf{U})\right)^2 + \varepsilon}. \tag{8}$$

A larger value of $E(\mathbf{U})$ indicates a stronger and more dominant local boundary structure, while a smaller value corresponds to weak or ambiguous structural evidence.

Since the absolute scale of $E(\mathbf{U})$ may vary across samples and models, we use its normalized form:

$$\hat{\mathbf{E}}(\mathbf{U}) = \frac{E(\mathbf{U})}{\max E(\mathbf{U}) + \varepsilon}, \tag{9}$$

where the maximum is computed over the spatial region of each training sample. This normalization allows the magnitude term to focus on relative local structural saliency rather than absolute Sobel response scale.

Local Structure Orientation. LSO describes the dominant geometric orientation of the local boundary structure. Instead of explicitly performing eigendecomposition of the 2×2 structure tensor, we use the following double-angle representation:

$$\mathbf{O}(\mathbf{U}) = \frac{\left(J_{xx}(\mathbf{U}) - J_{yy}(\mathbf{U}),\, 2J_{xy}(\mathbf{U})\right)}{\sqrt{\left(J_{xx}(\mathbf{U}) - J_{yy}(\mathbf{U})\right)^2 + \left(2J_{xy}(\mathbf{U})\right)^2 + \varepsilon}}. \tag{10}$$

This representation encodes orientation rather than signed vector direction, and is therefore invariant to direction reversal. This property is desirable for semantic boundary modeling, since the same local boundary pattern should have a consistent orientation regardless of which opposite direction is chosen along the boundary.

By substituting $\mathbf{U} = \mathbf{P}$ and $\mathbf{U} = \mathbf{Y}^{\mathrm{sm}}$, we obtain the LS representations for prediction and target:

$$\hat{\mathbf{E}}_{\mathrm{p}} = \hat{\mathbf{E}}(\mathbf{P}), \quad \hat{\mathbf{E}}_{\mathrm{y}} = \hat{\mathbf{E}}(\mathbf{Y}^{\mathrm{sm}}), \quad \mathbf{O}_{\mathrm{p}} = \mathbf{O}(\mathbf{P}), \quad \mathbf{O}_{\mathrm{y}} = \mathbf{O}(\mathbf{Y}^{\mathrm{sm}}). \tag{11}$$

3.4 LS consistency loss

Based on the above representation, LS regularization enforces consistency between prediction and target in both magnitude and orientation. Since not all boundary-buffer pixels provide equally reliable structural information, we use the normalized target magnitude to modulate the supervision strength:

$$\mathbf{W}_{\mathrm{ls}} = \mathbf{B} \odot \hat{\mathbf{E}}_{\mathrm{y}}, \tag{12}$$

where $\odot$ denotes element-wise multiplication. This weighting restricts the LS loss to boundary-buffer regions and assigns larger weights to pixels with stronger target structural evidence.

The LSM consistency loss is defined as

$$\mathcal{L}_{\mathrm{ls}}^{\mathrm{mag}} = \frac{\sum_{u,v} \mathbf{W}_{\mathrm{ls}}(u,v) \left| \hat{\mathbf{E}}_{\mathrm{p}}(u,v) - \hat{\mathbf{E}}_{\mathrm{y}}(u,v) \right|}{\sum_{u,v} \mathbf{W}_{\mathrm{ls}}(u,v) + \varepsilon}. \tag{13}$$

This term encourages the prediction to reproduce the target's local structural strength around semantic boundaries.

The LSO consistency loss is defined as

$$\mathcal{L}_{\mathrm{ls}}^{\mathrm{ori}} = \frac{\sum_{u,v} \mathbf{W}_{\mathrm{ls}}(u,v) \left[1 - \langle \mathbf{O}_{\mathrm{p}}(u,v), \mathbf{O}_{\mathrm{y}}(u,v) \rangle \right]}{\sum_{u,v} \mathbf{W}_{\mathrm{ls}}(u,v) + \varepsilon}, \tag{14}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product between two normalized LSO descriptors. This term encourages the predicted local boundary orientation to be aligned with the target orientation.

The final LS loss combines the two complementary consistency terms:

$$\mathcal{L}_{\mathrm{ls}} = \mathcal{L}_{\mathrm{ls}}^{\mathrm{mag}} + \mathcal{L}_{\mathrm{ls}}^{\mathrm{ori}}. \tag{15}$$

Therefore, LS regularization provides joint supervision on both the strength and orientation of local semantic transitions.

3.5 Overall training objective

The final training objective is formulated as

$$\mathcal{L} = \mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{ls}} \mathcal{L}_{\mathrm{ls}}, \tag{16}$$

where $\mathcal{L}_{\mathrm{ce}}$ denotes the standard pixel-wise CE loss, and λ_{ls} controls the contribution of LS regularization. In this work, λ_{ls} is selected from $\{0.03, 0.1, 0.3, 1.0\}$. The proposed LS loss is used only during training. During inference, the segmentation model remains unchanged and follows the same procedure as the CE baseline, introducing no additional inference cost.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate the proposed method on four datasets, namely **ADE20K** (Zhou et al., 2017), **COCO** (Lin et al., 2014), **Cityscapes** (Cordts et al., 2016), and **ISPRS-Potsdam** (Rottensteiner et al., 2012). ADE20K, COCO, and Cityscapes are widely used semantic segmentation benchmarks in generic computer vision, covering scene parsing, object-centric settings, and urban street scenes, respectively. For these three datasets, we use their official train/validation splits. For ISPRS-Potsdam, a high-resolution remote sensing benchmark with fine structural details, we randomly construct a 6:4 train–validation split once and keep it fixed across all experiments.

Models. Our method is evaluated on VFM-based segmentation models with different pretrained backbones and decoders. Specifically, we consider DINOv2-Small/Base (Oquab et al., 2024), DINOv3-Small/Base (Siméoni et al., 2025), and RADIOv2.5-Base (Heinrich et al., 2025) as backbones, combined with Linear, UPerNet (Xiao et al., 2018), and DPT (Ranftl et al., 2021) decoders.

Training protocol and evaluation metrics. In the main experiments, each VFM backbone and decoder are jointly fine-tuned end-to-end. We use AdamW with $\beta = (0.9, 0.999)$, weight decay 0.01, backbone/decoder learning rates $5 \times 10^{-6}/2 \times 10^{-4}$, and polynomial decay with power 0.9. Training uses four A100 GPUs; per-GPU batch sizes are 8 for ADE20K, COCO, and ISPRS-Potsdam, and 4 for Cityscapes. CE and CE+LS share initialization, sampling, augmentation, and optimization settings within each configuration. Table 1 reports actual patch-aligned crops and training budgets. LS adds no trainable parameters and is discarded at inference. Trainable parameter counts and frozen-encoder controls appear in Appendix G.

The boundary-buffer size is fixed globally to $k_b = 3$, with 3×3 Sobel filters; target smoothing uses a 3×3 mean filter. Only λ_{ls} is selected from $\{0.03, 0.1, 0.3, 1.0\}$ on validation data (Appendix D); buffer-size sensitivity appears in Appendix I. Main comparisons share seed 12345, with three-seed results in Appendix E. We report mIoU and Boundary mIoU (B-mIoU) (Cheng et al., 2021), using a boundary width of 2% of image height and width.

Table 1: Actual crops, class counts, and training budgets. Global batch sizes are 32, except 16 for Cityscapes.

Dataset	Classes	DINOv2 crop	DINOv3 / RADIO crop	Epochs
ADE20K	150	518×518	512×512	60
COCO	81	518×518	512×512	20
Cityscapes	19	798×798	800×800	180
ISPRS-Potsdam	6	518×518	512×512	60

4.2 Ablation Study

We first conduct an ablation study to examine the contribution of different components in the proposed LS regularization. Specifically, we evaluate its magnitude term LSM, orientation term LSO, and their combination, i.e., the full LS loss, on ADE20K and ISPRS-Potsdam with DINOv2-Small/Base + UPerNet. As shown in Table 2, both LSM and LSO improve over the CE baseline, indicating that local-structure supervision provides useful information beyond pixel-wise classification. The magnitude term yields larger B-mIoU gains, suggesting that LSM is effective in strengthening boundary responses. The orientation term also brings consistent improvements, showing that LSO helps regularize the directional organization of local boundary structures.

Table 2: Ablation study of different components in LS, including the magnitude component LSM, the orientation component LSO, and their combination (LSM+LSO, i.e., the full LS loss)

Loss Component	ADE20K DINOv2-S + UPerNet		ADE20K DINOv2-B + UPerNet		ISPRS-Potsdam DINOv2-S + UPerNet		ISPRS-Potsdam DINOv2-B + UPerNet	
	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU
CE	49.50	38.41	54.50	43.19	81.09	59.03	81.53	60.24
+LSM	50.02 (+0.52)	39.50 (+1.09)	55.14 (+0.64)	44.54 (+1.35)	81.29 (+0.20)	60.16 (+1.13)	81.57 (+0.04)	60.98 (+0.74)
+LSO	49.72 (+0.22)	38.93 (+0.52)	54.66 (+0.16)	43.82 (+0.63)	81.43 (+0.34)	59.55 (+0.52)	82.02 (+0.49)	60.73 (+0.49)
+LSM+LSO (full LS)	**50.04** (+0.54)	**40.08** (+1.67)	**55.44** (+0.94)	**45.06** (+1.87)	**81.77** (+0.68)	**60.68** (+1.65)	**82.21** (+0.68)	**61.65** (+1.41)

Combining the two terms achieves the best overall performance across all four settings. The full LS loss improves B-mIoU over CE by $+1.67$, $+1.87$, $+1.65$, and $+1.41$, respectively, consistently outperforming either term alone. These results confirm that LSM and LSO are complementary: LSM enhances local transition strength, while LSO encourages coherent boundary orientation. Their joint use provides more complete local-structure supervision and makes LS an effective plug-in training regularizer without introducing inference overhead.

A practical cost analysis in Appendix C reports 16.15%/5.00% training-time and 21.26%/0.38% memory overhead on ADE20K/Cityscapes, with no inference overhead.

4.3 Model Generalization Experiments and Qualitative Analysis

4.3.1 Different VFM backbones with a unified decoder

We further evaluate the proposed LS regularization under different VFM backbones while fixing the decoder to UPerNet. Specifically, we consider five mainstream backbone variants, namely DINOv2-

Table 3: Comparison under different VFM backbones with a unified decoder (UPerNet). CE+LS denotes CE augmented with the proposed local structure regularization. Values in parentheses indicate the improvement over the corresponding CE baseline. Best results under each metric are in bold.

Backbone	Variant	Method	ADE20K		COCO		Cityscapes		ISPRS-Potsdam	
			mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU
DINOv2	Small	CE	49.50	38.41	64.21	44.21	81.79	77.56	81.09	59.03
		CE+LS	50.04 (+0.54)	40.08 (+1.67)	64.66 (+0.45)	44.80 (+0.59)	82.27 (+0.48)	78.23 (+0.67)	81.77 (+0.68)	60.68 (+1.65)
DINOv2	Base	CE	54.50	43.19	67.28	47.97	83.55	79.79	81.53	60.24
		CE+LS	**55.44** (+0.94)	**45.06** (+1.87)	**67.74** (+0.46)	**49.41** (+1.44)	**83.93** (+0.38)	**80.25** (+0.46)	**82.21** (+0.68)	**61.65** (+1.41)
DINOv3	Small	CE	47.33	35.41	62.23	41.46	80.43	75.84	79.25	55.96
		CE+LS	48.29 (+0.96)	37.00 (+1.59)	62.55 (+0.32)	42.33 (+0.87)	80.77 (+0.34)	76.31 (+0.47)	80.17 (+0.92)	57.76 (+1.80)
DINOv3	Base	CE	51.96	40.51	66.69	46.49	82.51	78.56	80.85	58.49
		CE+LS	53.43 (+1.47)	42.49 (+1.98)	67.18 (+0.49)	47.60 (+1.11)	82.96 (+0.45)	79.18 (+0.62)	81.62 (+0.77)	59.96 (+1.47)
RADIO-v2.5	Base	CE	49.98	38.68	63.07	41.92	81.48	77.05	80.55	57.59
		CE+LS	51.29 (+1.31)	40.55 (+1.87)	63.88 (+0.81)	43.24 (+1.32)	81.98 (+0.50)	77.94 (+0.89)	81.13 (+0.58)	59.27 (+1.68)

Small, DINOv2-Base, DINOv3-Small, DINOv3-Base, and RADIOv2.5, and conduct experiments on all four datasets: ADE20K, COCO, Cityscapes, and ISPRS-Potsdam. Following UniMatch V2 (Yang et al., 2025), the CE baseline on Cityscapes uses the OHEM variant (Shrivastava et al., 2016).

The results are reported in Table 3. LS consistently improves both mIoU and B-mIoU across all tested VFM backbones and datasets, with more pronounced gains on boundary-sensitive metrics. The best B-mIoU improvements reach 1.98 on ADE20K, 1.44 on COCO, 0.89 on Cityscapes, and 1.80 on ISPRS-Potsdam, while the mIoU gains reach up to 1.47, 0.81, 0.50, and 0.92, respectively. Under the best-performing DINOv2-Base + UPerNet setting, CE+LS achieves the best performance on all four datasets. These results demonstrate that LS effectively enhances local boundary-structure modeling and generalizes across different pretrained visual representations.

The consistent boundary gains across strong VFM backbones show that pretrained region-level semantics alone are not sufficient for producing coherent boundary structures in dense prediction, which further supports our motivation to explicitly supervise local structures along semantic boundaries.

4.3.2 Different decoders with a unified backbone

We further evaluate the compatibility of LS with different segmentation decoders by fixing the backbone to DINOv2-Base and comparing Linear, UPerNet, and DPT. These decoders represent different levels of prediction capacity: Linear is a lightweight projection head, UPerNet emphasizes multi-scale contextual aggregation, and DPT is designed for detail-preserving dense prediction from transformer features.

Table 4: Comparison under different decoders with a unified backbone (DINOv2-Base). CE+LS denotes CE augmented with the proposed local structure regularization. Values in parentheses indicate the improvement over the corresponding CE baseline. Best results under each metric are in bold.

Decoder	Method	ADE20K		COCO		Cityscapes		ISPRS-Potsdam	
		mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU
Linear	CE	53.98	42.67	66.17	46.32	82.92	79.06	80.98	59.13
	CE+LS	54.42 (+0.44)	43.28 (+0.61)	66.27 (+0.10)	47.18 (+0.86)	83.20 (+0.28)	79.51 (+0.45)	81.27 (+0.29)	60.40 (+1.27)
UPerNet	CE	54.50	43.19	67.28	47.97	83.55	79.79	81.53	60.24
	CE+LS	**55.44** (+0.94)	**45.06** (+1.87)	**67.74** (+0.46)	**49.41** (+1.44)	83.93 (+0.38)	80.25 (+0.46)	**82.21** (+0.68)	61.65 (+1.41)
DPT	CE	53.40	42.05	66.23	46.32	84.82	81.24	81.71	60.76
	CE+LS	54.17 (+0.77)	43.27 (+1.22)	66.52 (+0.29)	48.02 (+1.70)	**85.11** (+0.29)	**81.64** (+0.40)	82.13 (+0.42)	**61.85** (+1.09)

As shown in Table 4, UPerNet achieves the best results on ADE20K and COCO, where object-level features are important, while DPT performs the best on Cityscapes and obtains the highest B-mIoU on ISPRS-Potsdam, reflecting its advantage in preserving fine spatial details. More importantly, LS consistently improves all decoder baselines, especially on boundary-sensitive metrics. B-mIoU improves by 0.61–1.87 on ADE20K, 0.86–1.70 on COCO, 0.40–0.46 on Cityscapes, and 1.09–1.41 on ISPRS-Potsdam, while mIoU also increases consistently. The gains are generally larger with UPerNet and DPT than with Linear, suggesting that stronger decoders can benefit more from the proposed local-structure supervision.

4.3.3 Interpretability and qualitative analysis

To further examine the effect of LS on local boundary structures, we visualize qualitative results under the best-performing DINOv2-Base + UPerNet setting in Fig. 3. Compared with CE, CE+LS produces stronger boundary LSM responses and more coherent LSO fields aligned with the GT, leading to clearer boundaries and more consistent local geometric structures. This indicates that LS enhances boundary-structure learning beyond pixel-wise classification.

Physical interpretation of LSM and LSO. A semantic boundary corresponds to a spatial transition between class probabilities. LSM and LSO capture two complementary properties: *how strong and distinct the transition is*, and *in which spatial direction it occurs*. Sobel operators measure horizontal and vertical probability changes in each semantic channel. The structure tensor aggregates these responses into a sign-invariant description of local transition geometry. **LSM** measures the strength of the dominant anisotropic transition: a sharp, directionally coherent transition produces a strong response, whereas a weak or ambiguous transition generally produces a lower response. **LSO** describes the dominant direction of probability change, which is normal to the boundary rather than tangent to the contour. Its double-angle representation treats opposite gradient directions as equivalent, since they describe the same unoriented boundary. For example, across a vertical boundary, class probabilities mainly change horizontally. The horizontal Sobel response therefore dominates, and LSO indicates a horizontal transition axis corresponding to the vertical contour. Together, LSM and LSO regularize transition strength and directional consistency. Although represented as spatial maps, both descriptors depend on neighboring probabilities and aggregate information across semantic channels, thereby constraining local structure beyond independent pixel classification. Further derivations appear in Appendix B.

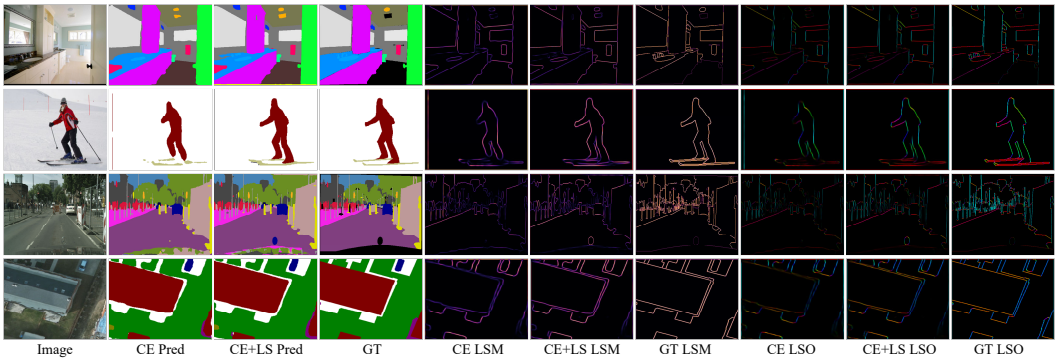


Figure 3: Visual comparison of CE and CE+LS under the DINOv2-Base + UPerNet setting. Rows 1–4 correspond to ADE20K, COCO, Cityscapes, and ISPRS-Potsdam, respectively. LSM and LSO are visualized following the magnitude and orientation indicators in Fig. 1. LS produces stronger boundary LSM responses and more coherent LSO fields aligned with the GT.

4.4 Comparison with Boundary- and Topology-related Losses

After verifying that LS generalizes across different VFM backbones and decoders, we further compare it with representative boundary- and topology-related losses to examine whether explicit local-structure regularization provides a stronger supervision signal. These baselines include Boundary loss (Kervadec et al., 2019), DistanceMap loss (Caliva et al., 2019), clDice (Shit et al., 2021), ABL (Wang et al., 2022), Boundary DoU (Sun et al., 2023), BUS (Choe et al., 2024), and Log-Hausdorff loss (Rao et al., 2025). For BUS, we use its dilation-erosion contrastive objective as a supervised structure-aware baseline. All methods are added to the same CE baseline, with loss weights selected on the validation set and identical training protocols for fair comparison.

As shown in Table 5, existing boundary- and topology-related losses do not consistently improve the CE baseline across datasets. Some methods improve specific metrics on certain datasets, but their gains are often limited or unstable, and several losses even lead to performance degradation. In contrast, LS consistently improves all four datasets in terms of both mIoU and B-mIoU, with particularly clear gains on boundary-sensitive B-mIoU. These results demonstrate that LS provides

more stable and effective boundary-aware supervision than existing boundary- and topology-related losses, confirming the robustness and effectiveness of the proposed local structure regularization.

Appendices H and K additionally evaluate native segmentation models and direct Sobel matching. Additional results in Appendix F show that LS is also complementary to region-level losses such as Dice and Lovász, bringing further gains when combined with them.

Table 5: Comparison with different boundary- and topology-related losses. For each dataset, we use the backbone–decoder combination corresponding to the best baseline result in Tables 3 and 4. Values in parentheses indicate the improvement over the CE baseline. Best results are in bold.

Boundary/Topology Loss	ADE20K DINOv2-B + UPerNet		COCO DINOv2-B + UPerNet		Cityscapes DINOv2-B + DPT		ISPRS-Potsdam DINOv2-B + DPT	
	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU
CE	54.50	43.19	67.28	47.97	84.82	81.24	81.71	60.76
+Boundary (Kervadec et al., 2019)	54.33 (-0.17)	43.16 (-0.03)	67.11 (-0.17)	47.94 (-0.03)	84.56 (-0.26)	80.95 (-0.29)	81.66 (-0.05)	60.88 (+0.12)
+DistanceMap (Caliva et al., 2019)	53.68 (-0.82)	42.03 (-1.16)	66.85 (-0.43)	47.64 (-0.33)	83.52 (-1.30)	79.82 (-1.42)	81.55 (-0.16)	60.30 (-0.46)
+clDice (Shit et al., 2021)	53.54 (-0.96)	41.72 (-1.47)	66.29 (-0.99)	46.51 (-1.46)	83.12 (-1.70)	79.08 (-2.16)	80.84 (-0.87)	58.52 (-2.24)
+ABL (Wang et al., 2022)	54.91 (+0.41)	43.44 (+0.25)	67.03 (-0.25)	47.87 (-0.10)	84.62 (-0.20)	81.00 (-0.24)	81.82 (+0.11)	60.91 (+0.15)
+Boundary DoU (Sun et al., 2023)	55.11 (+0.61)	44.03 (+0.84)	67.40 (+0.12)	48.28 (+0.31)	85.05 (+0.23)	81.51 (+0.27)	81.98 (+0.27)	61.38 (+0.62)
+BUS (Choe et al., 2024)	51.69 (-2.81)	39.50 (-3.69)	63.71 (-3.57)	44.09 (-3.88)	80.60 (-4.22)	76.20 (-5.04)	76.36 (-5.35)	51.63 (-9.13)
+Log-Hausdorff (Rao et al., 2025)	54.30 (-0.20)	43.06 (-0.13)	66.99 (-0.29)	47.74 (-0.23)	84.68 (-0.14)	80.97 (-0.27)	81.53 (-0.18)	59.94 (-0.82)
+LS (ours)	**55.44** (+0.94)	**45.06** (+1.87)	**67.74** (+0.46)	**49.41** (+1.44)	**85.11** (+0.29)	**81.64** (+0.40)	**82.13** (+0.42)	**61.85** (+1.09)

5 Conclusion

In this paper, we proposed LS regularization for semantic segmentation. Unlike conventional pixel-wise supervision, LS explicitly models semantic boundaries as local transition structures and regularizes them through magnitude and orientation consistency derived from structure-tensor analysis. Extensive experiments across diverse datasets, VFM backbones, and decoders show that LS consistently improves boundary-sensitive performance while also enhancing overall mIoU. Comparisons with existing boundary losses further demonstrate the stability and effectiveness of LS.

One limitation is that the current LS formulation uses fixed structural scales and loss weights, while boundary patterns may vary across object sizes and scene complexity. Future work will explore adaptive LS estimation conditioned on local boundary complexity or prediction uncertainty. Nevertheless, LS already provides a simple, architecture-agnostic mechanism for improving local structure learning in semantic segmentation without changing the deployed inference model.

Acknowledgements and Disclosures

Funding. This work was supported by both the Munich Center for Machine Learning (MCML) and the Science and Technology Development Fund (FDCT), Macao SAR, under the International Collaborative Research Grant (0001/2025/AIJ).

Competing interests. The authors declare no competing interests.

References

Tai An, Weiqiang Huang, Da Xu, Qingyuan He, Jiacheng Hu, and Yujia Lou. A deep learning framework for boundary-aware semantic segmentation. In *2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)*, pages 886–890. IEEE, 2025.

Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4413–4421, 2018.

Francesco Caliva, Claudia Iriondo, Alejandro Morales Martinez, Sharmila Majumdar, and Valentina Pedoia. Distance map loss penalty term for semantic segmentation. *arXiv preprint arXiv:1908.03679*, 2019.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *International Conference on Learning Representations (ICLR)*, 2023.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C. Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15334–15342, 2021.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, 2022.

Seun-An Choe, Ah-Hyung Shin, Keon-Hee Park, Jinwoo Choi, and Gyeong-Moon Park. Open-set domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23943–23953, 2024.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.

Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. Radiov2.5: Improved baselines for agglomerative vision foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Hoel Kervadec, Jihene Bouchtiba, Christian Desrosiers, Eric Granger, Jose Dolz, and Ismail Ben Ayed. Boundary loss for highly unbalanced segmentation. In *International conference on medical imaging with deep learning*, pages 285–296. PMLR, 2019.

Hoel Kervadec, Jihene Bouchtiba, Christian Desrosiers, Eric Granger, Jose Dolz, and Ismail Ben Ayed. Boundary loss for highly unbalanced segmentation. *Medical Image Analysis*, 67:101851, 2021.

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 *European conference on computer vision*, pages 740–755. Springer, 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.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021.

Amog Rao, Ananya Shukla, Jia Bhargava, Yangming Ou, and Rina Bao. Spatial prior-guided boundary and region-aware 2d lesion segmentation in neonatal hypoxic ischemic encephalopathy. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 520–530. Springer, 2025.

Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sebastien Benitez, and Uwe Breitkopf. The isprs benchmark on urban object classification and 3d building reconstruction. 2012.

Suprosanna Shit, Johannes C. Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien P. W. Pluim, Ulrich Bauer, and Bjoern H. Menze. cldice – a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16560–16569, 2021.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Qiya Song, Feng Mo, Kexin Ding, Lin Xiao, Renwei Dian, Xudong Kang, and Shutao Li. Mcfnet: Multiscale cross-domain fusion network for HSI and LiDAR data joint classification. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–12, 2025.

Qiya Song, Yiqiang Xie, Yuan Sun, Renwei Dian, and Xudong Kang. Robust remote sensing image-text retrieval with noisy correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9732–9741, 2026.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7262–7272, 2021.

Fan Sun, Zhiming Luo, and Shaozi Li. Boundary difference over union loss for medical image segmentation. In *International conference on medical image computing and computer-assisted intervention (MICCAI)*, pages 292–301. Springer, 2023.

Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5229–5238, 2019.

Chi Wang, Yunke Zhang, Miaomiao Cui, Peiran Ren, Yin Yang, Xuansong Xie, Xian-Sheng Hua, Hujun Bao, and Weiwei Xu. Active boundary loss for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2397–2405, 2022.

Bo Wen, Haochen Zhang, Dirk-Uwe G Bartsch, William Freeman, Truong Nguyen, and Cheolhong An. Topology-preserving image segmentation with spatial-aware persistent feature matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5762–5771, 2025.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12077–12090, 2021.

Lihe Yang, Zhen Zhao, and Hengshuang Zhao. Unimatch v2: Pushing the limit of semi-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):3031–3048, 2025.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Supplementary Material for Beyond Pixel-wise Supervision: Local Structure Regularization for Semantic Segmentation

A Background on the Structure Tensor

The structure tensor is a classical tool for characterizing local geometric organization in images. Given a scalar image field I, its local gradients I_x and I_y describe first-order variations along the horizontal and vertical directions. The corresponding structure tensor is commonly defined as

$$\mathbf{J}(I) = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \tag{17}$$

or as its locally averaged form within a neighborhood. This tensor summarizes the distribution of local gradient energy and provides a second-order description of local image structure. In particular, its anisotropy indicates whether a local region contains a dominant transition pattern, while its principal orientation reflects the dominant geometric direction of that transition.

In semantic segmentation, however, the quantity of interest is no longer a scalar intensity field, but a semantic field defined over multiple classes. Both the model prediction and the target label can be represented in this form: the prediction is a probability map, while the target is converted into a smoothed one-hot map. Therefore, to make structure-tensor analysis applicable to semantic segmentation, we extend it from a scalar image field I to a generic multi-channel semantic field $\mathbf{U} \in \mathbb{R}^{K\times H\times W}$, where $\mathbf{U}$ can denote either the predicted probability map or the smoothed target map. For each class channel, spatial gradients are computed independently, and their gradient energies are then accumulated across classes:

$$J_{xx}(\mathbf{U}) = \sum_{k=0}^{K-1} U_{x,k}^2, \quad J_{yy}(\mathbf{U}) = \sum_{k=0}^{K-1} U_{y,k}^2, \quad J_{xy}(\mathbf{U}) = \sum_{k=0}^{K-1} U_{x,k} U_{y,k}. \tag{18}$$

This class-aware formulation allows the tensor to capture semantic transition structures rather than raw intensity variations. Based on these tensor components, the proposed LS regularization derives two complementary cues: Local Structure Magnitude (LSM), which measures the strength of local anisotropic semantic transitions, and Local Structure Orientation (LSO), which describes their dominant geometric orientation. By matching LSM and LSO between prediction and target fields within boundary-buffer regions, LS provides end-to-end supervision for local boundary-structure learning in deep segmentation models. Therefore, LS can be viewed as an adaptation of classical structure-tensor analysis from scalar image geometry to multi-channel semantic probability fields, making it suitable for differentiable boundary-structure regularization in deep segmentation models.

B Mathematical Interpretation and Scope of LS

For a symmetric positive-semidefinite tensor $\mathbf{J} = \left[\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right]$, the characteristic equation yields

$$\mu_{1,2} = \frac{a + c \pm \sqrt{(a-c)^2 + 4b^2}}{2}. \tag{19}$$

Thus the unstabilized LSM equals the eigenvalue gap $\mu_1 - \mu_2$, whereas the trace $a + c$ equals total gradient energy. A strong response indicates directional dominance. For a principal eigenvector axis at angle θ, diagonalizing $\mathbf{J}$ gives

$$a - c = (\mu_1 - \mu_2)\cos 2\theta, \qquad 2b = (\mu_1 - \mu_2)\sin 2\theta. \tag{20}$$

Dividing by the eigenvalue gap yields the double-angle descriptor. When both gaps are nonzero and numerical stabilization is negligible, the orientation penalty is

$$1 - \langle \mathbf{O}_\mathrm{p}, \mathbf{O}_\mathrm{y} \rangle = 1 - \cos\big(2(\theta_\mathrm{p} - \theta_\mathrm{y})\big) = 2\sin^2(\theta_\mathrm{p} - \theta_\mathrm{y}). \tag{21}$$

It is unchanged by reversing either axis by π. The principal gradient axis describes the transition normal, rather than the contour tangent.

When the eigenvalue gap vanishes, orientation is undefined. The stabilizer ε avoids division by zero, so the descriptor need not have exactly unit norm near degenerate tensors. The angular

identity above holds for nondegenerate tensors in the unstabilized limit. Target-magnitude weighting reduces the contribution of weak target structures, although smoothing, stabilization, and sample-wise normalization affect its numerical value. LS does not guarantee exact topology preservation or recovery of details absent from the target. Its interpretability concerns output-space geometry, rather than individual pretrained feature channels.

C Training Computational Complexity of LS

We evaluate the computational and practical overhead of LS using DINOv2-Base + UPerNet on ADE20K and Cityscapes. Table 6 summarizes the profiled FLOPs, measured training time, and GPU memory consumption. FLOPs are reported per input image, while training time is measured per epoch. All experiments are conducted on four A100 GPUs.

The additional LS computation consists of channel-wise smoothing, Sobel filtering, structure-tensor construction, and magnitude/orientation consistency evaluation. These operations scale linearly with the number of semantic classes and image pixels, with complexity $\mathcal{O}(KHW)$.

Table 6: Computational and practical overhead of LS under DINOv2-Base + UPerNet. FLOPs are profiled per image, while training time is measured per epoch on four A100 GPUs. Relative increases are reported for training time and GPU memory.

Measurement	ADE20K	Cityscapes
Input size	518×518	798×798
Number of classes	150	19
CE configuration (GFLOPs/image)	155.44	480.80
CE+LS configuration (GFLOPs/image)	157.25	481.34
CE training time (s/epoch)	260	100
CE+LS training time (s/epoch)	302	105
Relative training-time increase	16.15%	5.00%
CE GPU memory (MB)	24,826	42,334
CE+LS GPU memory (MB)	30,104	42,494
Relative GPU-memory increase	21.26%	0.38%
Additional inference cost	0	0

As shown in Table 6, LS increases training time by 16.15% on ADE20K and 5.00% on Cityscapes, with corresponding GPU-memory increases of 21.26% and 0.38%. ADE20K incurs greater overhead despite its smaller input size because its 150 semantic channels require more probability-field operations and intermediate storage than the 19 channels of Cityscapes. FLOPs and practical training costs capture different aspects of computation: memory access and GPU-kernel execution costs are not fully reflected in FLOP counts. LS introduces no trainable parameters and is discarded during inference, leaving the deployed architecture and inference procedure unchanged.

D Sensitivity Analysis of λ_{ls}

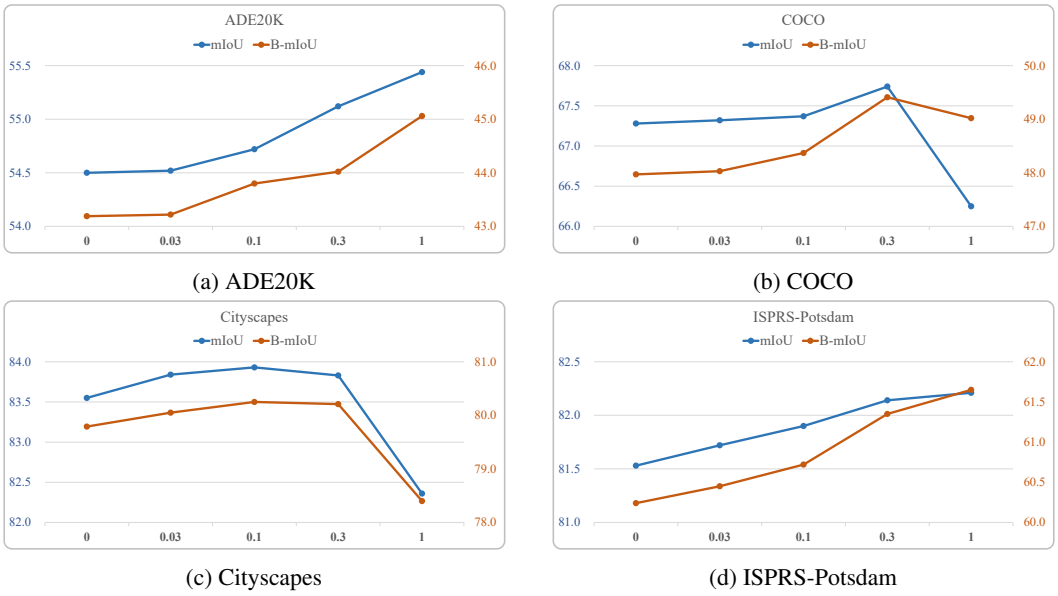
The coefficient λ_{ls} controls the strength of the proposed LS regularization. To examine its effect, we evaluate $\lambda_{\text{ls}} \in \{0.03, 0.1, 0.3, 1.0\}$ under the DINOv2-Base + UPerNet setting on all four datasets, while keeping all other training settings unchanged. Both mIoU and Boundary mIoU (B-mIoU) are reported.

As shown in Table 7 and Fig. 4, LS is effective across a reasonably broad range of loss weights. For small to moderate values, i.e., $\lambda_{\text{ls}} \in \{0.03, 0.1, 0.3\}$, it consistently improves both mIoU and B-mIoU over the CE baseline on all datasets, indicating that local structure supervision provides a stable auxiliary training signal. The optimal value varies across datasets, with $\lambda_{\text{ls}} = 1.0$ performing best on ADE20K and ISPRS-Potsdam, $\lambda_{\text{ls}} = 0.3$ on COCO, and $\lambda_{\text{ls}} = 0.1$ on Cityscapes.

When λ_{ls} is too large, LS may become over-constraining. For example, $\lambda_{\text{ls}} = 1.0$ improves B-mIoU on COCO but decreases mIoU, and degrades both metrics on Cityscapes. Therefore, LS should be

Table 7: Sensitivity analysis of the LS loss weight λ_{ls} under the DINOv2-Base + UPerNet setting. CE denotes the baseline without LS regularization. Values in parentheses indicate the improvement over the CE baseline. Best results under each metric are in bold.

Loss Setting	ADE20K DINOv2-B + UPerNet		COCO DINOv2-B + UPerNet		Cityscapes DINOv2-B + UPerNet		ISPRS-Potsdam DINOv2-B + UPerNet	
	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU
CE	54.50	43.19	67.28	47.97	83.55	79.79	81.53	60.24
CE+LS, $\lambda_{ls} = 0.03$	54.52 (+0.02)	43.22 (+0.03)	67.32 (+0.04)	48.03 (+0.06)	83.84 (+0.29)	80.05 (+0.26)	81.72 (+0.19)	60.45 (+0.21)
CE+LS, $\lambda_{ls} = 0.1$	54.72 (+0.22)	43.80 (+0.61)	67.37 (+0.09)	48.37 (+0.40)	**83.93** (+0.38)	**80.25** (+0.46)	81.90 (+0.37)	60.72 (+0.48)
CE+LS, $\lambda_{ls} = 0.3$	55.12 (+0.62)	44.02 (+0.83)	**67.74** (+0.46)	**49.41** (+1.44)	83.83 (+0.28)	80.21 (+0.42)	82.14 (+0.61)	61.35 (+1.11)
CE+LS, $\lambda_{ls} = 1.0$	**55.44** (+0.94)	**45.06** (+1.87)	66.25 (-1.03)	49.02 (+1.05)	82.36 (-1.19)	78.40 (-1.39)	**82.21** (+0.68)	**61.65** (+1.41)



(a) ADE20K (b) COCO

(c) Cityscapes (d) ISPRS-Potsdam

Figure 4: Sensitivity curves of the LS loss weight λ_{ls} under the DINOv2-Base + UPerNet setting. The results show that LS improves boundary-sensitive performance across a broad range of λ_{ls} values, while overly large weights may over-constrain the prediction on some datasets.

balanced with the primary CE supervision. In practice, moderate values such as 0.1 and 0.3 provide stable improvements, and the final λ_{ls} is selected on the validation set for each setting.

E Multi-seed Results

Considering the substantial computational cost of training VFM-based segmentation models across multiple datasets, backbones, and decoders, the main experiments are conducted under a single but identical random seed for all compared methods. Since each comparison uses the same training protocol, data split, model setting, and random seed, the results remain relatively fair for evaluating the relative effect of different loss functions across datasets and model configurations.

To further examine the stability of the proposed LS regularization beyond a single-seed comparison, we additionally conduct three-seed experiments under representative DINOv2-based settings. Specifically, we evaluate DINOv2-Small/Base with UPerNet on ADE20K and ISPRS-Potsdam, covering both generic scene parsing and high-resolution remote sensing segmentation.

As shown in Table 8, CE+LS consistently outperforms the CE baseline across all tested settings in terms of both mIoU and B-mIoU. The improvements are particularly clear on the boundary-sensitive B-mIoU metric, while the standard deviations remain small across three seeds. These repeated runs support the consistency of gains in the tested settings; they do not establish statistical significance across all model configurations.

Table 8: Mean and standard deviation over three random seeds under DINOv2-based settings.

Backbone	Setting	Loss	mIoU	B-mIoU	Δ mIoU	Δ B-mIoU
DINOv2	ADE20K/Small-UPerNet	CE	49.37 ± 0.19	38.27 ± 0.20	–	–
		CE+LS ($\lambda_{ls} = 1.0$)	**49.99** ± 0.08	**39.89** ± 0.27	**+0.62**	**+1.62**
	ADE20K/Base-UPerNet	CE	54.33 ± 0.25	43.08 ± 0.16	–	–
		CE+LS ($\lambda_{ls} = 1.0$)	**55.36** ± 0.12	**45.10** ± 0.06	**+1.03**	**+2.02**
	ISPRS-Potsdam/Small-UPerNet	CE	81.06 ± 0.05	59.08 ± 0.07	–	–
		CE+LS ($\lambda_{ls} = 1.0$)	**81.75** ± 0.03	**60.55** ± 0.18	**+0.70**	**+1.47**
	ISPRS-Potsdam/Base-UPerNet	CE	81.46 ± 0.10	60.15 ± 0.13	–	–
		CE+LS ($\lambda_{ls} = 1.0$)	**82.19** ± 0.12	**61.54** ± 0.18	**+0.73**	**+1.39**

F Compatibility with Region-Level Losses

Although the proposed LS regularization directly improves the CE baseline, an important question is whether it is complementary to existing region-level losses. To answer this question, we further combine LS with Dice and Lovász (Berman et al., 2018) losses on top of the CE baseline. Here, CE serves as the baseline pixel-wise supervision, while Dice and Lovász are representative region-level losses that encourage region overlap and IoU-oriented optimization, respectively. In contrast, LS focuses on boundary-aware local structural consistency by explicitly regularizing the magnitude and orientation of local structures around object boundaries. Therefore, this experiment evaluates whether LS can provide additional structural supervision beyond existing region-level objectives.

Table 9: Compatibility of the proposed LS with representative region-level losses built upon the CE baseline. CE serves as the baseline pixel-wise loss, while Dice and Lovász are used as region-level losses. For each dataset, we use the backbone–decoder combination corresponding to the best baseline result in Tables 3 and 4. Values in parentheses indicate the improvement over the CE baseline. Best results are in bold.

Loss Setting	ADE20K DINOv2-B + UPerNet		COCO DINOv2-B + UPerNet		Cityscapes DINOv2-B + DPT		ISPRS-Potsdam DINOv2-B + DPT	
	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU	mIoU	B-mIoU
CE	54.50	43.19	67.28	47.97	84.82	81.24	81.71	60.76
+LS (ours)	55.44 (+0.94)	45.06 (+1.87)	67.74 (+0.46)	49.41 (+1.44)	85.11 (+0.29)	81.64 (+0.40)	82.13 (+0.42)	61.85 (+1.09)
+Dice	55.25 (+0.75)	44.89 (+1.70)	67.61 (+0.33)	49.21 (+1.24)	84.86 (+0.04)	81.21 (-0.03)	82.24 (+0.53)	61.39 (+0.63)
+Dice+LS (ours)	**56.08** (+1.58)	**46.05** (+2.86)	**68.09** (+0.81)	49.82 (+1.85)	85.15 (+0.33)	81.69 (+0.45)	**82.43** (+0.72)	**62.22** (+1.46)
+Lovász (Berman et al., 2018)	55.57 (+1.07)	45.14 (+1.95)	67.55 (+0.27)	49.23 (+1.26)	85.21 (+0.39)	81.71 (+0.47)	82.22 (+0.51)	61.51 (+0.75)
+Lovász+LS (ours)	55.75 (+1.25)	45.94 (+2.75)	67.94 (+0.66)	**50.09** (+2.12)	**85.29** (+0.47)	**81.85** (+0.61)	82.33 (+0.62)	62.09 (+1.33)

The results are reported in Table 9. First, adding LS to the CE baseline consistently improves both mIoU and B-mIoU across all datasets. The gains are especially clear on boundary-sensitive performance, with B-mIoU improvements of 1.87, 1.44, 0.40, and 1.09 points on ADE20K, COCO, Cityscapes, and ISPRS-Potsdam, respectively. This shows that local structural supervision provides useful information beyond the standard pixel-wise CE loss.

More importantly, LS remains effective when combined with region-level losses. Compared with CE+Dice, adding LS further improves B-mIoU by 1.16, 0.61, 0.48, and 0.83 points on ADE20K, COCO, Cityscapes, and ISPRS-Potsdam, respectively. Similarly, compared with CE+Lovász, adding LS brings additional B-mIoU gains of 0.80, 0.86, 0.14, and 0.58 points. These improvements demonstrate that LS is not simply duplicating the effect of Dice or Lovász. While Dice and Lovász mainly optimize region overlap or IoU-oriented objectives, LS explicitly constrains local boundary structures. Therefore, LS is complementary to existing region-level losses and can further enhance boundary-sensitive segmentation performance when integrated with them.

G Optimization Details and Frozen-Encoder Control

The main VFM experiments jointly optimize the backbone and decoder, using AdamW with $\beta = (0.9, 0.999)$, weight decay 0.01, backbone learning rate 5×10^{-6}, decoder learning rate 2×10^{-4}, and polynomial decay with power 0.9. Four A100 GPUs are used. Per-GPU batch sizes are 8 on

ADE20K, COCO, and ISPRS-Potsdam and 4 on Cityscapes. CE and CE+LS share initialization, sampling, and optimization within each configuration. Table 10 lists trainable parameters; decoder counts vary with feature dimension and class count. LS introduces no learnable parameters.

Table 10: Trainable parameter counts in the main end-to-end settings.

Component	Configuration	Trainable parameters
Backbone	DINOv2-S	22.1M
Backbone	DINOv2-B	86.6M
Backbone	DINOv3-S	21.6M
Backbone	DINOv3-B	85.7M
Backbone	RADIOv2.5-B	98.2M
Decoder	Linear	0.002–0.115M
Decoder	UPerNet	2.02–2.73M
Decoder	DPT	2.74–10.96M

Frozen-encoder control. On ISPRS-Potsdam, all 86.6M DINOv2-B parameters are frozen and the encoder remains in evaluation mode. Only the 2.71M UPerNet parameters are optimized, with learning rate 2×10^{-4}. Frozen CE and CE+LS share their training settings; LS uses $\lambda_{\text{ls}} = 1.0$. Table 11 shows gains of 0.73 mIoU and 1.52 B-mIoU with the frozen encoder, compared with 0.68 and 1.41 under end-to-end optimization. LS improves segmentation accuracy in both training regimes, while end-to-end fine-tuning achieves higher absolute accuracy.

Table 11: Frozen-encoder control: DINOv2-B + UPerNet on ISPRS-Potsdam.

Encoder	Objective	Trainable parameters	mIoU	B-mIoU
Fine-tuned	CE	89.3M	81.53	60.24
Fine-tuned	CE+LS	89.3M	82.21	61.65
Frozen	CE	2.71M	79.50	56.21
Frozen	CE+LS	2.71M	80.23	57.73

H Native Segmentation Models and Stronger Decoders

We evaluate SegFormer-B5 (Xie et al., 2021) and Swin-B + Mask2Former (Cheng et al., 2022), alongside DINOv2-B + UPerNet. All are fine-tuned end-to-end, with the same general AdamW/poly-decay strategy, data-processing and evaluation protocols, and best-checkpoint selection by validation mIoU. Within each model, CE and CE+LS share initialization and optimization; only the objective changes. Trainable parameter counts are approximately 84.7M, 106.9M, and 89.3M, respectively.

Swin-B supplies hierarchical multi-scale features directly to Mask2Former; a plain DINOv2 ViT would require an additional adapter. This experiment therefore tests the complete native Swin-B + Mask2Former architecture rather than isolating a decoder replacement under an identical backbone. LS is applied without architectural modifications.

Table 12: Native-model comparison. All reported metrics are percentages.

		CE		CE+LS	
Dataset	Model	mIoU	B-mIoU	mIoU	B-mIoU
ADE20K	DINOv2-B + UPerNet	54.50	43.19	55.44	45.06
	Swin-B + Mask2Former	51.95	43.55	53.57	45.32
	SegFormer-B5	49.67	40.09	50.17	41.83
ISPRS-Potsdam	DINOv2-B + UPerNet	81.53	60.24	82.21	61.65
	Swin-B + Mask2Former	80.81	59.36	81.40	60.60
	SegFormer-B5	77.40	53.65	78.06	55.52

On ADE20K, LS improves Mask2Former by 1.62 mIoU/1.77 B-mIoU and SegFormer by 0.50/1.74 points; on ISPRS-Potsdam the gains are 0.59/1.24 and 0.66/1.87. DINOv2-B + UPerNet with

LS obtains the highest mIoU on both datasets and the highest B-mIoU on ISPRS-Potsdam, while Mask2Former with LS has slightly higher ADE20K B-mIoU (45.32 versus 45.06). The consistent gains demonstrate that LS is compatible with both VFM-based and native segmentation architectures under the evaluated training protocol.

I Sensitivity to Boundary-Buffer Size

The default boundary-buffer kernel is $k_b = 3$ for all datasets and model configurations; it is not tuned separately. Using DINOv2-B + UPerNet, we evaluate $k_b \in \{3, 5, 7\}$ on ADE20K and ISPRS-Potsdam with $\lambda_{\text{ls}} = 1.0$, holding splits, batch sizes, GPU counts, and optimization schedules fixed. The results are presented in Table 13.

Table 13: Boundary-buffer sensitivity on ADE20K and ISPRS-Potsdam.

Dataset	Objective / k_b	mIoU	B-mIoU
ADE20K	CE	54.50	43.19
	CE+LS / 3 (default)	55.44	45.06
	CE+LS / 5	55.30	45.15
	CE+LS / 7	55.07	44.91
ISPRS-Potsdam	CE	81.53	60.24
	CE+LS / 3 (default)	82.21	61.65
	CE+LS / 5	82.19	61.58
	CE+LS / 7	82.30	61.59

All evaluated kernel sizes improve both mIoU and B-mIoU over CE on both datasets. On ISPRS-Potsdam, mIoU spans 82.19–82.30 and B-mIoU spans 61.58–61.65, corresponding to ranges of 0.11 and 0.07 percentage points. On ADE20K, B-mIoU gains range from 1.72 to 1.96 points, with $k_b = 3$ achieving the highest mIoU. These results indicate that LS remains effective across the evaluated buffer sizes. We retain the compact $k_b = 3$ globally to provide localized boundary supervision. The loss weight is the only dataset-selected LS hyperparameter; its sensitivity is analyzed in Appendix D.

J Cross-Class Aggregation, Target Smoothing, and Locality

Cross-class aggregation. Across a semantic transition, one class probability typically decreases as another increases. Summing gradient outer products captures their shared geometric transition while removing dependence on class order and gradient sign. The descriptor intentionally discards explicit class-pair identity: two differently labeled but geometrically similar boundaries may have similar LS descriptors. CE remains necessary to supervise semantic categories. At multi-class junctions, the aggregate tensor describes a dominant transition, rather than each pair independently. This class-agnostic geometry avoids dataset-specific transition graphs and uses $\mathcal{O}(KHW)$ operations; explicitly modeling every pair can require $\mathcal{O}(K^2HW)$ computation and larger intermediate memory. Class-pair-aware or channel-weighted variants remain possible extensions.

Target-field smoothing. One-hot annotations have abrupt transitions and rasterization artifacts. A 3×3 mean filter makes the auxiliary target field continuous and more stable. Predictions are already continuous softmax probabilities; further smoothing could attenuate weak transitions that LS should expose. CE still uses the original labels, including thin structures. The symmetric filter smooths around the annotated contour, while fixed-scale averaging may attenuate very thin structures or fine corners. The original labels continue to provide precise pixel-wise CE supervision.

Local support of the structure tensor. Classical tensors often apply an additional spatial average to J_{xx}, J_{yy}, J_{xy}. Here each class gradient already uses a 3×3 Sobel neighborhood, and no additional averaging of gradient products is used. The objective is localized boundary supervision; a larger averaging region may mix nearby contours, corners, and multi-class junctions. Annotation noise is handled by target smoothing. The formulation therefore uses the Sobel neighborhood for local geometry estimation and target smoothing for annotation regularization.

Reading the orientation maps. For a vertical contour, probabilities vary mainly horizontally, so J_{xx} dominates J_{yy} and the principal transition axis is horizontal. Orientation colors encode an axis angle over $[0, \pi)$: similar colors indicate aligned local transitions; opposite signed gradients represent the same boundary axis. The magnitude map highlights dominant anisotropic transitions, rather than assigning semantic identities.

Beyond independent pixel classification. LS remains a dense loss represented by spatial maps, but every descriptor depends on neighboring probabilities and jointly aggregates class-wise gradient statistics. It augments CE with neighborhood geometry rather than abandoning pixel-indexed optimization. A localized contour can still have a weak probability transition or inconsistent neighboring transition directions. Magnitude and orientation consistency target these two failure modes. Explicit topology preservation is not guaranteed.

K Direct Sobel-Gradient Matching Control

A direct Sobel baseline uses the same target smoothing, Sobel filters, and boundary-buffer support as LS, but matches class-wise horizontal and vertical gradients directly. LS instead forms their joint tensor statistics and matches magnitude and orientation. All methods share data splits, pretrained initialization, batch sizes, epochs, optimizer, and learning-rate schedule under DINOv2-B + UPerNet.

Table 14: Direct Sobel matching versus LS under the same backbone and decoder.

Dataset	Objective	mIoU	B-mIoU
ADE20K	CE	54.50	43.19
	CE+Sobel	54.61	43.55
	CE+LS	55.44	45.06
ISPRS-Potsdam	CE	81.53	60.24
	CE+Sobel	81.83	60.83
	CE+LS	82.21	61.65

Direct gradient matching improves both baselines, but LS exceeds it by 0.83 mIoU/1.51 B-mIoU on ADE20K and 0.38/0.82 on ISPRS-Potsdam. These results support a benefit beyond adding Sobel supervision alone in the tested settings.

Comparison with other boundary objectives. The main comparison adds each objective to CE, tunes loss weights on validation data, and fixes the training protocol within each dataset/model. Different losses target contour distance, overlap, topology, or morphology and need not provide equally effective gradients for these probability-field configurations. Sparse or perturbation-sensitive boundary supervision and competing optimization signals are possible explanations for degradation, rather than directly measured causes. LS acts on soft probabilities through differentiable local operations. The comparison characterizes the effectiveness of these objectives under the same segmentation and validation-based tuning protocol.

L Probability-Field Evaluation Details

We conduct a component-wise probability-field analysis on ADE20K and ISPRS-Potsdam using DINOv2-Base + UPerNet. LSM Field and LSO Field are computed using the magnitude and orientation discrepancy measures defined in Eqs. 13 and 14, respectively. Lower values indicate better agreement between predicted and target structures. SDF Match evaluates boundary alignment using signed distance fields, with higher values indicating better agreement between predicted and ground-truth boundaries. For the two field metrics, the one-hot ground truth is smoothed using a 3×3 mean filter. All methods use the same ground-truth-defined boundary support to ensure a consistent comparison.

As shown in Table 15, LSM and LSO exhibit distinct and complementary effects. Adding LSM alone reduces magnitude discrepancy from 0.6849 to 0.6293 on ADE20K and from 0.7152 to 0.6689 on ISPRS-Potsdam. Adding LSO alone reduces orientation discrepancy from 0.4853 to 0.4148 and from

Table 15: Component-wise probability-field analysis using DINOv2-Base + UPerNet. Lower LSM/LSO Field discrepancies and higher SDF Match indicate better agreement with the target.

Dataset	Objective	LSM Field ↓	LSO Field ↓	SDF Match ↑
ADE20K	CE	0.6849	0.4853	0.6460
	CE+LSM	0.6293	0.4939	0.6668
	CE+LSO	0.6860	0.4148	0.6535
	CE+LSM+LSO	0.6363	0.4221	0.6684
ISPRS-Potsdam	CE	0.7152	0.3212	0.6990
	CE+LSM	0.6689	0.3361	0.7066
	CE+LSO	0.7128	0.2930	0.7039
	CE+LSM+LSO	0.6513	0.2937	0.7158

0.3212 to 0.2930, respectively. These component-wise changes are consistent with the quantities directly regularized by the two terms.

Combining LSM and LSO achieves the highest SDF Match on both datasets, increasing it from 0.6460 to 0.6684 on ADE20K and from 0.6990 to 0.7158 on ISPRS-Potsdam. Although the combined objective does not minimize every individual discrepancy, it provides the best overall boundary alignment among the evaluated variants. These results support the complementary roles of LSM in regularizing transition strength and LSO in regularizing transition direction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction clearly state that the paper proposes Local Structure (LS) regularization to enhance fine-grained detail delineation in semantic segmentation, which models semantic boundary neighborhoods using structure-tensor-based Local Structure Magnitude (LSM) and Local Structure Orientation (LSO). The claims are supported by experiments on four datasets, multiple VFM backbones, different decoders, and comparisons with representative boundary/topology-related losses.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The conclusion discusses a limitation of the current LS formulation, namely that it uses fixed structural scales and loss weights, while boundary patterns may vary across object sizes and scene layouts. The paper also points to adaptive LS estimation based on local boundary complexity or prediction uncertainty as future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper does not present theoretical theorems or formal proofs. Rather, it models the proposed LS regularization by adapting and extending the classical theory of structure tensor. The mathematical formulations are used to define the proposed LS regularization, including boundary-buffer construction, structure-tensor-based LSM/LSO representations, and the final loss.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper describes the datasets, VFM backbones, decoders, evaluation metrics, loss formulations, and hyperparameter selection strategy used in the experiments. The submitted code further provides an avenue to reproduce the proposed LS regularization.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The code has been uploaded as part of the submission material, and the experiments are conducted on publicly available datasets, including ADE20K, COCO, Cityscapes, and ISPRS-Potsdam. The paper cites the original dataset and model sources, and the submitted code provides the implementation of the proposed LS regularization.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The paper specifies the datasets, data splits, evaluated backbones and decoders, optimizer, evaluation metrics, and the search range used to select the LS loss weight. All methods are trained and evaluated under the same protocol with the same random seed (12345) to ensure fair comparison.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes].

Justification: Appendix E reports mean and standard deviation across three random seeds for DINOv2-Small/Base with UPerNet on ADE20K and ISPRS-Potsdam. These describe run-to-run variability, not confidence intervals or formal significance tests. Other main comparisons use the same single seed (12345).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: Training uses four A100 GPUs. Appendix C reports per-epoch time and GPU memory for CE and CE+LS on ADE20K and Cityscapes, and Appendix G reports trainable parameters. Resource measurements are not available for every experimental configuration or for the total project, so this disclosure remains partial.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research uses publicly available semantic segmentation datasets and existing pretrained visual foundation models, and does not involve private data, human subjects, or deceptive data collection. The study is conducted for improving general semantic segmentation methodology and conforms to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No].

    Justification: The paper focuses on a general semantic segmentation training regularizer and does not include a dedicated broader-impact discussion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk pretrained generative models, language models, or scraped datasets. It proposes a training regularization loss for semantic segmentation and evaluates it on standard public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The paper cites the original sources for all datasets, pretrained VFM backbones, decoders, and baseline losses used in the experiments. The datasets and pretrained models are used for academic research following their intended benchmark and research usage.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper does not introduce a new dataset, benchmark, pretrained model, or other standalone asset. The main contribution is a training regularization method for semantic segmentation, whose implementation is included in the submitted code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human subjects, user studies, crowdsourcing, or collection of personal data, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method does not involve LLMs as an important, original, or non-standard component. LLMs were used only when necessary for language polishing, editing, or formatting, and their use did not affect the methodology, experiments, results, or scientific claims of the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.