SPAM 2: EFFICIENT SUPERPIXEL ANYTHING USING JOINT OBJECT FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent superpixel methods based on foundation segmentation models have pushed accuracy further by constraining superpixels to lie within high-level objects. However, this gain comes with a high computational cost, which limits their use in practical or large-scale vision pipelines. In this work, we introduce SPAM 2 (SuperPixel Anything Model 2), an efficient object-aware superpixel framework that notably replaces the costly traditional iterative clustering by a fast trainable local pixel-to-seed assignment and leverages lightweight foundation segmentation models to constrain superpixels within objects. To further exploit high-level information, SPAM 2 fuses superpixel and object features through a joint representation before final assignments. We also introduce a new supervision strategy based directly on oracle superpixel maps to recover more regular regions from a denser training signal. Comprehensive experiments on standard benchmarks and a downstream superpixel-based task show that SPAM 2 achieves state-of-the-art superpixel accuracy, while its lightweight variant is an order of magnitude faster than other object-based methods, making it suitable for dedicated computer vision pipelines. Code and pretrained models will be made available.

1 INTRODUCTION

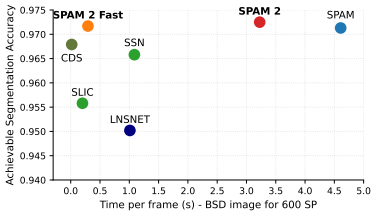
Superpixel segmentation is a standard tool in computer vision and image processing. It aims to group pixels into connected and locally coherent regions, providing a compact representation of the image content. By reducing the number of visual elements while preserving most image structures, superpixels are commonly used as a preprocessing step or as intermediate representations in many pipelines, including semantic segmentation (Mei et al., 2025; Zhang et al., 2026), optical flow (Liu et al., 2019), hierarchical image classification (Park et al., 2025), and transformer-based representation learning (Ke et al., 2024; Lew et al., 2024).

Contrary to standard object segmentation, superpixels seek to follow image boundaries, while satisfying an implicit regularity constraint: superpixels should remain compact, connected, and of comparable size (Stutz et al., 2018). This dual objective makes the problem intrinsically ambiguous across the literature (Giraud & Clément, 2026). Highly regular partitions are stable and easy to interpret, but may fail to capture thin structures or weak object contours, while more accurate segmentations are often produced by irregular, noisy regions or even fragmented regions.

Classical approaches mostly rely on low-level features such as color similarity and spatial proximity. Among them, SLIC (Achanta et al., 2012) is one of the most influential methods, formulating superpixel generation as a locally constrained clustering problem in a joint color-spatial feature space. While simple, training-free, and widely used, such methods rely on local appearance and can struggle with weak boundaries, illumination changes, or strong intra-object texture variations.

Deep learning-based methods have been introduced to overcome these limitations by replacing handcrafted features with learned representations (Jampani et al., 2018; Xu et al., 2024). By exploiting higher-level visual features, they generally improve boundary adherence, especially around complex object structures, but at the expense of lower shape regularity, leading to noisy superpixels. As image oversegmentations, superpixels are expected to partition all image regions at a given scale. This raises the question of how to effectively combine high-level semantic representations, which facilitate the capture of complex objects, with accurate, regular, and noise-robust delineation of homogeneous regions.

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



Property	SPAM	SPAM 2
Object-constrained	✓	✓
Superpixel algorithm	Iterative k-means clustering	Trainable local assignment
Joint object features	✗	✓
Fast version	✗	✓

Figure 1: **Comparison between object-constrained methods SPAM (Walther et al., 2025) and SPAM 2.** SPAM 2 improves SPAM accuracy using a trainable local assignment of superpixels on joint object features and also proposes a fast version (SPAM 2 Fast).

Recently, segmentation foundation models such as SAM (Kirillov et al., 2023) have opened a new direction by providing strong semantic-agnostic object priors. In particular, SPAM (Walther et al., 2025) showed that incorporating such priors into superpixel clustering can help capture complex object structures while reducing noisy region assignments that arise from relying solely on low- or high-level visual features. However, SPAM only partially leverages foundation model information, relying on a frozen object mask, disconnected from the superpixel clustering process. Moreover, its practical use remains limited by the inference cost of the external object segmentation pipeline (Kirillov et al., 2023), which restricts applicability in real-time or resource-constrained settings. At the same time, superpixels are regaining interest as content-adaptive representations in large-scale vision pipelines (Ke et al., 2024; Mei et al., 2025), highlighting the need for methods that combine high segmentation quality with strong computational efficiency.

Contributions. In this work, we introduce SPAM 2 (SuperPixel Anything Model 2), an efficient and accurate object-guided superpixel framework. SPAM 2 preserves the core advantage of object-constrained superpixels, i.e., preventing leakage across object regions, while significantly improving both accuracy and efficiency (see Fig. 1). To this end, we replace the costly iterative clustering stage with a direct, trainable local assignment formulation, enabling efficient pixel-to-seed grouping. In addition, we revisit the integration of low- and high-level cues through an early joint feature design that better aligns object-level and superpixel formation. This architecture also enables a lightweight configuration, up to 15x faster than SPAM (Walther et al., 2025) at comparable segmentation quality.

Our main contributions are summarized as follows:

- We introduce **SPAM 2** and its lightweight version **SPAM 2 Fast**, an efficient object-guided superpixel framework that replaces costly clustering with a lightweight, trainable local assignment mechanism driven by object priors.
- We propose a joint object features architecture that integrates local superpixel features with foundation model object cues, enabling more consistent and structure-aware assignments.
- We show that regularity can be efficiently enforced via supervision derived from oracle superpixel maps, introducing a direct form of knowledge distillation from superpixel-level annotations.
- We demonstrate through extensive experiments, including ablations, standard benchmarks, runtime analysis and downstream semantic segmentation, that SPAM 2 achieves superior accuracy while being much faster than prior object-guided approaches. In addition to multi-scale performance curves, we introduce, to our knowledge for the first time for superpixels, statistical significance tests on per-image scores, confirming that these gains are consistent across images.

2 Related Works

Traditional superpixel algorithms. Traditional superpixel methods mainly rely on low-level cues, such as color similarity, texture, and spatial proximity, to produce compact and connected over-segmentations. Among them, SLIC (Achanta et al., 2012) remains one of the most influential approaches, performing a local k-means clustering in a joint color/spatial feature space. SLIC has

served as a basis for many methods that further improve the speed–quality trade-off by introducing contour constraints (Zhang et al., 2016), more complex feature spaces (Liu et al., 2016; Chen et al., 2017), or by reducing computational cost (Achanta & Süsstrunk, 2017; Bobbia et al., 2021). Other traditional approaches use graph-based algorithms (Liu et al., 2011; Benesova & Kottman, 2014; Kang et al., 2020; Zhou et al., 2023), coarse-to-fine clustering (Van den Bergh et al., 2012) or watershed (Machairas et al., 2015). These approaches are lightweight, predictable, and naturally produce regular regions. However, their reliance on local appearance cues limits their ability to capture meaningful object boundaries, especially when contours are weak or illumination varies (Wu & Yan, 2026). As a result, classical superpixels often preserve geometric regularity but may poorly align with object structures.

Deep learning-based superpixels. Deep learning-based methods tend to address these limitations by replacing hand-crafted cues with learned visual representations. SSN (Jampani et al., 2018) introduced a differentiable SLIC-like formulation, where a convolutional network predicts dense features, later grouped through soft clustering. SFCN (Yang et al., 2020) proposed to directly predict soft assignments to nearby superpixel candidates. Following methods such as AINet (Wang et al., 2021) introduced a boundary-aware loss, BINet (Zhao et al., 2024) a bio-inspired network architecture, CDS (Xu et al., 2024) inter-pixel correlations within a contrastive learning to improve assignment robustness, and (Juybari et al., 2026) a graph-Laplacian loss that encourages spatial regularity and connectivity during training. Recent reviews such as (Barcelos et al., 2024; Wu & Yan, 2026) provide a detailed analysis of the superpixel methods literature. These methods generally improve object adherence and superpixel quality metrics at the expense of partition regularity. To address this, SPAM (Walther et al., 2025) has proposed to leverage foundation segmentation model such as SAM (Kirillov et al., 2023) in a complementary manner. High-level objects constrain the clustering, improving boundary adherence while allowing superpixels to operate at a finer scale, to maintain a regular structure within objects. However, SPAM relies on a heavy object segmentation model, limiting its applicability in resource-constrained, real-time or large-scale settings.

Segment Anything task. The Segment Anything Model (SAM) (Kirillov et al., 2023) provides powerful semantic-agnostic object priors through large-scale foundation model training. However, SAM produces independent object masks rather than a dense image partition, making it complementary to rather than a replacement for superpixel segmentation. In fact, SAM-based decompositions remain substantially below dedicated superpixel methods on standard superpixel benchmarks (Giraud & Clément, 2026). Nevertheless, its object-level cues are particularly valuable for guiding superpixel formation, motivating object-aware approaches such as SPAM. Several SAM variants have been proposed to reduce inference cost. They either accelerate the image encoder through lightweight architectures or distillation (Zhang et al., 2023a; Xiong et al., 2024; Zhang et al., 2024), reduce decoder evaluations via more selective prompting (Zhang et al., 2023b; Fu et al., 2024), or replace the prompt-based pipeline with a single-shot segmentation model (Zhao et al., 2023).

3 SUPERPIXEL ANYTHING MODEL 2

We present SPAM 2, a fast object-aware superpixel segmentation framework (Fig. 2). Inspired by (Yang et al., 2020), we formulate superpixel segmentation as a local pixel-to-seed classification, where each pixel is assigned in a single feed-forward pass to one of its neighboring seeds (Sec. 3.1). The superpixel model is combined with a pretrained foundation object model providing high-level features and a prior object segmentation map. These features are fused with the low-level superpixel features to guide the assignment (Sec. 3.3), while the prior map prevents superpixels from spanning multiple objects through a test-time mask correction step (Sec. 3.2). We further introduce an oracle-superpixel supervision strategy to recover more regular and interpretable regions (Sec. 3.4).

The previous SPAM model (Walther et al., 2025) also uses a prior object map, but relies on costly iterative clustering and exploits object information only as a final constraint. In contrast, SPAM 2 predicts assignments in a single feed-forward pass and additionally uses high-level object features to guide them. Finally, as SPAM 2 is agnostic to the choice of superpixel and object encoders, we study different pretrained backbones and derive two main variants targeting accuracy and efficiency.

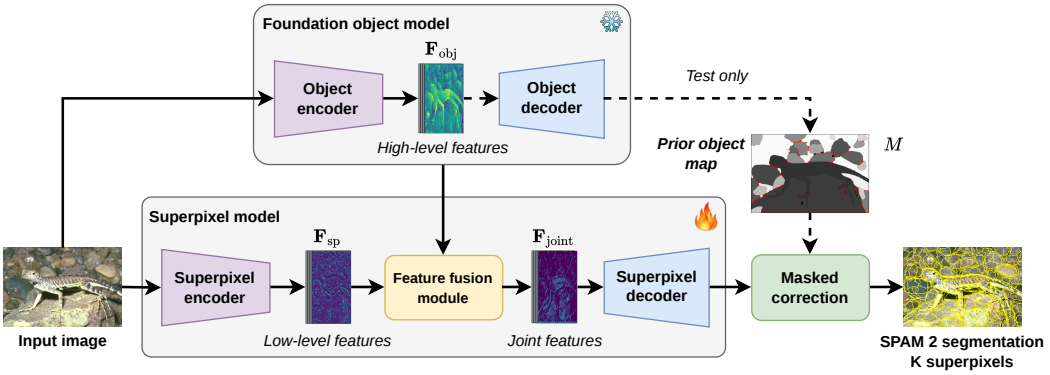


Figure 2: **Overview of Superpixel Anything Model 2 (SPAM 2)**. A superpixel encoder extracts low-level local features, while a foundation object model produces high-level object features and a prior object map. These features are fused into a joint representation, from which a decoder predicts a local 3×3 superpixel assignment distribution for each pixel. At test time, the prior object map corrects this prediction to ensure that superpixels remain within the same object, preventing cross-object leakage. Superpixels are obtained in a single feed-forward pass, without iterative clustering.

3.1 Superpixel model as local assignment

Superpixel model. The superpixel model of SPAM 2 is a lightweight fully-convolutional encoder–decoder network that predicts a superpixel decomposition in a single forward pass, in contrast to the iterative clustering used in SPAM (Walther et al., 2025). We build on the architecture of CDS (Xu et al., 2024), itself derived from the SFCN (Yang et al., 2020) design. The encoder extracts a dense low-level feature map $\mathbf{F}_{\text{sp}}$ from the input image, encoding color, texture and spatial cues, while the decoder maps these features to a local pixel-to-seed assignment (detailed below).

Superpixel decoder. We formulate superpixel segmentation as a local pixel-to-seed classification problem. Given a target number of superpixels K, the image is first partitioned into a regular grid of initial seeds $\mathcal{S} = \{s_1, \ldots, s_K\}$, to which pixels are then assigned. For efficiency, we restrict the prediction to a small spatial neighborhood: each pixel p can only be assigned to its closest seed on the grid or to its 8 neighboring seeds, forming the standard 3×3 candidate set $\mathcal{N}(p)$ with $|\mathcal{N}(p)| \leq 9$. The superpixel decoder predicts a dense assignment tensor $\mathbf{A} \in \mathbb{R}^{9 \times HW}$, where $\mathbf{A}_j(p)$ is the assignment weight of pixel p to the j-th seed in $\mathcal{N}(p)$, normalized by a softmax over the nine candidates (see Fig. 3). The unconstrained assignment is then obtained by selecting the most likely local seed.

Training and loss function. The superpixel model of SPAM 2 is trained with the CDS objective (Xu et al., 2024). The superpixel loss combines a semantic reconstruction term with a position regularization term:

$$\mathcal{L}_{\text{sp}} = \mathcal{L}_{\text{sem}} + w_p \, \mathcal{L}_{\text{pos}}, \tag{1}$$

where $\mathcal{L}_{\text{sem}}$ enforces consistency with the ground-truth segmentation, $\mathcal{L}_{\text{pos}}$ favors spatially compact assignments, and w_p is a balancing parameter set to 0.003.

The objective also includes content-disentanglement terms. An auxiliary modality, which is in practice the Sobel-filtered image, is processed in parallel to predict an assignment $\mathbf{A}_{\text{aux}}$, and the two corresponding local content features are aligned through a Kullback-Leibler divergence $\mathcal{L}_{\text{align}}$, while $\mathcal{L}_{\text{MI}}$ reduces the mutual information between the two modalities. The full objective is:

$$\mathcal{L} = \lambda(\mathcal{L}_{\text{sp}}(\mathbf{A}) + \mathcal{L}_{\text{sp}}(\mathbf{A}_{\text{aux}})) + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{MI}}, \tag{2}$$

where λ is empirically set to 0.005.

3.2 Object model and masked correction

Foundation object model. The local assignment of Sec. 3.1 relies only on local appearance and may therefore let a superpixel leak across high-level objects. Following SPAM (Walther et al.,

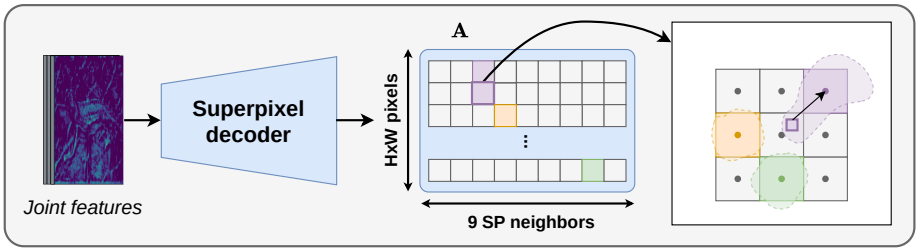


Figure 3: **Superpixel decoder with local assignment.** From the joint features of the superpixel and object encoders, a single forward pass predicts, for every pixel, a soft assignment over its 9 neighboring superpixel seeds (a $HW \times 9$ map). Each pixel is thus classified to one of the 9 candidate seeds arranged on a regular grid, yielding the dashed superpixel boundaries shown on the right.

2025), we prevent this by exploiting a prior object map M obtained from a foundation object model such as SAM (Kirillov et al., 2023). This prior is applied at inference time to correct the predicted assignment, without any additional forward pass (details below). As SPAM 2 is agnostic to the choice of this object model, it can be swapped to trade segmentation quality for speed, which we exploit to derive two main configurations. The default **SPAM 2** uses a SAM ViT-L encoder with grid-based automatic mask generation, favoring object quality, while **SPAM 2 Fast** instead combines a lightweight EfficientViT (Zhang et al., 2024) encoder with a MobileSAMv2 (Zhang et al., 2023b) decoder, drastically reducing inference time while preserving accurate priors. We analyze this accuracy–efficiency trade-off across different object models in Sec. 4.2.

Mask-constrained reassignment. At inference time, the prior object map M is used to correct the initial local assignment map S produced by the superpixel decoder. For each superpixel, we first identify the object label that covers most of its pixels. Pixels assigned to a superpixel dominated by a different object are then considered inconsistent. These pixels are temporarily unassigned and relabeled by propagating valid neighboring superpixel labels, while restricting propagation to pixels belonging to the same object region. The process is repeated until no further pixel can be reassigned. If an object has no seed, a new superpixel label is created. This correction requires no additional forward pass and should be interpreted as a leakage-reduction step rather than a hard constraint over the whole image. Ambiguous pixels that would result from overlapping object proposals in the foundation segmentation model (red pixels in the prior object map of Fig. 2) keep their initial assignment from the superpixel decoder. Finally, a standard spatial connectivity enforcement is performed by merging small disconnected components only into valid neighboring superpixels, as in (Walther et al., 2025). This process is illustrated in Appendix B.3.

3.3 Joint object-superpixel features

While the masked correction is applied only at inference, after the assignment is predicted, SPAM 2 also learns to inject object-level information into the superpixel model during training, fusing local superpixel features and object encoder features before decoding.

Fusion module. Let $\mathbf{F}_{\text{sp}}$ be the dense feature map produced by the superpixel encoder and $\mathbf{F}_{\text{obj}}$ the intermediate features extracted by the object encoder. As these may differ in spatial resolutions and channel dimensions, we adapt them with two projection modules ϕ_{sp} and ϕ_{obj}, then fuse them into a joint representation using a feature fusion module g:

$$\mathbf{F}_{\text{joint}} = g\big(\phi_{\text{sp}}(\mathbf{F}_{\text{sp}}),\, \phi_{\text{obj}}(\mathbf{F}_{\text{obj}})\big). \tag{3}$$

The superpixel decoder then predicts the local assignment $\mathbf{A}$ from the joint features $\mathbf{F}_{\text{joint}}$. In practice, we implement g as a FiLM (Perez et al., 2018) module: the projected object features predict per-channel scale and shift parameters that modulate the superpixel features. Alternative fusion strategies for g are compared in Appendix B.1.

Training strategy. The superpixel model is first trained alone, without the feature fusion module: the decoder predicts directly from $\mathbf{F}_{\text{sp}}$. In a second stage, we plug in the feature fusion module and finetune only its parameters, while freezing the superpixel encoder and decoder and object encoder.

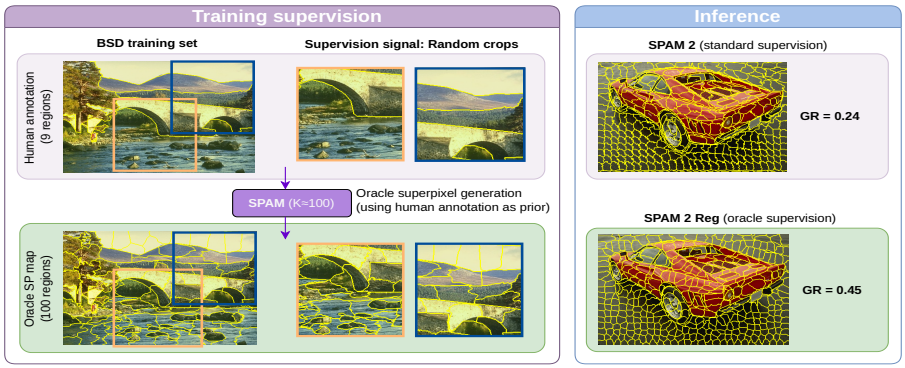


Figure 4: **Oracle superpixel supervision.** Standard deep superpixel methods are supervised only with object-level BSD annotations (top row), whereas we also derive oracle superpixel maps using the original SPAM (Walther et al., 2025) (bottom row) using human annotation as object prior to get denser training signals. At inference, the model finetuned with oracle maps yields more regular decompositions (higher Global Regularity (GR) scores), while preserving object consistency.

As the decoder remains fixed, the fusion module must produce joint features compatible with the already learned assignment, thereby preserving the local superpixel structure while enriching it with object information. We found this two-stage scheme significantly more effective than training the whole architecture jointly, which over-relies on the object signal and degrades accuracy at high superpixel counts. Training settings are detailed in Appendix A.3.

3.4 Oracle superpixel supervision for regularity

Local assignment can reduce superpixel regularity. We therefore introduce a regularity-oriented training strategy. Instead of only using object-level human annotations as supervision, we propose to finetune the model directly with oracle coarse-scale superpixel maps, i.e. denser decompositions that already respect the precisely annotated objects. This provides a denser training signal, mainly composed of regular regions. In practice, we generate these oracle maps with the original object-based method SPAM (Walther et al., 2025), using the BSD train ground-truth segmentation as its prior object map and around 100 superpixels to balance object awareness and regularity. These oracle maps remain constrained by the annotated objects while inheriting the regular and interpretable structure of SPAM. We then finetune SPAM 2 on these label maps, which encourages more regular decompositions while preserving object consistency. This acts as a superpixel-level knowledge distillation, transferring SPAM's regular structure to SPAM 2 without introducing any additional computation at inference time. We denote the resulting models **SPAM 2 Reg** and **SPAM 2 Fast Reg**. Their accuracy–regularity trade-off is quantified in Appendix B.2.

4 Experimental Validations

We first present the experimental protocol (Sec. 4.1). We then analyze the contribution of each SPAM 2 component through an extensive ablation study (Sec. 4.2), before comparing its performance to state-of-the-art superpixel methods on standard datasets and metrics (Sec. 4.3), as well as on a downstream semantic segmentation task (Sec. 4.4).

4.1 Experimental protocol

Datasets. We evaluate SPAM 2 on five standard datasets covering natural, indoor, urban, and biomedical domains. BSD (Martin et al., 2001) is the standard superpixel benchmark, with natural images and fine human annotations; NYUv2 (Silberman et al., 2012) targets indoor scenes; and SBD (Gould et al., 2009) provides semantic object annotations at diverse object scales. KITTI (Alhaija et al., 2018) extends the evaluation to driving scenes, while DRIVE (Staal et al., 2004) contains biomedical images with thin, irregular retinal vessels (detailed descriptions in Appendix A.1). Together, these datasets assess the generalization of SPAM 2 across image domains.

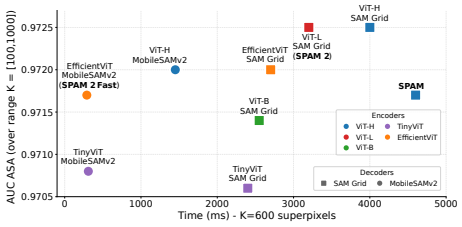


Figure 5: **Object-model accuracy–runtime trade-off**. AUC ASA on BSD versus runtime across object encoder–decoder configurations.

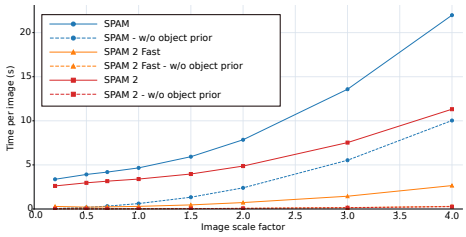


Figure 6: **Runtime scalability over image size.** Scale 1 corresponds to the original BSD image resolution (321×481).

Evaluation metrics. Evaluating superpixel methods is known to be challenging, as different metrics capture complementary, and sometimes conflicting, properties (Stutz et al., 2018; Giraud & Clément, 2026). Following these guidelines, we focus on robust measures of segmentation quality: Achievable Segmentation Accuracy (ASA) for object adherence, Boundary Recall versus Boundary Precision (BR/BP) for contour quality, and Global Regularity (GR) for shape consistency. Each metric is reported as a function of the number of generated superpixels for fair comparison, and may be summarized by its area under the curve (AUC) over a common superpixel range $[100, 1000]$, as in (Juybari et al., 2026; Walther et al., 2026). Metric definitions are detailed in Appendix A.2.

4.2 Ablation Study

Choice of object model. Fig. 5 evaluates the accuracy–runtime trade-off on the BSD of several object models providing both priors and features to SPAM 2. AUC are computed on a superpixel range of K=[100,1000] and time for K=600. Using the same ViT-H + SAM Grid pipeline as SPAM (Walther et al., 2025), our framework is already $1.5\times$ faster with higher AUC ASA. ViT-L provides the best grid-based accuracy (0.9725) at lower runtime, and is therefore used in SPAM 2. As SAM Grid remains the main bottleneck, we combine MobileSAMv2 (Zhang et al., 2023b) with EfficientViT for SPAM 2 Fast, reaching 0.9717 AUC ASA in 290 ms, i.e., up to $15\times$ faster than SPAM and $10\times$ faster than SPAM 2. We thus retain SPAM 2 for accuracy and SPAM 2 Fast for efficiency. Note that we mainly use SAM 1 as it is well suited to our class-agnostic full-image partitioning context, while SAM 2 (Ravi et al., 2025) is video oriented and yields weaker priors or SAM 3 (Carion et al., 2026) targets concept-level prompting. The lightweight variants of SAM 1 also enable the design SPAM 2 Fast. The computational time gaps also increase with both image size and superpixel scale. Fig. 6 further evaluates scalability with image size. While SPAM runtime increases rapidly, SPAM 2 scales substantially better. The curves without object prior isolate the superpixel frameworks and highlight the efficiency gain of the proposed local assignment over iterative clustering. Appendix B.2 compares runtime across the number of superpixels, while Appendix B.4 isolates the superpixel-stage cost.

Superpixel model components. We evaluate the contribution of each component of SPAM 2 in Fig. 7. We compare the full model against three ablated configurations: without the masked correction, without the joint module, and without both. Note that without joint and mask inference takes about 20ms per image on the BSD. We also report performance for the lightweight SPAM 2 Fast version. The masked correction consistently improves ASA over the baseline without object guidance. This shows that the object constraint introduced in SPAM integrates effectively into the local assignment framework of SPAM 2, preventing assignments that would cross object regions. Adding the joint features module further improves accuracy, particularly in the low-superpixel regime where regions are more prone to crossing object boundaries. This confirms the benefit of combining local superpixel cues with higher-level object features. Using both masked correction and joint features yields the best performance across all scales, while SPAM 2 Fast preserves most of the gains at a much lower computational cost.

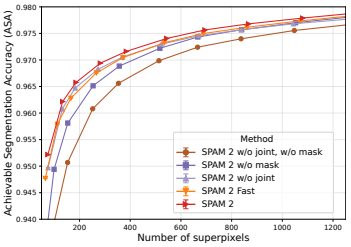


Figure 7: **Superpixel model components ablations.**

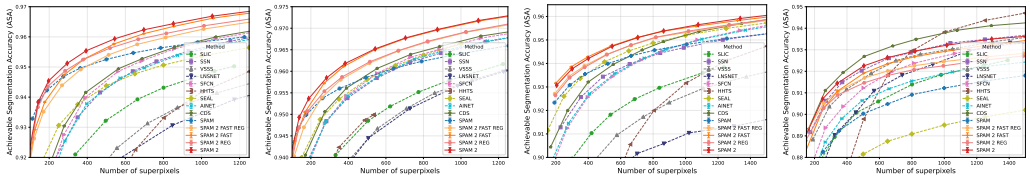


Figure 8: **Quantitative results on various datasets.** The segmentation accuracy of SPAM 2 and its variants are consistently the best or among the best methods of the state-of-the-art.

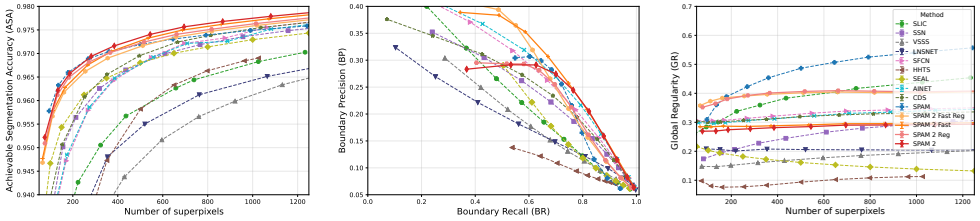


Figure 9: **Quantitative results on BSD.** Accuracy (left), contour quality (center) and regularity (right) results for SPAM 2 models compared to state-of-the-art methods.

4.3 Comparison to state-of-the-art methods

Quantitative results. Fig. 8 compares the ASA of SPAM 2 with traditional and deep-learning-based superpixel methods as the number of superpixels increases on NYUv2, SBD, KITTI, and DRIVE. On NYUv2 and SBD, SPAM 2 achieves the highest ASA across the plotted range, while SPAM 2 Fast remains close, particularly on SBD where their curves nearly overlap and is even 1st on KITTI. The slightly lower results on DRIVE can be explained by the reliability of the object priors on this dataset: the foundation model is likely less suited to thin retinal vessels, which reduces the benefit of object guidance. Regularized variants generally attain lower ASA than their non-regularized counterparts, illustrating the accuracy-regularity trade-off. The complete AUC comparison is reported with full ASA, contour, and GR curves in Appendix C.1. Fig. 9 provides a more detailed analysis on BSD. See also Fig. 1 (left) for a summary over accuracy and runtime. SPAM 2 is consistently among the most accurate methods over the full range of superpixel numbers, with a clear advantage at higher resolutions. SPAM 2 Fast remains very close to the full model, showing that most of the accuracy gain is preserved with a lightweight pipeline. The regularized variants slightly reduce ASA but clearly improve GR, leading to a more balanced accuracy-regularity compromise.

Significance analysis. Superpixel methods are generally evaluated with standard average curves over multiple scales. Recent works also report AUC performance (Juybari et al., 2026; Walther et al., 2026) over these curves to strengthen the evaluation. Yet, since superpixel segmentation accuracy lies in a high-accuracy regime, improvements in average metrics can appear numerically small, making it important to assess whether they reflect consistent gains across images rather than a few isolated cases (Walther et al., 2026). We propose to introduce significance tests to assess the superiority of superpixel methods. To do so, we use pairwise Wilcoxon signed-rank tests on the per-image ASA scores interpolated at a given scale. Tab. 1 reports the results on BSD at an intermediate scale $K = 500$. Both SPAM 2 and SPAM 2 Fast significantly outperform all compared methods, with p-values $< 1e^{-3}$. Although the absolute differences in ASA may be small, the improvements appear to be highly consistent across the dataset and cannot be explained by variations on a small image subset. Hence, such statistical testing is particularly relevant in this high-accuracy regime, and to the best of our knowledge is introduced for the first time for superpixel segmentation. Corresponding tests for NYUv2, SBD, KITTI, and DRIVE are reported in Appendix C.2.

Qualitative examples. Qualitative results are shown in Fig. 10. SPAM 2 produces segmentations comparable to other deep learning-based methods in terms of object adherence, while generating more regular superpixels than approaches such as (Tu et al., 2018). The regularized variant (SPAM 2 Reg) further improves shape consistency and homogeneity, yielding more interpretable partitions. Further examples on the other datasets are in Appendix C.3.

Table 1: **Statistical significance of ASA gains on BSD.** Two-sided Wilcoxon signed-rank tests on per-image ASA interpolated at $K = 500$. Cells report p-values; green/red indicate significant gains/losses for the row method, and $\sim$ a non-significant difference.

	SLIC	SEAL	SSN	SFCN	LNSNET	AINET	VSSS	CDS	HHTS	SPAM	SPAM 2 Fast Reg	SPAM 2 Fast	SPAM 2 Reg	SPAM 2
SLIC(Achanta et al., 2012)	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\sim .374$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SEAL(Tu et al., 2018)	$\uparrow < .001^*$	–	$\sim .273$	$\sim .186$	$\uparrow < .001^*$	$\sim .122$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SSN(Jampani et al., 2018)	$\uparrow < .001^*$	$\sim .273$	–	$\sim .915$	$\uparrow < .001^*$	$\sim .526$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SFCN(Yang et al., 2020)	$\uparrow < .001^*$	$\sim .186$	$\sim .915$	–	$\uparrow < .001^*$	$\sim .960$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
LNSNET(Zhu et al., 2021)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
AINET(Wang et al., 2021)	$\uparrow < .001^*$	$\sim .122$	$\sim .526$	$\sim .960$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
VSSS(Zhou et al., 2023)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
CDS(Xu et al., 2024)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
HHTS(Chang et al., 2024)	$\sim .374$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM(Walther et al., 2025)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow .009^*$	$\downarrow < .001^*$	$\downarrow .049^*$	$\downarrow < .001^*$
SPAM 2 Fast Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow .009^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Fast	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .049^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$
SPAM 2	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–
Method	SLIC	SEAL	SSN	SFCN	LNSNET	AINET	VSSS	CDS	HHTS	SPAM	SPAM 2 Fast Reg	SPAM 2 Fast	SPAM 2 Reg	SPAM 2

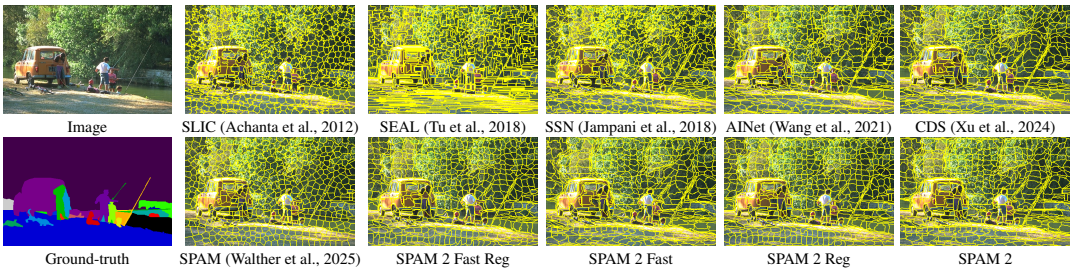


Figure 10: **Qualitative evaluation of superpixel methods on BSD and NYUv2 examples.** The number of requested superpixels is set to $K = 650$ for BSD. SPAM 2 generates accurate superpixels in all variants compared to state-of-the-art methods.

4.4 Downstream Semantic Segmentation

We evaluate the downstream impact of our superpixels using semantic segmentation with CAST (Ke et al., 2024), which replaces fixed ViT patches with superpixel-based segment tokens. Following the standard VOC transfer setting, CAST-S is pretrained on unlabeled COCO and finetuned on PASCAL VOC. We keep the model and training protocol fixed and vary only the superpixel decomposition, using the same requested number K for all methods. Tab. 2 shows that SPAM 2 reaches 68.94 mIoU, improving over SEEDS by +1.85 points and over CDS and SPAM by about +1.0 point. The fast variant also reaches 68.53 mIoU (+1.43 over SEEDS). SPAM 2 provides more effective segment tokens by reducing cross-object mixing (qualitative example in Appendix C.3).

Table 2: **Downstream semantic segmentation results** on PASCAL VOC using CAST (Ke et al., 2024). mIoU results are reported with gain over the default method (SEEDS).

Method	**mIoU (gain)**
SEEDS (Van den Bergh et al., 2012)	67.09
SLIC (Achanta et al., 2012)	66.50 (-0.59)
AINet (Wang et al., 2021)	67.81 ($+0.72$)
CDS (Xu et al., 2024)	67.92 ($+0.83$)
SPAM (Walther et al., 2025)	67.91 ($+0.82$)
SPAM 2 Fast	68.53 ($+1.43$)
SPAM 2	68.94 ($+\mathbf{1.85}$)

5 Conclusion

We introduced SPAM 2, an efficient object-guided superpixel framework that replaces iterative clustering with direct local pixel-to-seed assignment. Fused object features and masked correction preserve accuracy while substantially reducing inference cost. Object information is leveraged both as a hard assignment constraint and as a feature-level prior through a joint object-superpixel representation. Overall, SPAM 2 improves the practical use of object-guided superpixels by removing the computational bottleneck of clustering-based approaches, while maintaining state-of-the-art segmentation performance and improving downstream tokenization quality. Beyond efficiency, SPAM 2 provides new insights into how object-level knowledge can be distilled into superpixel formation via oracle pseudo-supervision, a step toward efficiently integrating foundation model priors into fine-grained superpixels for large-scale vision pipelines.

AI USE STATEMENT

Generative AI tools were used for translation and to assist with drafting and editing parts of the manuscript for clarity and readability and to support code prototyping during development. The tools were not used to generate experimental data, design the methodology, formulate mathematical claims, or interpret the experimental results. All AI-assisted content was reviewed by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Upon acceptance, we will publicly release the full source code for training, inference and evaluation, together with the pretrained weights of all SPAM 2 variants (SPAM 2, SPAM 2 Fast, SPAM 2 Reg and SPAM 2 Fast Reg), enabling their direct use as off-the-shelf superpixel tokenizers. The foundation object models rely on publicly available checkpoints, and all training and evaluation settings are detailed in Appendix A.

REFERENCES

Radhakrishna Achanta and Sabine Süsstrunk. Superpixels and polygons using simple non-iterative clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4895–4904, 2017.

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars M. Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 2018.

Isabela Borlido Barcelos, Felipe de Castro Belém, Leonardo de Melo João, Zenilton K. G. do Patrocínio Jr., Alexandre Xavier Falcão, and Silvio Jamil Ferzoli Guimarães. A comprehensive review and new taxonomy on superpixel segmentation. *ACM Computing Surveys*, 2024.

Wanda Benesova and Michal Kottman. Fast superpixel segmentation using morphological processing. In *Conference on Machine Vision and Machine Learning*, 2014.

Serge Bobbia, Richard Macwan, Yannick Benezeth, Keisuke Nakamura, Randy Gomez, and Julien Dubois. Iterative boundaries implicit identification for superpixels segmentation: a real-time approach. *IEEE Access*, 9:77250–77263, 2021.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. 2026.

Thomas V Chang, Simon Seibt, and Bartosz von Rymon Lipinski. Hierarchical histogram threshold segmentation-auto-terminating high-detail oversegmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3195–3204, 2024.

Jiansheng Chen, Zhengqin Li, and Bo Huang. Linear spectral clustering superpixel. *IEEE Transactions on Image Processing*, 26:3317–3330, 2017.

Jianhai Fu, Yuanjie Yu, Ningchuan Li, Yi Zhang, Qichao Chen, Jianping Xiong, Jun Yin, and Zhiyu Xiang. Lite-SAM is actually what you need for segment everything. In *European Conference on Computer Vision (ECCV)*, 2024.

Rémi Giraud and Michaël Clément. Superpixel segmentation: A long-lasting ill-posed problem. *Computer Vision and Image Understanding*, 2026.

Rémi Giraud, Vinh-Thong Ta, and Nicolas Papadakis. Evaluation framework of superpixel methods with a global regularity measure. *Journal of Electronic Imaging*, 26(6):061603, 2017.

Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *International Conference on Computer Vision (ICCV)*, 2009.

Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *European Conference on Computer Vision (ECCV)*, pp. 363–380, 2018.

Jeremy Juybari, Josh Hamilton, Shuvra Das, Chaofan Chen, Andre Khalil, and Yifeng Zhu. Differentiable laplacian matrix guided superpixel segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Xuejing Kang, Lei Zhu, and Anlong Ming. Dynamic random walk for superpixel segmentation. *IEEE Transactions on Image Processing*, 29:3871–3884, 2020.

Tsung-Wei Ke, Sangwoo Mo, and X Yu Stella. Learning hierarchical image segmentation for recognition and by recognition. In *International Conference on Learning Representations (ICLR)*, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023.

Jaihyun Lew, Soohyuk Jang, Jaehoon Lee, Seungryong Yoo, Eunji Kim, Saehyung Lee, Jisoo Mok, Siwon Kim, and Sungroh Yoon. Superpixel tokenization for vision transformers: Preserving semantic integrity in visual tokens. *arXiv:2412.04680*, 2024.

Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2097–2104, 2011.

Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4571–4580, 2019.

Y.-J. Liu, C.-C. Yu, M.-J. Yu, and Y. He. Manifold SLIC: A fast method to compute content-sensitive superpixels. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 651–659, 2016.

Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Decenciere. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, 2015.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision (ICCV)*, 2001.

Jieru Mei, Liang-Chieh Chen, Alan Yuille, and Cihang Xie. SPFormer: Enhancing vision transformer with superpixel representation. *Transactions on Machine Learning Research*, 2025.

Seulki Park, Youren Zhang, Stella X. Yu, Sara Beery, and Jonathan Huang. Visually consistent hierarchical image classification. In *International Conference on Learning Representations (ICLR)*, 2025.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

Alexander Schick, Mika Fischer, and Rainer Stiefelhagen. An evaluation of the compactness of superpixels. *Pattern Recognition Letters*, 43:71–80, 2014.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, 2012.

J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, and B. van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509, 2004.

David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018.

Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, and Jan Kautz. Learning superpixels with segmentation-aware affinity loss. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 568–576, 2018.

Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. SEEDS: Superpixels extracted via energy-driven sampling. In *European Conference on Computer Vision (ECCV)*, pp. 13–26, 2012.

Julien Walther, Rémi Giraud, and Michaël Clément. Superpixel anything: A general object-based framework for accurate yet regular superpixel segmentation. In *British Machine Vision Conference (BMVC)*, 2025.

Julien Walther, Rémi Giraud, and Michaël Clément. Supervise: A unified framework for standardized and reproducible superpixel evaluation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2026.

Yaxiong Wang, Yunchao Wei, Xueming Qian, Li Zhu, and Yi Yang. AINet: Association implantation for superpixel segmentation. In *International Conference on Computer Vision (ICCV)*, 2021.

Chong Wu and Hong Yan. A survey of superpixel methods and their applications. *Signal Processing: Image Communication*, 2026.

Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, Raghuraman Krishnamoorthi, and Vikas Chandra. EfficientSAM: Leveraged masked image pretraining for efficient segment anything. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Sen Xu, Shikui Wei, Tao Ruan, and Lixin Liao. Learning invariant inter-pixel correlations for superpixel generation. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 6351–6359, 2024.

Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13964–13973, 2020.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight SAM for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023a.

Chaoning Zhang, Dongshen Han, Sheng Zheng, Jinwoo Choi, Tae-Ho Kim, and Choong Seon Hong. MobileSAMv2: Faster segment anything to everything. *arXiv:2312.09579*, 2023b.

Yongxia Zhang, Xuemei Li, Xifeng Gao, and Caiming Zhang. A simple algorithm of superpixel segmentation with boundary constraint. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

Yuxuan Zhang, Shuchang Wang, Xiaoman Liu, Zhenbo Shi, Zhidong Yu, Wei Song, and Wei Yang. Rethinking adversary in semantic segmentation: An out-of-distribution perspective. In *European Conference on Computer Vision (ECCV)*. Springer, 2026.

Zhuoyang Zhang, Han Cai, and Song Han. EfficientViT-SAM: Accelerated segment anything model without accuracy loss. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024.

Tingyu Zhao, Bo Peng, Yuan Sun, Daipeng Yang, Zhenguang Zhang, and Xi Wu. Rethinking superpixel segmentation from biologically inspired mechanisms. *Applied Soft Computing*, 156: 111467, 2024.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv:2306.12156*, 2023.

Pei Zhou, Xuejing Kang, and Anlong Ming. Vine spread for superpixel segmentation. *IEEE Transactions on Image Processing*, 32:878–891, 2023.

Lei Zhu, Qi She, Bin Zhang, Yanye Lu, Zhilin Lu, Duo Li, and Jie Hu. Learning the superpixel in a non-iterative and lifelong manner. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1225–1234, 2021.

A EXPERIMENTAL SETUP

This section details the datasets, evaluation metrics, and training procedure. Section B examines model design and efficiency, while Sec. C provides the full benchmark comparison and additional quantitative and qualitative results.

A.1 DATASET DETAILS

We evaluate SPAM 2 on five datasets spanning natural images, indoor scenes, driving scenes, and biomedical imagery. This diversity is useful for a superpixel benchmark because the difficulty is not limited to object-boundary adherence: a method must also preserve small structures, remain stable under strong perspective changes, and produce meaningful regions when the target objects are extremely thin. Fig. 11 shows representative images and annotations, while Tab. 3 summarizes the evaluation splits.

BSD, SBD, and NYUv2 cover generic object and scene understanding settings. KITTI introduces strong perspective effects and large variations in apparent object scale. DRIVE is qualitatively different from the other benchmarks: its binary annotations describe thin retinal vessels, for which a small boundary displacement can affect contour metrics substantially. Results on DRIVE should therefore be read as a test for generalization to thin structures.

Table 3: **Summary of the datasets used in our evaluation.**

				Number of objects				
Dataset	**Domain**	**Test images**	**Average size**	**Mean**	**Min**	**Max**	**GT/image**	**Notes**
BSD (Martin et al., 2001)	Outdoor scenes	200	321x481	22	3	69	4-8	Fine multi-user annotations
SBD (Gould et al., 2009)	Outdoor scenes	477	320x240	20	2	135	1	PASCAL-style segmentation masks
NYUV2 (Silberman et al., 2012)	Indoor scenes	399	608x448	25	5	79	1	Depth files available
KITTI (Alhaija et al., 2018)	Driving	200	1242x375	21	6	40	1	Driving scenes
DRIVE (Staal et al., 2004)	Medical images of retina	20	565x584	2	2	2	1	Binary medical annotations of thin structures

A.2 EVALUATION METRICS DETAILS

Object segmentation (ASA). A central criterion when evaluating a superpixel segmentation is how well superpixels align with the objects in the image. Object segmentation metrics therefore quantify the average overlap between the generated superpixels and the objects in a ground-truth segmentation. Among the existing measures, the most commonly used one is Achievable Segmentation Accuracy (ASA) (Liu et al., 2011). ASA assigns each superpixel to the ground-truth object with which it has the largest intersection, sums these best overlaps over all superpixels, and normalizes the result by the image size:

$$\mathrm{ASA}(\mathcal{S}, \mathcal{G}) = \frac{1}{|I|} \sum_{S_k} \max_{G_j} |S_k \cap G_j|. \tag{4}$$

Regularity (GR). Regularity is another important aspect of superpixel segmentation. As shown in (Giraud et al., 2017), the level of regularity can significantly affect segmentation performance. Consequently, methods that explicitly enforce regular superpixels can obtain lower segmentation scores than approaches that allow stretched, thin, or visually ambiguous regions.

To evaluate regularity, the GR metric (Giraud et al., 2017) was introduced as a more robust alternative to circularity or compactness (CO) (Schick et al., 2014). GR combines two terms: a Shape Regularity Criteria (SRC), which evaluates each superpixel independently based on convexity, balance, and contour smoothness, and a Smooth Matching Factor (SMF), which measures the consistency of shapes across the segmentation. The resulting metric is defined as:

$$\mathrm{GR}(\mathcal{S}) = \frac{1}{\sum_{S_k \in \mathcal{S}} |S_k|} \sum_{S_k \in \mathcal{S}} |S_k| \, \mathrm{SRC}(S_k) \, \mathrm{SMF}(S_k). \tag{5}$$

Contour detection (BR vs BP). Contour detection results can differ from object segmentation results. For example, thin objects may contribute many contour pixels despite covering a small area,

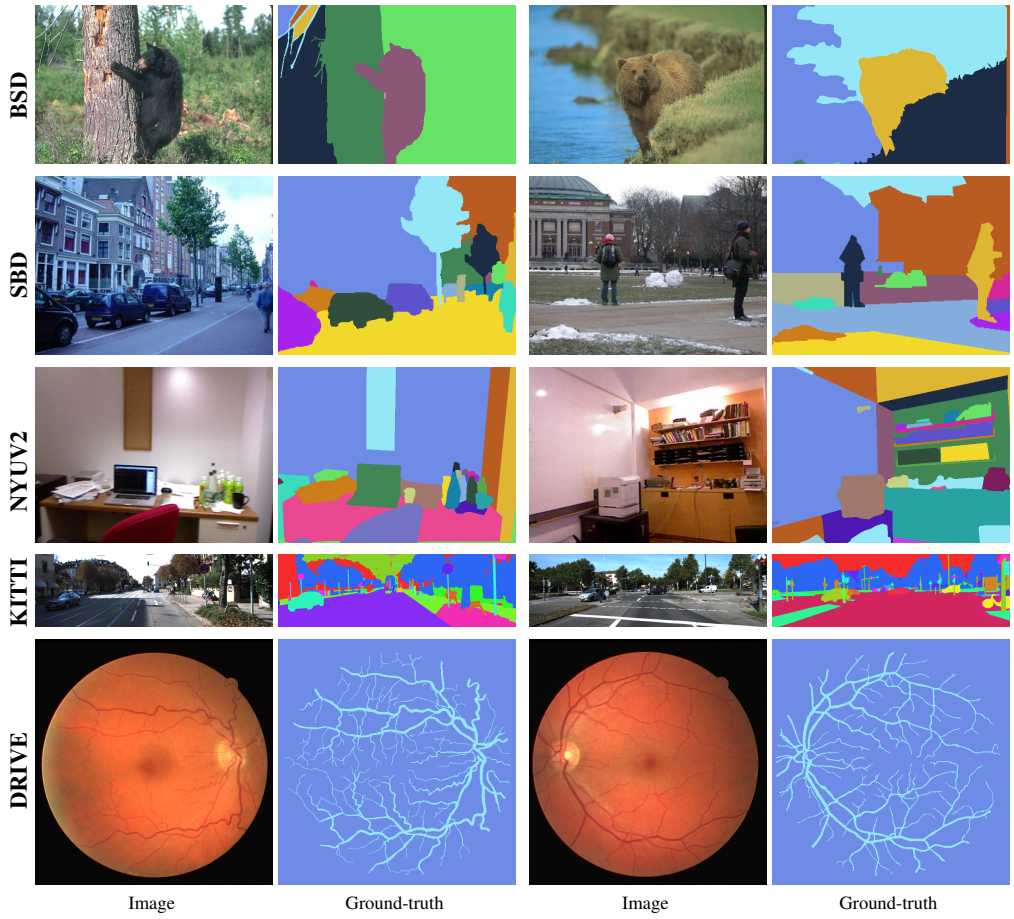


Figure 11: **Examples from the datasets used in our experimental validations:** BSD, SBD, NYUV2, KITTI, and DRIVE.

while large objects can be well captured in terms of area even when their boundaries are not accurately aligned with the ground-truth. The standard metric for this aspect is Boundary Recall (BR), which measures how well the ground-truth contours $\mathcal{B}(\mathcal{G})$ are detected by the superpixel boundaries $\mathcal{B}(\mathcal{S})$:

$$\mathrm{BR}(\mathcal{S},\mathcal{G}) = \frac{1}{|\mathcal{B}(\mathcal{G})|} \sum_{p\in\mathcal{B}(\mathcal{G})} \delta[\min_{q\in\mathcal{B}(\mathcal{S})} \|p-q\| < \epsilon], \tag{6}$$

where $\delta[a] = 1$ if a is true and 0 otherwise, and ϵ is a distance threshold set to 2 pixels (Liu et al., 2011; Van den Bergh et al., 2012).

Following (Stutz et al., 2018; Giraud et al., 2017), BR should be reported together with Boundary Precision (BP), which measures the proportion of superpixel boundary detections that correspond to true object contours. This prevents noisy superpixel boundaries from being favored by recall alone. Therefore, Fig. 9 jointly reports BR and BP through BR/BP curves, rather than considering either metric independently.

For the AUC results in Tab. 5, we report the AUC of BR $\times$ AUC of BP. Since BR increases and BP decreases with the superpixel scales, and their curves do not share the same start and end points, we summarize their joint behavior with this product instead of directly integrating a BR/BP curve.

A.3 Training settings

Like other deep superpixel methods, SPAM 2 is trained on the BSD dataset (Martin et al., 2001). Following CDS (Xu et al., 2024), the superpixel predictor is first trained to learn local assignment features that are consistent with image structure, using a batch size of 8 for 250k iterations. We then finetune the feature fusion module for 30k iterations with the same supervision strategy. The regularity-oriented variants are finetuned for 30k iterations with the oracle superpixel maps (Sec. 3.4). In all stages, we use a polynomial learning-rate schedule with an initial learning rate of 5×10^{-4} and 5000 warmup iterations. As in (Xu et al., 2024), we set weight $w_p = 0.003$ (equation 1) to balance assignment accuracy and regularity constraints. All models were trained and evaluated on an NVIDIA RTX A5000.

B Model design and efficiency

B.1 Joint object-superpixel feature fusion

The joint representation introduced in the main paper combines the low-level superpixel stream with intermediate features from the object encoder. Because the two backbones can have different spatial resolutions and channel counts, the object features are first resized and projected to the same 32-channel space as the superpixel features. Fig. 12 details the architectures of the four fusion strategies considered in our ablation. In Fig. 13 we evaluate these different fusion strategies. Overall, all three approaches achieve very similar segmentation performance, improving the one of SPAM 2 without using object features. This suggests that the gains of SPAM 2 primarily stem from the availability of object-level information rather than from a specific fusion mechanism.

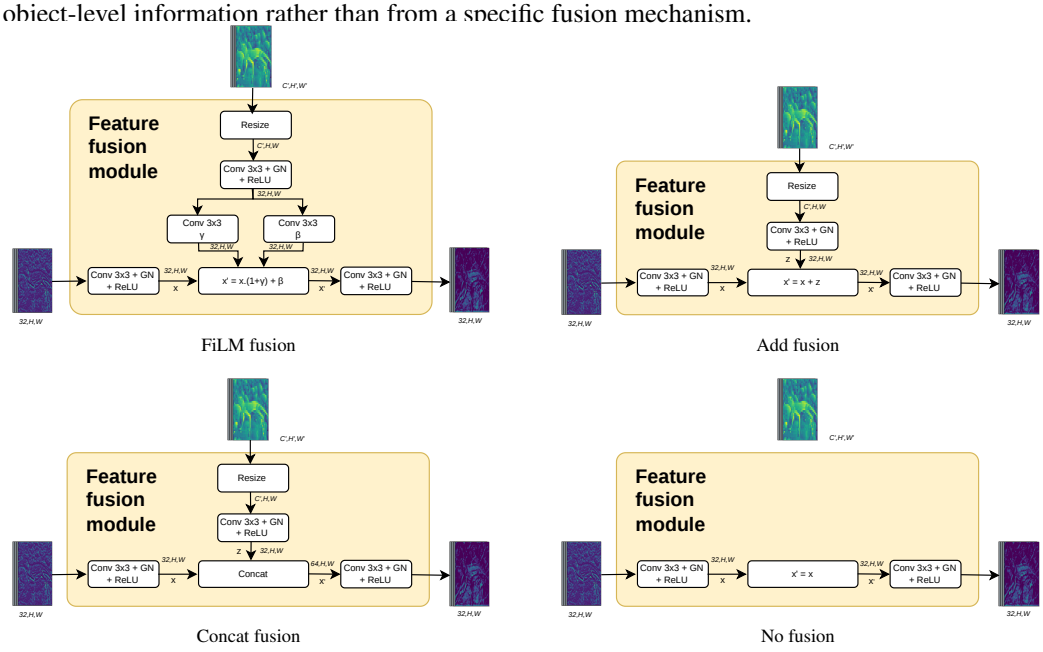


Figure 12: **Fusion configurations.** Details of the four fusion strategies compared in Fig. 13.

With FiLM fusion, the object stream predicts per-channel scale and bias parameters γ and β, which modulate the superpixel features as $x' = x \odot (1+\gamma) + \beta$. This is the configuration used by the final SPAM 2 model. It preserves the superpixel feature basis while allowing the object representation to selectively amplify, attenuate, or shift channels.

B.2 Ablation object model

Choice of object model. Fig. 14 compares runtime across the requested number of superpixels. From $K = 150$ to 3000, SPAM rises from about 4.4 to 5.7 seconds per image, while SPAM 2 stays near 3.3–3.4 seconds and SPAM 2 Fast near 0.3–0.4 seconds. Without object masks, the gap isolates

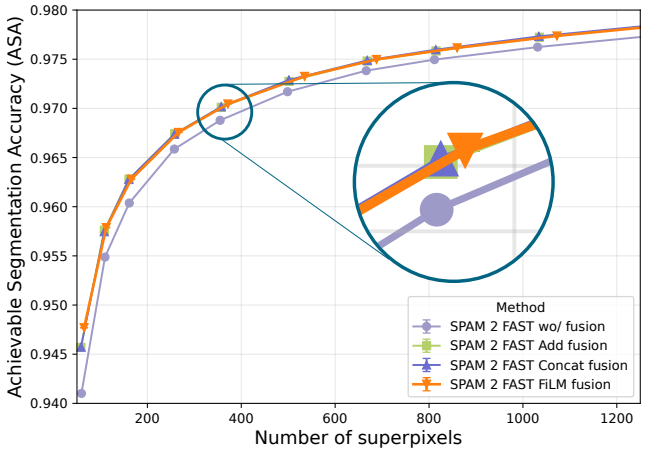


Figure 13: **Impact of the fusion module on accuracy.**

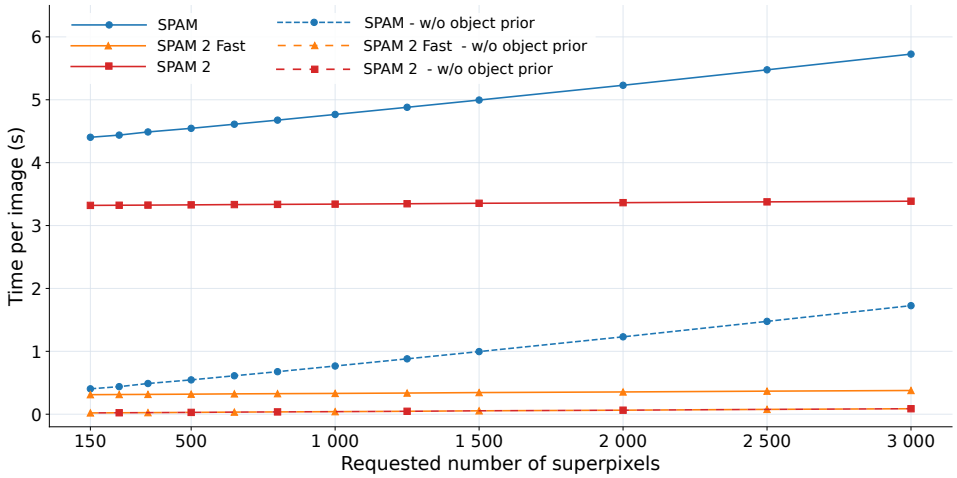


Figure 14: **Runtime scalability over the requested number of superpixels.** The evaluation is done on the BSD images of size 321×481.

the superpixel stage: SPAM rises from about 0.4 to 1.7 seconds, whereas both SPAM 2 variants remain below 0.1 seconds. We use SAM 1 as it remains the most suitable foundation model for our class-agnostic, automatic full-image setting. SAM 2 is primarily video-oriented and yielded weaker priors in our experiments, while SAM 3 targets promptable concept-level segmentation. Moreover, SAM 1 provides lightweight distilled variants that enable our fast configuration.

Regularity from oracle superpixels. We finally study the regularity of the generated superpixels, which must balance object adherence with compact and regular shapes. As shown in Fig. 15, our proposed oracle supervision (Sec. 3.4) markedly improves regularity: SPAM 2 Reg and SPAM 2 Fast Reg reach a substantially higher Global Regularity (GR) than SPAM 2, while keeping the accuracy (ASA) close. As the oracle maps are more regular inside each object yet still constrained by the annotations, the model learns to produce smoother, more homogeneous regions without sacrificing object adherence.

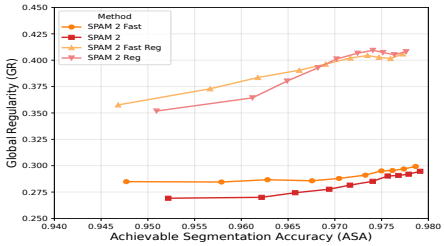


Figure 15: **Accuracy-regularity trade-off with oracle superpixel map supervision.**

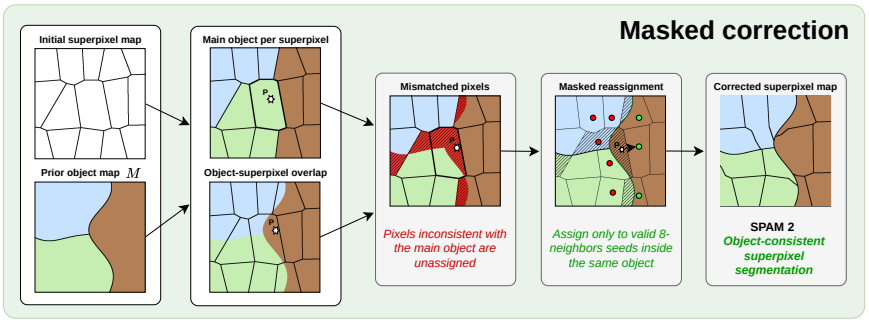


Figure 16: **Pipeline of the masked correction.** From the initial superpixel assignment and the prior object map, we determine each superpixel's main object as the one with most overlap (left). Pixels whose object label disagrees with their superpixel's main object are flagged as mismatched and unassigned (center, red). Each is then reassigned among its 8 neighboring seeds, restricted to the ones inside the same object (green valid, red invalid) and excluding its own seed. The result is an object-consistent superpixel segmentation whose boundaries respect the prior object map (right).

B.3 Details on masked correction algorithm

Masked correction. Fig. 16 shows how the masked correction identifies pixels inconsistent with their superpixel's main object and reassigns them locally to obtain an object-consistent superpixel segmentation.

B.4 Runtime comparison

Tab. 4 compares the runtime of SPAM (Walther et al., 2025) and our SPAM 2 while excluding the object computation step, in order to isolate the cost of producing the final superpixel segmentation from the object cues. In SPAM, this stage is dominated by the object-based iterative clustering, which accounts for 750 ms out of a total runtime of 751 ms. By contrast, SPAM 2 replaces this costly optimization with an assignment-based framework, reducing the clustering time to 17 ms. The additional encoder, decoder, and joint assignment steps introduce only a small overhead, while the connectivity step remains unchanged. Overall, for $K = 1000$ on BSD, SPAM 2 reduces the runtime from 751 ms to 19.5 ms. This confirms that an important efficiency gain comes from removing the iterative clustering bottleneck.

Table 4: **Runtime comparison excluding object segmentation model.** We compare SPAM's object-based iterative clustering with our assignment-based framework. Times are reported in milliseconds (ms) for $K = 1000$ and averaged over BSD.

Framework	SP Encoder	SP Decoder	Joint	SP algorithm	Total time (ms)
SPAM (Walther et al., 2025)	1	0	0	750	**751.0**
SPAM 2	1	1	0.5	17	**19.5**

C Extended benchmark results

C.1 Benchmark results

We compare SPAM 2 with standard traditional and deep-learning-based superpixel methods in Tab. 5. Overall, SPAM 2 obtains the best accuracy among the evaluated methods on most datasets, reaching the highest ASA on BSD, NYUv2 and SBD, and ranking second on KITTI and DRIVE. For contour quality, we report BR×BP, computed as AUC(BR)×AUC(BP). SPAM 2 and its fast variant rank either first or second on most datasets, indicating that they recover object boundaries while keeping precise, low-noise contours. On DRIVE, SPAM 2 ranks second, just behind CDS. This dataset is more challenging for object-guided methods, as foundation model priors are likely less reliable on thin retinal vessels: the original SPAM, which fully relies on them, is strongly

Table 5: **Quantitative results comparison across datasets and metrics.** AUC results over the range [100,1000]. **Best**, second, and *third* best values are highlighted.

	Segmentation accuracy - ASA					Contour detection - BR × BP					Regularity - GR				
Method	BSD	NYUV2	SBD	KITTI	DRIVE	BSD	NYUV2	SBD	KITTI	DRIVE	BSD	NYUV2	SBD	KITTI	DRIVE
SLIC (Achanta et al., 2012)	0.9558	0.9278	0.9456	0.9231	0.9092	0.07193	0.06172	0.09301	0.05632	0.07895	0.3854	0.4146	0.4344	*0.3428*	**0.6184**
SEEDS (Van den Bergh et al., 2012)	0.9539	0.9203	0.9366	0.932	0.9187	0.05942	0.05364	0.08217	0.06973	0.1249	0.2161	0.2052	0.2475	0.2053	0.3452
SSN (Jampani et al., 2018)	0.9658	0.9384	0.9548	0.9395	*0.9256*	0.09136	0.07695	0.1055	**0.07906**	0.1232	0.2530	0.2258	0.2808	0.2446	0.3234
SEAL (Tu et al., 2018)	0.9666	0.9412	0.9567	0.9455	0.8875	0.07718	0.06647	0.09828	0.07141	0.09751	0.1622	0.1386	0.1306	0.1677	0.1162
SFCN (Yang et al., 2020)	0.9651	0.9406	0.9554	0.9389	0.9185	0.08709	0.07519	0.1057	0.0654	0.1086	0.3286	0.3073	0.3287	0.2976	0.3600
LNSNET (Zhu et al., 2021)	0.9502	0.9077	0.9399	0.8956	0.914	0.06774	0.05325	0.09306	0.04182	0.1052	0.2041	0.1836	0.2380	0.1303	0.3664
AINET (Wang et al., 2021)	0.9651	0.9380	0.9551	0.9392	0.9113	0.09212	0.07678	0.1107	0.07088	0.09347	0.3203	0.3022	0.3248	0.2881	0.3633
VSSS (Zhou et al., 2023)	0.9430	0.9136	0.9408	0.9136	0.9225	0.07579	0.07064	0.1101	0.05961	0.1372	0.1732	0.2079	0.1927	0.1785	0.3637
CDS (Xu et al., 2024)	0.9679	0.9422	0.9571	0.9435	**0.9309**	0.09467	0.07929	0.1099	0.07525	**0.1514**	0.3176	0.2929	0.3152	0.2887	0.3273
HHTS (Chang et al., 2024)	0.9471	0.8762	0.935	0.8984	0.913	0.0697	0.06077	0.09439	0.07442	*0.1341*	0.0960	0.0862	0.1137	0.0667	0.0701
SPAM (Walther et al., 2025)	0.9713	0.9519	0.9582	0.9431	0.9052	*0.09594*	0.08402	0.105	0.07271	0.05953	**0.4777**	**0.4463**	**0.4824**	**0.4976**	0.6093
SPAM 2 Fast Reg	0.9706	0.9517	0.9604	*0.9478*	0.9199	0.0915	0.08784	0.1137	0.07384	0.09558	*0.3977*	*0.3556*	0.3808	0.3502	0.4787
SPAM 2 Fast	0.9717	0.9545	0.9629	**0.9502**	0.9231	0.1007	0.09652	**0.1234**	*0.07559*	0.1053	0.2913	0.2628	0.2856	0.2440	0.3261
SPAM 2 Reg	*0.9714*	*0.9541*	*0.9609*	0.9478	0.9180	0.09307	*0.09057*	*0.1143*	0.07154	0.0934	0.3999	0.3515	*0.3869*	0.3328	*0.5023*
SPAM 2	**0.9725**	**0.9565**	**0.9633**	0.9500	0.9261	**0.1021**	**0.09827**	0.123	0.07675	0.1221	0.2842	0.2618	0.2837	0.2382	0.3205

affected, whereas SPAM 2 remains competitive thanks to its learned local assignment. The regularized variants are slightly less accurate here, as such thin and elongated structures naturally favor more irregular superpixels. Finally, for regularity (GR), the original SPAM remains the most regular method, owing to its rather costly iterative clustering on object-based seeds. The regular versions of SPAM 2, which distill this regularity from SPAM through oracle superpixel maps, rank second or third on most datasets, while keeping a much better accuracy-runtime trade-off.

C.2 Additional quantitative results

Fig. 17 provides the full curves for the four datasets that are summarized only by AUC values in the main paper. Across SBD and NYUv2, the full and fast SPAM 2 variants consistently occupy the high-accuracy regime, while the regularized variants move the trade-off toward higher GR. KITTI shows the same overall behavior despite its strong perspective changes. DRIVE is the most challenging out-of-domain case: its long, thin vessels favor methods that tolerate highly elongated and less regular regions, explaining why the accuracy and contour ordering differs from the natural-image benchmarks. We also report the rankings of all evaluated methods across datasets in Tab. 6.

To complement the average quantitative comparison, Fig. 18 reports the distribution of ASA scores across BSD images for different scales. Beyond improving the mean accuracy, SPAM 2 shifts the overall distribution upward, with higher quartiles than previous methods. This shows that the improvements are consistent across the dataset and are not driven by a limited number of favourable images.

Tabs. 7–10 report two-sided paired Wilcoxon tests on per-image ASA interpolated at $K = 500$. Each cell compares its row method with its column method: green/red indicate a significant gain/loss, white indicates no significant difference, and gray marks self-comparisons. All four SPAM 2 variants outperform SPAM on NYUv2, SBD, KITTI, and DRIVE ($p < 0.001$). On BSD (Tab. 1), SPAM 2 and SPAM 2 Fast also outperform SPAM ($p < 0.001$), while the regularized variants remain on par with it: SPAM 2 Reg is still significantly better ($p = 0.049$), and SPAM 2 Fast Reg shows a small difference in favor of SPAM ($p = 0.009$; AUC ASA of 0.9706 vs. 0.9713), as these variants deliberately trade a small amount of accuracy for higher regularity, at a much lower computational cost than SPAM. The full model also outperforms CDS on NYUv2, SBD, and KITTI ($p < 0.001$), whereas CDS outperforms it on DRIVE ($p < 0.001$). The high number of non-significant results on DRIVE may be due to the limited size of the dataset, which contains only 20 images and therefore provides less statistical power.

C.3 Additional qualitative results

We provide additional visual examples in Figs. 19, 20, 21 and 22 to complement the quantitative evaluation. These examples illustrate how the different methods behave on images with varied content, object scales, and boundary complexity. Beyond global metric values, visual comparisons make it possible to assess whether the generated superpixels follow meaningful image structures, preserve fine details, and remain spatially coherent.

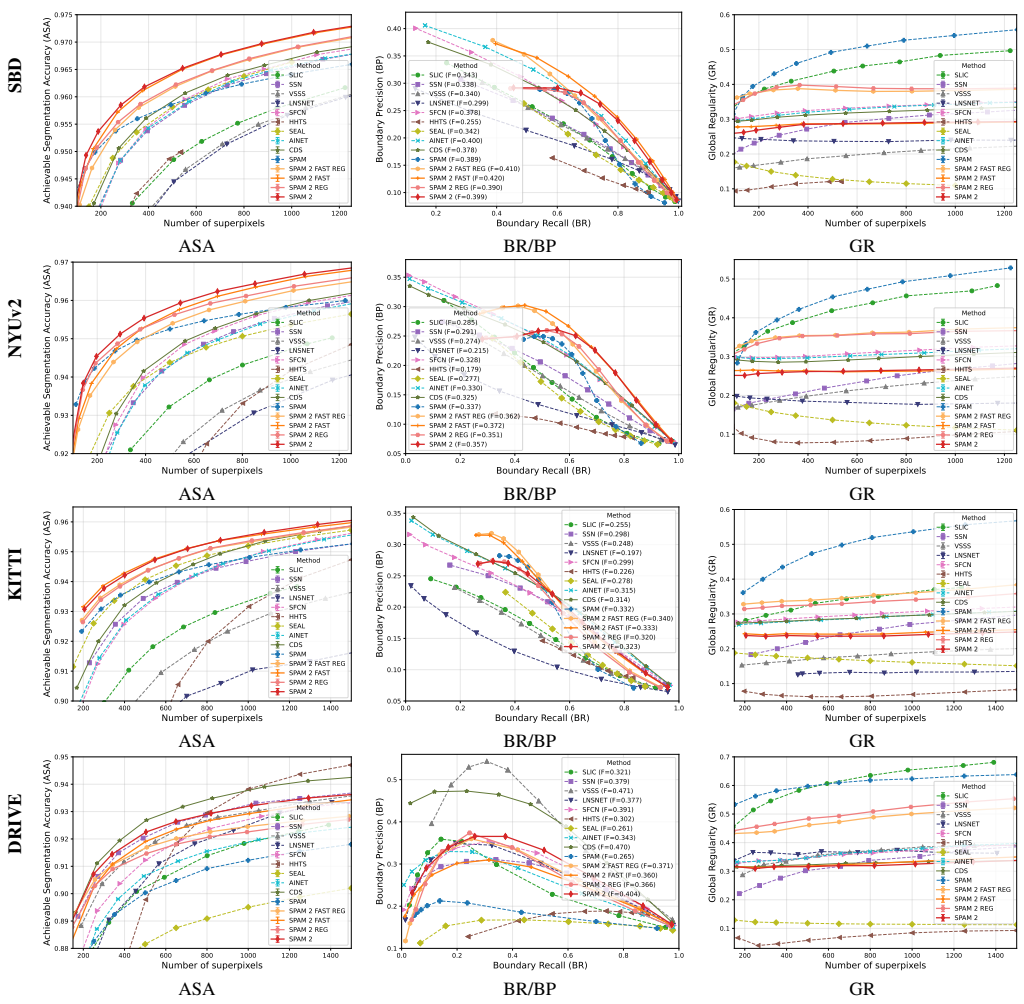


Figure 17: **Detailed quantitative results on additional datasets.** Rows correspond to datasets and columns report ASA as a function of the requested number of superpixels, the BR–BP contour trade-off, and GR as a function of the requested number of superpixels. We also provide F-measure for BR–BP curves as in Stutz et al. (2018).

In particular, the qualitative results highlight the trade-off between boundary adherence and regularity. Methods that strongly follow image contours may better capture small or elongated structures, but can also produce irregular regions. Conversely, more regular superpixels are often easier to interpret visually, but may miss fine object boundaries. The examples therefore provide a complementary view of the behavior summarized by the ASA, contour, and GR metrics.

We also provide in Fig. 23, a qualitative example of semantic segmentation on the downstream task evaluated in Sec. 4.4.

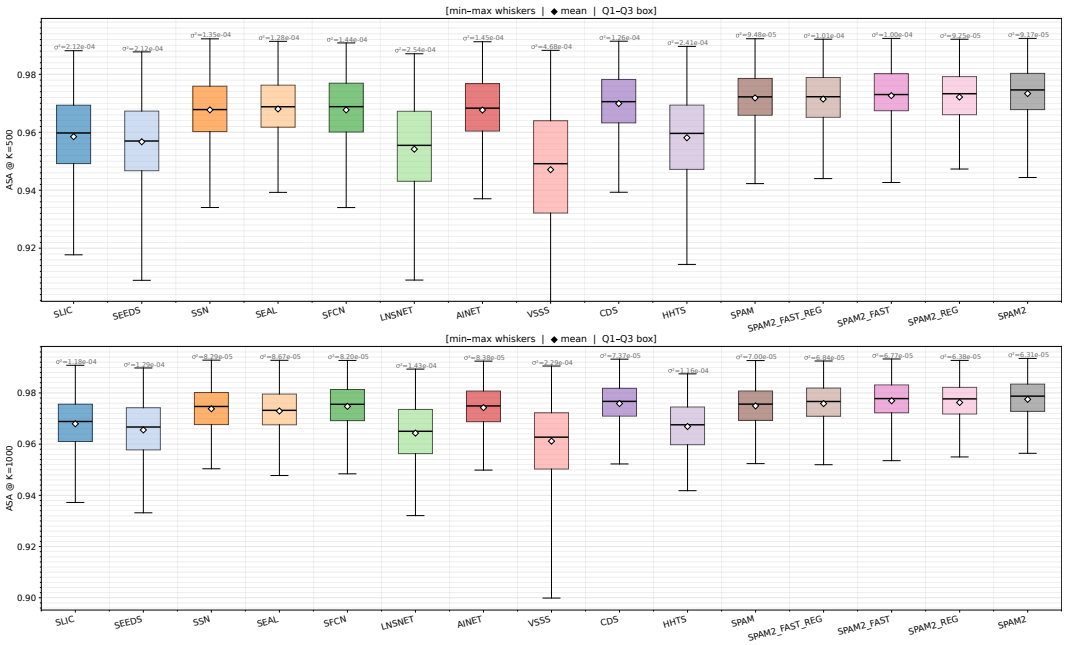


Figure 18: **Distribution of segmentation accuracy on BSD.** Box plots report ASA at 500 superpixels (top) and 1000 superpixels (bottom) for each compared method. Boxes indicate the first and third quartiles, whiskers the minimum and maximum values, and diamonds the mean. SPAM 2 variants exhibit the highest overall distributions, with consistently higher quartiles than previous methods. Note that the gap on average accuracy between SPAM and SPAM 2 variants is on par with the gap between other deep learning-based methods.

Table 6: **Method rankings by selected metric AUC across datasets.** Rank 1 is best for each metric and dataset.

	ASA					AUC(BR) × AUC(BP)					GR				
Method	BSD	NYUV2	SBD	KITTI	DRIVE	BSD	NYUV2	SBD	KITTI	DRIVE	BSD	NYUV2	SBD	KITTI	DRIVE
SLIC (Achanta et al., 2012)	11	11	11	12	13	12	12	14	14	14	4	2	2	3	1
SEEDS (Van den Bergh et al., 2012)	12	12	14	11	7	15	14	15	11	4	11	12	11	11	9
SSN (Jampani et al., 2018)	8	9	10	8	3	8	7	9	1	5	10	10	10	8	12
SEAL (Tu et al., 2018)	7	7	7	5	15	10	11	11	9	10	14	14	14	13	14
SFCN (Yang et al., 2020)	10	8	8	10	8	9	9	8	12	7	5	5	5	5	8
LNSNET (Zhu et al., 2021)	13	14	13	15	10	14	15	13	15	9	12	13	12	14	5
AINET (Wang et al., 2021)	9	10	9	9	12	6	8	5	10	12	6	6	6	7	7
VSSS (Zhou et al., 2023)	15	13	12	13	5	11	10	6	13	2	13	11	13	12	6
CDS (Xu et al., 2024)	6	6	6	6	1	4	6	7	4	1	7	7	7	6	10
HHTS (Chang et al., 2024)	14	15	15	14	11	13	13	12	5	3	15	15	15	15	15
SPAM (Walther et al., 2025)	4	4	5	7	14	3	5	10	7	15	1	1	1	1	2
SPAM 2 Fast Reg	5	5	4	3	6	7	4	4	6	11	3	3	4	2	4
SPAM 2 Fast	2	2	2	1	4	2	2	1	3	8	8	8	8	9	11
SPAM 2 Reg	3	3	3	4	9	5	3	3	8	13	2	4	3	4	3
SPAM 2	1	1	1	2	2	1	1	2	2	6	9	9	9	10	13

Table 7: **Statistical significance of ASA gains on NYUv2.** Two-sided Wilcoxon signed-rank tests on per-image ASA interpolated at $K = 500$. Cells report p-values; green/red indicate significant gains/losses for the row method, and $\sim$ a non-significant difference.

SLIC(Achanta et al., 2012)	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SEAL(Tu et al., 2018)	$\uparrow < .001^*$	–	$\uparrow .005^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\sim .060$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SSN(Jampani et al., 2018)	$\uparrow < .001^*$	$\downarrow .005^*$	–	$\downarrow < .001^*$	$\uparrow < .001^*$	$\sim .345$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SFCN(Yang et al., 2020)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
LNSNET(Zhu et al., 2021)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
AINET(Wang et al., 2021)	$\uparrow < .001^*$	$\sim .060$	$\sim .345$	$\downarrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
VSSS(Zhou et al., 2023)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
CDS(Xu et al., 2024)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
HHTS(Chang et al., 2024)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM(Walther et al., 2025)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Fast Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Fast	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$
SPAM 2	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–
Method	SLIC	SEAL	SSN	SFCN	LNSNET	AINET	VSSS	CDS	HHTS	SPAM	SPAM 2 Fast Reg	SPAM 2 Fast	SPAM 2 Reg	SPAM 2

Table 8: **Statistical significance of ASA gains on SBD.** Two-sided Wilcoxon signed-rank tests on per-image ASA interpolated at $K = 500$. Cells report p-values; green/red indicate significant gains/losses for the row method, and $\sim$ a non-significant difference.

SLIC(Achanta et al., 2012)	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\sim .762$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SEAL(Tu et al., 2018)	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\sim .111$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow .007^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SSN(Jampani et al., 2018)	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\uparrow < .001^*$	$\sim .075$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SFCN(Yang et al., 2020)	$\uparrow < .001^*$	$\sim .111$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
LNSNET(Zhu et al., 2021)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow .030^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
AINET(Wang et al., 2021)	$\uparrow < .001^*$	$\downarrow < .001^*$	$\sim .075$	$\downarrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
VSSS(Zhou et al., 2023)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow .030^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
CDS(Xu et al., 2024)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow .004^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
HHTS(Chang et al., 2024)	$\sim .762$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM(Walther et al., 2025)	$\uparrow < .001^*$	$\uparrow .007^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow .004^*$	$\uparrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Fast Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow .008^*$	$\downarrow < .001^*$
SPAM 2 Fast	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow .009^*$
SPAM 2 Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .008^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$
SPAM 2	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .009^*$	$\uparrow < .001^*$	–
Method	SLIC	SEAL	SSN	SFCN	LNSNET	AINET	VSSS	CDS	HHTS	SPAM	SPAM 2 Fast Reg	SPAM 2 Fast	SPAM 2 Reg	SPAM 2

Table 9: **Statistical significance of ASA gains on KITTI.** Two-sided Wilcoxon signed-rank tests on per-image ASA interpolated at $K = 500$. Cells report p-values; green/red indicate significant gains/losses for the row method, and $\sim$ a non-significant difference.

SLIC(Achanta et al., 2012)	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SEAL(Tu et al., 2018)	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .037^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SSN(Jampani et al., 2018)	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SFCN(Yang et al., 2020)	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow .003^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
LNSNET(Zhu et al., 2021)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
AINET(Wang et al., 2021)	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow .003^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
VSSS(Zhou et al., 2023)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
CDS(Xu et al., 2024)	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
HHTS(Chang et al., 2024)	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM(Walther et al., 2025)	$\uparrow < .001^*$	$\downarrow .037^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Fast Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\downarrow < .001^*$	$\sim .249$	$\downarrow < .001^*$
SPAM 2 Fast	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow .009^*$
SPAM 2 Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .249$	$\downarrow < .001^*$	–	$\downarrow < .001^*$
SPAM 2	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow .009^*$	$\uparrow < .001^*$	–
Method	SLIC	SEAL	SSN	SFCN	LNSNET	AINET	VSSS	CDS	HHTS	SPAM	SPAM 2 Fast Reg	SPAM 2 Fast	SPAM 2 Reg	SPAM 2

Table 10: **Statistical significance of ASA gains on DRIVE.** Two-sided Wilcoxon signed-rank tests on per-image ASA interpolated at $K = 500$. Cells report p-values; green/red indicate significant gains/losses for the row method, and $\sim$ a non-significant difference.

SLIC(Achanta et al., 2012)	–	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow .044^*$	$\sim .058$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\sim .143$	$\sim .245$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SEAL(Tu et al., 2018)	$\downarrow < .001^*$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow .003^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SSN(Jampani et al., 2018)	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .048^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .076$	$\uparrow < .001^*$	$\downarrow .036^*$
SFCN(Yang et al., 2020)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .090$	$\downarrow < .001^*$	$\uparrow .004^*$	$\uparrow < .001^*$	$\downarrow .002^*$	$\downarrow < .001^*$	$\sim .114$	$\downarrow < .001^*$
LNSNET(Zhu et al., 2021)	$\uparrow .044^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\sim .202$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\sim .083$	$\uparrow .006^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
AINET(Wang et al., 2021)	$\sim .058$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\sim .202$	–	$\downarrow .002^*$	$\downarrow < .001^*$	$\uparrow .044^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
VSSS(Zhou et al., 2023)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow .048^*$	$\sim .090$	$\uparrow < .001^*$	$\uparrow .002^*$	–	$\downarrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .927$	$\sim .330$	$\sim .261$	$\downarrow .006^*$
CDS(Xu et al., 2024)	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$
HHTS(Chang et al., 2024)	$\sim .143$	$\uparrow .003^*$	$\downarrow < .001^*$	$\downarrow .004^*$	$\sim .083$	$\downarrow .044^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\sim .522$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM(Walther et al., 2025)	$\sim .245$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow .006^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\sim .522$	–	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Fast Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\uparrow .002^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .927$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–	$\downarrow .011^*$	$\uparrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Fast	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .076$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .330$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .011^*$	–	$\uparrow < .001^*$	$\downarrow < .001^*$
SPAM 2 Reg	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\sim .114$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\sim .261$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\downarrow < .001^*$	$\downarrow < .001^*$	–	$\downarrow < .001^*$
SPAM 2	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .036^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow .006^*$	$\downarrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	$\uparrow < .001^*$	–
Method	SLIC	SEAL	SSN	SFCN	LNSNET	AINET	VSSS	CDS	HHTS	SPAM	SPAM 2 Fast Reg	SPAM 2 Fast	SPAM 2 Reg	SPAM 2

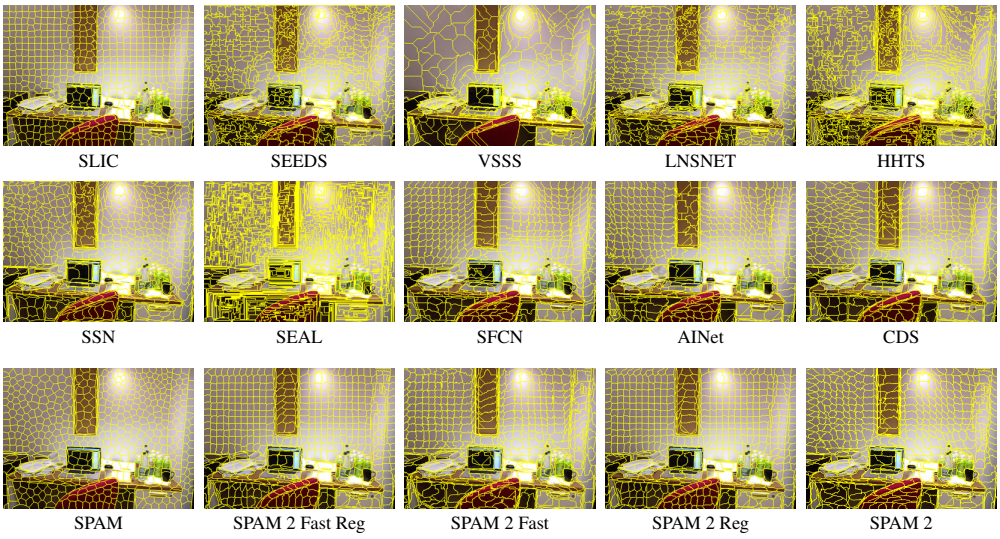


Figure 19: **Qualitative comparison on NYUv2.** Superpixel boundaries with $K = 650$ requested superpixels. Object-guided variants reduce leakage across furniture and wall boundaries, while regularized variants produce smoother regions in uniform areas.

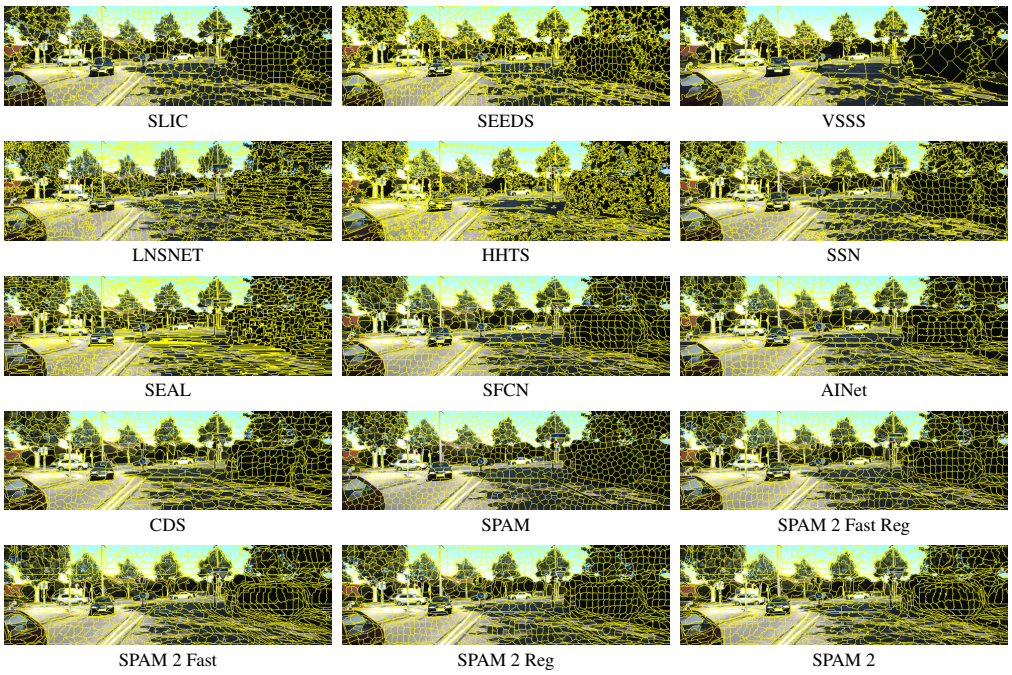


Figure 20: **Qualitative comparison on KITTI.** Superpixel boundaries with $K = 1000$ requested superpixels. The example illustrates robustness to wide images, perspective, and large changes in apparent object scale.

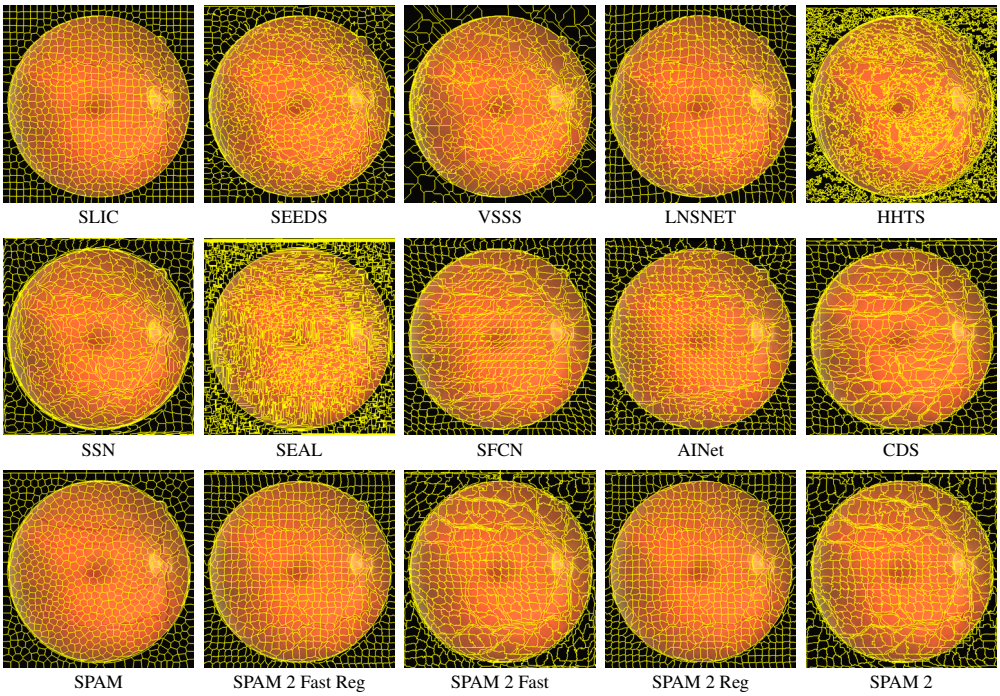


Figure 21: **Qualitative comparison on DRIVE.** Superpixel boundaries with $K = 800$ requested superpixels. Methods differ strongly in their willingness to form elongated regions along the retinal vessels, highlighting the tension between contour adherence and geometric regularity.

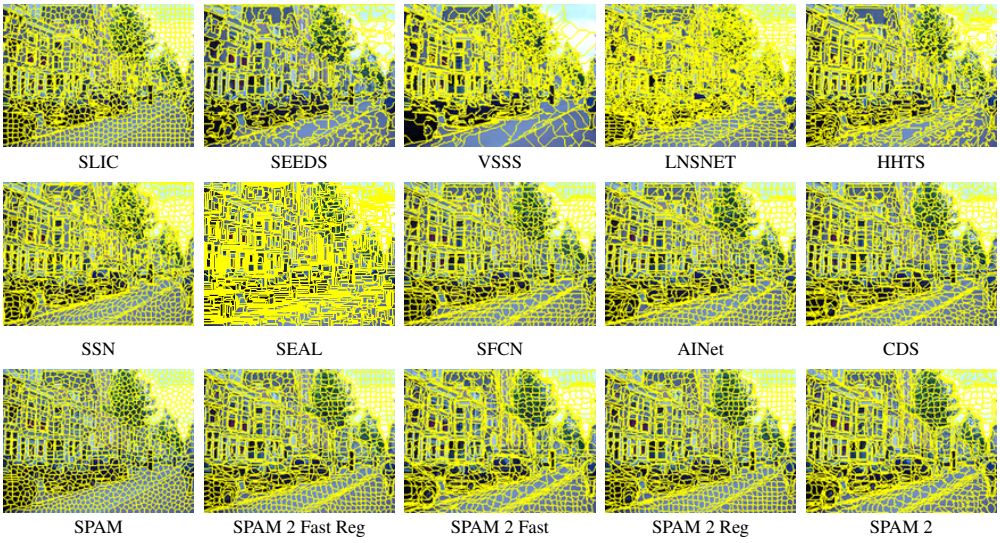


Figure 22: **Qualitative comparison on SBD.** Superpixel boundaries with $K = 1250$ requested superpixels. SPAM 2 variants preserve the dense object contours while the regularized variants favor more homogeneous regions inside objects.

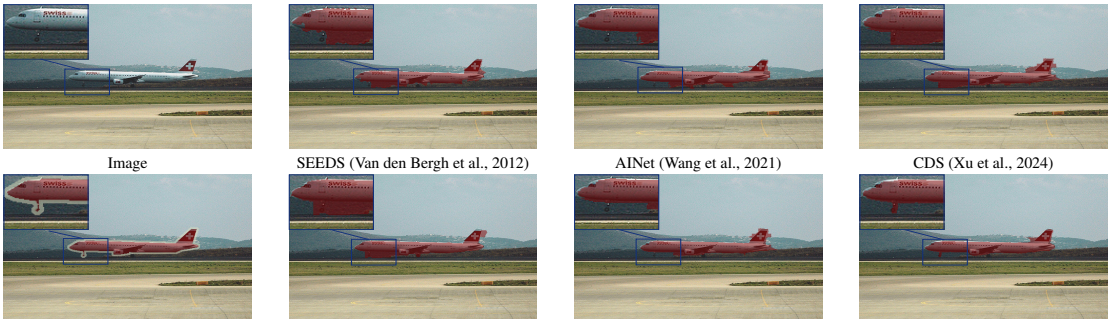


Figure 23: **Qualitative downstream segmentation example.** Semantic segmentations with CAST (Ke et al., 2024) using different superpixel tokenizers. Superpixels from SPAM 2 produce tighter and cleaner object boundaries (see zoomed nose).