Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Semantic boundary refinement for traffic salient object segmentation using residual logit correction

Run Yuan
Hunan Automotive Engineering Vocational University
Haonan Long
ilovezzptc@163.com
Hunan Automotive Engineering Vocational University
Wei Deng
Guangzhou Civil Aviation College
Xueli Wang
Hunan Automotive Engineering Vocational University

Research Article

Keywords: autonomous driving, traffic salient object segmentation, semantic boundary refinement, residual correction, boundary evaluation, TSOD10K

Posted Date: November 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11383472/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Semantic boundary refinement for traffic salient object segmentation using residual logit correction

Run Yuan[1], Haonan Long[1*], Wei Deng[2], Xueli Wang[1]
[1]Hunan Automotive Engineering Vocational University, Zhuzhou, Hunan, China
[2]Guangzhou Civil Aviation College, Guangzhou, China
∗Corresponding author: Email addresses: ilovezzptc@163.com (Haonan Long)

Abstract: Traffic salient object segmentation requires complete regions and reliable contours for pedestrians, vehicles, signs, and obstacles, yet driving scenes contain shadows, lane markings, reflections, occlusion, and small targets that make these two goals conflict. We study this problem as task-aware semantic boundary refinement rather than generic edge detection. The supervised boundary is explicitly defined as the morphological boundary of the binary traffic-salient mask; it therefore excludes non-salient image gradients but does not encode a hierarchy of safety importance. EdgeRefine augments DeepLabV3-ResNet50 with a zero-initialized residual correction module in mask-logit space. In an 18run study on TSOD10K with three seeds per setting, the selected consistency weight of 0.01 changes ODS from 0.4875 ± 0.0016 to 0.4906 ± 0.0026,Boundary IoU from 0.5847±0.0020 to 0.5881±0.0004, and edge MAE from 0.004621 ± 0.000030 to 0.004496 ± 0.000008. The same setting slightly decreases Dice and increases Chamfer distance, and most paired 95% confidence intervals for overlap metrics include zero; the gains are therefore modest rather than universally significant. A SegFormer-B0 portability study yields larger improvements in Dice, Boundary F1, and Boundary IoU, while zero-shot DUTS-TE evaluation exposes backbone-dependent transfer: the SegFormer variant improves most region and overlap metrics but not Chamfer, whereas the DeepLabV3 variants generally degrade. Condition-wise HD95and directed-distance diagnostics show that local overlap improvements can coexist with distant false contour fragments. These results position EdgeRefine as a lightweight, model-portable boundary correction mechanism with measurable benefits and explicit geometric and cross-domain limitations.

Keywords: autonomous driving, traffic salient object segmentation, semantic boundary refinement, residual correction, boundary evaluation, TSOD10K

1. Introduction

Autonomous driving systems must distinguish pedestrians, vehicles, traffic signs, cones, and unexpected obstacles from visually complex road scenes. Traffic salient object segmentation (TSOS) addresses this need by predicting foreground regions according to the benchmark's traffic-saliency annotations. For this task, a useful prediction must preserve both the extent of the salient foreground and the location of its contour.

Contour errors can affect the spatial interpretation of a scene even when most foreground pixels are correctly classified. A displaced vehicle contour changes the apparent free space, a broken pedestrian silhouette reduces object integrity, and fragments associated with shadows or lane markings introduce irrelevant structure. These errors are particularly difficult around small targets, occlusions, and adjacent road users, where foreground selection and geometric precision interact.

Early saliency formulations emphasized visual attention and appearance contrast, as illustrated by Itti and Koch [1] and Achanta et al. [2]. In the present traffic task, however, the annotated foreground determines which structures should be retained. The resulting boundary is therefore a semantic target: it follows the selected foreground rather than every visible transition in the image. This distinction matters because a weak-contrast pedestrian contour may belong to the target mask, whereas a prominent road marking or reflection may not.

Improving region prediction alone does not directly address this distinction. A largely correct mask can still contain displaced contours, merged foreground regions, or isolated fragments. Conversely, stronger responses to image edges can introduce boundaries unrelated to the annotated foreground. The central design problem is therefore to use boundary information to correct a semantic mask while keeping the correction aligned with the same foreground definition.

We define a task-aware semantic boundary as the morphological contour of the binary traffic-salient mask. Here, task awareness refers to the source of the supervision, not to an additional estimate of driving risk. All annotated foreground pixels receive equal status, and the binary mask does not preserve instance separation where adjacent foreground objects merge. This explicit scope makes it possible to study mask-aligned

correction without conflating it with generic edge detection or risk-ranked perception.
Figure 1 illustrates the motivating errors on representative TSOD10K scenes. The first row shows incomplete recovery of a small pedestrian beside a dominant vehicle; the second shows locally merged or displaced contours around adjacent road users; and the third shows unstable contour continuity under occlusion and irregular pedestrian–vehicle geometry. These examples motivate a correction mechanism that uses semantic context as well as contour evidence. The remaining discrepancies also show why local boundary overlap and geometric distance must be evaluated separately.
This study asks whether a compact boundary-guided module can refine an existing semantic mask without replacing its encoder or disrupting its initial prediction. It also examines whether the resulting changes persist across random seeds, apply to a Transformer backbone, and transfer beyond the traffic benchmark. Section 2 reviews the relevant segmentation, boundary-refinement, and edge-detection methods to establish the specific methodological differences.

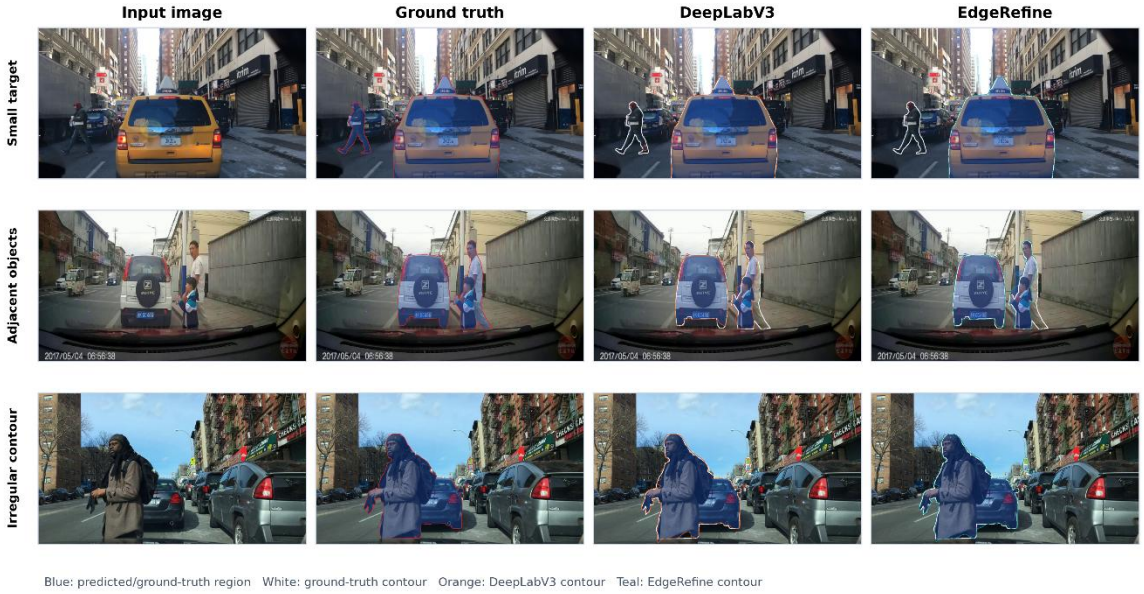


Figure 1: *Motivation on representative TSOD10K scenes. Blue denotes the salient region, white the ground-truth contour, orange the DeepLabV3 contour, and teal the EdgeRefine contour. The three rows expose complementary failure modes: omission of a small safety-relevant target, boundary displacement or merging between adjacent objects, and unstable contour continuity around irregular pedestrian–vehicle silhouettes. Remaining white–teal discrepancies also make clear that EdgeRefine does not solve all geometric errors.*

We address these questions with EdgeRefine, a zero-initialized residual correction module operating in mask-logit space. The contributions are as follows:
(1) We formulate traffic-salient boundary refinement using an explicit binary, mask-derived target and combine a semantic mask prior with a task-aware boundary response through residual logit correction. Zero initialization preserves the initial semantic prediction before the correction is learned.
(2) We attach the correction design to DeepLabV3-ResNet50 and SegFormer-B0 while retaining each semantic encoder and the same foreground definition, thereby examining applicability across CNN and Transformer representations.
(3) We evaluate six settings with three random seeds each in an 18-run controlled study. The analysis includes a consistency-weight sweep, paired confidence intervals, A100 latency and complexity measurements, HD95 and Chamfer diagnostics, ranked improvement and failure cases, and zero-shot testing on all 5,019 DUTS-TE images. This protocol exposes metric-dependent benefits, uncertainty, and cross-domain limitations..

2. Related Work

2.1 Traffic and General Salient Object Segmentation

Deep salient object detection (SOD) has developed several ways to recover complete foreground while

preserving local detail. BASNet [3] combines region and boundary supervision, EGNet [4] uses edge guidance, and U2-Net [5] captures multiscale context through nested U-shaped structures. Transformer-based VST [6] and its more efficient successor VST++ [7] extend global feature interaction, while dichotomous image segmentation (DIS) [8] and ICON [9] emphasize fine segmentation and foreground integrity. More recent approaches explore two-dimensional prompt learning in VSCode [10], gated progressive optimization in GPONet [11], and Mamba-based representations in Samba [12]. These developments show that contour quality is already an established concern in SOD; the distinction of the present study is not the use of boundary supervision alone, but its integration into an explicit residual correction of semantic mask logits.

Traffic-specific SOD additionally ties foreground selection to road-scene annotations. TFGNet by Jia et al. [13] develops traffic-oriented feature interaction and guidance fusion, while Qiu et al. [14] introduce TSOD10K and Tramba to model visual saliency together with driving relevance. These studies provide the task context and benchmark for the present work. EdgeRefine takes the annotated traffic-salient foreground as given and studies how to refine its contour using a compact module. It neither introduces a new saliency-ranking criterion nor assumes that binary annotations encode different levels of safety importance.

2.2 Boundary-Aware Semantic Segmentation

Semantic segmentation architectures provide the region representations on which boundary refinement operates. FCN [15] and U-Net [16] established dense prediction and encoder–decoder designs; residual learning [17], feature pyramids [18], pyramid pooling [19], and Dilated Residual Networks [20] developed complementary approaches to feature extraction and spatial context. DeepLabV3+ [21] further combines atrous processing with decoder-based recovery of spatial detail. Subsequent architectures extend these representations through ViT [22], Swin Transformer [23], and SegFormer [24], or through convolutional designs such as ConvNeXt [25] and SegNeXt [26]. Mask2Former [27] and OneFormer [28] address broader segmentation formulations, PIDNet [29] targets real-time semantic segmentation, and Mamba [30] introduces selective state-space modeling as a sequence-modeling foundation. These architectural advances are relevant to the choice of semantic representation, but encoder design and explicit contour correction remain distinct methodological questions.

Boundary-aware approaches can be distinguished by their supervision, correction location, and geometric assumptions. Gated-SCNN [31] integrates a dedicated shape stream with semantic features. SegFix [32] predicts boundary displacement directions for post-hoc label correction, whereas PointRend [33] adaptively re-predicts uncertain points. Boundary loss [34] introduces distance-based supervision, and Boundary IoU [35] provides an evaluation measure sensitive to contour quality. EdgeRefine instead supervises a boundary head using the binary foreground mask and uses its response to guide residual changes in the original mask logits. This avoids constructing displacement labels or adaptively selecting points, but it does not provide an explicit distance-minimization mechanism. Consequently, improved contour overlap need not imply reduced geometric displacement, a distinction examined in the experiments.

2.3 Generic Edge Detection

Generic edge detection targets perceptual discontinuities rather than the boundary of a selected salient foreground. HED [36] uses deeply supervised multiscale prediction, RCF [37] aggregates convolutional features, and BDCN by He et al. [38] learns edges through a bidirectional cascade. DexiNed [39] uses a densely connected architecture, PiDiNet [40] emphasizes efficient extraction through pixel differences, and EDTER [41] introduces Transformer-based edge modeling. Recent methods address complementary issues: DiffusionEdge [42] formulates crisp edge prediction through diffusion, RankED [43] uses ranking-based losses to address imbalance and annotation uncertainty, and MuGE [44] predicts edges at multiple granularities. Their localization mechanisms are informative, but perceptual edge labels are not interchangeable with traffic-salient boundary labels. A detector may correctly recover road markings, facade details, or reflections that lie outside the annotated foreground. EdgeRefine therefore derives its boundary supervision from the task mask rather than substituting a generic edge map for the semantic target.

2.4 Conceptual Positioning

Table 1 summarizes the main conceptual difference. The proposed contribution is not a new generic edge detector. It is a compact semantic-mask correction mechanism whose boundary labels and final output are both tied to traffic-salient foreground.

Table 1: *Conceptual comparison with representative edge and boundary-aware methods."Task aligned"*

means that the supervised boundary is derived from the same foreground definition as the final traffic-saliency mask.

Method family	Supervision source	Predicted target auxiliary	Correction/refinement domain	Inference role	Task aligned
HED/RCF[13, 14]	Human-labeled perceptual edges	Generic edge probability	None; standalone edge map	Direct edge output	No
BASNet/EGNet [3, 4]	Saliency masks and/or salient edges	SOD boundary features	Coupled SOD decoder or loss	Joint region prediction	Partly
Gated-SCNN [8]	Semantic labels and shape targets	Shape-stream features	Feature/logit fusion	Joint semantic prediction	Dataset dependent
SegFix [9]	Boundary displacement labels	Direction and distance	Post-hoc label displacement	Corrected semantic labels	Dataset dependent
PointRend [10]	Segmentation labels	Uncertain point labels	Adaptive point re-prediction	Refined region logits	Dataset dependent
Boundary loss[11]	Ground-truth distance transform	No auxiliary output required	Training objective only	No extra inference branch	Dataset dependent
EdgeRefine(Ours)	Binary traffic-salient masks	Mask-derived salient boundary	Zero-initialized residual mask logits	Refined mask salient	Yes

3. Method

3.1 Overall Framework

Given an RGB image $I \in R^{H \times W \times 3}$ and a binary traffic-salient mask $Y \in \{0,1\}^{H \times W}$, the model predicts refined mask logits M_r and boundary logits E. DeepLabV3-ResNet50 is the principal semantic trunk because Its atrous spatial pyramid pooling (ASPP) supplies a strong multi-scale region prior. A mask head predicts coarse logits M, a boundary head predicts E from the same high-level features, and EdgeRefine predicts a residual correction to M. Figure 2 presents the data flow, while Figure 3 decomposes ASPP, the boundary head, the residual module, and the objective. To test architectural portability rather than redefine the method, we also attach the same boundary and residual heads to the deepest SegFormer-B0 feature map.

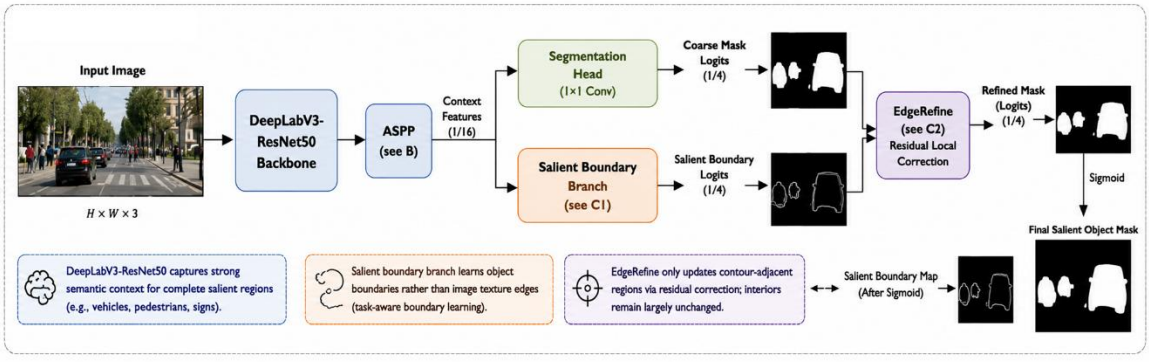


Figure 2: *Overall EdgeRefine-DeepLabV3 framework. DeepLabV3-ResNet50 supplies the coarse traffic-salient mask and shared semantic features. A mask-derived boundary head predicts the task-aware contour, and the zero-initialized EdgeRefine path applies residual correction to mask logits.*

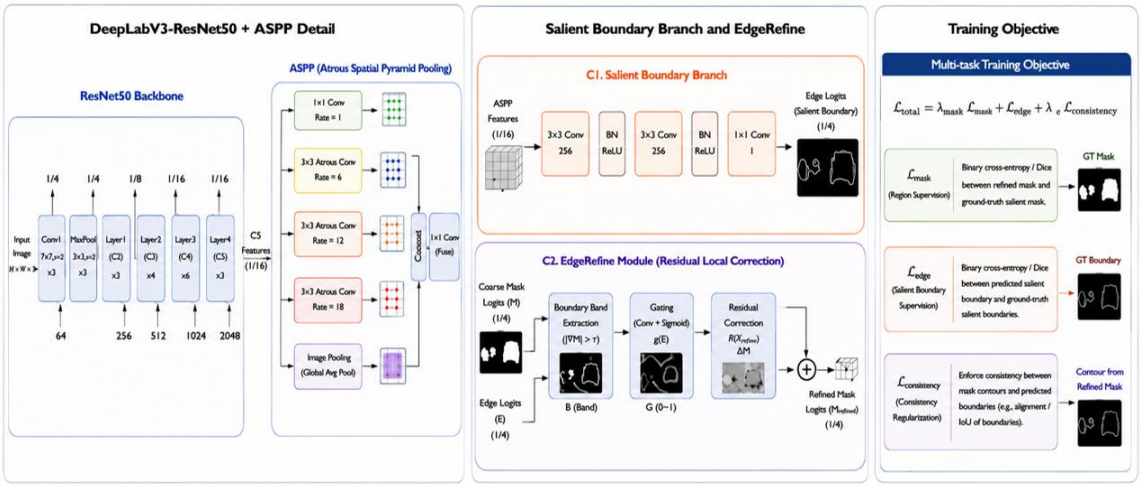


Figure 3: *Core module decomposition. The figure separates (i) the DeepLabV3-ResNet50 and ASPP semantic trunk, (ii) salient-boundary generation and prediction, (iii) edge-gated residual logit correction, and (iv) the joint objective. The consistency term is evaluated as an optional weight λc, not assumed to be universally beneficial.*

3.2 Task-Aware Salient Boundary Generation

We define a task-aware semantic boundary as the morphological boundary of the benchmark's binary foreground mask. With a 3 × 3 rectangular structuring element K3, the boundary target is:

$$Y_b = Dilate(Y; K_3) - Erode(Y; K_3) \quad (1)$$

The finite-width band is generated after nearest-neighbor mask resizing and contains only foreground-background transitions induced by the TSOS annotation. Thus, lane markings or shadows are negative examples unless they belong to an annotated salient object. This definition is task aware in its label source, but not instance aware or risk ranked: two adjacent foreground instances may share a merged binary region, and all annotated foreground objects receive equal supervision. These are explicit scope limitations of the current formulation.

The salient-boundary head maps high-level semantic features to boundary logits E. It uses a 1×1 channel projection followed by two 3×3 convolutional blocks and a 1 ×1 output layer. Unlike a generic edge detector, the head is optimized only against Yb and is used as guidance rather than as the final task output.

3.3 EdgeRefine Residual Correction

Let $g(E) = \sigma(E)$ be the predicted semantic-boundary response. The residual input combines the coarse mask logits, boundary logits, and their 8 gated interaction:

$$X_r = \text{Concat}(M, E, M \odot g(E)) \quad (2)$$

A compact residual head $R(\cdot)$, containing two 3×3 convolution–normalization–activation blocks and a 1×1 output layer, predicts a mask-logit correction:

$$M_r = M + R(X_r) \odot (1 + g(E)) \quad (3)$$

The multiplicative factor increases correction amplitude near predicted semantic boundaries while allowing a lower-amplitude correction elsewhere. It is therefore boundary enhanced rather than mathematically restricted to a narrow contour band.

The final 1×1 layer of R is initialized with zero weights and zero bias. At the beginning of fine-tuning, $R(X_r) = 0$ and Eq. (3) is an identity mapping, so the refined logits exactly match the pretrained semantic logits. Gradients can then learn conservative deviations from the baseline instead of perturbing the established mask prediction at initialization. We use this as an optimization safeguard; we do not claim that zero initialization is an independently verified regularizer.

3.4 Training Objective

The mask and boundary branches are supervised by binary cross-entropy and Dice losses. The optional consistency term compares a differentiable boundary extracted from $\sigma(M_r)$ with $\sigma(E)$. The complete

objective is:

$$\mathcal{L} = \mathcal{L}_m^{\text{bce}} + 1.5\mathcal{L}_m^{\text{dice}} + 0.25\mathcal{L}_b^{\text{bce}} + 0.5\mathcal{L}_b^{\text{dice}} + \lambda_c\mathcal{L}_{\text{cons}} \quad (4)$$

We evaluate $\lambda_c \in \{0,0.01,0.05,0.10\}$ with three seeds each. The results in Section 4.4 do not support consistency as a universally beneficial component, so it is treated as an optional training term. The balanced configuration used for detailed diagnosis is $\lambda_c = 0.01$; direct boundary logits remain guidance only, and the final semantic boundary is extracted from $\sigma(M_r)$.

4. Experiments

4.1 Datasets and Protocols

The principal benchmark is TSOD10K introduced by Qiu et al. [7]. We use 9,897 training images, 1,100 validation images, and the official 2,756 image TSOD10K-TE test set. All models are trained on the same split files and evaluated at 512×512 resolution. To test cross-domain behavior without target-domain adaptation, we additionally evaluate frozen TSOD10K check-points on all 5,019 DUTS-TE image–mask pairs. DUTS-TE is a general SOD set rather than a traffic-specific benchmark; its purpose here is to stress transfer and expose domain dependence, not to replace TSOD10K as the target task.

4.2 Implementation and Statistical Reporting

The DeepLabV3 controlled matrix starts from the same saved DeepLabV3-ResNet50 checkpoint. Every setting is fine-tuned for 8 epochs with batch size 24, AdamW, learning rate 10^{-4}, weight decay 10^{-4}, mixed precision, and gradient clipping at 1.0. Training augmentation consists of horizontal flipping with probability 0.5; brightness and contrast perturbation in [0.85,1.15] with probability 0.35 each; and color perturbation in [0.9,1.1] with probability 0.25. The matrix contains a semantic-only control, an edge-head-only model, and EdgeRefine with four consistency weights, each run with seeds 41, 42, and 43. The selected checkpoint maximizes the average of validation Dice and Boundary F1.

SegFormer-B0 is initialized from the public MiT-B0 weights and trained for 16 epochs with batch size 16 and learning rate 6×10^{-5}. EdgeRefine SegFormer-B0 is initialized from that trained semantic baseline and fine-tuned for another 16 epochs under the same optimizer settings. This experiment tests whether the module can operate on a modern Transformer representation; because the refined model receives an additional fine-tuning stage, it is reported as a portability study rather than a strict causal ablation.

All main experiments use PyTorch 2.1 with CUDA 11.8 on one NVIDIA A100-PCIE-40GB GPU. We report three-seed means and sample standard deviations for the controlled DeepLabV3 matrix. Paired 95% confidence intervals are computed from seed-wise differences with Student's t distribution ($n = 3$). Given the small sample size and multiple correlated metrics, these intervals are descriptive; we avoid binary significance claims.

4.3 Evaluation Metrics

Region quality is measured by MAE, F_β, Dice, and IoU. Boundary overlap is measured by Boundary F1 (B-F1) and Boundary IoU (B-IoU) with a two- pixel tolerance. The threshold-sweep edge protocol reports ODS, OIS, AP, edge MAE, and precision/recall at the ODS threshold. Chamfer distance and HD95 measure geometric displacement and are lower-is-better. S-measure, maximum E-measure, and weighted F-measure are added for the DUTS-TE SOD protocol. We report overlap and distance metrics together because a method can improve tolerance-based overlap while creating a small number of distant false boundary points.

4.4 Three-Seed Controlled Study

Table 2 and Figure 4(a–b) summarize the controlled TSOD10K-TE results across six settings, each evaluated using three random seeds. All settings were initialized from the same saved DeepLabV3 checkpoint and fine-tuned for eight epochs. Relative to the DeepLabV3 control, the selected EdgeRefine configuration ($\lambda_c = 0.01$) increases the reported mean ODS from 0.4875 to 0.4906 (+0.0031), OIS from 0.5366 to 0.5400 (+0.0034), and Boundary IoU from 0.5847 to 0.5881 (+0.0034). Edge MAE decreases from 0.004621 to 0.004496 (−0.000125), indicating improved edge-prediction accuracy. These improvements are accompanied by a small decrease in Dice from 0.8490 to 0.8478 (−0.0012) and an increase in Chamfer distance from 14.971 to 17.126 (+2.155 pixels). Thus, refinement improves selected threshold-based boundary metrics

without uniformly improving region overlap or geometric distance.

Table 2: *Three-seed controlled study on TSOD10K-TE (mean ± sample standard deviation). All settings start from the same DeepLabV3 checkpoint and receive the same eight-epoch fine-tuning budget. Higher values indicate better performance except for edge MAE and Chamfer distance.*

Setting	ODS↑	OIS↑	Edge MAE↓	Dice↑	B-F1↑	B-IoU↑	Chamfer↓
DeepLabV3 control	0.4875 ± 0.0016	0.5366 ± 0.0009	0.004621 ± 0.000030	**0.8490 ± 0.0012**	0.8089 ± 0.0040	0.5847 ± 0.0020	**14.971 ± 1.137**
Edge head only	0.4909 ± 0.0031	0.5366 ± 0.0028	0.004552 ± 0.000038	0.8443 ± 0.0028	0.8090 ± 0.0015	0.5854 ± 0.0017	18.198 ± 2.204
EdgeRefine, $\lambda_c = 0$	0.4893 ± 0.0011	0.5385 ± 0.0006	0.004466 ± 0.000071	0.8458 ± 0.0009	**0.8126 ± 0.0032**	0.5876 ± 0.0017	17.225 ± 1.629
EdgeRefine, $\lambda_c = 0.01$	0.4906 ± 0.0026	**0.5400 ± 0.0015**	0.004496 ± 0.000008	0.8478 ± 0.0015	0.8095 ± 0.0033	0.5881 ± 0.0004	17.126 ± 0.649
EdgeRefine, $\lambda_c = 0.05$	**0.4909 ± 0.0020**	0.5393 ± 0.0026	**0.004452 ± 0.000026**	0.8458 ± 0.0027	0.8099 ± 0.0042	**0.5883 ± 0.0037**	18.120 ± 2.116
EdgeRefine, $\lambda_c = 0.10$	0.4906 ± 0.0006	0.5398 ± 0.0001	0.004459 ± 0.000023	0.8461 ± 0.0013	0.8111 ± 0.0016	0.5882 ± 0.0005	17.388 ± 0.483

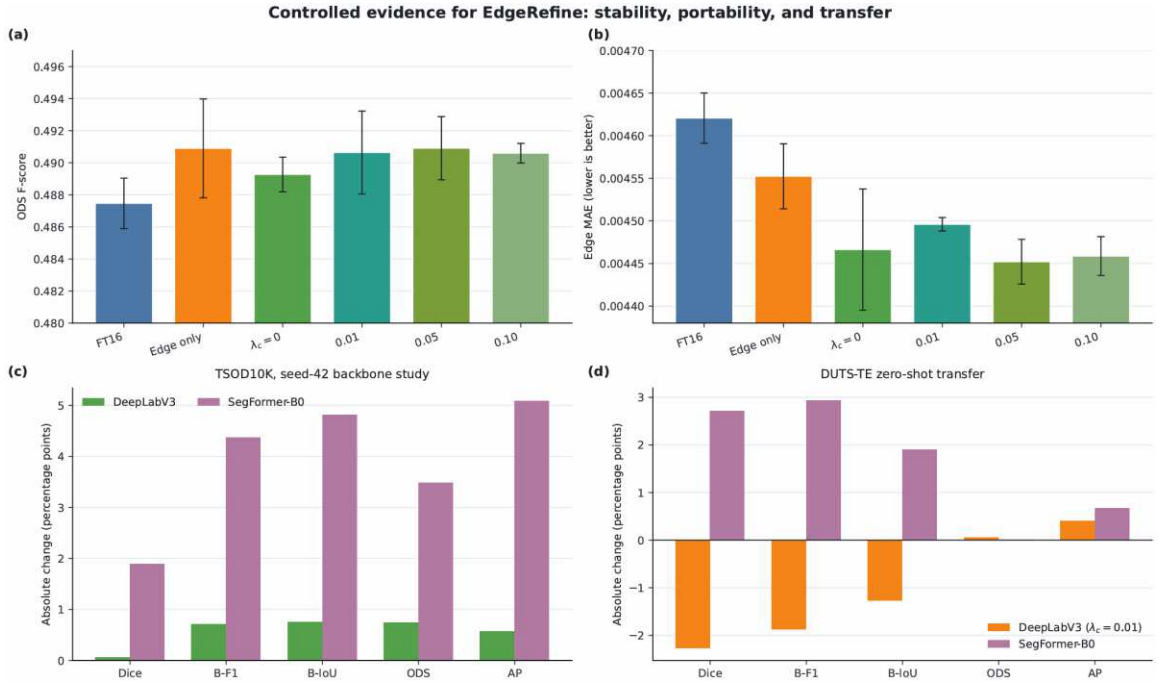


Figure 4. Controlled experimental evidence for EdgeRefine. (a) Three-seed ODS means and standard deviations for the six DeepLabV3 controlled settings listed in Table 2. (b) Three-seed edge MAE for the same settings (lower is better). (c) Changes in Dice, Boundary F1, Boundary IoU, ODS, and AP for the seed-42 DeepLabV3 and SegFormer-B0 comparisons in Table 3; positive values indicate improvements. (d) Corresponding changes in the zero-shot DUTS-TE evaluation in Table 4, highlighting backbone-dependent transfer rather than universal gains. Changes in (c) and (d) are expressed in percentage points.

The reported seed-paired 95% confidence intervals are [−0.00699, 0.01334] for ODS, [−0.00213, 0.00890] for OIS, [−0.00086, 0.00751] for Boundary IoU, [−0.00779, 0.00536] for Dice, and [−1.609, 5.919] for Chamfer distance. All of these intervals include zero. The reported interval for edge MAE is [−0.0001929, −0.0000563], while the interval for region precision is [0.00014, 0.01792]; both exclude zero. The intervals are based on seed-wise paired differences and are reported as descriptive uncertainty estimates because only three seeds are available. Small differences between interval midpoints and differences of displayed means can arise from rounding.

The consistency-weight sweep exhibits no monotonic trend, as shown in Table 2 and Figure 4(a–b). The $\lambda_c = 0$ setting has the highest mean Boundary F1 (0.8126), $\lambda_c = 0.05$ has the highest mean Boundary IoU (0.5883) and lowest mean edge MAE (0.004452), whereas $\lambda_c = 0.01$ attains the highest mean OIS

(0.5400) and better Dice preservation than the other EdgeRefine settings. We therefore retain $\lambda_c = 0.01$ as a balanced configuration for subsequent diagnosis, not as a universally optimal setting. The consistency objective remains optional rather than a core contribution.

4.5 Backbone Portability and Computational Cost

As shown in Table 3 and Figure 4(c), the seed-42 DeepLabV3 comparison shows that EdgeRefine raises Dice from 0.8482 to 0.8488, Boundary F1 from 0.8073 to 0.8128, Boundary IoU from 0.5848 to 0.5885, and ODS from 0.4857 to 0.4931. The corresponding Chamfer distance, however, increases from 16.2607 to 16.5679 pixels, reiterating the trade-off between boundary overlap and geometric displacement. EdgeRefine adds 0.568 million parameters and approximately 5.007 G MACs to DeepLabV3; synchronized batch-1 inference latency increases from 8.229 to 9.346 ms (+1.117 ms, based on the table's rounded figures).
Table 3 and Figure 4(c) also report the SegFormer-B0 portability study. Attaching the refinement module raises Dice from 0.8047 to 0.8237 (+0.0190), Boundary F1 from 0.7305 to 0.7742 (+0.0437), Boundary IoU from 0.4726 to 0.5208 (+0.0482), and ODS from 0.3813 to 0.4161 (+0.0348); Chamfer decreases from 14.6947 to 14.2159 (−0.4788 pixels). This gain requires 0.101 million additional parameters, approximately 2.742 G extra MACs, and 0.949ms additional measured latency. As explained in Section 4.2 and the table caption, the refined SegFormer model undergoes an additional training stage and hence is not a strict equal-budget causal ablation. These results demonstrate implementation portability and encouraging performance, while leaving the precise causal contribution of the refinement module subject to a matched-training comparison.

Table 3. Seed-42 backbone portability and computational cost on TSOD10K-TE. The SegFormer-B0 pair involves an additional refinement stage and is not a strict equal-epoch ablation. Latency is synchronized A100 batch-1 inference at 512×512.

Method	Dice ↑	B-F1 ↑	B-IoU ↑	Chamfer ↓	ODS ↑	AP ↑	Params (M)↓	MACs (G) ↓	Latency (MS) ↓
DeepLabV3 control	0.8482	0.8073	0.5848	**16.2607**	0.4857	0.4192	39.634	164.118	**8.229**
EdgeRefine-DeepLabV3	**0.8488**	**0.8128**	**0.5885**	16.5679	**0.4931**	**0.4250**	40.202	169.125	9.346
SegFormer-B0	0.8047	0.7305	0.4726	14.6947	0.3813	0.3015	**3.714**	**6.764**	9.234
EdgeRefine-SegFormer-B0	**0.8237**	**0.7742**	**0.5208**	**14.2159**	**0.4161**	**0.3524**	3.815	9.506	10.183

4.6 Zero-Shot Cross-Dataset Evaluation

Table 4 and Figure 4(d) report zero-shot transfer to all 5,019 DUTS-TE images using frozen TSOD10K-trained checkpoints, without target-domain fine-tuning. Relative to the DeepLabV3 semantic control, every evaluated DeepLabV3 boundary variant in this table reduces Dice, Boundary F1, and Boundary IoU. In the selected $\lambda_c = 0.01$ configuration, Dice decreases from 0.4767 to 0.4541 (−0.0226), Boundary IoU decreases from 0.1730 to 0.1602 (−0.0128), and Chamfer increases from 69.435 to 130.701 (+61.266 pixels). ODS improves slightly from 0.1519 to 0.1524 (+0.0005). These results caution against interpreting the source-domain boundary improvements as uniformly transferable to general salient-object segmentation.
In contrast, the SegFormer comparison in Table 4 and Figure 4(d) shows better zero-shot region and overlap scores following refinement: Dice increases from 0.5923 to 0.6194 (+0.0271), Boundary F1 from 0.3651 to 0.3945 (+0.0294), Boundary IoU from 0.2015 to 0.2205 (+0.0190), S-measure from 0.7081 to 0.7207 (+0.0126), and weighted F-measure from 0.6176 to 0.6366 (+0.0190); MAE decreases from 0.0958 to 0.0906 (−0.0052). Nevertheless, Chamfer increases from 47.706 to 52.919 (+5.213 pixels). Thus, cross-domain transfer depends on both the backbone and the evaluation metric. Table 4 covers the configurations explicitly listed; no zero-shot result for $\lambda_c = 0.10$ is reported here, so no conclusion is drawn about that setting.

Table 4. Zero-shot evaluation on DUTS-TE using frozen TSOD10K checkpoints, without target-domain training. Higher values indicate better performance except MAE and Chamfer distance.

Method	Dice ↑	MAE ↓	B-F1 ↑	B-IoU ↑	Chamfer ↓	ODS ↑	S-measure ↑	Weighted-F ↑
DeepLabV3	0.4767	0.1124	0.3366	0.1730	69.435	0.1519	0.6396	0.4960

control								
Edge head only	0.4321	0.1165	0.2944	0.1513	157.587	0.1478	0.6221	0.4449
EdgeRefine, $\lambda_c = 0$	0.3541	0.1180	0.2713	0.1410	216.913	0.1451	0.5870	0.3999
EdgeRefine, $\lambda_c = 0.01$	0.4541	0.1138	0.3179	0.1602	130.701	0.1524	0.6335	0.4564
EdgeRefine, $\lambda_c = 0.05$	0.4193	0.1129	0.3019	0.1534	163.358	0.1482	0.6190	0.4412
SegFormer-B0	0.5923	0.0958	0.3651	0.2015	**47.706**	0.2053	0.7081	0.6176
EdgeRefine-SegFormer-B0	**0.6194**	**0.0906**	**0.3945**	**0.2205**	52.919	**0.2054**	**0.7207**	**0.6366**

4.7 Geometric Diagnosis of the Chamfer Trade-Off

To examine why tolerance-based boundary overlap can improve while Chamfer distance increases, we evaluated the EdgeRefine-DeepLabV3 model trained with random seed 42 and $\lambda_c = 0.01$ on individual TSOD10K-TE images. The analysis considers directed boundary distances, HD95, the proportion of predicted boundary pixels outside the matching tolerance, and image groups defined by foreground-area fraction, the number of foreground connected components, and an image-gradient texture proxy. These groupings describe observable mask and image characteristics; they do not provide direct object-scale, occlusion, or scene-complexity annotations.

Figure 5 shows that the changes are condition dependent. In images with a foreground-area fraction of at least 10%, Boundary IoU increases by 0.887 percentage points, while Chamfer distance and HD95 decrease by 0.394 and 2.10 pixels, respectively. For images with a foreground-area fraction below 2%, Boundary IoU increases by 0.639 percentage points and the distant false-boundary ratio decreases from 0.1941 to 0.1818; nevertheless, Chamfer distance and HD95 increase by 0.957 and 1.63 pixels, respectively. Single-component images gain 0.555 percentage points in Boundary IoU and show a 1.12-pixel reduction in HD95. In contrast, images with two or three connected components lose 0.350 percentage points in Boundary IoU, with increases of 1.53 pixels in Chamfer distance and 4.71 pixels in HD95. Chamfer distance decreases by 0.561 pixels in the high-texture group but increases by 1.96 pixels in the low-texture group.

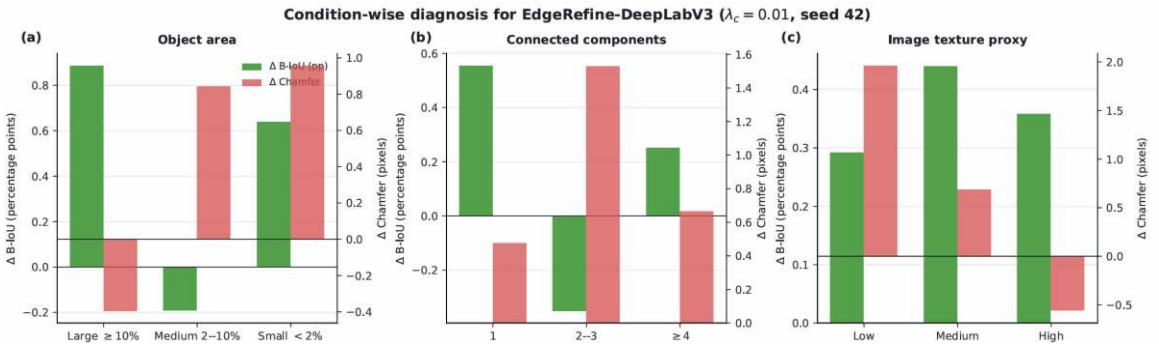


Figure 5: Condition-wise diagnosis for EdgeRefine-DeepLabV3 (λ_c = 0.01, seed 42). Green bars show the change in Boundary IoU (percentage points); red bars show the change in Chamfer distance (pixels). Positive green and negative red are preferred. Object area is measured as foreground fraction, connected components are computed from the binary mask, and texture is a quantile-based image-gradient proxy.

These observations indicate that improved boundary overlap within a fixed tolerance does not necessarily imply lower geometric distance. They are consistent with the presence of displaced or additional contour fragments in some images, a pattern further illustrated by the failure cases in Figure 8. The grouping variables are descriptive proxies, so the results should not be interpreted as establishing the causal effects of occlusion or object size.

4.8 Qualitative Error Analysis

Figure 6 illustrates representative TSOD10K-TE predictions and boundary error maps, highlighting both improvements in local contour continuity and remaining errors around small targets, reflections, and crowded

regions. Unlike the ranked comparisons in Figures 7 and 8, Figure 6 presents the input image, ground-truth boundary, EdgeRefine boundary prediction, and the associated error map.

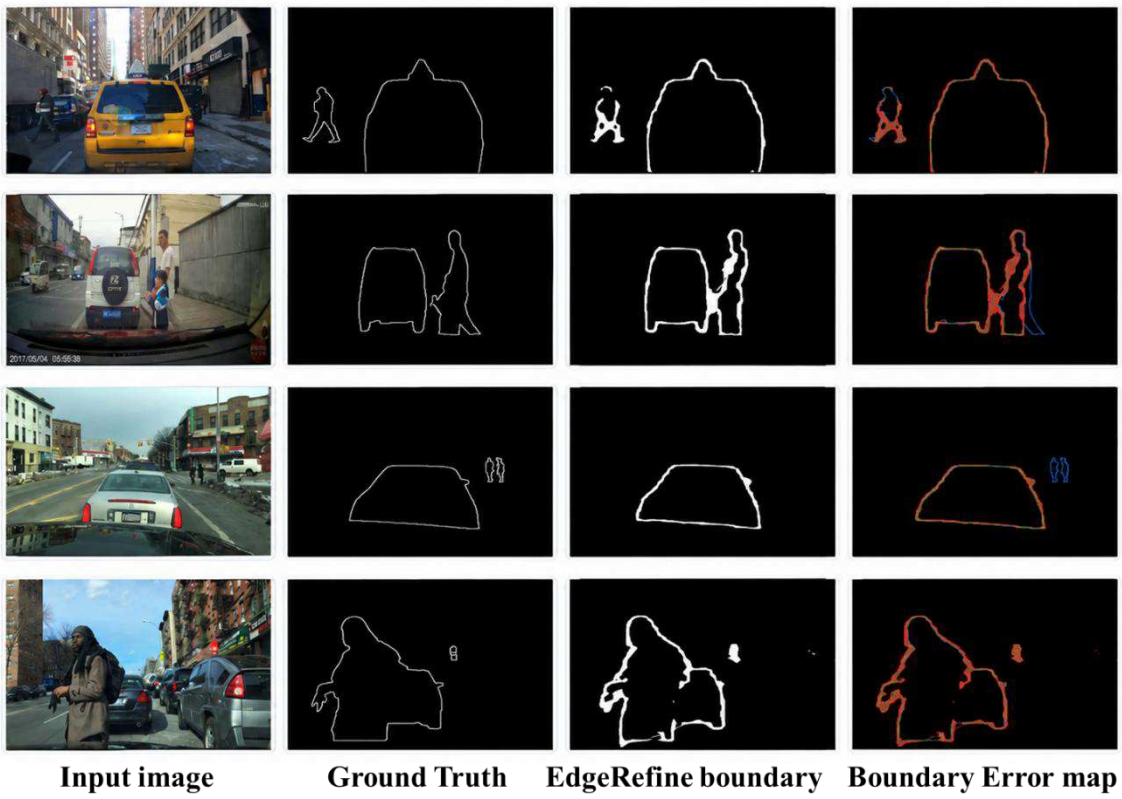


Figure 6: Representative error maps. Green denotes matched boundary, red extra predicted boundary, and blue missed ground-truth boundary. EdgeRefine often improves contour continuity, while small objects, reflections, and crowded scenes can still introduce missed or distant false segments.

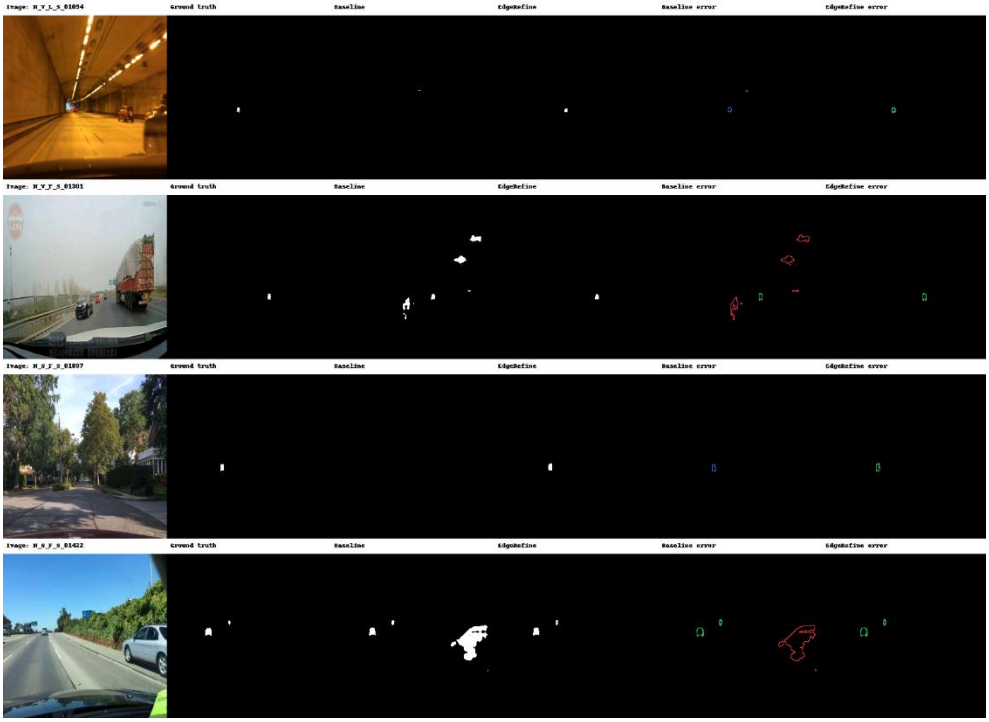


Figure 7: Four largest per-image Boundary-IoU improvements for EdgeRefine-DeepLabV3 (λ_c = 0.01, seed 42), ranked over the complete TSOD10K-TE set. The panels show image, ground truth, baseline prediction, EdgeRefine prediction, baseline error, and EdgeRefine error. These cases illustrate improved local contour coverage and suppression of some near-boundary errors.

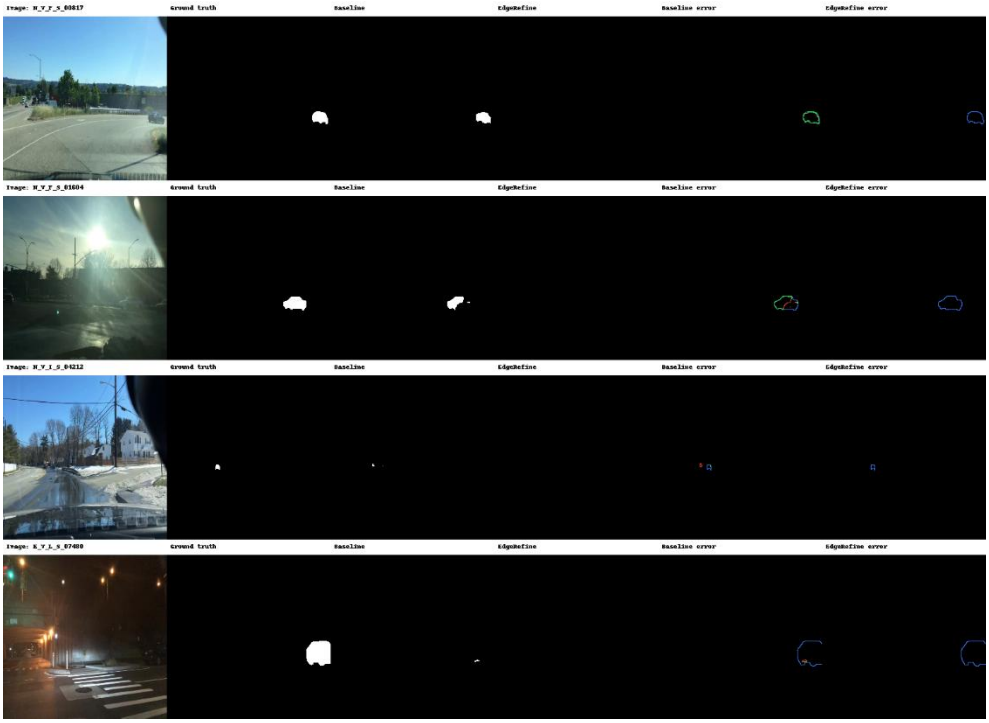


Figure 8: Four largest per-image Chamfer degradations under the same protocol. Despite locally plausible masks, sparse distant fragments or displaced contours increase geometric distance. These failure cases explain why overlap gains do not imply universal geometric improvement.

To complement these representative examples and reduce favorable-case selection bias, Figures 7 and 8 show cases ranked automatically using per-image changes over the full 2,756-image TSOD10K-TE test set. Figure 7 contains the four largest improvements in Boundary IoU, whereas Figure 8 contains the four largest degradations in Chamfer distance. Each ranked row shows the input image, ground truth, baseline prediction, EdgeRefine prediction, and the corresponding baseline and EdgeRefine error maps. Taken together, these visualizations show that local overlap improvements may coexist with isolated contour errors and should be interpreted alongside the quantitative results.

5. Discussion

5.1 What the Controlled Gains Support

The expanded controlled experiments support a more qualified conclusion than the initial single-run comparison. As summarized in Table 2 and Figure 4(a–b), the selected DeepLabV3 configuration ($\lambda_c = 0.01$) reduces mean edge MAE and provides modest mean improvements in ODS, OIS, and Boundary IoU relative to the semantic-only control, while mean Dice decreases and Chamfer distance increases. The paired confidence intervals for several overlap measures include zero; therefore, the observed gains should not be described as uniformly reliable across random seeds. The reported precision gain is consistent with a more conservative boundary prediction, although this interpretation remains metric specific.

Table 3 and Figure 4(c) also indicate that the correction module can be attached to SegFormer-B0 and is associated with improvements in region and boundary-overlap metrics. However, the refined SegFormer model undergoes an additional fine-tuning stage, so this experiment supports architectural applicability rather than a strict equal-training-budget causal comparison. Overall, EdgeRefine is best characterized as a lightweight task-aligned boundary correction mechanism with measurable but metric-dependent benefits, rather than as a method that consistently improves every evaluation criterion.

5.2 Why Boundary Overlap and Chamfer Disagree

Boundary F1 and Boundary IoU are evaluated with a matching tolerance and can reward improved correspondence along substantial portions of a predicted contour. Symmetric Chamfer distance instead averages nearest-neighbor boundary distances and can increase when additional predicted fragments lie far from the reference boundary. Consequently, a prediction may achieve better tolerance-based overlap on the dominant contour while performing worse on distance-based geometry.

The foreground-area and connected-component analyses in Figure 5, together with the ranked Chamfer degradation cases in Figure 8, are consistent with this explanation. In particular, the improvements in overlap metrics do not rule out sparse, spatially displaced false contour fragments. These observations motivate future evaluation of explicit distance-aware objectives or fragment suppression in combination with task-aware residual correction. They do not, by themselves, establish that any particular error source is the unique cause of the Chamfer degradation.

5.3 Scope and Generalization Limitations

The current task-aware boundary is binary and mask derived. It does not preserve instance separation when adjacent objects share foreground, does not distinguish pedestrians from vehicles, and does not encode risk priority. The DUTS-TE results in Table 4 show that transfer from traffic-scene masks to general salient-object masks is backbone dependent; in particular, the DeepLabV3 boundary-refinement variants do not consistently retain their source-domain gains. This pattern suggests sensitivity to the source–target domain shift, but does not on its own establish the mechanism of that sensitivity. The stronger SegFormer transfer indicates that representation and training protocol may matter, although the additional fine-tuning stage prevents a strict attribution to backbone alone. Evaluation on more independently annotated traffic datasets and adverse-weather subsets remains necessary before broad generalization can be claimed.

5.4 Deployment Considerations

The A100 measurements show an approximately 1 ms batch-1 latency increase for both backbones at 512 × 512. This overhead is small in the present profiling environment but does not establish real-time readiness in an autonomous-driving stack. Camera decoding, preprocessing, memory transfer, multi-sensor scheduling, and embedded accelerators are not represented. Model compression, lower-resolution boundary heads, and hardware-specific profiling are required before deployment claims can be made.

6. Conclusion

This study presented EdgeRefine-DeepLabV3 for task-aware boundary refinement of traffic salient-object masks. The method defines its boundary target directly from binary traffic-salient annotations, predicts a semantic boundary response, and applies zero-initialized residual correction in mask logit space. An 18-run, three-seed TSOD10K study shows modest average improvements in edge MAE, ODS, OIS, and Boundary IoU with overlapping uncertainty intervals for most overlap metrics and a persistent Dice–Chamfertrade-off. A SegFormer-B0 study demonstrates that the correction design is portable and can yield larger gains, while DUTS-TE zero-shot evaluation exposes backbone-dependent transfer and geometric limitations. The principal conclusion is therefore not that one configuration dominates all metrics, but that explicit task-aligned boundary correction is useful when its statistical uncertainty, geometric failure modes, and domain dependence are evaluated together.

Data Availability Statement

This study used the publicly available TSOD10K and DUTS-TE image datasets. The TSOD10K dataset is available through the official project repository at https://github.com/mj129/Tramba. The DUTS-TE dataset, which is the test subset of DUTS, is available from the official dataset website at https://saliencydetection.net/duts/. Access to and reuse of these datasets are subject to the terms specified by their respective providers. No new image dataset was collected for this study. Additional experimental records supporting the findings are available from the corresponding author upon reasonable request.

Author Contributions

Run Yuan: Conceptualization, methodology, writing—original draft.
Haonan Long: Supervision, validation, writing—review and editing.
Wei DENG: Writing—review and editing, methodology.
Xueli Wang: Data curation, experimental design, and analysis.
All authors have read and agreed to the published version of the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work was supported by the Hunan Provincial Natural Science Foundation under Grant No. 2025JJ80307.

Consent to Publish declaration

Not applicable

Consent to Participate declaration

Not applicable

Ethics declaration

Not applicable

Clinical trial number

Not applicable. This study does not involve any clinical trial.

Acknowledgments

The authors would like to thank the funding agencies for their support and all contributors involved in this research.

References

[1] Itti L, Koch C. A saliency-based search mechanism for overt and covert shifts of visual attention[J]. Vision research, 2000, 40(10-12): 1489-1506.
[2] R. Achanta, S. Hemami, F. Estrada and S. Susstrunk. "Frequency-tuned salient region detection," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 1597-1604, doi: 10.1109/CVPR.2009.5206596.
[3] X. Qin, Z. Zhang, C. Huang, et al. BASNet: Boundary-Aware Salient Object Detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7479-7489.
[4] J.-X. Zhao, J.-J. Liu, D.-P. Fan, et al. EGNet: Edge guidance network for salient object detection[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 8779-8788.
[5] X. Qin, Z. Zhang, C. Huang, et al. U2-Net: Going deeper with nested U-structure for salient object detection., 2020, 106[J]. DOI: https://doi. org/10.1016/j. patcog, 2020: 107404.
[6] N. Liu, N. Zhang, et al. Visual Saliency Transformer[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, 2021: 4722-4732.
[7] N. Liu, Z. Luo, N. Zhang, et al. VST++: Efficient and Stronger Visual Saliency Transformer[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(11): 7300-7316.
[8] X. Qin, H. Dai, X. Hu, et al. Highly Accurate Dichotomous Image Segmentation[C]//Computer Vision – ECCV 2022. Cham: Springer, 2022: 38-56.
[9] M. Zhuge, D.-P. Fan, et al. Salient Object Detection via Integrity Learning[J]. IEEE Transactions on

Pattern Analysis and Machine Intelligence, 2023, 45(3): 3738-3752.
[10] Z. Luo, N. Liu, W. Zhao, X. Yang, et al. VSCode: General Visual Salient and Camouflaged Object Detection with 2D Prompt Learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, 2024: 17169-17180.
[11] Y. Yi, N. Zhang, W. Zhou, et al. GPONet: A two-stream gated progressive optimization network for salient object detection[J]. Pattern Recognition, 2024, 150: 110330.
[12] J. He, K. Fu, X. Liu, et al. Samba: A Unified Mamba-based Framework for General Salient Object Detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, 2025: 25314-25324.
[13] N. Jia, Y. Sun, X. Liu. TFGNet: Traffic Salient Object Detection Using a Feature Deep Interaction and Guidance Fusion[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(3): 3020-3030.
[14] Y. Qiu, Y. Sun, J. Mei, L. Xiao, J. Xu, et al. Salient object detection in traffic scene through the TSOD10K dataset[J]. IEEE Transactions on Image Processing, 2025, 34: 7473-7488.
[15] J. Long, E. Shelhamer. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA: IEEE, 2015: 3431-3440.
[16] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation[C]//Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham: Springer, 2015: 234-241.
[17] K. He, X. Zhang, S. Ren, J. Sun, et al. Deep Residual Learning for Image Recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016: 770-778.
[18] T.-Y. Lin, P. Dollar, R. Girshick, et al. Feature Pyramid Networks for Object Detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE, 2017: 2117-2125.
[19] H. Zhao, J. Shi, X. Qi, X. Wang, et al. Pyramid Scene Parsing Network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE, 2017: 6230-6239.
[20] F. Yu, V.KOLTUN, T. FUNKHOUSER. Dilated Residual Networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE, 2017: 472-480.
[21] L.-C. Chen, Y. Zhu, et al. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation[C]//Computer Vision – ECCV 2018. Cham: Springer, 2018: 801-818.
[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[C]//International Conference on Learning Representations (ICLR). 2021.
[23] Z. Liu, Y. Lin, Y. Cao, H. Hu, et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, 2021: 10012-10022.
[24] E. Xie, W. Wang, Z. Yu, et al. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers[C]//Advances in Neural Information Processing Systems (NeurIPS). 2021, 34: 12077-12090.
[25] Z. Liu, H. Mao, C.-Y. Wu, et al. A ConvNet for the 2020s[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, 2022: 11976-11986.
[26] M.-H. Guo, C.-Z. Lu, Q. Hou, et al. SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation[C]//Advances in Neural Information Processing Systems (NeurIPS). 2022, 35: 1140-1156.
[27] B. Cheng, I. Misra, A. G. Schwing, et al. Masked-Attention Mask Transformer for Universal Image Segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, 2022: 1290-1299.
[28] J. Jain, J. Li, M.-T. Chiu, et al. OneFormer: One Transformer To Rule Universal Image Segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, 2023: 2989-2998.
[29] J. Xu, Z. Xiong, S. P. Bhattacharyya. PIDNet: A Real-Time Semantic Segmentation Network Inspired by PID Controllers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, 2023: 19529-19539.
[30] A. Gu, T. Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces[C]//First Conference on Language Modeling (COLM). 2024.
[31] T. Takikawa, D. Acuna, V. Jampani, S. Fidler, et al. Gated-SCNN: Gated Shape CNNs for Semantic

Segmentation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South): IEEE, 2019: 5229-5238.
[32] Y. Yuan, J. Xie, X. Chen, J. Wang, et al. SegFix: Model-Agnostic Boundary Refinement for Segmentation[C]//Computer Vision – ECCV 2020. Cham: Springer, 2020: 489-506.
[33] A. Kirillov, Y. Wu, K. He, et al. PointRend: Image Segmentation As Rendering[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, 2020: 9799-9808.
[34] H. Kervadec, J. Bouchtiba, et al. Boundary loss for highly unbalanced segmentation[J]. Medical Image Analysis, 2021, 67: 101851.
[35] B. Cheng, R. Girshick, P. Dollar, et al. Boundary IoU: Improving Object-Centric Image Segmentation Evaluation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, 2021: 15334-15342.
[36] S. Xie, Z. Tu. Holistically-Nested Edge Detection[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). Santiago, Chile: IEEE, 2015: 1395-1403.
[37] Y. Liu, M.-M. Cheng, X, et al. Richer Convolutional Features for Edge Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(8): 1939-1946.
[38] J. He, S. Zhang, M. Yang, et al. Bi-Directional Cascade Network for Perceptual Edge Detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA: IEEE, 2019: 3828-3837.
[39] X. Soria, E. Riba, et al. Dense extreme inception network for edge detection[J]. Pattern Recognition, 2023, 139: 109461.
[40] Z. Su, W. Liu, Z. Yu, et al. Pixel Difference Networks for Efficient Edge Detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, 2021: 5117-5127.
[41] M. Pu, Y. Huang, Y. Liu, et al. EDTER: Edge Detection with Transformer[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, 2022: 1402-1412.
[42] Y. Ye, K. Xu, Y. Huang, et al. DiffusionEdge: Diffusion Probabilistic Model for Crisp Edge Detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(7): 6675-6683.
[43] B. Cetinkaya, S. Kalkan. RankED: Addressing Imbalance and Uncertainty in Edge Detection Using Ranking-based Losses[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, 2024: 3239-3249.
[44] C. Zhou, Y. Huang, M. Pu, et al. MuGE: Multiple Granularity Edge Detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, 2024: 25952-25962.