Coordinated Training- and Inference-Time Boundary Optimization for Urban Scene Semantic Segmentation: An Empirical Analysis

Tong Bu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1092216489@qq.com

Yifei Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1310774954@qq.com

Hanxi Zhong
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: bi8bobi8bo0811@qq.com

Abstract—**Boundary displacement remains common around road edges, thin objects, and distant road users in urban scene semantic segmentation. We examine how boundary supervision during training combines with boundary correction at inference, using DeepLabV3 Plus with a ResNet-101 backbone. Training uses active boundary loss and weight averaging across three random seeds. Inference combines horizontal-flip augmentation with SegFix at native image resolution. Matched experiments compare cross-entropy and boundary-aware training under four inference protocols, measuring regional accuracy, boundary quality, per-class performance, and computational cost. On the Cityscapes validation set, active boundary loss gives positive gains across the three DeepLabV3 Plus training seeds and the four matched inference protocols. Horizontal flipping and SegFix provide further gains, and the complete configuration reaches 72.39 percent mean intersection over union. Experiments with SegFormer-B0 also show positive boundary F-score gains from active boundary loss in all three seeds. Its complete configuration increases mean intersection over union from 47.17 to 49.10 percent, indicating that the benefit is not limited to DeepLabV3 Plus. In the fully matched DeepLabV3 Plus experiments, the boundary F-score gain from active boundary loss remains positive but is smaller after SegFix. This is consistent with complementary effects and some overlap in the errors corrected by the two methods. The results provide an empirical basis for choosing training and inference settings when boundary accuracy and computational cost are both important.**

Keywords-Urban Scene Semantic Segmentation; Deeplabv3+; Active Boundary Loss; Test-Time Augmentation; Boundary Evaluation

I. INTRODUCTION

Urban scene semantic segmentation assigns pixel-level labels to roads, buildings, vehicles, pedestrians, and traffic infrastructure. It supports autonomous-driving perception, road-asset inspection, and urban visual analysis. Cityscapes provides high-resolution street images with fine pixel annotations and has supported developments in encoder-decoder architectures, multiscale context modeling, and boundary-aware methods [1]. However, mean intersection over union (mIoU) primarily measures regional overlap and is not always sensitive to local displacement along narrow structures, small objects, or the edges of large objects. For poles, traffic lights, traffic signs, pedestrians, and bicycles, small boundary errors can cause conspicuous shape loss or merging between classes.

Boundary errors in street scenes have several structural causes. Repeated backbone downsampling compresses the spatial support of poles, curbs, and distant objects; restoring the original output size cannot fully recover lost high-frequency contours. Class frequencies and object sizes are also highly imbalanced. Large road,

building, and sky regions dominate pixelwise losses, whereas traffic lights, riders, and bicycles contribute relatively few boundary pixels. In addition, single-view predictions are affected by local texture, occlusion, and asymmetric context. Even when coarse labels identify the correct regions, their contours may remain displaced. Boundary quality therefore depends jointly on the training objective, inference resolution, view fusion, and postprocessing.

DeepLabV3+ combines atrous spatial pyramid pooling with a lightweight decoder to recover spatial information while retaining a large receptive field [2]. Conventional cross-entropy (CE), however, remains a pixelwise classification objective: because boundary pixels constitute a small fraction of an image, optimization can favor large interiors. Active boundary loss (ABL) extracts boundaries from current predictions and learns directions toward ground-truth boundaries, introducing dynamic alignment constraints during training [3]. In contrast, SegFix replaces unreliable boundary labels with more reliable interior labels using independently predicted boundaries and directional offsets; it is model-agnostic inference postprocessing [4]. Their combined effects and the overlap between the regions they correct require examination under matched conditions.

Published boundary-optimization results often use different architectures, augmentations, and evaluation protocols. Differences in input resolution, random seeds, and boundary tolerances complicate comparisons. Additional training branches, model ensembles, and multiscale inference also incur computational costs. We therefore use paired controls to analyze the separate and combined effects of training-time boundary supervision, view fusion, and boundary postprocessing, together with their accuracy-cost tradeoffs.

Using matched models, data, and inference protocols, we examine how ABL gains change after horizontal-flip test-time augmentation (HFlip) and SegFix, which classes benefit, and the associated inference costs. CE and ABL are trained in pairs with three random seeds. Checkpoint weights are averaged separately for the two training settings. The final ablation uses all 500 Cityscapes validation images at the native resolution of 1024 by 2048 pixels.

The study has three parts. We first compare training-time boundary supervision with inference-time boundary correction under matched data, model, and evaluation settings. We then evaluate CE and ABL under four inference protocols, using seed-wise and per-class results, Boundary IoU, boundary F-score (Boundary F), and runtime measurements to examine component effects and interactions. Finally, three-seed SegFormer-B0 experiments test whether ABL and the complete inference configuration also improve a lightweight Transformer model.

II. Related Work

A. Urban Scene Semantic Segmentation

Fully convolutional networks extend classification networks to end-to-end dense prediction [5], while residual networks provide stable deep visual backbones [6]. PSPNet aggregates scene context at multiple scales through pyramid pooling [7]. DeepLabV3 enlarges the effective receptive field through parallel atrous convolutions in its atrous spatial pyramid pooling module [8]. DeepLabV3+ adds a decoder to recover low-level spatial detail and uses depthwise separable convolution in both encoder and decoder [2]. These methods balance contextual modeling with spatial recovery and are widely used for road-scene segmentation.

Subsequent work explores feature hierarchies, object-context relations, and efficiency. UPerNet uses feature pyramids to unify visual concepts at multiple levels [9]; HRNet maintains high-resolution representations and repeatedly fuses scales [10]; and OCRNet enriches pixel context with object-region representations [11]. For real-time applications, BiSeNet balances detail and speed through spatial and context paths [12], while PIDNet separates semantic, detail, and boundary branches using an analogy to proportional-integral-derivative control [13]. Among Transformers, SegFormer combines a hierarchical encoder with a lightweight multilayer-perceptron decoder [14]. Because backbone size and input resolution jointly affect accuracy and cost, we use DeepLabV3+ ResNet-101 for the main controlled

study and SegFormer-B0 for cross-architecture validation.

B. Boundary-Aware Training

Boundary supervision in semantic segmentation broadly includes auxiliary boundary branches, joint region-boundary objectives, and dynamic alignment. Gated-SCNN explicitly propagates contour information through a separate shape stream and gating [15]. Iterative pyramid context jointly optimizes semantic segmentation and semantic boundary detection [16]. Decoupled body-edge supervision separately models low-frequency interiors and high-frequency edges [17]. SBCB conditions multilevel backbone features on semantic boundary detection during training, with the auxiliary head removable at inference [18]. Such approaches offer expressive representations, but additional branches and multitask gradients can increase training sensitivity.

Loss-based approaches modify the optimization objective directly. Focal loss reweights difficult examples to address foreground-background imbalance [19]; Lovász-Softmax provides an optimizable surrogate for IoU [20]; boundary loss emphasizes contour geometry through distance transforms [21]; and approximate Hausdorff-distance losses target severe boundary deviations [22]. InverseForm learns inverse transformations between predicted and target boundaries [23]. Conditional boundary loss defines local, conditional objectives that attract same-class features and repel different-class features at boundary pixels [24]. These losses operate at different optimization scales, so direct combination may produce imbalanced gradient magnitudes or conflicting directions.

ABL derives directional supervision from current predicted boundaries. It converts the direction toward the nearest ground-truth boundary into a dynamic classification target, gradually moving predicted contours toward annotated ones [3]. Unlike fixed boundary weighting, it applies directional constraints near currently misaligned contours. We pair CE and ABL under identical source checkpoints, optimizers, seeds, and data orders to examine the effects of this supervision on regional accuracy and contour quality.

C. Test-Time Augmentation and Weight Averaging

Horizontal-flip test-time augmentation combines predictions from an image and its mirror view after restoring the flipped output to the original coordinates. Our implementation averages the class logits before label assignment. This requires two forward passes and can reduce variation between the two views. We compare its effects on CE and ABL checkpoints at a fixed input resolution.

Weight averaging combines multiple training solutions without increasing the number of models used at inference. Stochastic weight averaging averages parameters along an optimization trajectory to find wider optima [25]. Model soups extend this approach to fine-tuned models with a shared pretrained initialization [26]. We average checkpoints from three seeds with the same architecture, initial checkpoint, and training settings, forming separate CE and ABL soups. A three-model logit ensemble serves as the output-space comparison, allowing us to assess the accuracy and cost of parameter averaging versus multiple forward passes.

D. Inference-Time Boundary Correction and Evaluation

Postprocessing can refine predictions without retraining the main model. Fully connected conditional random fields use color and spatial similarity to model dense pixel relations, smoothing regions and aligning labels with image edges [27]. PointRend treats segmentation as pointwise rendering and adaptively refines uncertain locations [28]. SegFix predicts boundary locations and directions toward object interiors, then copies labels from more reliable interior pixels through a displacement field [4]. Since it does not depend on intermediate segmentation features, identical postprocessing can be applied to saved CE and ABL label maps.

mIoU describes regional overlap. When most pixels inside a large object are correct, shifting its contour by several pixels can have only a limited effect on regional IoU. Boundary IoU computes

overlap within ground-truth and predicted boundary neighborhoods and is more sensitive to contour errors [29]. Boundary F combines precision and recall after matching predicted and true boundaries within a tolerance. We report all three metrics, with fixed boundary widths and matching tolerances throughout the matched experiments.

E. Protocol Differences from Representative Published Results

Table I summarizes representative Cityscapes results for DeepLabV3+, Gated-SCNN, ABL, and SegFix, including backbones, data splits, and inference settings. Since training schedules and test protocols differ across studies, the table provides methodological context rather than a direct accuracy ranking.

Previous work reports gains from both ABL and SegFix on DeepLabV3 and identifies SegFix as a further postprocessing option for ABL predictions [3]. We examine whether the ABL gain remains after HFlip and SegFix, using checkpoints averaged across three seeds. The comparisons cover regional accuracy, boundary quality, per-class changes, and computational cost.

TABLE I. TABLE I. REPRESENTATIVE CITYSCAPES BOUNDARY-OPTIMIZATION METHODS AND PROTOCOLS

Study	Boundary intervention	Backbone/network	Split and inference	mIoU(%)
DeepLabV3+[2]	Decoder refinement	Xception-71	Test; no postprocessing	82.1
Gated-SCNN[15]	Training-time shape branch	WideResNet-38+ASPP	Validation; multiscale	80.8
DeepLabV3 baseline [3]	No dedicated boundary term	DeepLabV3	Validation; single scale	79.5
DeepLabV3+SegFix[3, 4]	Inference-time label displacement	DeepLabV3	Validation; single scale	80.5
DeepLabV3+IABL[3]	Training-time dynamic alignment	DeepLabV3	Validation; single scale	80.5
This study	Training-inference coordination	DeepLabV3+ R101	Validation; HFlip+SegFix	72.3881

III. Coordinated Boundary Optimization and Evaluation

A. Overall Pipeline

Fig. 1 shows the pipeline. For each of three fixed random seeds, models are fine-tuned for ten epochs from the same DeepLabV3+ ResNet-101 checkpoint using either CE or CE+ABL. We select the checkpoint with the highest validation mIoU from each run, then average the three selected checkpoints separately for CE and ABL. At inference, single-scale prediction or horizontal-flip logit averaging is performed at native resolution, with optional SegFix correction of the saved label maps. All configurations are evaluated on the same validation set.

Within each seed, CE and ABL start from the same checkpoint and differ only in the ABL weight. They are evaluated with the same input resolution, flip setting, and SegFix offsets. Checkpoint and offset hashes, data-split identifiers, and result archives are retained so that each reported difference can be traced to its paired runs.

B. Cross-Entropy and Active Boundary Loss

Let p_i denote the predicted probabilities for K classes at pixel i, and let y_i be its ground-truth label. With N valid pixels, CE is defined as

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N} \log \mathrm{p_i}\left(\mathrm{y_i}\right) \qquad (1)$$

ABL detects boundary points in the current semantic prediction and finds the direction toward the nearest true boundary for each point. The discretized direction becomes a prediction target derived from neighboring class distributions, encouraging the model to move predicted boundaries toward the ground truth during backpropagation. Retaining the core directional-alignment mechanism, we use the total loss

$$L = L_{\mathrm{CE}} + \lambda_{\mathrm{ABL}} L_{\mathrm{ABL}}, \quad \lambda_{\mathrm{ABL}} = 1.0. \qquad (2)$$

Let B_p and B_g denote the predicted and

ground-truth boundary sets. For $b \in B_p$, let b^* denote a nearest ground-truth boundary point. The desired direction can be described schematically as $d(b) = Q(b^* - b)$, where $Q(\cdot)$ discretizes direction. In the implementation, this target is obtained by comparing the ground-truth boundary distance map at the eight neighboring pixels and the center. KL divergences between neighboring class distributions provide directional scores, and directional CE uses $d(b)$ as the target. The loss is distance-weighted, with neighboring logits detached during gradient computation. The target changes with the current prediction and focuses supervision on boundaries that remain misaligned.

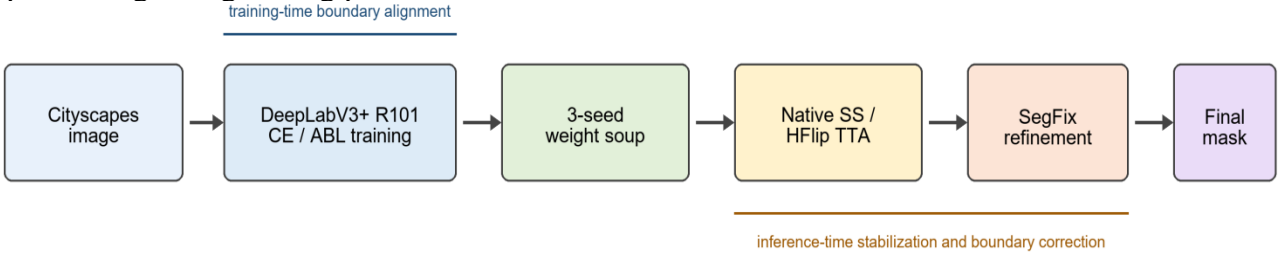

Figure 1. Coordinated training-time boundary alignment and inference-time correction.

ABL is optimized jointly with regional CE. CE governs semantic classes and interior consistency, whereas ABL supplies local contour-displacement constraints. The weight $\lambda_{\mathrm{ABL}} = 1.0$ was selected in preliminary controls and remains fixed across all three seeds. Both the best performance and the final-three-epoch average characterize training behavior.

The CE arm uses $\lambda_{\mathrm{ABL}} = 0$, and the ABL arm uses $\lambda_{\mathrm{ABL}} = 1$; all other settings are identical. Auxiliary semantic-boundary heads, attention modules, Lovász terms, and gradient-conflict projection are disabled in the final confirmation experiments, isolating ABL as the training difference.

C. Three-Seed Checkpoint Weight Averaging

Let θ_1, θ_2, and θ_3 be the parameters learned with three seeds under one training setting. Uniform weight averaging gives

$$\theta_{\mathrm{soup}} = \frac{\theta_1 + \theta_2 + \theta_3}{3}. \tag{3}$$

CE and ABL checkpoints are averaged separately. At half resolution, the ABL soup improves mIoU, Boundary IoU, and Boundary F over the CE soup, with positive per-class differences in all 19 classes. Averaging avoids selecting a single seed for deployment, although the resulting checkpoint performs slightly below the best individual model.

Weight averaging produces one parameter set and therefore retains single-model inference cost. The three-model logit ensemble instead requires three forward passes and averages their class scores before argmax. We compare the two approaches in terms of regional accuracy, boundary quality, and computational cost.

D. Native-Resolution Horizontal-Flip Augmentation

For an input image x, let $P(x)$ and $P(T(x))$ denote the class-score vectors, before softmax, for the original and horizontally flipped views. The flipped scores are mapped back to the original coordinates and averaged:

$$P_{\mathrm{HFlip}}(x) = \frac{P(x) + T^{-1}(P(T(x)))}{2}. \tag{4}$$

All outputs are restored to the native Cityscapes resolution of 1024 by 2048 pixels. HFlip does not update model parameters. Its effect is measured separately on CE and ABL checkpoints, making it an inference factor independent of the training loss.

Logits are averaged before argmax; the method does not vote between two discrete label maps. This retains scores for alternative classes and aligns the flipped output with the original image through T^{-1}. The two views need not agree in

scenes with asymmetric context, so we report both the classes that improve and those that decline, as well as the overall means.

E. SegFix Boundary Displacement Correction

SegFix uses pregenerated Cityscapes validation boundary and directional offsets to map a boundary position q to a more reliable interior position $q+\Delta q$, from which a label is sampled. We use the public Cityscapes semantic-offset files and apply identical correction to the CE/ABL and SS/HFlip predictions. Before execution, the offset archive hash, file count, validation-image names, and uniqueness are checked. Direction order, scale factor 2, bilinear sampling, border padding, and coordinate alignment follow the public implementation.

Let $S(q)$ be a coarse label, $M(q)$ the predicted boundary mask, and Δq the directional offset. Correction is summarized by $S'(q)=S(q)$ when $M(q)=0$, and $S'(q)=S(q+\Delta q)$ when $M(q)=1$. Because correction is concentrated near predicted contours, it can substantially improve Boundary IoU and Boundary F while leaving large interior semantic errors unresolved.

F. Inference Modes and Effect Decomposition

The basic mode is native-resolution single-scale inference with the ABL soup; the high-accuracy mode adds HFlip and SegFix. Performance changes are decomposed into training loss, input resolution, HFlip, and SegFix. ABL-minus-CE differences under the same inference protocol quantify training-time boundary supervision, while protocol differences on the same checkpoint quantify inference-stage effects.

$$I_M=[M(H+S)-M(H)]-[M(U+S)-M(U)] \tag{5}$$

For metric M, the interaction in (5) uses H, S, and U to denote HFlip, SegFix, and single-scale inference, respectively. A positive I_M means that SegFix gives a larger gain after HFlip; a negative value means a smaller gain. We calculate this difference-in-differences separately for the CE and ABL checkpoints.

G. Evaluation Design

We assess training stability using best-run values, means over the final three epochs, and sample standard deviations across three paired seeds. Eight matched conditions isolate the training and inference effects. Boundary IoU, Boundary F, per-class differences, and local examples describe where predictions change. Runtime and GPU memory characterize cost; file hashes and metric recomputation from saved predictions support reproducibility.

After the paired training runs, checkpoints are fixed for SS, HFlip, SegFix, and combined evaluation. Weight averaging is compared with logit ensembling. The final ablation keeps the model architecture, input resolution, data split, and evaluation parameters fixed while varying the training loss and inference protocol.

IV. Experiments and Results

A. Dataset, Implementation, and Metrics

We use the fine Cityscapes annotations: 2,975 training images and 500 validation images. All reported final results use the validation set; test labels are not used. DeepLabV3+ confirmation training is validated at 512 by 1024 pixels, whereas the final eight-condition ablation uses 1024 by 2048 pixels. The backbone is ResNet-101 with output stride 16. Each paired run lasts ten epochs, with initial learning rate 5×10^{-5}, polynomial decay, batch size 4, and seeds 1, 2, and 3. Each CE/ABL pair starts from the same checkpoint and differs only in the active boundary weight. The cross-architecture experiment uses ImageNet-pretrained SegFormer-B0, 513 by 513 random training crops, ten epochs, the same data split, and the prespecified final checkpoint for each of the three seeds. Evaluation is at native resolution with a 3-pixel boundary width and matching tolerance.

DeepLabV3+ is fine-tuned from an existing converged checkpoint for ten epochs using 512 by 1024 inputs. The experiments measure the incremental effect of adding ABL, with the data split, augmentation, learning-rate schedule, batch size, and training duration held constant across seeds. Best validation mIoU measures peak

performance, while the final-three-epoch mean describes late-training behavior. The best-mIoU checkpoint from each run is used for subsequent weight averaging and inference.

TABLE II. MAIN EXPERIMENTAL PROTOCOL

Item	Setting
Data split	Cityscapes: 2,975 train / 500 validation; 19 classes
Baseline	DeepLabV3+, ResNet-101, output stride 16
Confirmation training	10 epochs; initial learning rate 0.00005; polynomial decay; batch 4; seeds 1/2/3
Training factor	ABL loss weight: 0 for CE, 1 for ABL; otherwise matched
Final inference	1024 by 2048; SS or HFlip; optional SegFix
Boundary metrics	boundary width=3 px，match tolerance=3 px
Reported metrics	mIoU, Boundary IoU, Boundary F, and per-class IoU

mIoU is the arithmetic mean of the IoUs of 19 classes. Boundary IoU measures intersection and union within predicted and true boundary bands; Boundary F is the harmonic mean of boundary precision and recall with a 3-pixel tolerance. Because this width is fixed in pixels, absolute boundary scores are not directly compared across resolutions. All eight main ablations use the same native resolution, ensuring a common basis for comparison.

$$\mathrm{IoU}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k}, \tag{6}$$

$$\mathrm{mIoU} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{IoU}_k. \tag{7}$$

$$\mathrm{Boundary\ F} = \frac{2P_bR_b}{P_b + R_b}. \tag{8}$$

True positives TP , false positives FP , and false negatives FN exclude ignored labels, and $K = 19$. Boundary precision P_b and recall R_b use a 3-pixel tolerance. Three pixels represent different relative scales at 512 by 1024 and 1024 by 2048, so cross-resolution comparisons primarily describe mIoU changes. Boundary scores in the main eight-condition table share the same native-resolution scale.

Before inference, we verify all 2,975 training and 500 validation image-label pairs and the native image dimensions of 1024 by 2048 pixels. SHA-256 hashes identify the CE soup, ABL soup, and SegFix archive. Each validation image has one uniquely matched offset file. For each of the eight conditions, metrics computed during evaluation agree exactly with those recomputed from the 500 saved predictions.

B. Three-Seed Stability of Active Boundary Loss

Table III shows best-mIoU gains of 0.2333, 0.2133, and 0.2320 percentage points (pp) across the three seeds, with a mean of 0.2262 pp. The mean gains over the final three epochs are 0.2329 pp in mIoU and 0.5745 pp in Boundary F. Late-training IoU and Boundary F differences are positive for all 19 classes in every seed, supporting the use of these checkpoints in the subsequent averaging and inference experiments.

The sample standard deviations of the best-mIoU, late-training mIoU, and late-training Boundary F gains are 0.0112, 0.0008, and 0.0101 pp, respectively. The closely matched late-training differences indicate consistent relative effects across the three seeds.

The regional gain from ABL is below one percentage point, while its Boundary F gain is approximately 2.5 times the mIoU gain. This pattern is consistent with a mechanism focused on local contours. Because regional and boundary metrics respond differently, subsequent experiments report mIoU, Boundary IoU, and Boundary F together.

TABLE III. PAIRED THREE-SEED CE/ABL CONFIRMATION RESULTS (PERCENT AND PERCENTAGE-POINT GAINS)

Seed	CE best mIoU	ABL best mIoU	Best mIoU gain	Final-three-epoch mIoU gain	Final-three-epoch Boundary F gain
1	64.7490	64.9824	+0.2333	+0.2333	+0.5746
2	64.5737	64.7870	+0.2133	+0.2320	+0.5846
3	64.6854	64.9175	+0.2320	+0.2334	+0.5644
Mean ± sample SD	-	-	+0.2262±0.0112	+0.2329±0.0008	+0.5745±0.0101

C. Weight Averaging and Output Ensembling

The ABL soup improves over the CE soup by 0.2268 pp in mIoU, 0.4165 pp in Boundary IoU, and 0.5509 pp in Boundary F, with positive differences for all 19 classes on each metric. Logit ensembling adds only 0.0040 pp mIoU over the ABL soup, slightly reduces boundary scores, and requires three models. We therefore use a single weight-averaged checkpoint in the subsequent experiments. Its mIoU of 64.9612 percent is slightly below the best individual ABL checkpoint at 64.9824 percent.

The ABL soup exceeds the mean mIoU of its three source checkpoints by 0.0656 pp but remains 0.0212 pp below the best checkpoint. It avoids choosing a single seed for deployment without exceeding the best individual result. The three-model logit ensemble outperforms the ABL soup in only 10 of 19 classes for IoU and 8 of 19 for Boundary F, with slightly lower aggregate boundary scores. In this comparison, the small accuracy gain does not justify the threefold model-storage and forward-pass requirements.

D. Eight-Condition Native-Resolution Ablation

At native resolution, CE and ABL achieve 68.9459 and 69.0954 percent mIoU with single-scale inference. HFlip increases these to 70.6176 and 70.7509 percent, while SegFix applied to single-scale predictions yields 70.5472 and 70.6782 percent. Combining both inference enhancements gives 72.2792 percent for CE and 72.3881 percent for ABL. The latter also achieves the highest Boundary IoU, 25.9289 percent, and Boundary F, 62.3645 percent. The final result therefore improves all three metrics rather than trading regional accuracy for boundary quality.

Moving from half to native resolution increases CE and ABL mIoU by 4.2116 and 4.1342 pp, respectively. Model parameters are unchanged, so these gains are attributable to the change in input and evaluation resolution. All eight main experiments use 1024 by 2048 pixels; the 512 by 1024 results describe the training and checkpoint-fusion comparisons.

HFlip and SegFix have the same relative ordering for CE and ABL. HFlip alone gives slightly higher mIoU than SegFix alone, whereas SegFix gives higher Boundary IoU. Their combination gives the best values on all three metrics. Thus, the overall pattern of inference gains is similar under both training losses.

TABLE IV. CHECKPOINT FUSION AT HALF RESOLUTION (512 BY 1024; PERCENT)

Method	mIoU	Boundary IoU	Boundary F	mIoU gain over matched CE
CE soup	64.7343	19.5827	60.0305	-
ABL soup	64.9612	19.9993	60.5815	+0.2268
CE three-model logit ensemble	64.7384	19.5819	60.0201	-
ABL three-model logit ensemble	64.9652	19.9970	60.5724	+0.2267

TABLE V. EIGHT MATCHED NATIVE-RESOLUTION ABLATIONS (PERCENT)

Training/inference variant	mIoU	Boundary IoU	Boundary F
CE-SS	68.9459	17.4657	53.9541
ABL-SS	69.0954	17.8295	54.4984
CE-HFlip	70.6176	19.2453	57.4181
ABL-HFlip	70.7509	19.6158	57.9353
CE-SS+SegFix	70.5472	23.1386	58.9165
ABL-SS+SegFix	70.6782	23.5097	59.2708
CE-HFlip+SegFix	72.2792	25.5697	62.0517
ABL-HFlip+SegFix	72.3881	25.9289	62.3645

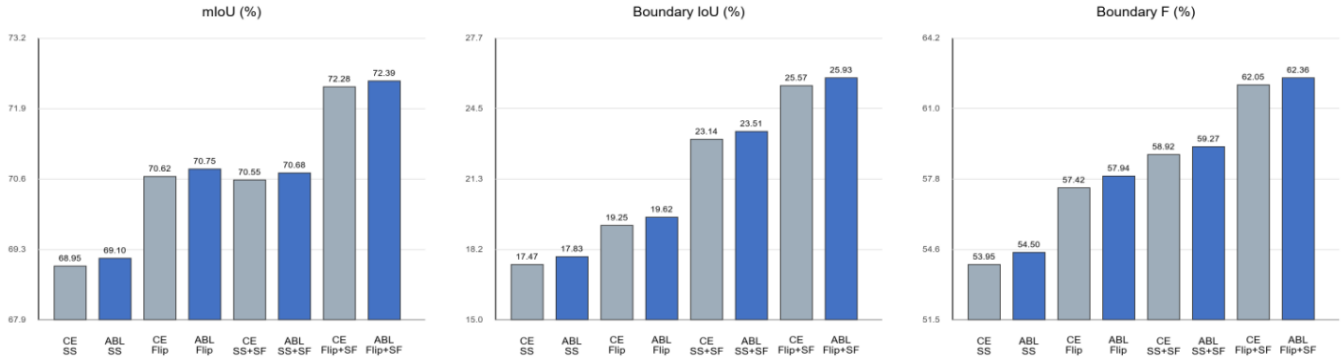

Figure 2. Regional and boundary metrics for CE and ABL under four inference protocols.

E. Component Contributions and Interactions

HFlip improves ABL mIoU by 1.6555 pp and benefits 17 of 19 classes, but bus and truck IoU decline by 0.5509 and 0.4872 pp, respectively. Its effect is therefore class-dependent. SegFix improves IoU, Boundary IoU, and Boundary F in all 19 classes for both ABL single-scale and HFlip predictions, with notable gains for pole, traffic sign, traffic light, person, and bicycle. The HFlip-SegFix mIoU interaction for ABL is approximately 0.0545 pp, a small positive value.

For ABL, HFlip adds 1.7863 pp Boundary IoU and 3.4369 pp Boundary F through two-view score averaging. SegFix produces larger boundary gains: when applied after ABL-HFlip, it adds 6.3131 pp Boundary IoU and 4.4292 pp Boundary F, alongside 1.6373 pp mIoU. Both operations can change labels near contours. The small positive interaction shows that their combined mIoU gain is slightly larger than the sum of their separate gains; it does not establish that they correct distinct errors.

The corresponding mIoU interaction for CE is approximately 0.0604 pp, close to the 0.0545 pp for ABL. SegFix therefore gains slightly more after HFlip than after single-scale inference under either training loss. This is consistent with a small benefit from combining view averaging with displacement correction.

F. Direct ABL Effects under Matched Inference Protocols

ABL improves regional and boundary metrics under all four protocols, with positive per-class IoU differences in all 19 classes. Its Boundary F gain nevertheless decreases from 0.5443 pp under single-scale inference to 0.3127 pp under the complete protocol. ABL and SegFix can therefore be used together, but the marginal ABL gain is smaller once inference-time correction is applied. This pattern is consistent with partial overlap in the boundary errors they correct, rather than a fully additive effect.

TABLE VI. MATCHED COMPONENT GAINS (PERCENTAGE POINTS)

Comparison	mIoU	Boundary IoU	Boundary F	Classes with positive IoU gain
ABL HFlip - ABL SS	+1.6555	+1.7863	+3.4369	17/19
ABL SS+SegFix - ABL SS	+1.5828	+5.6802	+4.7725	19/19
ABL HFlip+SegFix - ABL HFlip	+1.6373	+6.3131	+4.4292	19/19
ABL HFlip+SegFix - ABL SS	+3.2927	+8.0994	+7.8661	-

TABLE VII. ABL-MINUS-CE EFFECTS UNDER FOUR MATCHED PROTOCOLS (PERCENTAGE POINTS)

Inference protocol	mIoU	Boundary IoU	Boundary F	Classes with positive IoU gain
SS	+0.1495	+0.3638	+0.5443	19/19
HFlip	+0.1333	+0.3705	+0.5172	19/19
SS+SegFix	+0.1310	+0.3711	+0.3543	19/19
HFlip+SegFix	+0.1089	+0.3592	+0.3127	19/19

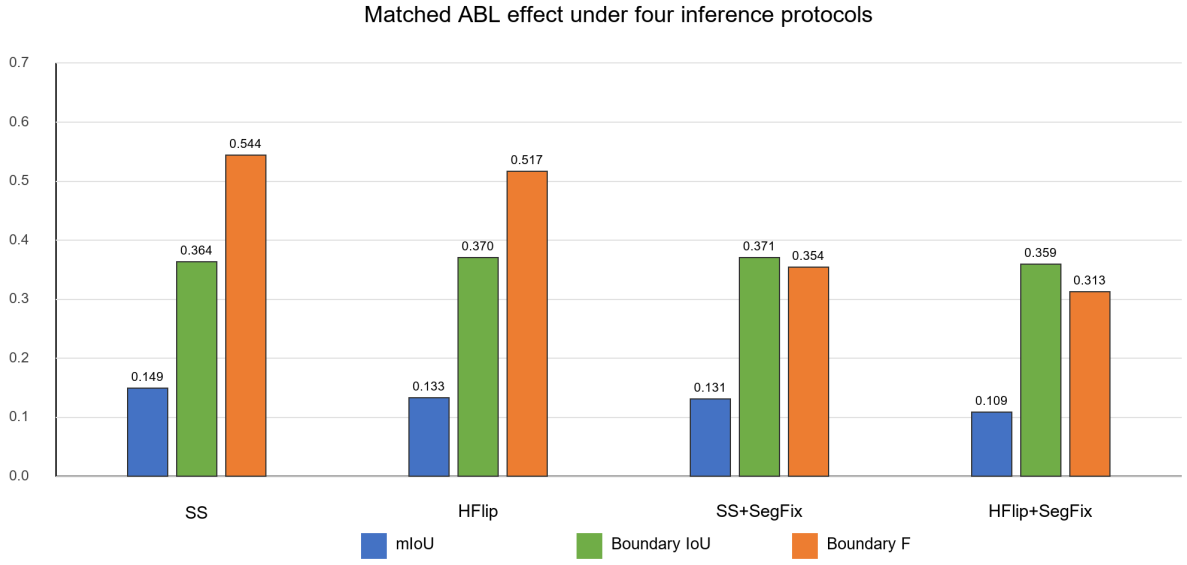

Figure 3. Matched ABL gains over CE under different inference protocols.

Without SegFix, the ABL-minus-CE Boundary F gain changes from 0.5443 pp with SS to 0.5172 pp with HFlip. Adding SegFix reduces these gains to 0.3543 and 0.3127 pp, respectively. Boundary IoU gains remain near 0.36 to 0.37 pp. The reduced ABL effect is therefore clearer in Boundary F than in Boundary IoU. Both metrics still show a positive ABL gain after postprocessing, but neither alone establishes which corrected pixels overlap.

G. Per-Class and Qualitative Analysis

Under the complete protocol, ABL improves IoU in all 19 classes. Pole has the largest gain, 0.6628 pp, while sky has the smallest, 0.0192 pp; sidewalk, person, and bicycle also improve. Boundary F gains are positive for every class. Fig. 5 shows three qualitative examples in which HFlip improves the continuity of small objects and SegFix further corrects road edges, pedestrian contours, and local displacements near traffic infrastructure. Large interior classification errors can remain because the correction focuses on boundary neighborhoods.

Fine boundary differences are difficult to see in full-scene images at page scale. For Fig. 6, we use a fixed-size sliding window to select the region with the largest net increase in correctly labeled pixels in each of two examples with relatively large gains. The input, ground truth, and three predictions are cropped at identical coordinates. In these selected regions, HFlip reduces breaks in thin objects and isolated noise, while SegFix

moves road, vehicle, or pedestrian contours closer to the ground truth.

Fig. 7 compares pixel correctness relative to ground truth between ABL-SS and ABL-HFlip-SegFix. Green denotes incorrect-to-correct transitions, red correct-to-incorrect transitions, light gray persistently correct pixels, and dark gray persistently incorrect pixels. Examples with larger and smaller net gains show both corrections near boundaries and new errors in occluded or class-confused regions. All states are computed directly from predicted and true labels.

Fig. 8 shows examples with little or no improvement. In munster_000173, the complete pipeline produces a net loss of 3,762 correctly labeled pixels over the whole image. In lindau_000018, the complete ABL and CE configurations differ by only five correctly labeled pixels. Thus, positive aggregate and per-class gains do not imply improvement in every image. Residual errors include displaced contours, broken thin objects, merged adjacent classes, incorrect region labels, and errors involving occlusion or rare vehicles. Contour displacement and broken thin objects are more directly related to boundary correction; the other errors also depend on the underlying semantic predictions.

TABLE VIII. PER-CLASS ABL-MINUS-CE RESULTS UNDER THE COMPLETE PROTOCOL (PERCENT AND PERCENTAGE POINTS)

Class	CE IoU	ABL IoU	IoU gain	Boundary F gain
road	97.32	97.35	+0.036	+0.375
sidewalk	81.21	81.37	+0.159	+0.552
building	91.20	91.23	+0.034	+0.166
wall	51.36	51.39	+0.027	+0.302
fence	55.55	55.59	+0.040	+0.176
pole	51.84	52.51	+0.663	+0.747
traffic light	62.83	62.89	+0.059	+0.118
traffic sign	73.92	74.04	+0.126	+0.171
vegetation	92.15	92.18	+0.032	+0.143
terrain	63.93	64.00	+0.078	+0.331
sky	94.47	94.49	+0.019	+0.088
person	78.92	79.08	+0.153	+0.373
rider	56.34	56.42	+0.081	+0.178
car	93.00	93.10	+0.101	+0.447
truck	66.89	66.96	+0.073	+0.330
bus	76.79	76.90	+0.118	+0.290
train	56.81	56.88	+0.070	+0.417
motorcycle	56.90	56.96	+0.051	+0.327
bicycle	71.88	72.03	+0.147	+0.413

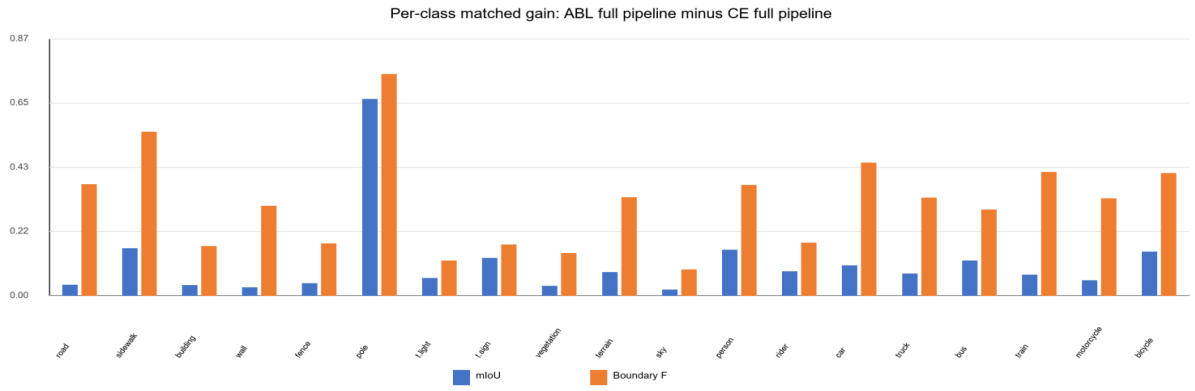

Figure 4. Per-class ABL gains over CE under the complete inference protocol.

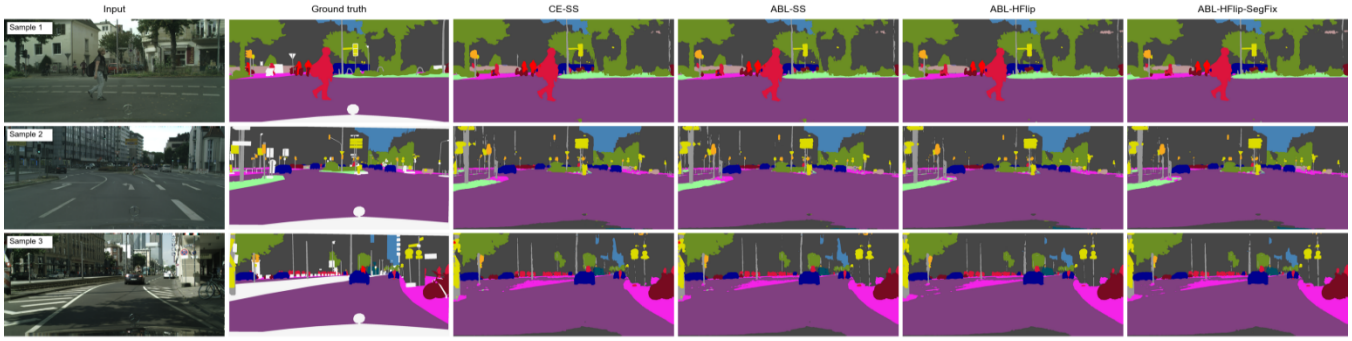

Figure 5. Inputs, ground truth, and CE/ABL predictions under different inference modes.

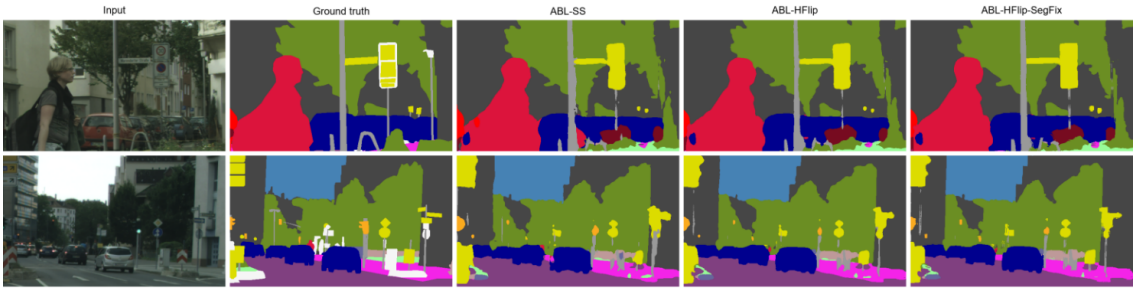

Figure 6. Enlarged local boundary comparisons from saved predictions.

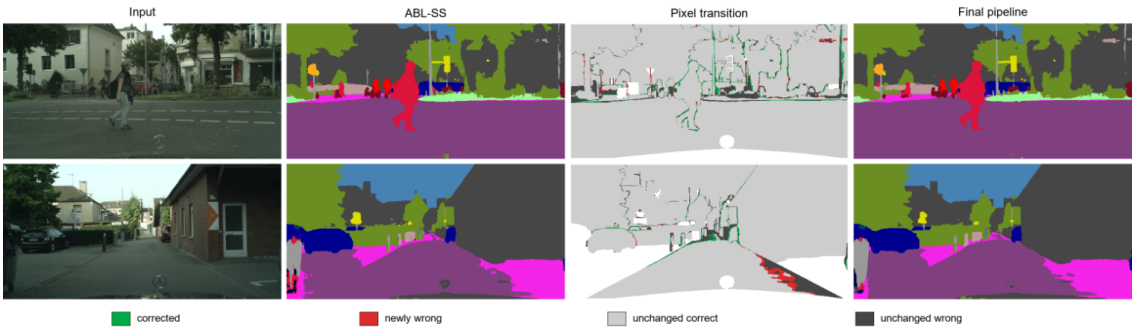

Figure 7. Pixel-correctness transitions from ABL single-scale inference to the complete pipeline.

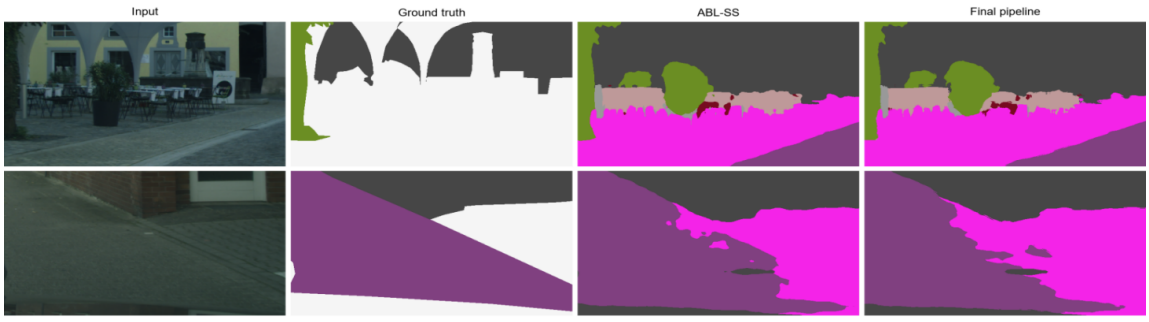

Figure 8. Local failure cases and residual errors in the complete pipeline.

H. Computational Cost and Reproducibility

HFlip requires two forward passes, and SegFix additionally depends on external boundary offsets. ABL-SS achieves 69.0954 percent mIoU with one forward pass, whereas ABL-HFlip-SegFix reaches 72.3881 percent at higher inference cost. The main stages of the server pipeline take approximately 36.9 minutes. Consistency checks pass for the 500 validation images, 500 matched offsets,

checkpoint hashes, and metric recomputation from saved predictions.

The shared original/flip inference stage processes 500 images in 442.9 s for CE and 428.1 s for ABL, or approximately 0.89 and 0.86 s per image. SegFix processes 2,000 prediction maps across four conditions in 856.2 s, approximately 0.43 s per map. These measurements include data access, caching, and server overhead. Since CE and ABL have identical architectures, the observed timing difference should not be interpreted as an architectural speed advantage; cache state and execution order may contribute.

ABL-SS requires neither extra main-model forward passes nor boundary-offset files. HFlip adds one mirror-view pass, and SegFix uses dataset-matched offsets to refine contours. The complete configuration may be useful for offline map updates, road inspection, or batch analysis. Its suitability for real-time edge deployment would require end-to-end latency measurements on the target hardware, including any boundary-offset model.

TABLE IX. ACCURACY, COST, AND EXTERNAL DEPENDENCIES

ABL inference mode	Main-model passes	External offsets	mIoU	Boundary F	Use case
SS	1	No	69.0954	54.4984	Low additional cost
HFlip	2	No	70.7509	57.9353	Accuracy without an external model
SS+SegFix	1	Required	70.6782	59.2708	Boundary-quality priority
HFlip+SegFix	2	Required	72.3881	62.3645	Highest offline accuracy

I. Cross-Architecture Validation with SegFormer-B0

We also evaluate ImageNet-pretrained SegFormer-B0 with three seeds to test whether the combined approach benefits a lightweight Transformer backbone. Both training settings use AdamW, learning rate 6×10^{-5}, weight decay 0.01, batch size 4, and ten epochs. The final-epoch checkpoint is selected in advance for each run, with no averaging across seeds. The data split, native-resolution evaluation, and 3-pixel boundary settings are unchanged. We compare baseline SS, ABL SS, and HFlip+SegFix applied to the same ABL checkpoint. Flip fusion averages logits. Because initialization and training differ from the DeepLabV3+ experiments, we compare gains within each backbone rather than rank their absolute accuracy.

ABL SS improves mean mIoU by 0.0847 pp over baseline SS, but seed 2 decreases by 0.1747 pp. The regional gain is therefore not consistent across all three seeds. Mean Boundary IoU and Boundary F improve by 0.3102 and 0.5764 pp, respectively, and Boundary F improves in every seed. The more consistent ABL benefit on SegFormer-B0 is thus in boundary quality.

Adding HFlip and SegFix to the same ABL checkpoint further improves mIoU and Boundary F in all three seeds. The mean gains over ABL SS are 1.8540 and 4.7624 pp, respectively; the gains over baseline SS are 1.9387 and 5.3387 pp. Timing in Table X covers CUDA-synchronized network forward passes, flip fusion, and upsampling, but excludes data input/output, SegFix, and offset prediction. Parameter counts also exclude the external SegFix model. Network-stage time increases from 58.58 to 117.33 ms per image, a rise of approximately 100.3 percent; this is not end-to-end latency. These results show boundary gains on a second architecture, but the three configurations do not reproduce the full CE/ABL-by-inference ablation. They therefore cannot establish whether the marginal ABL gain also narrows after SegFix on SegFormer.

V. DISCUSSION

A. Component Contributions and Coordination

The initial 512 by 1024 CE soup achieves 64.7343 percent mIoU, whereas the complete ABL-HFlip-SegFix pipeline reaches 72.3881 percent. Restoring native resolution adds 4.2116 pp for CE; HFlip adds 1.6716 pp at native resolution; SegFix adds 1.6616 pp after CE-HFlip; and ABL adds 0.1089 pp over CE under the complete protocol. The final accuracy therefore

reflects input resolution, view fusion, boundary postprocessing, and training-time boundary supervision together.

In the DeepLabV3+ experiments, ABL gives positive training gains across three seeds. The averaged checkpoints also show positive gains under all four inference protocols and in all 19 classes. With SS, adding SegFix reduces the ABL-minus-CE Boundary F gain from 0.5443 to 0.3543 pp; with HFlip, it reduces the gain from 0.5172 to 0.3127 pp. Thus, ABL remains useful after SegFix, although its marginal Boundary F gain is smaller. Partial overlap in the errors corrected is a plausible explanation, not a direct pixel-level attribution established by these aggregate comparisons.

The 3.2927 pp mIoU gain from adding HFlip and SegFix to ABL-SS requires two-view inference and additional offset processing. Single-scale inference offers lower cost, while the complete configuration provides higher regional and boundary accuracy with more computation.

B. Comparison of Boundary-Optimization Alternatives

Preliminary experiments also tested Lovász with auxiliary boundary branches, different boundary weights, PCGrad and semantic-priority gradient projection, semantic-boundary conditioning, ABL combined with semantic-boundary supervision, and logit ensembling. Most combined variants improved less than ABL alone. These results are consistent with competing regional and boundary objectives, but performance differences alone do not identify the cause. Three-model logit ensembling adds only 0.0040 pp mIoU over the ABL soup and slightly reduces boundary scores, suggesting limited useful diversity among these checkpoints.

TABLE X. SEGFORMER-B0 CROSS-ARCHITECTURE VALIDATION (MEAN AND SAMPLE STANDARD DEVIATION; THREE SEEDS)

Configuration	mIoU(%)	Boundary IoU(%)	Boundary F(%)	Network time (ms/image)	Main-model parameters (M)
Baseline (SS)	47.1658±0.7053	10.8558±0.0712	34.6500±0.2955	58.61±0.03	3.719
ABL (SS)	47.2505±0.7030	11.1660±0.0545	35.2264±0.3047	58.58±0.03	3.719
ABL+HFlip+SegFix	49.1044±0.6677	15.1893±0.1084	39.9887±0.4002	117.33±0.13	3.719

TABLE XI. EXPERIMENTAL COMPARISON OF BOUNDARY-OPTIMIZATION ALTERNATIVES

Configuration	Observation	Interpretation
Lovász/boundary-residual combination	No stable advantage	Limited benefit from jointly optimizing regional and boundary objectives
Alternative boundary weights	Weight changes did not alter the overall trend	Possible sensitivity to loss scale
PCGrad/semantic-priority projection	Did not consistently outperform ABL alone	Additional gradient-processing complexity with limited gains
SBCB combined with ABL	Weaker than ABL alone	Possible overlap between boundary objectives
Three-model logit ensemble	mIoU gain 0.0040 pp; boundary scores slightly lower	Extra forward passes without corresponding boundary gains
ABL+soup+matched inference	Consistent direction across seeds and protocols	Balance among stability, accuracy, and inference cost

One possible explanation for the weaker combined-loss results is that several objectives emphasize the same difficult boundary pixels, changing their effective weight relative to interior pixels. Auxiliary boundary heads also depend on the feature levels and multitask weights used. For the three checkpoints studied here, parameter averaging gives a single deployable model, while output ensembling offers little additional accuracy. These observations do not imply that larger or more diverse ensembles would behave in the same way.

Within the tested configurations, ABL with checkpoint averaging retains single-model inference cost and can be combined with HFlip and SegFix. The results favor evaluating each added component under matched conditions, since more boundary losses or more models did not consistently produce larger gains.

C. Scope and Limitations

The experiments use the full Cityscapes validation set, with three-seed DeepLabV3+ training comparisons and four inference protocols for the averaged checkpoints. Three-seed SegFormer-B0 experiments provide a second architecture. ABL improves Boundary F on both, and the complete inference configuration gives further gains. However, ABL-only mIoU changes are not positive for every SegFormer seed. The experiments support boundary improvements on these two backbones, but do not isolate the effects of backbone capacity, initialization, or training schedule on regional accuracy.

SegFix uses Cityscapes-matched directional offsets, so its effectiveness depends on offset quality and data distribution. mIoU measures regional overlap, while Boundary IoU and Boundary F characterize local contour quality. A fixed 3-pixel tolerance does not have the same relative effect across object scales. We therefore interpret aggregate metrics together with per-class results and qualitative examples.

D. Practical Implications

Under a constrained computational budget, ABL single-scale inference provides the consistent boundary gains observed across DeepLabV3+ seeds and classes with one main-model forward pass. When offline accuracy takes priority, HFlip and SegFix increase mIoU from 69.0954 to 72.3881 percent. The same training-inference analysis can be applied to lightweight backbones for edge scenarios, but end-to-end latency must be remeasured on the target hardware.

The complete mode can support offline refinement when the data distribution is stable and reliable offsets are available. Without matched offsets, ABL-SS or ABL-HFlip reduces external dependencies. Changes in the model or image-acquisition equipment require reevaluation of CE/ABL differences and postprocessing under the new distribution and annotation scheme.

E. Future Directions

Future work should reproduce key CE/ABL comparisons before and after SegFix on additional convolutional and Transformer backbones and road-scene datasets such as CamVid and BDD100K. This would clarify relationships among backbone capacity, scene distribution, and boundary gains. The present cross-architecture evidence still comes from Cityscapes alone; cross-model and cross-dataset studies are needed to test how ABL effects change after boundary postprocessing.

VI. Conclusions

We evaluated ABL during training together with HFlip and SegFix at inference for urban scene semantic segmentation. In the DeepLabV3+ experiments, ABL gives positive gains across three training seeds; the averaged checkpoints retain gains under four inference protocols and in all 19 classes. The complete configuration reaches 72.3881 percent mIoU, 25.9289 percent Boundary IoU, and 62.3645 percent Boundary F. On SegFormer-B0, ABL raises mean Boundary F from 34.6500 to 35.2264 percent, with a positive change in every seed, although the mIoU changes are not all positive. The complete configuration reaches 49.1044 percent mIoU and 39.9887 percent Boundary F, with approximately twice the network-stage time, excluding SegFix. The matched DeepLabV3+ comparisons show that the ABL Boundary F gain remains positive but becomes smaller after SegFix, consistent with complementary effects and partial overlap in the errors corrected.

For the tested models, ABL single-scale inference improves boundary quality without an extra main-model forward pass. HFlip and SegFix provide higher accuracy at the cost of a second view and external boundary correction. Deployment decisions must also account for offset generation, postprocessing, and data transfer, which require end-to-end measurements. The small gains from additional losses and logit ensembling in our preliminary comparisons further show why computational cost should be assessed alongside accuracy.

Appendix A. Reproduction Checklist

Table XII summarizes the main experiment's data split, checkpoints, external offsets, and evaluation settings to support reproduction.

TABLE XII. MAIN-EXPERIMENT REPRODUCTION SETTINGS

Item	Experimental setting
Data	Cityscapes: 2,975 train, 500 validation; image-label pairs verified
Paired training	Seeds 1/2/3; CE and ABL share source checkpoint, optimizer, and data order
Inference protocol	1024 by 2048; batch 1; SS/HFlip; multiscale and model ensembling disabled
Model assets	SHA-256 identity checks for CE and ABL soups before execution
SegFix assets	Official offset_semantic.zip; 2,025 .mat files; unique matches for 500 validation offsets
Metrics	19-class mIoU; 3-pixel Boundary IoU band and Boundary F tolerance
Metric recomputation	Online and saved-prediction metrics agree exactly for all eight conditions (500 images each)
Environment	One 24 GiB GPU; PyTorch 2.1.0a0; CUDA 12.2 image
Archive	remaining_pipeline_results_LITE_20260811T135053Z.tar.gz

In the complete pipeline, shared original/flip inference takes 442.9 s for CE and 428.1 s for ABL. SegFix processes 2,000 predictions across four conditions in 856.2 s, and evaluation of the eight saved prediction sets takes 488.1 s. These times include data access, caching, and server overhead and represent costs in this experimental environment. Peak allocated CUDA memory during the model stage is approximately 1.45 GiB, excluding framework-reserved memory.

ACKNOWLEDGMENT

This article is supported by “Xi'an Technology University's National-level College Students' Innovation and Entrepreneurship Training Program in 2025 (Project Number: 202510702030)”.

REFERENCES

[1] M. Cordts et al., "The Cityscapes dataset for semantic urban scene understanding," in Proc. CVPR, 2016, pp. 3213-3223.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in Proc. ECCV, 2018, pp. 801-818.

[3] C. Wang et al., "Active boundary loss for semantic segmentation," in Proc. AAAI, vol. 36, no. 2, 2022, pp. 2397-2405.

[4] Y. Yuan, J. Xie, X. Chen, and J. Wang, "SegFix: Model-agnostic boundary refinement for segmentation," in Proc. ECCV, 2020, pp. 489-506.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proc. CVPR, 2015, pp. 3431-3440.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. CVPR, 2016, pp. 770-778.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in Proc. CVPR, 2017, pp. 2881-2890.

[8] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," arXiv:1706.05587, 2017.

[9] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in Proc. ECCV, 2018, pp. 418-434.

[10] J. Wang et al., "Deep high-resolution representation learning for visual recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3349-3364, 2021.

[11] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in Proc. ECCV, 2020, pp. 173-190.

[12] C. Yu et al., "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in Proc. ECCV, 2018, pp. 325-341.

[13] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "PIDNet: A real-time semantic segmentation network inspired by PID controllers," in Proc. CVPR, 2023, pp. 19529-19539.

[14] E. Xie et al., "SegFormer: Simple and efficient design for semantic segmentation with Transformers," in Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 12077-12090.

[15] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in Proc. ICCV, 2019, pp. 5229-5238.

[16] M. Zhen et al., "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in Proc. CVPR, 2020, pp. 13666-13675.

[17] X. Li et al., "Improving semantic segmentation via decoupled body and edge supervision," in Proc. ECCV, 2020, pp. 435-452.

[18] H. Ishikawa and Y. Aoki, "Boosting semantic segmentation by conditioning the backbone with semantic boundaries," Sensors, vol. 23, no. 15, Art. no. 6980, 2023.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proc. ICCV, 2017, pp. 2980-2988.

[20] M. Berman, A. R. Triki, and M. B. Blaschko, "The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in Proc. CVPR, 2018, pp. 4413-4421.

[21] H. Kervadec et al., "Boundary loss for highly unbalanced segmentation," in Proc. MIDL, 2019, pp. 285-296.

[22] D. Karimi and S. E. Salcudean, "Reducing the Hausdorff distance in medical image segmentation with

convolutional neural networks," IEEE Trans. Med. Imag., vol. 39, no. 2, pp. 499-513, 2020.

[23] S. Borse, Y. Wang, Y. Zhang, and F. Porikli, "InverseForm: A loss function for structured boundary-aware segmentation," in Proc. CVPR, 2021, pp. 5901-5911.

[24] D. Wu et al., "Conditional boundary loss for semantic segmentation," IEEE Trans. Image Process., vol. 32, pp. 3717-3731, 2023.

[25] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in Proc. UAI, 2018, pp. 876-885.

[26] M. Wortsman et al., "Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in Proc. ICML, 2022, pp. 23965-23998.

[27] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in Advances in Neural Information Processing Systems, vol. 24, 2011, pp. 109-117.

[28] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in Proc. CVPR, 2020, pp. 9799-9808.

[29] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary IoU: Improving object-centric image segmentation evaluation," in Proc. CVPR, 2021, pp. 15334-15342.