Auxiliary Supervision from Existing Resources for Urban Scene Understanding: Multi-Task Learning for Semantic and BEV Segmentation

February 2026

Haruya Ishikawa

A Thesis for the Degree of Ph.D. in Engineering

Auxiliary Supervision from Existing Resources for Urban Scene Understanding: Multi-Task Learning for Semantic and BEV Segmentation

February 2026

Graduate School of Science and Technology
Keio University

Haruya Ishikawa

Abstract

Urban scene understanding through dense prediction tasks such as semantic segmentation and bird's-eye-view (BEV) mapping is fundamental to autonomous driving systems, yet faces persistent challenges: prohibitive pixel-level annotation costs, imprecise object boundary delineation, and stringent deployment constraints. This thesis investigates how auxiliary supervision derived from existing resources—reinterpreted annotations, pretrained models, and unlabeled data—can improve segmentation performance without requiring additional annotation efforts or prohibitive computational overhead. The core principle is extracting value from available resources: (i) semantic boundaries derived from existing segmentation masks, (ii) boundary-based regularization with unlabeled data in label-scarce settings, and (iii) knowledge embedded in pretrained perspective-view (PV) models. Across three complementary frameworks, this thesis show how to leverage these signals while maintaining annotation and deployment efficiency—often with no inference-time overhead. Experiments on wide variety of urban scene datasets demonstrate consistent gains, most pronounced under semi-supervised learning (SSL) protocols and in unsupervised domain adaptation (UDA) settings.

Chapter 1 motivates the problem and formulates the research questions around the tension between annotation cost, segmentation accuracies, and efficiency under real-world constraints. Chapter 2 provides background on semantic segmentation and semantic boundary detection (SBD), reviews multi-task learning (MTL) and SSL for dense prediction, and introduces BEV mapping with commonly used datasets and evaluation metrics. Chapter 3 presents the Semantic-Boundary-Conditioned Backbone (SBCB), which conditions the backbone with boundaries derived on-the-fly from existing segmentation mask. A lightweight SBD head provides auxiliary supervision during training and is discarded at inference, yielding consistent improvements with zero test-time overhead. Chapter 4 introduces BoundMatch, extending boundary supervision to SSL via consistency regu-

larization on both segmentation and boundaries. Rather than deriving boundaries from noisy segmentation masks, BoundMatch learns them directly from hierarchical features and refines them with additional pipelines. The approach yields substantial gains under label scarcity across multiple datasets and backbones. Chapter 5 describes Perspective Cue Training (PCT) for multi-camera BEV segmentation, which leverages pseudo-labels from pretrained PV models (*e.g.* semantic segmentation and relative-depth teachers) to provide auxiliary supervision without BEV annotations for the target data. PCT delivers significant improvements in both SSL and UDA while preserving the inference cost of the base BEV model. Chapter 6 synthesizes the findings across SBCB, BoundMatch, and PCT, discusses limitations, and outlines future directions. Overall, this thesis offers a practical recipe for dense urban perception—conditioning backbones on boundaries and distilling cues from pretrained perspective-view models—to reduce annotation and deployment costs while improving segmentation across tasks and domains.

論文要旨

セマンティックセグメンテーション・鳥瞰図（Bird's-Eye-View; BEV）マッピングといった密な予測タスクによる都市シーン理解は自動運転システムの基盤である一方で、ピクセルレベルのアノテーションコストのコスト、物体境界の不正確な描出、そして厳しいデプロイ制約といった持続的な課題に直面している。本論文では、再解釈したアノテーション、事前学習モデル、膨大なラベルなしデータなどの既存の資源から得られる補助的な教師信号（Auxiliary Supervision）が、追加のアノテーション作業や過大な計算オーバーヘッドを伴うことなくセグメンテーション性能をどのように向上し得るかを検討する。中核となる原理は、利用可能な資源から価値を引き出すことにある：(i) 既存のセグメンテーションマスクから導出されるセマンティック境界、(ii) 少ラベル設定におけるラベルなしデータでの境界ベースの正則化、(iii) 事前学習済みパースペクティブビュー（PV）モデルに埋め込まれた知識。相補的な3つのフレームワークを通じて、アノテーションおよびデプロイ効率を維持しつつ、多くの場合は推論時オーバーヘッドなしで、これらのシグナルを活用する方法を示す。多様な都市シーンデータセットでの実験により一貫した性能向上を確認し、とくに半教師あり学習（Semi-Supervised Learning; SSL）プロトコルおよび教師なしドメイン適応（Unsupervised Domain Adaptation; UDA）に関連する設定で顕著である。

第1章では、アノテーションコスト・セグメンテーション精度・現実的制約下での効率の観点から、問題設定の動機付けと研究課題の説明をする。第2章では、セマンティックセグメンテーションとセマンティック境界検出（Semantic Boundary Detection; SBD）の背景を述べ、マルチタスク学習（Multi-Task Learning; MTL）および高密度予測におけるSSLを概観し、一般的に用いられるデータセットと評価指標とともにBEVマッピングを導入する。第3章では、Semantic-Boundary-Conditioned Backbone（SBCB）を提示する。これは、既存のセグメンテーションマスクから生成した境界によってバックボーンを条件付ける手法である。軽量な境界検出ヘッドが

学習中に補助的な信号を与え、推論時には取り外すことで、テスト時オーバーヘッドなしで改善を実現する。第4章では、BoundMatchを提案し、セグメンテーションと境界の双方に対する一貫性正則化を通じて境界監督をSSLへ拡張する。ノイズを含むセグメンテーションマスクから境界を導くのではなく、階層的特徴から境界を直接学習し、追加のパイプラインで精緻化する。複数データセットおよびバックボーンにわたり、少ラベル条件下で大幅な性能向上を得る。第5章では、マルチカメラBEVセグメンテーションに対するPerspective Cue Training（PCT）を述べる。これは、事前学習済みPVモデル（セマンティックセグメンテーションおよび相対深度の教師モデル）からの擬似ラベルを活用し、ターゲットデータにBEVアノテーションを用いずに補助的な信号を与える手法である。PCTは、ベースとなるBEVモデルの推論コストを維持したまま、SSLとUDAの双方で顕著な改善をもたらす。第6章では、SBCB・BoundMatch・PCTを横断的に総括し、限界点を議論するとともに今後の研究課題を展望する。総じて、本論文はタスクおよびドメインを横断してセグメンテーションを向上させつつ、アノテーションとデプロイのコストを削減する手法を提示する。

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation and Problem Statement

The rapid proliferation of camera-based perception systems in urban environments represents one of the most significant technological shifts in modern cities. From advanced driver assistance systems (ADAS) integrated into millions of vehicles to the expanding networks of traffic monitoring cameras and security systems, visual perception has become the primary sensory modality for understanding and managing urban spaces [1, 2]. These systems, despite their diverse applications and deployment contexts, share a fundamental requirement: the ability to accurately parse and understand complex urban scenes in real-time [3, 4].

At the heart of this visual understanding lies dense prediction tasks, particularly semantic segmentation, which assigns a semantic label to every pixel in an image [5, 6]. This pixel-level understanding enables critical capabilities across applications: distinguishing pedestrians from roadways for collision avoidance in vehicles, identifying traffic violations in monitoring systems, and detecting anomalies in security footage [7, 8, 9, 10]. The quality of these segmentation outputs directly impacts the safety and effectiveness of these systems, making accurate scene understanding not merely a technical challenge but a societal imperative.

Among urban perception applications, automotive systems present particularly stringent requirements due to their direct implications for human safety. The Society of Automotive

Engineers (SAE) defines a spectrum of driving automation from Level 0 (no automation) to Level 5 (full automation), with Levels 2 and 3 representing the current frontier of commercially deployed systems [11]. At these levels, the vehicle assumes partial or conditional control of driving tasks, making accurate environmental perception not merely beneficial but safety-critical [1].

Semantic segmentation plays a distinct role in this perception stack. While object detection identifies discrete entities—vehicles, pedestrians, cyclists—with bounding boxes for tracking and behavior prediction, semantic segmentation provides complementary dense scene understanding [12]. It delineates drivable areas, road boundaries, and the spatial extent of obstacles—information that bounding boxes alone cannot capture. This pixel-level understanding directly supports safety functions such as lane keeping assistance and emergency braking, where precise knowledge of road geometry and free space is essential [13, 1].

However, in modular autonomous driving architectures, perception outputs feed into downstream modules for trajectory prediction and path planning. This sequential processing introduces the risk of error propagation: inaccuracies in segmentation—particularly at object boundaries or in adverse conditions—can compound through subsequent stages, potentially affecting safety-critical decisions [14]. ISO 21448 (SOTIF) explicitly identifies such perception limitations as sources of safety risk [15].

For more advanced autonomous systems, bird's-eye-view (BEV) segmentation has emerged as a key representation, transforming multiple camera views into a unified top-down perspective for spatial reasoning and path planning [16]. While primarily developed for vehicular applications, the principles underlying multi-view perception extend to other urban camera systems, from intersection monitoring to wide-area surveillance [17].

The proliferation of these perception systems—from vehicles to infrastructure—underscores both their utility and a fundamental tension. The supervised learning paradigm that has driven progress in semantic segmentation requires extensive pixel-level annotations, creating a scalability challenge that intensifies with each new deployment context. A single urban scene image may require around an hour for accurate pixel-level labeling by trained annotators, and the cost multiplies when considering the diversity of urban environments, weather conditions, times of day, and camera configurations encountered in real-world deployments [3, 18]. This annotation bottleneck becomes particularly acute when we consider that modern vehicles alone can generate terabytes of visual data per

day, while the capacity for manual annotation remains fundamentally limited by human resources [19]. Making use of the existing volumes of unlabeled data becomes not just advantageous but essential for scaling urban scene understanding systems.

These annotation demands are amplified for advanced perception tasks. BEV segmentation, introduced earlier as a key representation for autonomous navigation, requires not only semantic labels but also precise sensor calibration and accurate 3D spatial annotations that maintain consistency across multiple camera viewpoints [20, 21]. The scarcity of such annotations makes BEV segmentation a particularly compelling target for methods that can reduce reliance on labeled data.

The challenge extends beyond mere volume. Urban scenes exhibit significant domain variations that necessitate diverse training data: a model trained on data from sunny California streets may fail catastrophically in snowy Boston conditions; segmentation learned from vehicle-mounted cameras may not transfer to surveillance cameras mounted at different heights and angles; and algorithms developed for daytime operation may struggle under nighttime illumination [22, 23]. Each new domain potentially requires additional annotated data, compounding the annotation burden.

Furthermore, the deployment constraints of these systems introduce additional complexity. Autonomous vehicles may have access to powerful onboard computing, but ADAS systems must operate within tight computational and power budgets. Traffic monitoring systems need to process multiple camera streams simultaneously, while edge devices in distributed surveillance networks operate under severe resource constraints. These diverse deployment scenarios demand accurate, but most importantly, efficient solutions that can adapt to varying computational capabilities without sacrificing performance.

This thesis addresses these challenges through a unified exploration of auxiliary supervision derived from existing resources. The core insight is that valuable supervisory signals already exist within our current data and models—they simply need to be extracted and utilized effectively. Rather than requiring new annotations or complex architectural modifications, we can reinterpret existing segmentation labels as semantic boundaries, leverage off-the-shelf pretrained models to generate pseudo-labels, and utilize the abundantly available unlabeled data through regularization techniques. The goal is to utilize existing data and models more effectively, rather than to develop new resources.

1.2 Challenges in Urban Scene Understanding

The challenges facing urban scene understanding systems extend across multiple dimensions, from the fundamental difficulties of visual perception in complex environments to the practical constraints of real-world deployment. Understanding these challenges is essential for motivating the approaches developed in this thesis.

1.2.1 The Annotation and Supervision Challenge

Table 1.1: Dataset size and annotation time for major public datasets for semantic segmentation and urban driving perception.

Dataset	**Size**	**Fine Annotation Time**
Cityscapes [3]	5K	∼90 min/image
Mapillary Vistas [24]	25K	∼94 min/image (∼15 minutes for QA)
BDD100K [18]	100K	∼75 min/image[1]
ApolloScape [25]	143,906	∼27 min/image
ADE20K [26]	20K	∼80 sec/instance
SemanticKITTI [27]	43,552	1.5∼4.5 hours per tile
NuScenes [28]	40K	7,937 hours total and 100K USD (according to [29])

The most immediate challenge facing urban perception systems is the cost and complexity of obtaining high-quality annotations, also known as ground-truth (GT) labels [3, 24]. In Tab. 1.1, we summarize the annotation times reported for several major public datasets used in urban scene understanding surveyed by [29].

Dense pixel-level labeling requires not only identifying object boundaries but also handling ambiguous cases such as partially occluded objects, transparent surfaces, and fine structures. For efficiency, instead of pixel-wise labeling, popular datasets often use polygonal annotations (*e.g.* using LabelMe [31]) that are later classified and rasterized into segmentation masks. Professional annotators must be trained to maintain consistency across diverse scenarios, and quality control processes are necessary to ensure annotation reliability [24]. Even with rigorous protocols, the inherent subjectivity in pixel-level labeling and human errors leads to noisy annotations that can hinder model training

[1] In [30], the authors re-annotated a few images to measure the annotation times. The official annotation time for BDD100K is not available.

(Fig. 1.2 potentially shows this occurring in a popular dataset). Recent semi-automated annotation tools have alleviated some of this burden, but they still are prone to errors and require substantial human oversight and correction.

This annotation burden becomes especially acute when considering the scale of data required for robust urban perception. While modern deep learning models benefit from large labeled datasets, the cost of pixel-wise annotation can reach several hours per image for complex urban scenes. The challenge has motivated the development of semi-supervised learning approaches that can leverage abundant unlabeled data alongside smaller labeled sets, reducing annotation requirements while maintaining competitive performance [32]. For instance, using only 1/16 of labeled data with appropriate semi-supervised techniques can achieve results comparable to fully supervised baselines [33].

Specialized urban perception tasks compound these challenges further. Bird's-eye-view segmentation (BEV), essential for autonomous navigation, requires understanding 3D spatial relationships from 2D multi-camera inputs. Creating BEV annotations necessitates careful sensor calibration (multi-camera rig and other sensors like LiDAR) and a complex 3D reconstruction pipeline, making the annotation process for datasets like NuScenes and Argoverse 2 significantly more resource-intensive than standard perspective view semantic segmentation [28, 34]. The scarcity of BEV annotations has made this domain particularly suitable for semi-supervised and weakly-supervised approaches that can utilize perspective view annotations or pseudo-labels to reduce the annotation burden.

Beyond annotation quantity, domain variation presents an additional challenge [18]. Urban environments exhibit substantial diversity across geographical locations, weather conditions, and temporal factors. Models trained on data from one domain often experience significant performance degradation when deployed in another—for example, transitioning from daytime to nighttime scenarios or from clear to adverse weather conditions. This domain shift problem intersects with the annotation challenge, as obtaining labeled data for every possible domain variation would be prohibitively expensive, further motivating the need for methods that can effectively utilize unlabeled data from diverse domains.

1.2.2 Technical Segmentation Challenges

Beyond annotation, urban scene segmentation faces inherent technical difficulties stemming from the complexity and variability of urban environments. Three categories of challenges

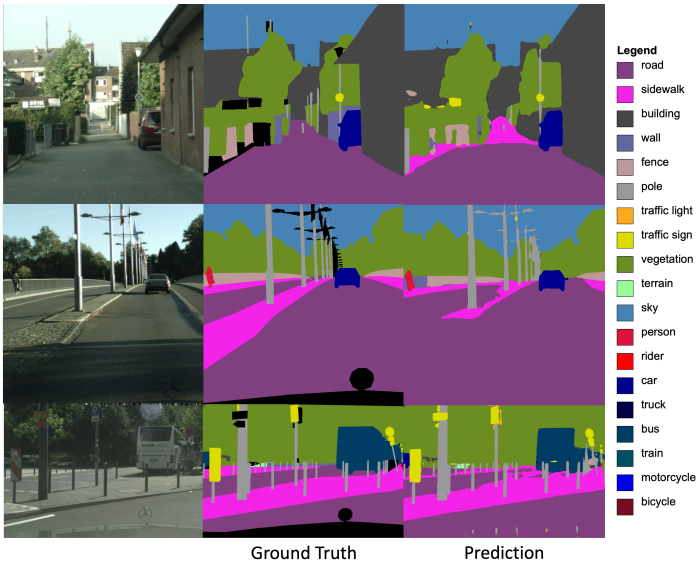


Figure 1.1: Boundary artifacts and road/sidewalk confusion in urban segmentation predictions.

are particularly relevant: boundary accuracy, small object segmentation, and robustness to environmental variations.

Object boundaries represent areas of maximum uncertainty in segmentation [35, 36, 37]. Current methods typically achieve over 95% accuracy in object interiors but can drop below 70% at boundaries [38]. Occlusion and overlapping objects further create ambiguous boundaries that challenge even human annotators. Fig. 1.1 illustrates this on Cityscapes [3]: prediction boundaries exhibit noticeable "blob"-like characteristics, and road/sidewalk boundaries are often misclassified. In autonomous driving systems, drivable area estimation relies on accurate segmentation of road surfaces and their boundaries; errors in this estimation directly affect path planning and free-space calculation [13]. ISO 21448 (SOTIF) identifies perception limitations as a source of safety risk, characterizing uncertainty in terms of existence uncertainty (whether objects are correctly detected) and state uncertainty (whether object properties are correctly estimated) [15, 39].

Urban scenes contain numerous small but critical objects that challenge existing segmentation approaches [40, 3, 18]. Traffic cones, poles, and signs occupy minimal image area but carry important semantic information for understanding road layout and identifying hazards. These objects are particularly vulnerable to class imbalance issues, where models tend to ignore or misclassify them in favor of dominant classes like road

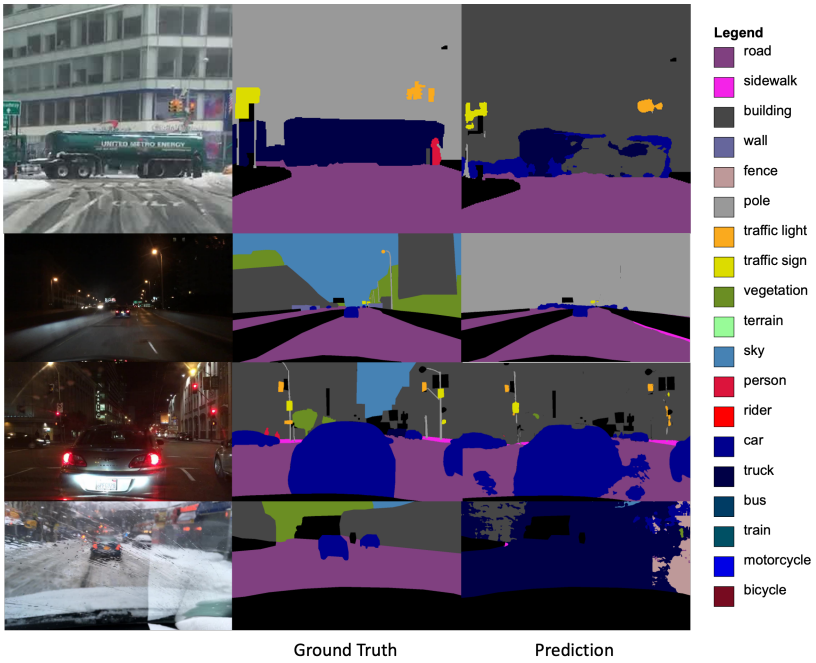


Figure 1.2: Environmental challenges (adverse weather, low lighting) and annotation errors in BDD100K.

and building. The problem is exacerbated in distant views where these objects may span only dozens of pixels. While bounding-box-based detection identifies the presence of such objects, pixel-level segmentation provides precise spatial extent information useful for free-space estimation in path planning. Beyond known object categories, segmentation-based approaches are particularly relevant for detecting unexpected road obstacles such as fallen cargo or debris, which may not conform to predefined detector classes and exhibit irregular shapes that bounding boxes cannot accurately represent [41, 42].

Finally, real-world urban scenes exhibit environmental and data quality challenges that compound the difficulties above. Fig. 1.2 illustrates examples from BDD100K: adverse weather conditions like rain and fog obscure object boundaries, nighttime scenes with low lighting complicate segmentation, and annotation errors in the dataset itself can mislead model training [43, 44]. In modular autonomous driving architectures, perception errors can propagate through prediction and planning modules [14, 45]. These challenges reinforce the need for methods that improve segmentation accuracy—particularly at boundaries and for small objects—while extracting maximum value from available supervision.

1.2.3 Deployment Constraints

The diversity of deployment scenarios for urban perception systems introduces stringent constraints that shape the development of practical solutions. Real-time processing requirements are perhaps the most universal constraint, with most applications requiring at least 10 Hz operation for basic functionality and 30+ Hz for smooth operation [46]. This translates to maximum processing times of 33–100 milliseconds per frame, including not just segmentation but also pre-processing, post-processing, and communication overhead [47].

Computational resources vary dramatically across deployment contexts. High-end autonomous vehicles may incorporate multiple NVIDIA GPUs with hundreds of TFLOPS of computing power, while ADAS systems in production vehicles typically rely on embedded processors with 10–100× less computational capability [48, 49]. Edge devices in smart city infrastructure may be even more constrained, operating on platforms with limited memory, power, and cooling [50]. This computational heterogeneity demands solutions that can scale across various requirements.

Energy efficiency, often overlooked in research settings, becomes crucial in many deployment scenarios. Battery-powered devices, vehicles seeking to maximize range, and large-scale deployments where energy costs are significant all require careful attention to computational efficiency. The energy cost of running complex neural networks continuously can be substantial, with some autonomous vehicle perception stacks consuming kilowatts of power [51].

1.3 Research Questions and Objectives

Given these challenges, this thesis investigates how auxiliary supervision from existing resources can address the fundamental tensions between annotation cost, model accuracy, and deployment efficiency in urban scene understanding. The research is guided by three core questions that explore different aspects of this central theme.

The first research question examines whether auxiliary supervision derived from existing annotations can improve segmentation performance without requiring additional labeling effort. Specifically, this thesis investigates whether semantic boundaries, which can be automatically extracted from existing segmentation

masks, provide complementary information that enhances the primary segmentation task. This question challenges the conventional approach of treating segmentation masks as monolithic annotations, suggesting instead that they contain derivable structural cues that can be exploited for improved learning.

The second research question explores how auxiliary tasks can provide regularization in label-scarce scenarios, particularly in semi-supervised learning settings where only a fraction of the data is annotated. This thesis investigates whether the multi-task learning framework, when extended to include boundary detection alongside segmentation, can provide additional constraints that prevent overfitting to limited labeled data and improve generalization to unlabeled samples. This question is particularly relevant given the annotation challenges discussed earlier, as it directly addresses the scalability problem by making use of the abundant unlabeled data available in urban environments.

The third research question investigates whether auxiliary supervision can bridge tasks with asymmetric annotation availability—specifically, whether readily available models pretrained on datasets from annotation-rich domains can provide supervision for annotation-scarce but semantically related tasks. This thesis explores whether pseudo-labels generated from pretrained models on individual perspective views from multi-camera systems can provide useful supervision for learning BEV representations, potentially eliminating the need for expensive BEV annotations in semi-supervised and domain adaptation scenarios. This question extends the auxiliary supervision concept across different representation spaces and examines the transferability of knowledge between tasks/views.

These research questions translate into specific objectives that guide the technical development in this thesis:

- First, this thesis aims to develop practical frameworks that extract maximum value from existing resources, whether those are segmentation annotations that can be reinterpreted as boundaries, pretrained models that can generate cost-effective pseudo-labels, and/or unlabeled data that can be leveraged through regularization techniques.

- Second, this thesis seeks to validate the effectiveness of these approaches across multiple urban segmentation scenarios, from fully-supervised to semi-supervised settings, and from single-view to multi-view tasks.

- Third, this thesis prioritizes maintaining deployment efficiency while improving accuracy, ensuring that performance gains do not come at the cost of practical applicability.

- Finally, this thesis aims to provide empirical insights that can guide the community in understanding when and how auxiliary supervision provides benefits.

1.4 Contributions and Thesis Organization

This thesis makes several contributions to the field of urban scene understanding, spanning theoretical insights, technical innovations, and practical implementations. These contributions are organized around three case studies that explore complementary aspects of auxiliary supervision from existing resources.

The technical contributions center on three novel frameworks that instantiate the auxiliary supervision principle in progressively challenging contexts.

SBCB (Semantic-Boundary-Conditioned Backbone, Chapter 3) establishes the foundation by demonstrating that semantic boundary detection serves as an effective auxiliary task for improving segmentation in fully-supervised settings. The contributions are: (1) a multi-task framework where hierarchical backbone features are jointly supervised by segmentation and boundary objectives, (2) on-the-fly boundary label generation from existing segmentation masks, requiring zero additional annotation effort, (3) the design principle that auxiliary heads can be removed at inference, incurring no computational overhead, and (4) extensive validation showing consistent improvements across diverse architectures and datasets. By improving boundary accuracy without inference overhead, SBCB directly addresses the segmentation challenges identified in Sec. 1.2.2.

BoundMatch (Chapter 4) extends auxiliary boundary supervision to semi-supervised learning, where labeled data is scarce. The contributions are: (1) Boundary Consistency Regularized Multi-task Learning (BCRM), which enforces consistency on both segmentation and boundary predictions between teacher and student networks, (2) fusion modules (BSF, SGF) that enable bidirectional information flow between tasks while maintaining efficiency, (3) Harmonious Batch Normalization (HBN) to address training instabilities in EMA-based frameworks, and (4) comprehensive evaluation demonstrating that dual-task consistency provides stronger regularization than segmentation consistency alone. This is relevant for

Table 1.2: Summary of thesis contributions under the auxiliary supervision paradigm. SSL = semi-supervised learning, UDA = unsupervised domain adaptation, U = unlabeled data.

Work	**Regime**	**Target**	**Aux signal / labels (train)**
SBCB (Chapter 3)	Fully Sup.	PV-seg	Boundaries / full pixel GT
BoundMatch (Chapter 4)	SSL	PV-seg	Boundaries / few GT + U
PCT (Chapter 5)	SSL/UDA	BEV-seg	PV teachers / few BEV GT + U

deploying perception systems across diverse conditions where labeled data is scarce but consistent performance is required.

PCT (Perspective Cue Training, Chapter 5) applies auxiliary supervision to multi-camera BEV segmentation under semi-supervised learning and unsupervised domain adaptation. The contributions are: (1) a framework that leverages pretrained perspective view models (*e.g.* Mask2Former) to generate pseudo-labels as auxiliary supervision for BEV learning, (2) among the first systematic studies of semi-supervised learning for multi-camera BEV segmentation, introducing tailored augmentation strategies (CamDrop, BFD), and (3) camera-only domain adaptation that achieves competitive performance against methods requiring additional sensor modalities like LiDAR. Accurate BEV segmentation provides the spatial representation used by planning modules for trajectory generation and obstacle avoidance [16].

Tab. 1.2 and Fig. 1.3 summarize the three case studies and illustrate their relationships. The progression from SBCB to BoundMatch demonstrates that auxiliary boundary supervision, effective in fully-supervised settings, provides even stronger regularization when labels are scarce. The connection from BoundMatch to PCT shows that the underlying principle—conditioning shared representations with auxiliary task supervision—generalizes beyond boundary detection to other auxiliary tasks and target domains. In PCT, perspective view segmentation serves as the auxiliary task, and the benefits manifest as improved BEV representations for both semi-supervised learning and domain adaptation.

The thesis is organized as follows. **Chapter 2** provides background on semantic segmentation, boundary detection, multi-task learning, and semi-supervised learning. **Chapters 3–5** present the three case studies in sequence, each building on insights from the previous. **Chapter 6** synthesizes the findings and discusses future directions. The appendices provide implementation details and extended results.

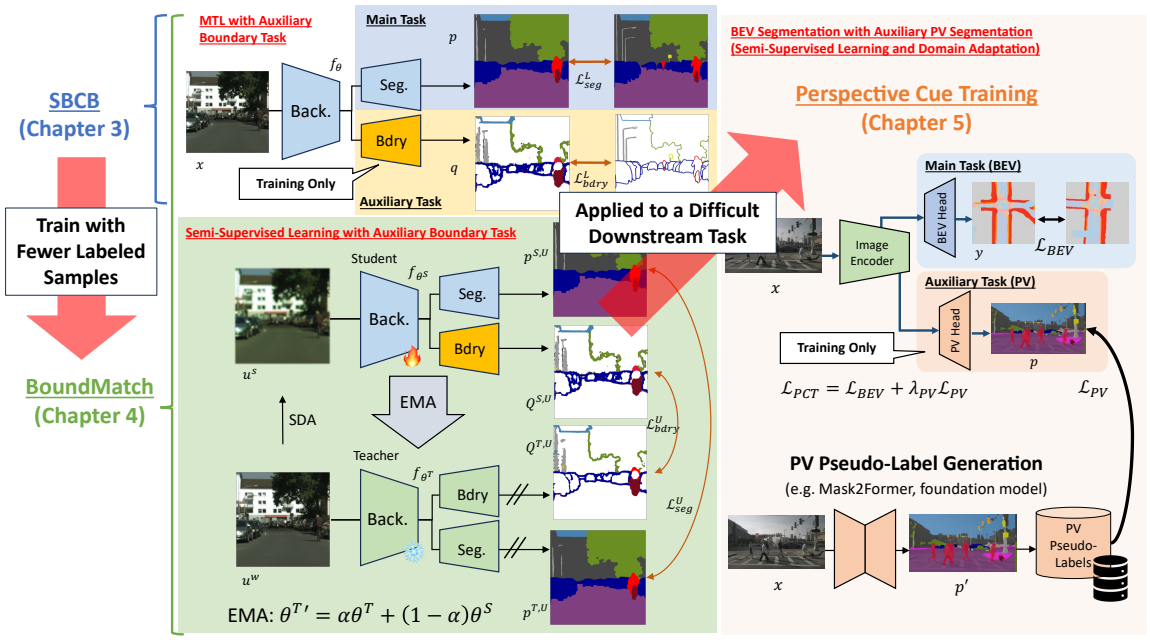


Figure 1.3: Overview of the three frameworks. SBCB conditions the backbone with boundary detection at zero inference cost; BoundMatch extends this to semi-supervised learning via dual-task consistency; PCT applies the principle to BEV segmentation using perspective view pseudo-labels. Auxiliary heads can be removed at inference for efficiency.

From a theoretical perspective, this thesis provides a unified framework for understanding how auxiliary supervision from existing resources enhances urban perception tasks. The empirical analyses demonstrate the complementarity between primary and auxiliary tasks across different architectures, datasets, and training paradigms. The thesis also establishes evaluation protocols considering both standard metrics and boundary-specific measures.

From a practical standpoint, this thesis provides comprehensive benchmarking across varying architectures (CNNs to transformers), datasets (Cityscapes, BDD100K, nuScenes), and learning regimes (fully-supervised to domain adaptation). Detailed ablation studies isolate each component's contribution, and deployment strategies balance accuracy-efficiency trade-offs from edge devices to full models.

1.5 Publication List

The publication list of the author and the corresponding chapters in the thesis are as follows:

- Haruya Ishikawa and Yoshimitsu Aoki, *Boosting Semantic Segmentation by Conditioning the Backbone with Semantic Boundaries. Sensors*, 23, 2023. (Chapter 3)
- Haruya Ishikawa and Yoshimitsu Aoki, *BoundMatch: Boundary Detection Applied to Semi-Supervised Segmentation. IEEE Access*, 13, 172776–172798, 2025. (Chapter 4)
- Haruya Ishikawa, Takumi Iida, Yoshinori Konishi, and Yoshimitsu Aoki, *PCT: Perspective Cue Training Framework for Multi-Camera BEV Segmentation.* In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 13253–13260, 2024. (Chapter 5)

1.6 Summary

Urban scene understanding through dense prediction represents a critical capability for numerous applications, from autonomous vehicles to smart city infrastructure. However, the traditional supervised learning paradigm faces fundamental scalability challenges due to expensive pixel-level annotation requirements, domain diversity, and deployment constraints. This thesis explores how auxiliary supervision derived from existing resources—segmentation annotations, easily accessible pretrained models, and unlabeled data—can address these challenges without requiring additional annotation efforts or prohibitive computational overhead.

Through three complementary studies, this thesis demonstrates that auxiliary supervision from existing resources—whether reinterpreting segmentation annotations as boundaries or leveraging pretrained model knowledge by utilizing abundant unlabeled multi-camera data—can significantly improve segmentation performance across various settings. The frameworks developed in this thesis—SBCB for boundary conditioning, BoundMatch for semi-supervised learning with boundary regularization, and PCT for perspective-supervised BEV segmentation—provide practical solutions that balance accuracy improvements with deployment efficiency. Extensive empirical validation across diverse datasets

confirms the broad applicability of these approaches while honestly acknowledging their limitations.

As urban environments become increasingly instrumented with visual sensors and the demand for accurate scene understanding continues to grow, the ability to extract maximum value from existing resources becomes ever more critical. This thesis contributes to this goal by establishing auxiliary supervision as a practical and effective paradigm for improving urban scene understanding, providing both immediate technical solutions and longer-term insights that can guide future research in this vital area.

2

Background and Related Work

2.1 Semantic Segmentation

2.1.1 Evolution and Core Methods

Pixel-wise classification tasks have long been central to computer vision [52]. Edge detection, one of the earliest forms of dense prediction, identifies boundaries between regions through intensity discontinuities. Classical edge detection methods like Canny edges produce binary boundary maps, while segmentation methods like active contours and level-set models [53, 54] demonstrated how edge information could guide region delineation.

Semantic segmentation extends this foundation by assigning every pixel a class label, transforming an input image $I \in \mathbb{R}^{H \times W \times 3}$ into a dense label map $Y \in \{0, 1, \ldots, C-1\}^{H \times W}$. Unlike edge detection's binary outputs, semantic segmentation requires both regional understanding and precise boundary localization. This pixel-level understanding is fundamental for urban scene parsing, where precise delineation between classes directly impacts safety-critical decisions.

Fully Convolutional Networks (FCN) [5] transformed semantic segmentation by enabling end-to-end learning for dense prediction. By replacing fully connected layers with convolutional operations, FCNs could process arbitrary-sized inputs and produce correspondingly-sized outputs. The key challenge was recovering spatial resolution after downsampling in feature extraction [55].

Encoder-decoder architectures addressed this resolution loss. SegNet [56] introduced

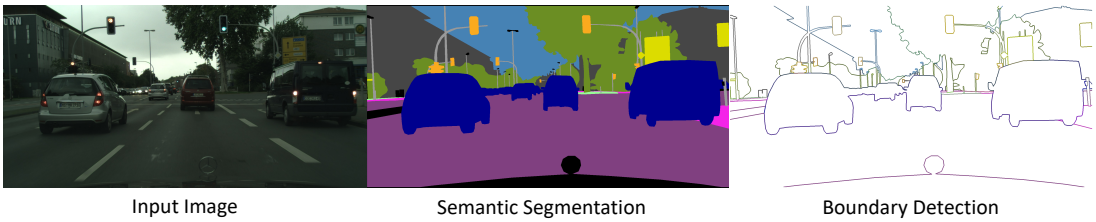


Figure 2.1: Example of semantic segmentation (middle) and semantic boundary detection (right) on an urban driving scene. Semantic boundary detection produces multi-label boundary maps where each channel corresponds to a specific class boundary.

pooling indices for upsampling, while U-Net established skip connections that preserve fine-grained details. These skip connections proved crucial for maintaining boundary precision—a recurring theme in segmentation architecture design [57, 58].

Multi-scale context modeling brought further improvements. PSPNet's Pyramid Pooling Module aggregates context at multiple scales [59], while the DeepLab series pioneered atrous convolutions for larger receptive fields without resolution loss [60, 61]. DeepLabV3+ combines atrous spatial pyramid pooling with a decoder module, achieving effective balance between global context and local detail [62].

Attention mechanisms enable explicit relationship modeling across spatial and channel dimensions [63, 64, 65, 66, 67, 68, 69, 70]. OCRNet [71] computes pixel-to-object relations, implicitly learning better boundary representations through object-aware context. The transformer paradigm, introduced through SegFormer [72] and the MaskFormer series [73, 74], reformulates segmentation as mask classification, achieving state-of-the-art results with architectural simplicity.

Efficient architectures address deployment constraints. BiSeNet's dual-pathway design separates spatial and context processing, while mobile architectures (MobileNetV2/V3) enable edge deployment through depthwise separable convolutions and neural architecture search [75, 76, 77, 78, 79].

2.1.2 Edge and Boundary Detection

While semantic segmentation evolved toward regional classification, edge and boundary detection continued advancing as a complementary task. HED [80] introduced multi-scale learning through side outputs at different network depths for improved edge detection.

Unlike classical methods, learned edge detection could capture semantic meaning alongside geometric discontinuities.

The convergence of segmentation and boundary detection produced semantic boundary detection—identifying not just any boundary, but boundaries between specific semantic classes. This task bridges pure edge detection and semantic segmentation, requiring both precise localization and semantic understanding [81] as shown in Fig. 2.1. Unlike semantic segmentation, semantic boundary detection (SBD) is formulated as multi-label pixel-wise classification, where each class has its own boundary channel (*i.e.* boundaries can overlap).

CASENet [82] pioneered SBD with a convolutional network making use of the rich hierarchical representations in deep networks inspired by HED. The architecture extracts hierarchical features: early layers (also called "sides") provide detailed edge features while deeper layers contribute semantic understanding. DFF [83] and DDS [84] extended this with adaptive fusion and deep supervision, respectively.

Traditionally, boundary detection approaches relied on manual annotation or pre-computed edges from segmentation masks. However, as explored in Chapter 3, on-the-fly (OTF) boundary generation from segmentation masks offers a flexible and efficient alternative.

2.2 Multi-Task Learning and Boundary-Aware Methods

2.2.1 Multi-Task Learning Foundations

Multi-task learning (MTL) enables neural networks to learn multiple related tasks simultaneously, leveraging shared representations to improve generalization [85]. In dense prediction tasks, MTL exploits the inherent relationships between different visual modalities; for example, depth, surface normals, and semantic segmentation [86, 87, 88]. This shared learning can improve individual task performance while reducing computational redundancy through parameter sharing.

The predominant MTL architecture employs hard parameter sharing: a shared encoder extracts common features, with task-specific decoders producing specialized outputs [89, 87]. This design naturally fits hierarchical vision models where early layers capture

low-level features useful across tasks, while deeper layers specialize. Through shared parameters, related tasks provide complementary supervision that acts as a regularizer, reducing overfitting risk while enabling mutual enhancement through feature sharing [90, 91].

However, MTL introduces the challenge of task balancing. Different tasks converge at different rates and scales, leading to gradient imbalances that can cause negative transfer—where joint training performs worse than single-task learning [92, 93]. Kendall *et al.* [91] addressed this through uncertainty-based weighting, automatically balancing losses based on task-specific homoscedastic uncertainty. Alternative approaches include gradient normalization [92], dynamic weight averaging [94], and task-specific learning rates [95].

Auxiliary tasks serve a distinct role in MTL: rather than being objectives of interest themselves, they provide additional supervision to improve the main task [96]. PSPNet demonstrated this principle by adding an auxiliary FCN head at an intermediate layer, providing additional gradient flow and regularization during training [59]. This auxiliary supervision helps combat gradient vanishing in deep networks while encouraging the learning of multi-scale representations. Critically, auxiliary tasks can be discarded at inference, adding no computational overhead—a principle central to efficient boundary-aware methods.

The selection of auxiliary tasks significantly impacts effectiveness. Taskonomy [97] revealed task affinities through large-scale empirical analysis, showing that geometric tasks (*i.e.* depth, normals) and semantic tasks (*i.e.* segmentation, classification) form natural clusters. Boundary detection occupies a unique position: it bridges low-level edge detection and high-level semantic understanding, making it particularly effective as an auxiliary task for segmentation—an insight leveraged across boundary-aware architectures. This thesis investigates this synergistic relationship, especially for semantic boundary detection (SBD) as an auxiliary task to enhance semantic segmentation in both fully-supervised and semi-supervised settings (Chapter 3 and Chapter 4).

2.2.2 Boundary-Aware Segmentation

The integration of boundary information into semantic segmentation has evolved along two primary paths: architectural fusion approaches that explicitly combine boundary and segmentation streams, and implicit methods that encourage boundary awareness through

loss functions or post-processing. This distinction reflects a fundamental trade-off between representation power and computational efficiency.

2.2.2.1 Multi-Task Learning Approaches

Owing to the fact that segmentation and boundary detection are inherently complementary tasks, multi-task learning (MTL) has been the predominant strategy for boundary-aware segmentation.

Most common boundary-aware methods utilize binary boundary detection as an auxiliary task to enhance semantic segmentation [98, 99, 77, 100, 101].

GSCNN [98] established the two-stream paradigm for boundary-aware segmentation. The architecture comprises a regular stream for semantic segmentation and a shape stream for boundary detection, connected through a novel gated convolutional layer along with Canny edge features. The dual-task learning with explicit feature fusion demonstrated significant improvements in boundary quality.

The two-stream design has been extensively adopted in subsequent works which utilize boundary auxiliary task [77, 102, 101, 100]. For example, STDC [77] incorporates a Detail Head that processes high-resolution features to preserve fine details, supervised by boundary-like detail ground-truth generated through Laplacian operators. PIDNet [100] employs a three branch network to parse detailed, context, and boundary information separately, balancing the effects of each branch through PID controller inspired modeling.

Methods that do not require a separate boundary detection branch have also been proposed. DecoupleSegNet [99] proposed to decouple features into body and boundary components to handle inner object consistency and fine-grained boundaries jointly.

Semantic boundary detection (SBD) as an auxiliary task has also been explored [103, 102, 104]. RPCNet [103] introduced the first joint semantic segmentation and semantic boundary detection model using the iterative refinement between the two tasks. By recursively exchanging information between tasks across multiple scales, the model progressively refines both outputs. CSEL [102] proposed three branch approach for joint semantic segmentation and SBD, where the additional branch outputs affinity matrix. The affinity matrix and raw boundary detection results are used to refine the segmentation feature map through recurrent message passing by dynamic graph propagation. Mobile-Seed [104] adapts joint training between semantic segmentation and SBD for

lightweight architectures through efficient boundary-semantic fusion. The bidirectional flow of information—segmentation informing boundary detection and vice versa—exemplifies the complementary nature of these tasks and results in significant performance gains.

In Chapter 3, we extensively explore the use of SBD as an auxiliary task for semantic segmentation using a simple auxiliary head. This auxiliary head operates only during training, adding no inference cost—demonstrating the practical value of training-time MTL. Furthermore, in Chapter 4, this auxiliary SBD supervision is extended to semi-supervised settings, leveraging boundary information for stronger regularization when training with unlabeled data.

2.2.2.2 Post-Processing and Implicit Methods

Not all boundary-aware methods require architectural modifications. Post-processing approaches refine segmentation outputs using learned or heuristic boundary corrections, operating on the principle that segmentation errors concentrate near boundaries.

SegFix [105] exemplifies model-agnostic refinement, training a separate network to correct boundary errors in segmentation outputs. By learning to replace uncertain boundary predictions with more reliable interior labels, SegFix improves various base models without architectural constraints. However, this independence comes at the cost of additional inference computation and separate training requirements.

DeepStrip [106] focuses on high-resolution boundary refinement through strip pooling and boundary-aware loss functions. The method processes boundaries at native resolution while maintaining computational efficiency through sparse processing. SDN [107] reformulates boundary refinement as an anisotropic diffusion process, iteratively sharpening boundaries through learned diffusion coefficients.

Implicit boundary awareness can also be induced through specialized loss functions. Active Boundary Loss [108] weights pixel contributions based on distance to boundaries, forcing models to focus on challenging boundary regions. InverseForm Loss [109] penalizes topological inconsistencies that often manifest as boundary errors. These loss-based approaches offer the advantage of architectural independence but typically show smaller improvements than explicit boundary modeling.

While these boundary-aware methods demonstrate various strategies for improving segmentation quality, they largely operate within fully-supervised settings and are often

tied to specific architectures. The potential for boundary supervision as a general training principle—applicable across architectures and training paradigms—remains an open area for exploration, which is explored in Chapter 3.

2.3 Semi-Supervised Semantic Segmentation

2.3.1 Problem Formulation and Motivation

Dense pixel-level annotation is prohibitively expensive—a single image or an urban driving scene could take at most around 90 minutes of expert annotation [3, 18, 110, 24]. Semi-supervised semantic segmentation (SS-SS) addresses this bottleneck by leveraging abundantly available unlabeled data alongside limited labeled examples.

Given a small labeled dataset $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{N_L}$ and a larger unlabeled dataset $\mathcal{D}_U = \{x_j\}_{j=1}^{N_U}$ where $N_U \gg N_L$, the goal is to learn a segmentation model that generalizes well by exploiting both data sources. Semi-supervised methods augment the standard supervised loss with an unsupervised term: $\mathcal{L} = \mathcal{L}_{sup} + \lambda \mathcal{L}_{unsup}$, where λ balances the two objectives.

2.3.2 Taxonomy of Semi-Supervised Approaches

Semi-supervised segmentation methods fall into five main categories [111]:

Adversarial methods use discriminators to distinguish real from predicted segmentation maps, providing adversarial supervision for unlabeled data [112, 113]. Despite early promise, training instability limits practical adoption.

Pseudo-labeling generates labels for unlabeled data using model predictions, then retrains with these pseudo-labels [114, 115, 116]. The main challenge is confirmation bias—models reinforce their own errors.

Contrastive Learning learns discriminative features by contrasting similar and dissimilar pixel representations [117, 118]. While effective for features space boundary learning, these methods require careful sampling strategies.

Consistency Regularization (CR) enforces prediction stability under perturbations, based on the smoothness assumption that nearby points share labels [32, 119, 120, 121, 33,

122, 123, 124, 125, 126]. This paradigm dominates current research due to its simplicity and effectiveness.

Hybrid methods combine multiple paradigms, leveraging complementary strengths [127, 128, 129, 130, 131, 132].

Among these approaches, consistency regularization has emerged as the dominant paradigm in semi-supervised semantic segmentation. Unlike adversarial training's instability issues or contrastive learning's complex sampling requirements, CR methods offer both conceptual simplicity and strong empirical performance. Even state-of-the-art hybrid methods frequently build upon CR as their foundation, underscoring its central role. Given this prominence, we now examine the consistency regularization framework in detail, beginning with the Mean Teacher approach that established many of its core principles.

2.3.3 Consistency Regularization Framework

The consistency regularization paradigm leverages the smoothness assumption—that semantically similar inputs should produce similar predictions—to learn from unlabeled data [133]. While various implementations exist, the Mean Teacher framework [32] provides the foundational architecture that subsequent methods build upon.

2.3.3.1 Mean Teacher Framework

Mean Teacher maintains two networks: a student model updated via gradient descent and a teacher model whose weights are an exponential moving average (EMA) of the student's weights as shown in Fig. 2.2. The EMA update ($\theta^T \leftarrow \alpha\theta^T + (1-\alpha)\theta^S$) creates a temporally ensembled teacher that produces more stable predictions; where θ^S and θ^T are the student's and teacher's model parameters respectively.

For unlabeled data, the student must produce consistent predictions with the teacher despite receiving perturbed inputs, while the teacher receives clean inputs. This asymmetry forces the student to learn robust representations invariant to perturbations.

2.3.3.2 Perturbation Strategies

Consistency regularization effectiveness depends critically on perturbation design. Modern approaches apply perturbations at multiple levels:

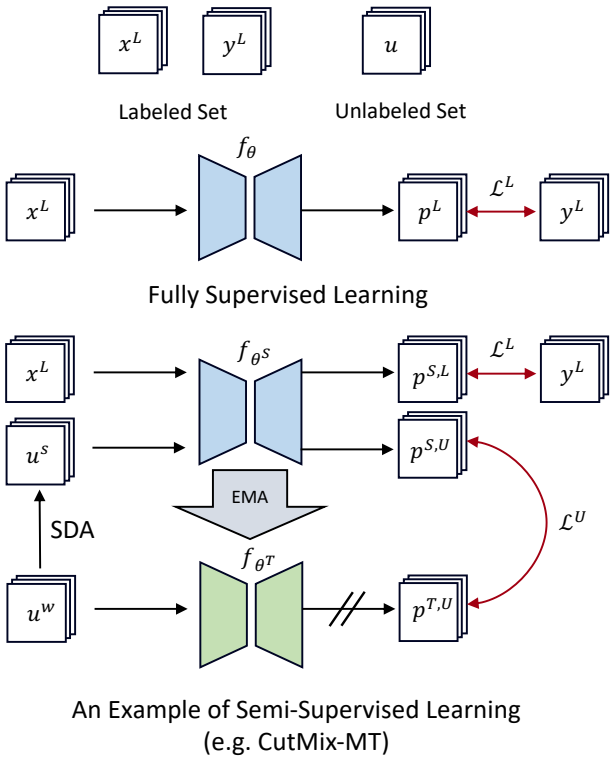


Figure 2.2: Illustration of the Teacher-Student framework for consistency regularization. The teacher generates pseudo-labels from weakly augmented inputs (u^w) to supervise the student processing strongly augmented inputs (u^s). Dashed lines indicate detached gradients.

Input perturbations transform images directly which is the most common approach [119, 134, 120, 121]. CutMix-MT [119] mixes rectangular regions between images, while ClassMix [134] uses semantic-aware masks that preserve object regions. AugSeg [120] applies intensive photometric and geometric transformations, demonstrating that stronger augmentations improve performance.

Feature perturbations introduce noise within networks. CCT [135] perturbs intermediate features across multiple auxiliary decoders, enforcing internal consistency. This encourages robust representations throughout the network hierarchy rather than just at the output.

Network perturbations leverage model diversity [136, 137, 136]. CPS [138] trains two networks with different initializations, using cross-pseudo-supervision where each network's predictions supervise the other. This exploits the complementary errors of

Figure 2.3: Bird's eye view (BEV) segmentation transforms multi-camera images (left) into a unified top-down semantic map (right).

differently initialized models.

Recent approaches have combined various perturbation strategies for further gains [33, 123]. The NRCR framework [125] identifies key mechanisms for noise robustness in consistency regularization: label refinement through confidence filtering, strong regularization via consistency, multi-view learning from different perturbations, robust loss functions, and selective sampling strategies. The multi-view learning principle—where different perturbations of the same input provide complementary supervision—can be extended beyond input-level perturbations to task-level perspectives. By introducing an auxiliary boundary detection task alongside semantic segmentation, we create another view of the same input data, where geometric and semantic interpretations provide mutual regularization. This multi-task perspective is particularly valuable at object boundaries where pseudo-labels are least reliable, as boundary detection offers complementary geometric supervision precisely where semantic predictions are most uncertain. Chapter 4 explores this in more detail.

2.4 Bird's Eye View Segmentation

2.4.1 Task and Methods

Bird's eye view (BEV) segmentation transforms multi-camera street-view images into a unified top-down semantic map, providing critical spatial understanding for autonomous navigation as shown in Fig. 2.3. Unlike perspective view segmentation where each camera

is processed independently, BEV segmentation must reason about 3D space from 2D observations, handle occlusions between views, and fuse information from multiple cameras into a coherent representation [16]. This task is formulated as multi-label pixel-wise classification, where each BEV grid cell can contain multiple semantic classes (*e.g.* a pedestrian crossing overlapping with drivable area), distinguishing it from standard semantic segmentation's mutually exclusive labels [139].

2.4.1.1 Camera-to-BEV Transformation Challenges

The fundamental challenge lies in the view transformation: converting perspective images with depth ambiguity into metrically accurate BEV representations. This ill-posed problem requires reasoning about 3D geometry from 2D projections, where a single image pixel corresponds to a ray of potential 3D points. Early geometric methods used Inverse Perspective Mapping (IPM), assuming flat ground planes to establish pixel-to-BEV correspondences [140]. While computationally efficient, IPM fails catastrophically on non-planar surfaces, producing severe distortions for elevated objects like vehicles or pedestrians.

The transition to learning-based methods addressed these limitations through data-driven 3D reasoning. Methods like PON [141] and VPN [142] emerged to learn implicit geometric transformations, using convolutional networks to map image features directly to BEV space.

2.4.1.2 Lifting-Based Methods

The breakthrough in multi-camera BEV segmentation came with lifting-based approaches that explicitly reason about depth. LSS (Lift-Splat-Shoot) [143] established the foundational paradigm: "lift" 2D features into 3D using predicted depth distributions and "splat" them into BEV by pooling. By predicting depth distributions rather than point estimates, LSS captures the inherent uncertainty in depth estimation from monocular images. Multi-camera fusion occurs naturally in 3D space, though overlapping regions and calibration errors introduce additional challenges.

BEVDepth [144] enhanced LSS with explicit depth supervision from LiDAR, demonstrating that accurate depth prediction significantly improves BEV segmentation. The method introduced camera-aware depth prediction that accounts for varying intrinsics across cameras. BEVFusion [139] unifies camera and LiDAR features in a shared BEV

space that fully preserves geometric and semantic information, enabling efficient sensor fusion. Furthermore, BEVFusion also improves camera-only BEV segmentation through improved BEV pooling.

2.4.1.3 Cross-View Attention and Transformer Methods

Parallel to lifting-based approaches, cross-view attention methods learn implicit geometric transformations through attention mechanisms. CVT (Cross-View Transformers) [145] proposed cross-view attention mechanism for learning BEV representations by using positional embeddings to map features across views without explicit geometric modeling.

GKT [146] improved CVT's geometry awareness by incorporating geometric priors into the attention mechanism. The method guides cross-view attention using camera geometry, reducing the learning burden and improving generalization. These geometric cues act as inductive biases that constrain the attention patterns to geometrically plausible correspondences.

BEVFormer [147] brought the transformer paradigm to BEV perception, using deformable attention to aggregate features from multi-camera images into BEV queries.

The multi-camera nature introduces unique complexities: features must be aligned across cameras with different intrinsics, overlapping views create redundancy that must be resolved, and weather or lighting conditions can affect cameras asymmetrically (*e.g.* sun glare on forward cameras while rear cameras remain unaffected). These challenges compound the difficulty of learning robust BEV representations.

2.4.2 Learning with Limited BEV Labels

Creating BEV annotations requires precise 3D localization and careful projection to the ground plane, making it substantially more expensive than perspective view labeling. A single nuScenes [28] keyframe requires annotating across six synchronized cameras with consistent 3D semantics—a process taking hours per frame. While semi-automatic approaches have been proposed, they still require significant human verification to ensure quality and annotating every scenario quickly becomes cumbersome [21, 20]. This annotation bottleneck severely limits the scale of BEV datasets and motivates research into learning with limited supervision.

The scarcity of BEV annotations contrasts sharply with the abundance of perspective view resources. While BEV annotation requires complex 3D reconstruction and multi-camera calibration, perspective view segmentation benefits from extensive datasets (Cityscapes, BDD100K) and mature pretrained models. This asymmetry presents an opportunity: perspective view models could potentially provide supervisory signals for BEV learning, leveraging the semantic correspondence between views despite the geometric transformation. While existing approaches have explored the use of pretrained backbones on 2D vision tasks for 3D perception tasks, leveraging these vision tasks as auxiliary signals for label scarce scenarios remain unexplored. This direction—using readily available perspective view resources as auxiliary supervision for annotation-scarce BEV tasks—forms the basis of the approach explored in Chapter 5.

2.4.2.1 Semi-Supervised BEV Segmentation

Despite extensive research in semi-supervised 2D segmentation (Section 2.3), semi-supervised learning for BEV segmentation remains largely unexplored. The challenge lies in the domain gap between readily available perspective view labels and required BEV annotations. Simply applying 2D semi-supervised methods fails to address the unique challenges of view transformation and multi-camera fusion.

SkyEye [148] explored self-supervised learning for monocular BEV segmentation, leveraging frontal-view video annotations. However, this approach is limited to voxel-based representations and requires complex multi-step training pipeline which includes training a depth estimation model separately. Concurrent work [149] proposed rotation-based augmentation for BEV semi-supervised learning which only works for monocular view settings.

The prior semi-supervised BEV segmentation methods are constrained to particular model architectures and introduce complex training pipelines that only work under the monocular view setting. PCT, in Chapter 5, aims to provide a simple model agnostic framework that aims to leverage the abundant perspective view resources to improve BEV segmentation under multi-camera setting.

2.4.2.2 Domain Adaptation for BEV

Domain shifts in BEV segmentation are particularly severe due to compounding effects: geographical differences in road layouts, weather variations affecting cameras asymmetrically, and dramatic lighting changes between day and night [23]. A model trained on daytime data from one city may fail catastrophically when deployed at night in another location.

DualCross [23] pioneered UDA for BEV segmentation through cross-modal knowledge distillation. A teacher model trained on source domain LiDAR data guides adaptation to the target camera domain. The two-stage training process first aligns features across modalities, then performs target-specific refinement. While effective, this approach requires expensive LiDAR data in the source domain, limiting applicability.

DA-BEV [150] is a concurrent work for camera-only domain adaptation particularly aimed at 3D object detection. The method applies query-based adversarial learning for image and BEV features for regularization.

Similar to semi-supervised BEV segmentation, domain adaptation for BEV remains underexplored. In Chapter 5, this thesis proposes a simple yet effective framework that leverages off-the-shelf pretrained perspective view models to facilitate domain adaptation for BEV segmentation without requiring additional sensors. Furthermore, it is a training algorithm and does not impose any architectural constraints making it widely applicable to multiple BEV architectures.

2.5 Datasets and Evaluation Metrics

2.5.1 Key Datasets

2.5.1.1 Semantic Segmentation Datasets

Cityscapes [3] serves as the primary benchmark for urban scene understanding, containing 5,000 finely annotated images (2,975 training, 500 validation, 1,525 test) with 19 semantic classes. The dataset provides additional coarse annotations for 20,000 images, though these are rarely used in practice due to quality inconsistencies.

BDD100K [18] offers unprecedented scale and diversity with 100,000 driving videos capturing varied weather conditions, times of day, and geographic locations. The semantic

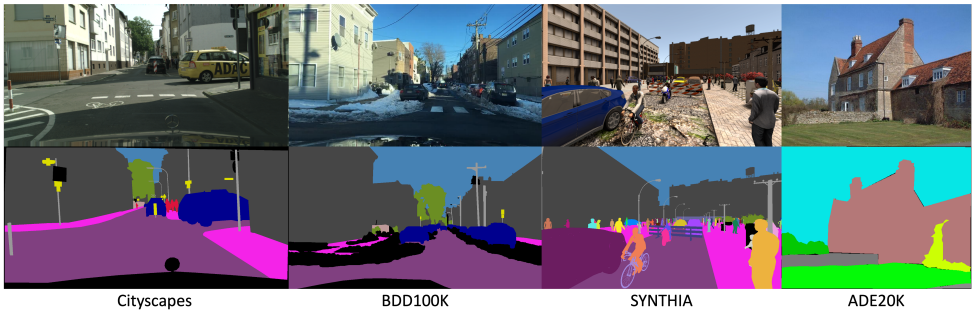


Figure 2.4: Samples from segmentation datasets used in this thesis: Cityscapes, BDD100K, SYNTHIA, and ADE20K (left to right).

segmentation subset contains 10,000 images with 19 classes compatible with Cityscapes, enabling cross-dataset evaluation. This diversity makes BDD100K particularly valuable for domain adaptation studies and robust model training.

SYNTHIA [151] provides synthetic data with pixel-perfect annotations generated from computer graphics. The RAND subset contains 13,400 images with instance-level annotations, offering precise boundaries unachievable through manual annotation. This synthetic nature enables controlled experimentation with boundary quality, as annotation noise is eliminated.

Pascal VOC 2012 [152] and **ADE20K** [26] represent general scene understanding benchmarks. Pascal VOC contains 10,582 images across 21 classes with notable boundary annotation inconsistencies. ADE20K provides 20,210 training and 2,000 validation images spanning 150 fine-grained categories, testing scalability to complex taxonomies.

2.5.1.2 Bird's Eye View Datasets

nuScenes [28] established the standard for multi-camera BEV perception, providing 1,000 scenes (700 training, 150 validation, 150 test) with synchronized 6-camera coverage. Each keyframe includes 3D bounding boxes and semantic maps projected to BEV space. The dataset's structured splits enable standardized evaluation for semi-supervised learning and domain adaptation, with established protocols for day/night and city-to-city shifts.

Argoverse2 [34] scales BEV perception with 1,000 sequences and high-definition maps across six cities. The sensor suite includes seven ring cameras providing 360° coverage with higher resolution than nuScenes. Geographic diversity enables challenging cross-city domain adaptation experiments.

2.5.2 Evaluation Metrics

2.5.2.1 Standard Segmentation Metrics

The mean Intersection over Union (mIoU) serves as the primary metric for semantic segmentation, measuring the overlap between predicted and ground-truth regions:

$$\text{IoU}_c = \frac{|P_c \cap G_c|}{|P_c \cup G_c|} = \frac{TP_c}{TP_c + FP_c + FN_c} \tag{2.1}$$

where P_c and G_c denote predicted and ground-truth pixels for class c, with the mean computed as:

$$\text{mIoU} = \frac{1}{C} \sum_{c=0}^{C-1} \text{IoU}_c \tag{2.2}$$

While mIoU effectively captures overall segmentation quality, it exhibits known insensitivity to boundary errors [153, 38]. Additional metrics like Region Over-segmentation Measure (ROM) and Region Under-segmentation Measure (RUM) [154] quantify segmentation fragmentation but are omitted here for brevity.

2.5.2.2 Boundary-Specific Metrics

Semantic Boundary Detection uses the mean F-score (mF) at optimal dataset scale (ODS), evaluating each class boundary separately [81, 80, 82]:

$$\text{mF} = \frac{1}{C} \sum_{c=0}^{C-1} \max_t F_c(t) \tag{2.3}$$

where $F_c(t)$ is the F-score (harmonic mean between precision and recall) for class c at threshold t, optimized per-class. The evaluation uses a computationally expensive bipartite matching algorithm to strictly associate predicted and ground-truth boundary pixels. For ODS, we iterate over all possible threshold values for each class and apply a single chosen threshold to all images in the dataset. The threshold that yields the highest F-score for each class is selected, and the mean is then computed across classes.

Boundary F1 Score (BF1) evaluates boundary quality of *segmentation predictions* by measuring precision and recall with a distance tolerance [153]:

$$\text{Precision} = \frac{1}{|P_b|} \sum_{p \in P_b} \mathbb{1}[\min_{g \in G_b} d(p, g) < \theta] \tag{2.4}$$

$$\text{Recall} = \frac{1}{|G_b|} \sum_{g \in G_b} \mathbb{1}[\min_{p \in P_b} d(p, g) < \theta] \tag{2.5}$$

where P_b and G_b are predicted and ground-truth boundary pixels, $d(\cdot, \cdot)$ is Euclidean distance, and θ is the tolerance threshold (typically 3 pixels). The F-score combines these as:

$$\text{BF}_\theta = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.6}$$

Morphological dilation is used for efficiency instead of the computationally expensive bipartite matching.

Another F-score-based boundary evaluation, referred to as the **Boundary F-score**, was originally introduced for the video object segmentation task on the DAVIS dataset [155]. The evaluation protocol is conceptually similar, but the underlying implementation differs significantly, so the two metrics do not necessarily produce identical values. The Boundary F-score was later popularized in semantic segmentation through the evaluation of GSCNN [98], and methods comparing against GSCNN commonly adopt the same implementation for consistency.

Chapter 3 uses the boundary F-score to stay consistent with [98], while Chapter 4 uses the more efficient BF1 implementation introduced in [156].

Boundary IoU (BIoU) or Trimap IoU provide computationally efficient boundary evaluation [156]. BIoU restricts IoU computation to boundary regions:

$$\text{BIoU} = \frac{1}{C} \sum_{c=0}^{C-1} \frac{\sum_{(i,j) \in B_k(G)} [P_{c,i,j} \wedge G_{c,i,j}]}{\sum_{(i,j) \in B_k(G)} [P_{c,i,j} \vee G_{c,i,j}]} \tag{2.7}$$

where $B_k(G) = G \oplus \text{MinPool}_k(G)$ defines the boundary mask via morphological operations where k is the tolerance threshold. More details are in Sec. 4.4.1.3.

2.5.2.3 BEV Segmentation Metrics

BEV segmentation employs multi-label formulation where pixels can belong to multiple classes (*e.g.* vehicles on drivable area). Each class is evaluated independently using binary IoU:

$$\text{IoU}_c = \frac{\sum_{i,j} \hat{Y}_{c,i,j} \cdot Y_{c,i,j}}{\sum_{i,j} \max(\hat{Y}_{c,i,j}, Y_{c,i,j})} \tag{2.8}$$

where $\hat{Y}, Y \in \{0, 1\}^{H \times W \times C}$ are predicted and ground-truth binary masks. The final metric averages across classes: $\text{mIoU}_{\text{BEV}} = \frac{1}{C} \sum_{c=0}^{C-1} \text{IoU}_c$.

2.6 Positioning of the methods in this thesis

Semantic-Boundary-Conditioned Backbone (SBCB) framework (Chapter 3) provides a systematic empirical analysis of semantic boundary supervision in fully-supervised settings. While semantic boundaries have been explored in prior joint learning methods like RPCNet [103] and CSEL [102]—which achieve higher accuracy through complex bidirectional refinement—SBCB demonstrates that even simple auxiliary supervision during training can yield consistent improvements. Unlike GSCNN's permanent architectural modifications [98] or SegFix's post-processing overhead [105], SBCB operates purely at training time with zero inference cost. The framework's value lies not in architectural novelty but in establishing that conditioning backbones with boundary signals provides reliable gains across diverse architectures, from CNNs to Vision Transformers.

BoundMatch (Chapter 4) introduces dual-task consistency regularization for semi-supervised segmentation, where both segmentation and boundary detection undergo independent consistency regularization. This multi-task perspective aligns with NRCR's analysis [125] that multi-view learning—where different interpretations of the same data provide complementary supervision—enhances noise robustness in consistency regularization. Unlike CFCG [123] and CW-BASS [115] which derive boundaries from noisy pseudo-labels, or BoundaryMatch [157] which still depends on segmentation outputs, BoundMatch learns boundaries independently from hierarchical features while applying consistency regularization to both tasks. The bidirectional fusion modules (BSF and SGF) enable mutual refinement between tasks, creating a framework where boundary and segmentation provide distinct yet complementary regularization signals. This dual-task consistency approach proves particularly effective under severe label scarcity, where the additional regularization from the auxiliary task helps prevent overfitting.

Perspective Cue Training (PCT) framework (Chapter 5) addresses both semi-supervised learning (SSL) and unsupervised domain adaptation (UDA) for multi-camera BEV segmentation—settings where BEV annotations are scarce or entirely absent. Despite the critical importance of BEV perception, prior work on semi-supervised BEV segmentation remains extremely limited, with existing methods constrained to monocular settings or specific architectures. While SkyEye [148] and X-Align [158] also utilize pretrained segmentation models, they focus on fully-supervised or self-supervised monocular scenarios. PCT systematically explores how pretrained perspective view models—trained on

abundant PV datasets—can serve as auxiliary teachers for multi-camera BEV learning. Unlike DualCross [23] which requires expensive LiDAR supervision or DA-BEV's complex adversarial training [150], PCT demonstrates that off-the-shelf PV models provide effective auxiliary supervision through simple multi-task learning. This represents the first comprehensive study of SSL for multi-camera BEV segmentation, showing that pretrained models can substantially improve performance in both semi-supervised and domain adaptation scenarios.

Together, these three frameworks explore complementary aspects of auxiliary supervision from existing resources. SBCB establishes the empirical value of boundary conditioning in fully-supervised settings, demonstrating consistent if modest gains through training-time auxiliary tasks. BoundMatch extends this principle to label-scarce scenarios through dual-task consistency regularization, where independent boundary learning provides an additional regularization view that enhances noise robustness. PCT shifts from within-task to cross-task auxiliary supervision, systematically investigating how pretrained PV models can address the severe annotation scarcity in BEV segmentation across both SSL and UDA settings. While each framework addresses different challenges, they converge on a common principle: existing resources—whether they are segmentation masks that can be reinterpreted, pretrained models that can be repurposed, or vast amounts of unlabeled data—contain valuable signals that, when properly extracted, enhance urban scene understanding without prohibitive costs.

3

Conditioning the backbone with semantic boundaries

This chapter presents the first of three studies exploring auxiliary supervision from existing resources, focusing on the fully-supervised setting where all training data has complete annotations. We demonstrate that semantic boundaries, which can be automatically derived from existing segmentation masks at zero additional annotation cost, provide complementary supervision that systematically improves segmentation quality across diverse architectures. This study establishes the foundational principle that existing annotations contain derivable structural cues that, when properly extracted and utilized, enhance the primary task without requiring additional labeling effort or inference overhead.

3.1 Introduction

Sec. 1.2.2 identified a fundamental tension in urban scene understanding: while segmentation accuracy in object interiors exceeds 95%, performance at boundaries—the critical regions for downstream decision-making—drops below 70% [38]. This boundary imprecision manifests as fragmented pedestrian detections, uncertain vehicle delineations, and ambiguous road edges that can compromise system safety. The challenge is particularly acute given the deployment constraints outlined in Sec. 1.2.3, where solutions must balance accuracy improvements with computational efficiency.

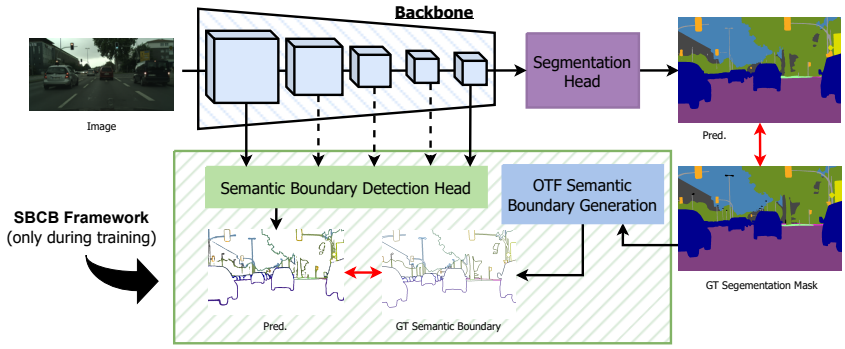


Figure 3.1: Overview of the SBCB framework. The SBD head is integrated into the backbone during training, supervised by ground-truth boundaries generated on-the-fly from segmentation masks. The SBD head is removed at inference.

This chapter addresses the first research question posed in Sec. 1.3: *Can auxiliary supervision derived from existing annotations improve segmentation performance without requiring additional labeling effort?* We investigate this through the lens of semantic boundary detection (SBD), recognizing that every segmentation mask inherently contains boundary information that current training paradigms discard. The key insight is that boundaries represent a complementary view of the same annotation—while segmentation focuses on regional classification, boundaries explicitly model the transitions between classes where uncertainty is highest.

We present the Semantic Boundary Conditioned Backbone (SBCB) framework, which leverages this complementarity through hierarchical multi-task learning. Unlike the boundary-aware methods surveyed in Sec. 2.2.2 that require architectural modifications or post-processing, SBCB operates purely as a training-time auxiliary task, as shown in Fig. 3.1. The framework conditions the backbone network to learn boundary-aware features by jointly optimizing for segmentation and semantic boundary detection, where the latter is supervised by boundaries generated on-the-fly from existing segmentation masks. Critically, during inference, the boundary detection head can be removed entirely, resulting in zero additional computational overhead—a key consideration for the deployment scenarios discussed in Sec. 1.2.3.

Through extensive experiments on urban driving datasets including Cityscapes [3], BDD100K [18], and SYNTHIA [151], we demonstrate that SBCB consistently improves segmentation quality with average gains of 1.2% mIoU and 2.6% boundary F-score. The

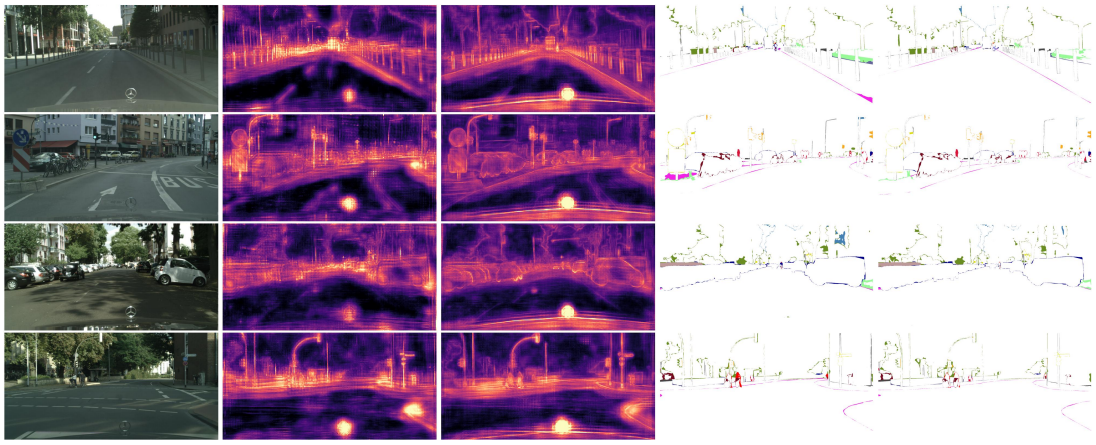


Figure 3.2: Visualization of backbone features (L_2 norm) and segmentation errors for DeepLabV3+ with and without SBCB. Columns from left: input, features without SBCB, features with SBCB, errors without SBCB, errors with SBCB. SBCB produces more boundary-aware features and reduces errors near boundaries.

contributions of this work are:

- A training framework that systematically improves segmentation through auxiliary boundary supervision derived entirely from existing annotations
- Extensive analysis revealing how boundary conditioning, especially semantic boundary conditioning, enhances backbone feature representations, particularly around object edges, as illustrated in Fig. 3.2
- Comprehensive validation across diverse architectures and datasets, demonstrating the generality of boundary conditioning
- An efficient on-the-fly boundary generation algorithm that integrates seamlessly with modern training pipelines
- Open-source codebase for reproducibility [1].

This chapter provides empirical evidence that auxiliary supervision with boundaries provides benefits to semantic segmentation, showcasing that simply reinterpreting segmentation masks as semantic boundaries can yield significant performance gains without additional annotation costs or inference overhead.

[1]Source code: `https://github.com/haruishi43/boundary_boost_mmseg`

3.2 Technical Context and Positioning

Having established the comprehensive background in Chapter 2, this section positions SBCB within the landscape of boundary-aware segmentation methods and identifies the specific technical gaps it addresses. Rather than repeating the literature survey, we focus on three critical aspects: the unexploited potential in existing approaches, the architectural considerations that enable our framework, and our distinct positioning relative to prior work.

3.2.1 Gaps in Existing Approaches

The survey in Sec. 2.2.2 reveals three categories of boundary-aware methods: architectural fusion approaches like GSCNN [98], post-processing methods like SegFix [105], and implicit loss-based techniques [108]. While these methods demonstrate the value of boundary information, they leave critical gaps that SBCB addresses.

First, architectural fusion approaches (Sec. 2.2.2.1) require permanent modifications to the segmentation architecture. GSCNN's shape stream with Canny Edge auxiliary feature decreases the FPS from 12.3 down to 8.4 compared to the DeepLabV3+ baseline. This significantly hinders the deployment constraints identified in Sec. 1.2.3, where inference efficiency is paramount. Similarly, DecoupleSegNet [99] requires specialized decoders that cannot be easily integrated into existing deployed models. Recent performant methods like CSEL [102] utilize three-branch networks. While these methods improve segmentation, they do so at the cost of increased complexity and inference time.

Second, post-processing methods (Sec. 2.2.2.2) operate on fixed segmentation outputs, unable to influence the feature learning process. SegFix [105], while model-agnostic, requires training a separate refinement network and adds additional inference costs. Implicit methods like ABL [108] introduce a boundary-aware loss function, but the effectiveness is often lower compared to auxiliary supervision. For example, GSCNN [98] improves boundary F-score by 3.9%, while ABL only achieves a 1.4% improvement over the baseline. More fundamentally, these methods cannot address the root cause—that the backbone features themselves lack boundary awareness.

Third, existing multi-task learning approaches in segmentation typically use auxiliary tasks that are semantically distant from the primary objective. As shown in the Taskonomy study [97], task affinity matters significantly for positive transfer. While PSPNet's auxiliary

FCN head [59] provides additional supervision, it essentially duplicates the primary task rather than providing complementary information. SBCB leverages the natural complementarity between regional segmentation and boundary detection, two faces of the same underlying scene structure.

3.2.2 Architectural Considerations

The effectiveness of SBCB rests on two key architectural insights derived from the edge detection literature (Sec. 2.1.2) and modern segmentation architectures (Sec. 2.1.1).

Hierarchical Feature Utilization: The success of HED [80] and CASENet [82] demonstrated that boundaries manifest at multiple scales—fine details in early layers and semantic understanding in deeper layers. Modern segmentation backbones naturally produce such hierarchical features, yet popular methods utilize only the final representations. SBCB exploits all hierarchical levels through the generalized SBD head, ensuring that early layers maintain edge sensitivity while deeper layers learn boundary semantics—similar to deep supervision.

Training-Test Asymmetry: A critical insight from auxiliary task learning [96] is that training and inference objectives need not align. While joint training shapes the learned representations, the auxiliary components can be discarded at test time. This principle, underutilized in segmentation, enables SBCB to gain the benefits of multi-task learning without inference penalties. The boundary detection head acts as scaffolding during construction—essential for shaping the structure but removable once complete.

3.2.3 Positioning of SBCB

SBCB occupies a unique position in the design space of boundary-aware segmentation methods. Unlike the architectural modifications reviewed in Sec. 2.2.2.1, SBCB requires no permanent changes to the segmentation model. Unlike the post-processing methods in Sec. 2.2.2.2, it directly influences feature learning. Unlike generic auxiliary tasks, it leverages the proven complementarity between segmentation and boundary detection [97].

Most critically, SBCB represents a *framework* rather than a specific architecture. While GSCNN, DecoupleSegNet, and others propose particular network designs, SBCB provides a training methodology applicable to any hierarchical backbone. This generality

is demonstrated through successful application to CNNs (ResNet, HRNet, ConvNeXt), mobile architectures (MobileNet, BiSeNet), and Vision Transformers (SegFormer).

The framework also addresses a practical consideration often overlooked in academic work: integration with existing deployed systems. Organizations with mature segmentation pipelines cannot easily adopt new architectures requiring retraining, revalidation, and recertification. SBCB allows these systems to improve their existing models through retraining alone, maintaining architectural compatibility while enhancing performance.

This positioning—as a general, efficient, training-time framework—distinguishes SBCB from prior work and establishes the foundation for its extensions to semi-supervised (Chapter 4) settings, where the flexibility to work with diverse architectures and data scenarios becomes even more valuable.

3.3 Approach

The Semantic Boundary Conditioned Backbone (SBCB) framework, as illustrated in Fig. 3.1, enhances semantic segmentation by incorporating a semantic boundary detection (SBD) head into the backbone network during training. This SBD head receives multi-scale features from selected stages of the backbone and is supervised using ground-truth (GT) semantic boundaries, dynamically generated on-the-fly using the GT segmentation masks. Remarkably, during inference, when the task does not require semantic boundary information, the SBD head can be omitted, resulting in a semantic segmentation model with no increase in parameters.

To introduce the SBCB framework thoroughly, we explain the systematic approach in the following sections. In Sec. 3.3.1, we comprehensively review existing SBD architectures while introducing the specific SBD heads utilized in our experiments. Moving forward, in Sec. 3.3.2, we delve into the details of the framework by applying the SBCB approach to two prominent backbone networks, namely DeepLabV3+ and HRNet. In Sec. 3.3.3, we elucidate the On-The-Fly (OTF) semantic boundary generation module, which is a pivotal component, endowing this framework with remarkable flexibility and ease of use. Finally, in Sec. 3.3.4, we expound upon the loss function employed within the SBCB framework, which plays a crucial role in optimizing and training the model effectively.

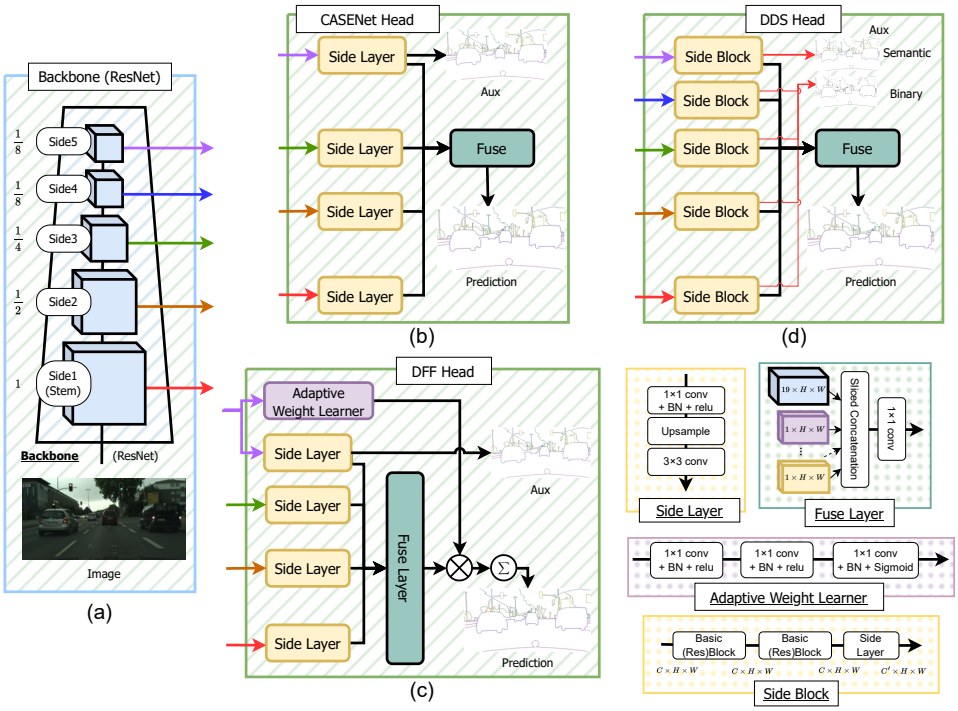


Figure 3.3: Overview of SBD architectures: (a) backbone multi-level features, (b) CASENet with Side Layers and Fuse Layer, (c) DFF with Adaptive Weight Learner, (d) DDS with deeper Side Blocks and deep supervision.

3.3.1 Semantic Boundary Detection Heads

This section presents an overview of significant SBD models based on Convolutional Neural Networks (CNN) that have emerged over the years. Understanding these SBD heads is crucial for comprehending their application in the SBCB framework and the conducted experiments. Additionally, we highlight some effective modifications we have made during our reimplementation. Furthermore, we introduce the "Generalized" versions of these SBD heads, which are utilized in the SBCB framework.

CASENet. The CASENet architecture [82], proposed by Yu et al., presents a novel nested design without deep supervision on ResNet [159]. The architecture, shown in Fig. 3.3b, modifies the ResNet backbone to capture higher-resolution features (detailed in Sec. 3.5.10). In each stage of the backbone (excluding stage 4), the features are passed into the Side Layer, comprising a 1×1 convolutional kernel followed by a deconvolutional layer, increasing the resolution to match the input image. Throughout this chapter, we interchangeably use the terms "Stage" and "Side." "Side" is often used in SBD-related

literature, which includes the Stem. The last Side Layer (Side 5) produces an $N_{cat} \times H \times W$ tensor, while the other Side Layers (Side 1 to 4) generate $1 \times H \times W$ outputs, where N_{cat} is the number of categories, and H and W are the height and width of the image. The outputs of the Side Layers are then processed by a Fuse layer, which performs sliced concatenation of each feature, resulting in a $(4 \times N_{cat}) \times H \times W$ feature. This feature is further processed by a 1×1 convolution kernel to produce an $N_{cat} \times H \times W$ logit, supervised using the ground truth semantic boundaries. Additionally, the output of the last Side Layer is also supervised with the same ground truth semantic boundaries, acting as an auxiliary signal. Further details on the semantic boundary supervision loss $\mathcal{L}_{\text{SBD}}$ for the Fuse Layer and the last Side Layer are explained in Sec. 3.3.4.

In our implementation, we observed checkerboard artifacts in the original Side Layer outputs. To address this, we replaced the Side Layers with bilinear upsampling followed by a 3×3 convolutional kernel, as depicted in Fig. 3.3. This modification was adapted from techniques introduced for generative models using deconvolution [160], and we ensured it did not increase the number of parameters.

DFF. The DFF architecture, introduced in [83], enhances the CASENet model by incorporating the Adaptive Weight Learner. This addition refines the output of the Fuse layer using attentive weights. As depicted in Fig. 3.3c, the Fuse layer produces sliced concatenated features. Instead of using a standard 1×1 convolutional kernel, the Adaptive Weight Learner calculates weights, which are then applied to the tensor and summed to produce an output tensor of size $N_{cat} \times H \times W$.

DDS. DDS [84] is the latest method that surpasses CASENet and DFF. It introduces a deeper Side Block composed of two ResNet Basic Blocks followed by a Side Layer. Fig. 3.3d shows an overview of the network. Unlike CASENet, DDS explicitly supervises all Side Blocks, with the final output supervised by semantic boundaries and earlier outputs supervised by binary boundaries.

Generalized SBD heads. To enable seamless integration within the SBCB framework, we introduce a generalized SBD head that can be applied to diverse backbone networks and segmentation architectures. This SBD head, referred to as the Generalized SBD head, is illustrated in Fig. 3.4. Within our framework, we achieve this generalization by incorporating flexible Side and Fuse layers, accommodating any of the previously mentioned SBD heads (CASENet, DFF, and DDS). The Side Layer can be adapted from CASENet's Side Layers or DDS's Side Blocks, while the Fuse Layer can take the form

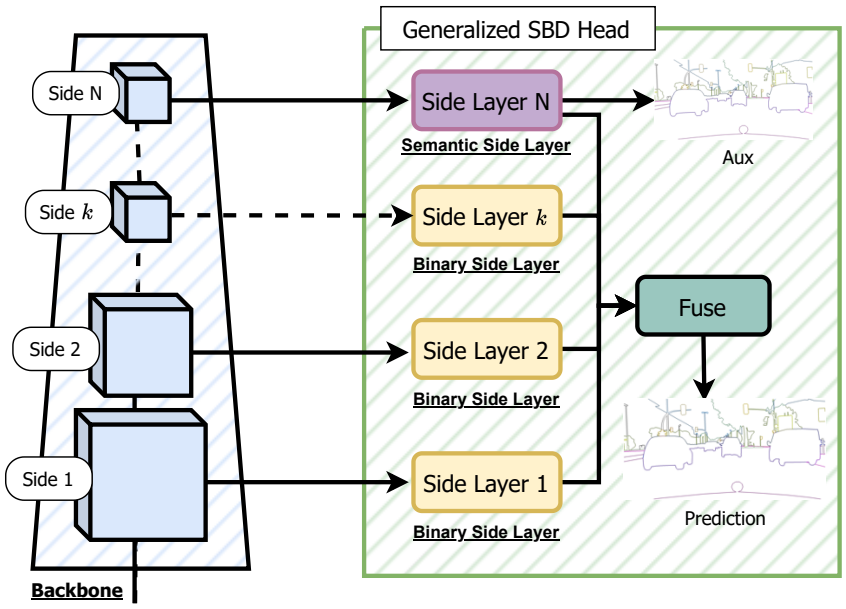


Figure 3.4: Overview of the Generalized SBD Head, extending CASENet to accommodate varying numbers of sides. The last Side Layer (Semantic Side) outputs N_{cat} channels; earlier Side Layers (Binary Sides) output single channels.

of either CASENet's Fuse Layer or DFF's Fuse Layer with Adaptive Weight Learner. Moreover, our approach allows for the manipulation of the number of Sides, offering versatility to the framework. Specifically, in DDS, the Nth side output is supervised using semantic boundaries, while binary boundaries supervise the earlier side outputs. This adaptability empowers the Generalized SBD head to seamlessly integrate with different segmentation architectures and backbones, providing enhanced flexibility for the SBCB framework.

As a formal definition, the features obtained from the kth stage backbone is $\mathbf{S}_k$ and the features obtained from the kth Side Layer is $\mathbf{B}_k$. S_{SBD} represents a set of semantic boundary predictions, and S_{Bin} represents a set of binary boundary predictions. For CASENet and DFF, $S_{\text{SBD}} = \{\mathbf{B}_N, \mathbf{B}_{\text{fuse}}\}$, where $\mathbf{B}_N$ represents the last side output and $\mathbf{B}_{\text{fuse}}$ represents the final fused prediction as shown in Fig. 3.4. For DDS, we supervise $S_{\text{SBD}} = \{\mathbf{B}_N, \mathbf{B}_{\text{fuse}}\}$ and $S_{\text{Bin}} = \{\mathbf{B}_k, \dots \mathbf{B}_2, \mathbf{B}_1\}$. Concretely, the Generalized SBD head

can be defined as follows:

$$\{\mathbf{S}_N, \ldots, \mathbf{S}_k, \ldots, \mathbf{S}_1\} = \text{Backbone}(\mathbf{I}) \tag{3.1}$$

$$\mathbf{B}_k = \text{SideLayer}_k(\mathbf{S}_k) \tag{3.2}$$

$$\mathbf{B}_{\text{fuse}} = \text{Fuse}(\{\mathbf{B}_N, \ldots, \mathbf{B}_k, \ldots, \mathbf{B}_1\}), \tag{3.3}$$

where $\mathbf{I}$ is the input image.

3.3.2 SBCB Framework

In this section, we will demonstrate the application of the SBD heads we reviewed in Sec. 3.3.1 within the SBCB framework. As mentioned earlier in Sec. 3.3, the SBCB framework incorporates an SBD head into the backbone, with each multi-scale feature directed to different Side Layers. Certain crucial factors need to be considered to ensure the ease of implementation across various backbones.

Is the feature with the largest resolution passed to the first Side Layer? To capture boundary details effectively, the first Side Layer must receive features with the largest resolution. Hence, we follow the SBD architecture convention and utilize the backbone's stem if possible ($\mathbf{B}_1$). Generally, a feature resolution of 1 or 1/2 of the input resolution suffices for the first Side Layer.

Which backbone features to pass to which Side Layer? When applying the SBCB to hierarchical backbones like ResNet, earlier stages ($\mathbf{B}_1 \sim \mathbf{B}_{N-1}$) are best suited for the binary side layers, while the last stage ($\mathbf{B}_N$) naturally fits the semantic side layer. Fortunately, most semantic segmentation architectures utilize hierarchical backbones, making the application of the SBCB framework straightforward. In cases such as HRNet, where features are hierarchical and branching out, it is crucial to incorporate all the features, typically by concatenating them.

Do the Side Layers receive features with sufficient resolution? Although semantic segmentation models generally work with higher input resolutions, some backbones may reduce feature resolution excessively. In such cases, it is beneficial to adjust the convolutional kernel's strides and dilations to increase the feature resolution. The goal is to ensure that the first side feature has a resolution of at least 1/2 of the input image. This technique, known as the "backbone trick," is discussed in detail in Sec. 3.5.10.

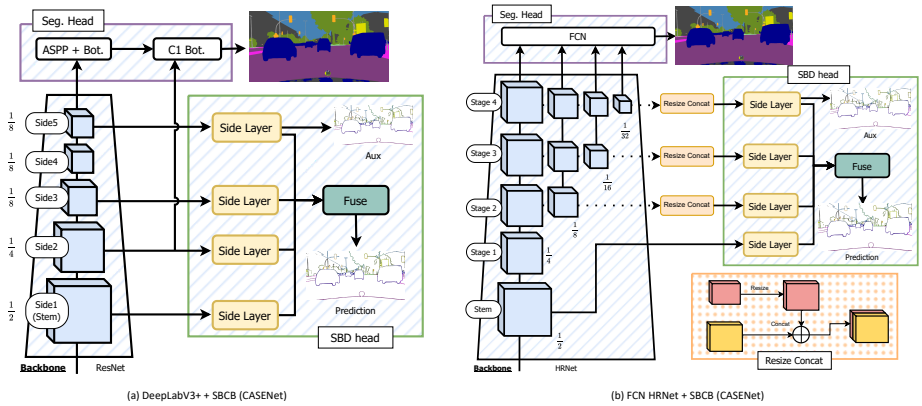
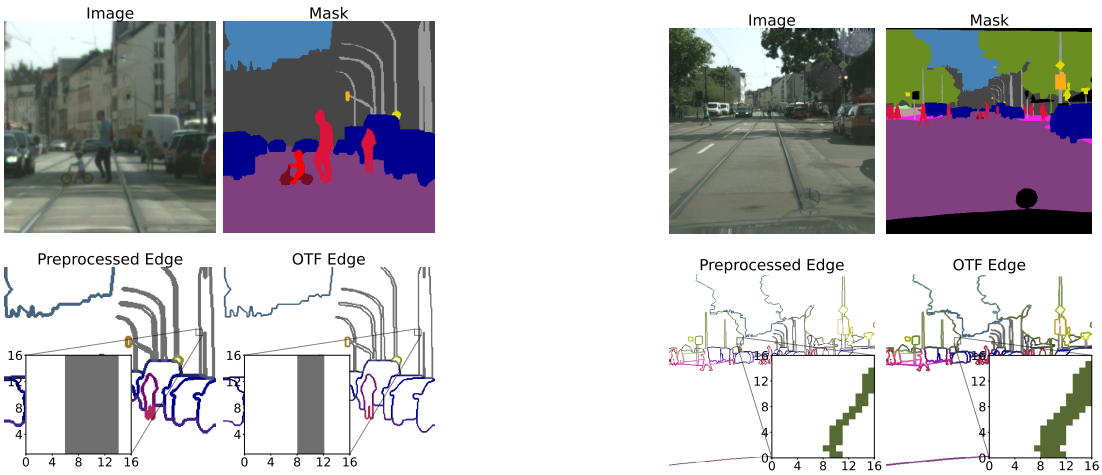


Figure 3.5: SBCB framework applied to (a) DeepLabV3+ and (b) HRNet with FCN head.

To enhance the comprehensiveness of the framework, we will present case studies demonstrating the application of the SBCB framework to popular architectures like DeepLabV3+ and HRNet. Furthermore, we will showcase how the SBCB framework can be seamlessly applied to other architectures, including those with heavily customized backbones. Detailed implementations and results of these case studies are provided in Sec. 3.6.

DeepLabV3+ + SBCB. Fig. 3.5a illustrates the CASENet head applied to DeepLabV3+. The architecture follows a similar design as the SBD architectures with ResNet, and we utilize the "backbone trick" when dealing with small input image sizes. Implementing various SBD heads on DeepLabV3+ is generally straightforward, and we can readily incorporate the DDS head by incorporating Side 4 features and replacing the Side Layers with Side Blocks.

HRNet + SBCB. The HRNet backbone consists of four stages, as depicted in Fig. 3.5b. Since the first stage already reduces the resolution to 1/4, we utilize the features from the stem for the first Side Layer. Unlike ResNet, HRNet maintains consistent feature resolutions across its stages while branching out into smaller resolutions in each stage. To effectively incorporate these features, we resize and concatenate the features from each stage before passing them through the Side Layer. By including all the features from each stage, we aim to achieve improved conditioning of the backbone for enhanced performance.



(**a**) Data rescaled by a factor of 2.

(**b**) Data rescaled by half.

Figure 3.6: Comparison of preprocessed boundaries (left) versus OTFGT boundaries (right) under different rescaling. OTFGT maintains consistent boundary widths regardless of scale.

3.3.3 On-the-fly Ground Truth Generation

For the SBD and edge detection tasks, boundaries are manually annotated by human annotators. In some datasets, like Cityscapes, automatic preprocessing scripts are provided to generate GT boundaries from semantic and optionally instance masks. These boundaries are generated before training and remain unchanged during training. On the other hand, for semantic segmentation tasks, it is a common practice to resize and rescale the GT masks during training to mitigate overfitting and introduce variations to the dataset. However, resizing the boundaries can result in inconsistent boundary widths, as illustrated in Fig. 3.6. This will lead to the model learning inconsistent boundary widths, which is undesirable.

To address this issue, we developed a straightforward semantic boundary generation algorithm called the on-the-fly (OTF) semantic boundary GT generation module (OTFGT), as illustrated in Fig. 3.7. This module takes a GT semantic segmentation mask $\mathbf{M}^{\mathrm{GT}}$ as input and produces a semantic boundary mask $\mathbf{B}^{\mathrm{GT}}$. For each category $c \in C$, where $\mathbf{M}_c^{\mathrm{GT}}$ is a binary 2D array representing the category, we calculate a binary 2D array of boundaries $\mathbf{B}_c^{\mathrm{GT}}$ using the equation:

$$\mathbf{B}_c^{\mathrm{GT}} = \mathrm{Thresh}_r(\mathrm{DT}(\mathbf{M}_c^{\mathrm{GT}}) + \mathrm{DT}(1 - \mathbf{M}_c^{\mathrm{GT}})). \tag{3.4}$$

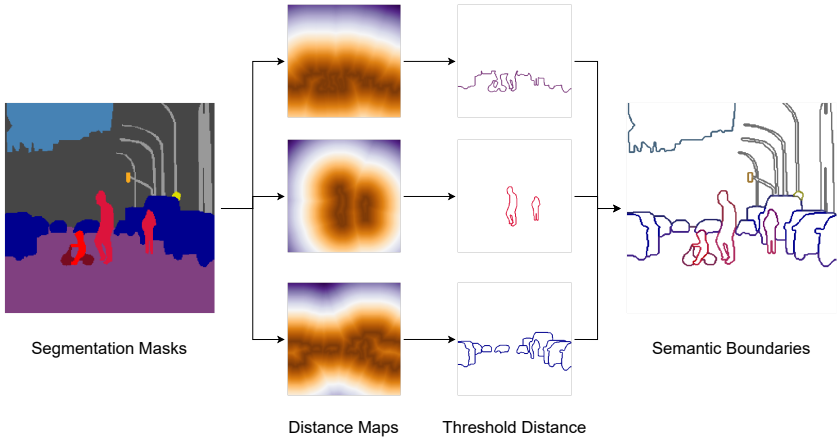


Figure 3.7: Overview of the OTFGT module. Distance transforms are applied to segmentation masks to obtain boundary maps, which are thresholded and concatenated to form the supervision tensor.

Here, DT is a Euclidean distance transform function that computes L_2 norm for each binary pixel using [161]. We obtain the outer distance with $\mathrm{DT}(\mathbf{M}_c^{\mathrm{GT}})$ and the inner distance with $\mathrm{DT}(1-\mathbf{M}_c^{\mathrm{GT}})$. We then add the two distances to acquire the distances from the mask boundaries. Thresh_r function thresholds the distance based on the radius r with the following condition:

$$\mathrm{Thresh}_r(d) = \begin{cases} 1 & d(i,j) \leq r \\ 0 & d(i,j) > r \end{cases}. \tag{3.5}$$

We repeat the algorithm to generate semantic boundaries of all categories C. For further details and Python code snippets, please refer to Sec. A.1.

3.3.4 Loss Functions.

The model generates segmentation and boundary maps with pre-defined semantic categories from an input image. For the segmentation map, we apply cross-entropy (CE) loss, denoted as $\mathcal{L}_{\mathrm{Seg}}$, on each pixel. As for the SBD head, binary cross-entropy (BCE) loss, $\mathcal{L}_{\mathrm{SBD}}$, is applied for multi-label boundaries S_{SBD}, following the approach in [82]. While CASENet and DFF utilize only multi-label boundaries for supervision, DDS introduces deep supervision of edges by supervising earlier side outputs (S_{Bin}) with binary boundary maps using BCE loss, $\mathcal{L}_{\mathrm{Bdry}}$ [84].

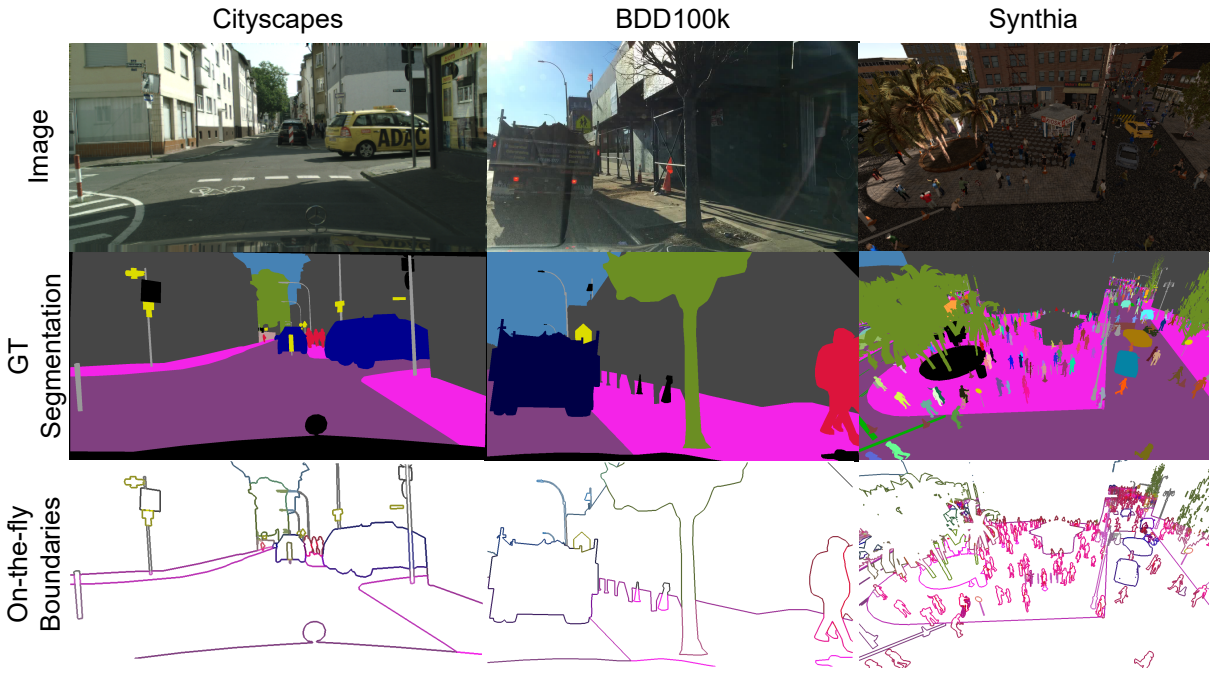


Figure 3.8: Sample images, segmentation masks, and OTF-generated semantic boundaries for Cityscapes, BDD100K, and Synthia datasets.

The overall loss function used is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{Seg}} + \alpha \sum^{S_{\text{SBD}}} \mathcal{L}_{\text{SBD}} + \beta \sum^{S_{\text{Bin}}} \mathcal{L}_{\text{Bdry}}, \tag{3.6}$$

where α and β are constants that balance the effects of losses from each task. We use the GT boundaries obtained from OTFGT ($\mathbf{B}^{\text{GT}}$) to supervise the boundaries.

3.4 Experiment Setup

3.4.1 Datasets

We employ four benchmark datasets—Cityscapes, BDD100K, SYNTHIA, and ADE20K—previously introduced in Sec. 2.5.1.1.

Cityscapes [3] serves as our primary benchmark for both semantic segmentation and semantic boundary detection. We primarily report quantitative evaluations on the validation set.

BDD100K [18] provides diverse driving scenes with label definitions consistent with Cityscapes. We use its semantic segmentation subset (10k images) to assess cross-domain generalization.
SYNTHIA [151] offers synthetic images with pixel-perfect annotations. We use the RAND subset to analyze performance under precise boundary supervision and in domain adaptation settings.
ADE20K [26] includes 150 semantic categories across diverse indoor and outdoor scenes. We use it to evaluate the scalability of our approach to complex, non-driving environments.

We visualize the semantic boundary GTs produced by OTFGT for each dataset in Fig. 3.8.

3.4.2 Evaluation Metrics

Segmentation Metrics. We evaluate the segmentation performances using the mean of intersection-over-union (mIoU). To assess the segmentation performance around the boundaries of the masks, we adopt the boundary F-score, following the approach in [98]. Unless explicitly stated, we use a pixel width of 3px for the boundary F-score.

Additionally, we employ the region-wise over-segmentation measure (ROM) and region-wise under-segmentation measure (RUM) recently proposed in [154]. ROM and RUM enable us to quantitatively measure the over- and under-segmentation characteristics of the models, providing a more objective evaluation compared to previous qualitative assessments. The values of ROM and RUM fall within the range of $[0, 1)$, where a value of 0 indicates no over- or under-segmentation, while higher values indicate increased over- or under-segmentation.
Boundary Detection Metrics. We follow [162] and adopt the maximum F-score (mF) at the optimal dataset scale (ODS) evaluated on the instance-sensitive "thin" protocol for SBD.

3.4.3 Implementation Details

Backbones and Segmentation Heads. For the ablation studies, we used DeepLabV3+ [62] with a ResNet-101 [159] backbone as the default segmentation architecture unless specified otherwise. We occasionally experimented with other popular alternatives such as PSPNet [59], DeepLabV3 [61], and HRNet [159]. For Sec. 3.6, we evaluated the SBCB

framework across a diverse set of segmentation architectures, including MobileNetV2 [78], BiSeNetV2 [76], ConvNeXt [163], and SegFormer [164].

Data Loading. For the Cityscapes dataset, unless specified otherwise, we use a unified training crop size of 512×1024, 40k training iterations, and a batch size of 8. For Synthia, BDD100K, and ADE20K datasets, we use a crop size of 640×640. During ablation studies in Sec. 3.5, we train for 80k iterations for Synthia, but reduce it to 40k iterations for experiments in Sec. 3.6 due to resource limitations. Common data augmentations such as random scaling (scale factors in $[0.5, 2.0]$), horizontal flip, and photo-metric distortions are applied.

Optimization. During training, we use the SGD optimizer with a momentum coefficient of 0.9 and a weight decay coefficient of 5×10^{-4}. The learning rate policy follows the "poly" learning rate decay, where the initial learning rate of 0.01 is multiplied by $(1 - \frac{\text{iter}}{\text{max\_iter}})^{\gamma}$ with $\gamma = 9$.

Loss. For our loss function in Eq. (3.6), we set $\alpha = 5$ and $\beta = 1$.

Inference. For the Cityscapes dataset, we conduct evaluations with single-scale whole inference. For Synthia and BDD100K datasets, slide inference is performed. In Sec. 3.6.3, we evaluate semantic segmentation performance using multi-scale and flip (MS+Flip) inference with scales of $[0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0]$.

Software and Hardware. All experiments are conducted using PyTorch, and the popular semantic segmentation framework "mmsegmentation" [165] is modified for our task. We ensure consistent software and hardware configurations for all experimental results. The models are trained on two NVIDIA A6000 GPUs.

3.5 Ablation Studies

In this section, we conduct ablation studies to analyze the impact of the SBCB framework from various perspectives. In Sec. 3.5.1, we compare different SBD heads and select the most suitable candidate for experimentation throughout the chapter. In Sec. 3.5.2, we explore the optimal side configuration to achieve the best performance with the SBCB framework. In Sec. 3.5.3, we investigate the effects of using instance-sensitive boundaries for supervision within the SBCB framework. In Sec. 3.5.4, we analyze the benefits of using the OTFGT module for generating semantic boundary GTs during training. In Sec. 3.5.5, we examine which semantic categories benefit the most from the SBCB framework. In

Table 3.1: Ablation studies comparing single-task baselines with the SBCB framework across different configurations and datasets.

a Results for Cityscapes.

Head	mIoU	mF (ODS)	Params.	GFLOPs
DeepLabV3+	79.5	–	60.2M	506
CASENet	–	63.7	42.5M	357
DFF	–	65.5	42.8M	395
DDS	–	73.4	243.3M	2079
SBCB (CASENet)	80.3	74.4	60.2M	508
SBCB (DFF)	80.2	74.6	60.5M	545
SBCB (DDS)	**80.6**	**75.8**	261.0M	2228

b Results with crop size of 769 × 769.

Head	mIoU	mF (ODS)	Params.	GFLOPs
DeepLabV3+	78.9	–	60.2M	506
CASENet	–	68.6	42.5M	357
DFF	–	68.9	42.8M	395
DDS	–	75.5	243.3M	2079
SBCB (CASENet)	80.3	74.0	60.2M	508
SBCB (DFF)	80.0	74.8	60.5M	545
SBCB (DDS)	**80.4**	**75.6**	261.0M	2228

c Results using HRNet-48 (HR48) backbone.

Head	mIoU	mF (ODS)	Params.	GFLOPs
FCN	80.5	–	65.9M	187
CASENet	–	75.7	65.3M	172
DFF	–	75.3	65.5M	210
DDS	–	78.9	89.0M	946
SBCB (CASENet)	**82.0**	78.9	65.9M	187
SBCB (DFF)	81.5	78.8	66.0M	221
SBCB (DDS)	81.0	**79.3**	89.5M	1012

d Results for BDD100K.

Head	mIoU	mF (ODS)
DeepLabV3+	60.0	–
CASENet	–	55.7
DFF	–	57.3
DDS	–	59.9
SBCB (CASENet)	61.4	56.6
SBCB (DFF)	62.0	58.1
SBCB (DDS)	**64.1**	**60.2**

e Results for Synthia.

Head	mIoU	mF (ODS)
DeepLabV3+	74.5	–
CASENet	–	61.0
DFF	–	64.8
DDS	–	**67.6**
SBCB (CASENet)	**75.9**	65.2
SBCB (DFF)	75.3	66.5
SBCB (DDS)	75.7	67.0

Sec. 3.5.6, we compare the SBCB framework with other auxiliary tasks to assess its effectiveness. In Secs. 3.5.7 and 3.5.8, we compare the SBCB framework with state-of-the-art multi-task and post-processing methods, demonstrating its ability to complement these methods and further improve segmentation quality. In Sec. 3.5.10, we investigate the effects of a simple yet effective modification to the backbone configuration, which improves segmentation and SBD performance. In Sec. 3.5.11, we analyze the effects of the SBCB framework on the task of SBD. Finally, in Secs. 3.5.12 and 3.5.13, we demonstrate that our framework enhances segmentation around boundaries and addresses over- and under-

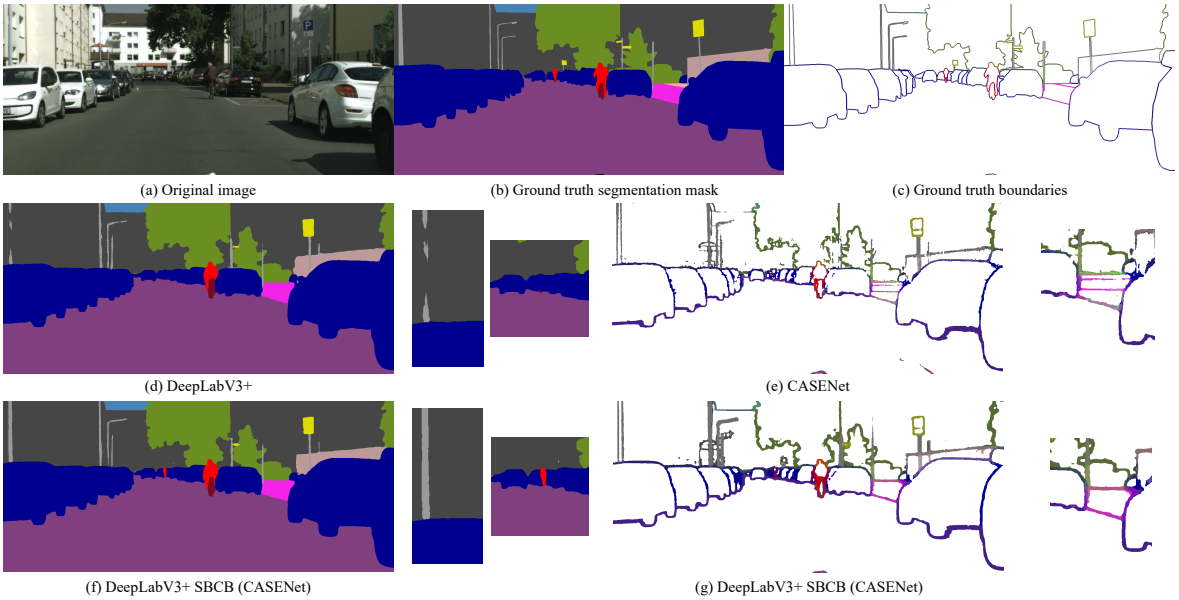


Figure 3.9: Qualitative results on Cityscapes: (a) input, (b) GT segmentation, (c) GT boundaries, (d) DeepLabV3+ baseline, (e) CASENet baseline, (f)–(g) SBCB outputs. SBCB improves detection of thin objects and reduces over-segmentation.

segmentation issues through boundary F-score and region-wise over-/under-segmentation measures (ROM and RUM) evaluations.

3.5.1 Which SBCB head to use?

In this section, we investigate the effects of using different semantic boundary detection (SBD) heads for the SBCB framework and determine the most suitable candidate for further evaluation.

Tab. 3.1a presents the performance of the DeepLabV3+ model trained using three different SBD heads: CASENet, DFF, and DDS, in comparison with single-task baseline models. All SBD heads integrated into the SBCB framework demonstrate improvements over the single-task DeepLabV3+ model. Joint training also contributes to enhancing the SBD metric (maximum F-score). Additionally, we provide information on the number of parameters and computational costs (GFLOPs) introduced during training by the SBD heads. DDS incurs higher costs, but it is the most effective of the three heads. However, the trade-off of using DDS over CASENet for the SBCB framework may not be advantageous in terms of performance gains, particularly when evaluating DDS on other datasets and

backbones.

Fig. 3.9 showcases qualitative results of applying the CASENet head to DeepLabV3+ compared to the baseline models. The additional semantic boundary supervision enables the model to detect smaller, thin objects more effectively. Furthermore, the SBCB framework enhances boundary detection by reducing artifacts and improving object perception. These results demonstrate the potential benefits of using SBD heads in the SBCB framework for semantic segmentation tasks.

Different crop size. We tested the SBD heads on a crop size of 769×769, another popular crop size in semantic segmentation. The results, shown in Tab. 3.1b, exhibit a similar trend to the results from Tab. 3.1a, with the CASENet head performing favorably.

Different backbone. We evaluated the effects of using the HRNet-48 (HR48) backbone, and the results are displayed in Tab. 3.1c. In this case, the CASENet head outperforms DDS and DFF by significant margins (1.0% and 0.5%, respectively). The CASENet head achieves an mF of 78.9%, identical to the heavy and inefficient single-task DDS model.

Different datasets. As performance can vary across datasets, we further evaluated the SBD heads on the BDD100K dataset and Synthia, as shown in Tabs. 3.1d and 3.1e, respectively. On the BDD100K dataset, the DDS head significantly outperforms the baseline model and CASENet head. The DFF head also performs better than the CASENet head for this dataset for the first time. On Synthia, however, the CASENet head performs better than DDS.

Choice of SBD Head: CASENet. Considering the additional parameters and computational costs, using the CASENet head proves to be beneficial. Moreover, the SBD head in the SBCB framework is only used as an auxiliary signal, and the CASENet head outperforms DDS in some results. While the DDS head may produce higher metrics when computational costs are not a concern, for the rest of the chapter, we use the CASENet head as our primary SBD head for the SBCB framework.

In Fig. 3.2, we present qualitative visualizations comparing DeepLabV3+ with and without the CASENet head. The feature maps obtained from the last stage of the backbone conditioned on SBD show boundary-aware characteristics, resulting in reduced segmentation errors, especially around the boundaries. This demonstrates the effectiveness of the SBCB framework in improving the segmentation performance by incorporating boundary information during the feature extraction process.

Table 3.2: Results using the ResNet-101 backbone with different side configurations on the Cityscapes validation split.

Head	**Sides**	mIoU	Δ
PSPNet		77.6	
	1 + 5	78.5	+0.9
	1 + 2 + 5	78.6	+1.0
	1 + 2 + 3 + 5	78.7	**+1.1**
	1 + 2 + 3 + 4 + 5	78.5	+0.9
DeepLabV3		79.2	
	1 + 5	79.8	+0.6
	1 + 2 + 5	79.9	**+0.7**
	1 + 2 + 3 + 5	79.9	**+0.7**
	1 + 2 + 3 + 4 + 5	79.4	+0.2
DeepLabV3+		79.5	
	1 + 5	80.1	+0.6
	1 + 2 + 5	80.1	+0.6
	1 + 2 + 3 + 5	80.3	+0.8
	1 + 2 + 3 + 4 + 5	80.5	**+1.0**

3.5.2 Which sides to supervise?

Tab. 3.2 presents the effect of using different side configurations for the CASENet head applied to the ResNet backbone. The original configuration, which includes Sides 1, 2, 3, and 5, performs the best on two models (PSPNet and DeepLabV3). However, on DeepLabV3+, the configuration 1+2+3+4+5 outperforms the original configuration by 0.2%. While the performance gains between configurations are negligible, it is worth noting that each model may have an optimal side configuration within the SBCB framework. For fairness and consistency, we choose the original configuration used by CASENet (Sides 1, 2, 3, and 5) to evaluate other models and benchmark our methods for further evaluation. Users of the SBCB framework should be aware that different models might benefit from different side configurations.

Table 3.3: Comparison of instance-sensitive (IS) and non-instance-sensitive (non-IS) boundary supervision on Cityscapes with ResNet-101 backbone.

Head	**Type**	mIoU	Δ
PSPNet		77.6	
	non-IS	78.7	+1.1
	IS	78.7	+1.1
DeepLabV3		79.2	
	non-IS	79.7	+0.5
	IS	79.9	+0.7
DeepLabV3+		79.5	
	non-IS	80.0	+0.5
	IS	80.3	+0.8

3.5.3 Influence of instance-sensitivity of semantic boundaries

Typically, labeling procedures for semantic segmentation requires annotators to produce polygons for each of the instances [3, 18], resulting in both semantic and instance segmentation masks. SBD benchmarks often prioritize instance-sensitive protocols where instance masks are used to add boundaries between instances when they overlap. However, we also consider the case where instance boundaries are not provided in the dataset by comparing instance-sensitive (IS) and non-instance-sensitive (non-IS) semantic boundaries as auxiliary objectives.

In Tab. 3.3, we show IoU metrics on the Cityscapes validation split using both IS and non-IS boundaries for supervision. Both types of boundaries lead to improvements over the baseline models. While IS boundaries yield slightly better performance, non-IS boundaries still provide noticeable benefits.

3.5.4 Effect of OTFGT

OTFGT makes the training process more robust by seamlessly handling random resizing and cropping. As stated previously, SBD datasets either use fixed scale or preprocess the datasets with a few pre-determined scales, leading to a lack of data augmentations and requiring extra storage to save the preprocessed ground-truths. We compare the effects of using prior approach of fixed scale SBD ground-truths with OTFGT on DeepLabV3+ with

Table 3.4: Comparison of OTFGT versus preprocessed boundaries on Cityscapes with ResNet-101 backbone.

Model	OTF	mIoU	Δ
PSPNet		78.5	
	✓	78.7	+0.2
DeepLabV3		79.5	
	✓	79.9	+0.4
DeepLabV3+		80.1	
	✓	80.3	+0.2

ResNet-101 backbone on the Cityscapes validation split. Following prior joint segmentation and semantic boundary detection works [103, 102], we applied random crop to the baseline MTL approach directly for a fair comparison.

Tab. 3.4 shows the results of using OTFGT compared to the traditional approach of loading preprocessed boundaries. The improvements brought by OTFGT are consistent throughout all three models. OTFGT and loading the preprocessed boundaries directly takes around 0.11 and 0.10 seconds respectively for the entire data pipeline per iteration. This is around 50% slower than the default training approach which takes only 0.06 seconds per iteration. Surprisingly, using OTFGT is much more effective, while maintaining the same level of computational cost as loading preprocessed boundaries. The traditional data pipeline for joint MTL—by loading preprocessed multi-label boundaries (*i.e.* decoding the encoded format [82]) and also applying data augmentation to the ground-truth boundaries—seem to add significant computational overhead. For training PSPNet, each iteration takes around 0.78 seconds while PSPNet+SBCB (CASENet) with OTFGT takes around 0.91 seconds, meaning around 16.7% overhead incurs with SBCB training. While the hefty overhead of the forward and backward pass by the boundary detection auxiliary head is difficult to reduce, improving the OTFGT may potentially reduce the iteration overhead to around 10%.

3.5.5 Does it improve all categories?

Tab. 3.5 presents the per-category IoU comparisons for each model. While the SBCB framework generally improves most categories, some categories exhibit worse IoU scores. Particularly, the categories “truck,” “bus,” and “train” are more affected, likely due to

Table 3.5: Per-category IoU on Cityscapes validation. Improvements (red) and drops (blue) relative to baseline.

Method	**SBCB**	mIoU	road	swalk	build.	wall	fence	pole	tlight	sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike
		77.6	98.0	83.9	92.4	49.5	59.3	64.5	71.7	79.0	92.4	64.2	94.7	81.8	60.5	95.0	77.8	89.1	80.1	63.4	77.9
PSPNet	✓	78.7	98.3	85.7	92.7	52.7	60.7	66.3	72.7	80.8	92.8	64.3	94.6	82.4	62.7	95.3	79.5	88.6	81.4	66.0	78.7
		+1.1	+0.3	+1.8	+0.3	+3.2	+1.4	+1.8	+1.0	+1.8	+0.4	+0.1	-0.1	+0.6	+2.2	+0.3	+1.7	-0.5	+1.3	+2.6	+0.8
		79.2	98.1	84.6	92.6	54.5	61.7	64.6	71.7	79.3	92.6	64.6	94.6	82.4	63.8	95.4	83.2	90.9	84.2	67.7	78.1
DeepLabV3	✓	79.9	98.4	86.4	93.0	55.3	63.7	66.8	72.9	80.4	94.9	65.4	94.9	83.3	65.9	95.5	81.9	92.3	81.3	68.2	78.9
		+0.7	+0.3	+1.8	+0.4	+0.8	+2.0	+2.2	+1.2	+1.1	+2.3	+0.8	+0.3	+0.9	+2.1	+0.1	-1.3	+1.4	-2.9	+0.5	+0.8
		79.5	98.1	85.0	92.9	53.2	62.8	66.5	72.1	80.4	92.7	64.9	94.7	82.8	63.6	95.5	85.1	90.9	82.2	69.4	78.4
DeepLabV3+	✓	80.3	98.3	85.9	93.4	65.7	65.6	68.5	73.0	81.4	92.8	66.1	95.3	83.3	65.6	95.5	81.3	88.3	78.1	68.7	78.8
		+0.8	+0.2	+0.9	+0.5	+12.5	+2.8	+2.0	+0.9	+1.0	+0.1	+1.2	+0.6	+0.5	+2.0	0.0	-3.8	-2.6	-4.1	-0.7	+0.4

their relatively low number of samples and potential confusion with the "car" category. To address this, additional techniques such as Online Hard Example Mining (OHEM) could be employed during training to focus on difficult samples and improve the overall performance on challenging categories with irregular boundaries and low sample counts.

From the table, it is evident that the SBCB framework significantly enhances the segmentation performance for categories characterized by complex shapes, such as "vegetation", "terrain", "person", and "bike". The incorporation of boundary-aware features in the backbone contributes to the segmentation head's improved capability in accurately predicting the boundaries of these categories, consequently leading to superior segmentation results. This observation is visually apparent in Fig. 3.2, where the segmentation errors surrounding the edges are notably reduced compared to the baseline DeepLabV3+ model, particularly evident in scenarios such as people riding bikes in the bottom row.

One of the challenges in semantic segmentation arises from the presence of overlapping classes, which can occur due to occlusion or see-through objects. When occlusion is present, the predictions near the point of occlusion often become uncertain. This is evident in Fig. 3.9, where objects like "poles" occlude a considerable portion of a building. The baseline model exhibits fragmentation in the prediction of the thin pole, as the larger wall provides more certainty in its segmentation. In contrast, the SBCB framework yields a cleaner segmentation mask, thanks to its ability to comprehend object boundaries more effectively.

Similarly, in scenarios involving see-through categories, such as the fence overlapping with a building in the same figure, the model faces a challenging task in making accurate predictions. However, the SBCB framework proves beneficial by generating a cleaner segmentation mask, leveraging its proficiency in understanding object boundaries more robustly.

The Cityscapes dataset contains annotations with inherent noise, particularly pronounced in categories featuring irregular boundaries, which inherently poses challenges to the segmentation task. This noise in annotations could potentially contribute to the observed variations in IoU scores across categories. Nevertheless, our belief is that the SBCB framework can still enhance segmentation results, even in the presence of noisy annotations, as long as the generated GT boundaries offer valuable guidance during the model's training process. Although detecting such improvements purely based on quantitative metrics like IoU may be difficult, the qualitative outcomes presented in Fig. 3.2,

Table 3.6: Comparison of backbone conditioning methods (FCN, BBCB, SBCB) on (a) Cityscapes and (b) Synthia using ResNet-101 backbone.

a Results on the Cityscapes validation split.

Head	FCN	BBCB	SBCB	Param. (M)	mIoU	Δ
PSPNet				65.58	77.6	
	✓			+2.37	78.3	+0.7
		✓		+0.01	78.1	+0.5
			✓	+0.05	**78.7**	+1.1
	✓	✓		+2.37	79.1	+1.5
	✓		✓	+2.41	**79.4**	+1.8
DeepLabV3				84.72	79.2	
	✓			+2.37	79.3	+0.1
		✓		+0.01	79.6	+0.4
			✓	+0.05	**79.9**	+0.7
	✓	✓		+2.37	**80.1**	+0.9
	✓		✓	+2.41	**80.1**	+0.9
DeepLabV3+				60.20	79.5	
	✓			+2.37	79.7	+0.2
		✓		+0.01	79.9	+0.4
			✓	+0.05	**80.3**	+0.8
	✓	✓		+2.37	**80.6**	+1.1
	✓		✓	+2.41	80.5	+1.0

b Results on the Synthia dataset.

Head	FCN	BBCB	SBCB	mIoU	Δ
PSPNet				70.5	
	✓			70.1	−0.4
		✓		70.7	+0.2
			✓	**71.7**	+1.2
	✓	✓		70.7	+0.2
	✓		✓	**71.6**	+1.1
DeepLabV3				70.9	
	✓			70.6	−0.3
		✓		70.7	−0.2
			✓	**71.9**	+1.0
	✓	✓		70.5	−0.4
	✓		✓	**71.0**	+0.1
DeepLabV3+				72.4	
	✓			72.0	−0.4
		✓		72.1	−0.3
			✓	**73.5**	+1.1
	✓	✓		72.3	−0.1
	✓		✓	**73.5**	+1.1

along with additional metrics like ROM and RUM (as discussed in Sec. 3.5.13), allow us to discern reduced fragmentation and enhanced accuracy in the segmentation masks.

3.5.6 Comparisons of different auxiliary signals

In PSPNet [59], the authors introduced an FCN head to the fourth stage (one before the last stage) in the backbone to stabilize training and improve segmentation metrics. The auxiliary FCN head is trained on the same segmentation task as the main head and has been widely adopted in open-source projects such as `mmseg`.

Although not commonly used, various approaches have explored using binary edge and boundary detection as an auxiliary task for semantic segmentation. Despite the difference in tasks, it has been found that the learned features in the edge detection head can be fused into the segmentation head.

In this section, we compare the SBCB framework with the mentioned auxiliary techniques, namely "FCN" and "Binary Boundary Conditioned Backbone (BBCB)." Note that BBCB is the SBCB framework applied to binary boundary detection. We apply FCN, BBCB, and SBCB to three popular segmentation heads (PSPNet, DeepLabV3, and DeepLabV3+) with ResNet-101 as the backbone. The results on the Cityscapes validation

split are shown in Tab. 3.6a.

While all auxiliary signals improve IoU, the models trained using the SBCB framework consistently perform the best. The improvements of SBCB compared to BBCB are around twice, demonstrating the importance of the SBD task. FCN shows significant gains of 0.7% when applied to PSPNet, but FCN has minimal impact on the other models. Both BBCB and SBCB complement FCN, achieving higher IoU results. Additionally, it is essential to consider the additional parameters introduced by these auxiliary signals during training. While SBCB and BBCB only add thousands of parameters, FCN adds $2.37M$ parameters. Considering the performance gains and the additional parameters, it is evident that boundary-based auxiliary signals offer more benefits than FCN.

We also evaluate the same models and auxiliary heads on the Synthia dataset as shown in Tab. 3.6b. Surprisingly, FCN and BBCB do not provide significant performance gains and even have worse metrics than the baselines. However, SBCB improves upon the baseline by over 1%. It is plausible that the features learned using FCN could have conflicted with the main heads. In contrast to the noisy annotations in Cityscapes, Synthia contains precise segmentation masks rendered from a CG engine instead of human annotations. The classes "human" and "bike" in Synthia have small and thin segmentation masks, which adds to the difficulty. Although features learned by FCN complemented the main head features in Cityscapes, it appears that the FCN learned to derive a conflicted segmentation map for Synthia. The larger number of parameters in FCN compared to SBCB or BBCB might have contributed to this issue. Surprisingly, BBCB did not perform as well as expected because it focuses on low-level features without explicitly modeling high-level semantics.

The SBCB framework conditions the backbone with SBD, a challenging task that focuses on low-level features while requiring high-level features for accurate boundary detection. The hierarchical modeling of the SBD task in the SBCB framework leads to better improvement in segmentation metrics compared to using FCN or binary boundaries as auxiliary signals.

3.5.7 Comparisons with SegFix

In Tab. 3.7, we compare our framework with SegFix [105], a popular post-processing method. We obtained the results for SegFix by using the open-source code, which refines

Table 3.7: Comparison of SBCB with SegFix post-processing on Cityscapes validation.

Model		mIoU	Δ
PSPNet		77.6	
	+SegFix	78.8	+1.2
	+SBCB	78.7	+1.1
	+SBCB + FCN	79.4	+1.8
	+SBCB + SegFix	79.7	+2.1
	+SBCB + FCN + SegFix	**80.3**	**+2.8**
DeepLabV3		79.2	
	+SegFix	80.3	+1.1
	+SBCB	79.9	+0.7
	+SBCB + FCN	80.1	+0.9
	+SBCB + SegFix	80.8	+1.6
	+SBCB + FCN + SegFix	**81.0**	**+1.8**
DeepLabV3+		79.5	
	+SegFix	80.4	+0.9
	+SBCB	80.3	+0.8
	+SBCB + FCN	80.6	+1.1
	+SBCB + SegFix	81.0	+1.5
	+SBCB + FCN + SegFix	**81.2**	**+1.7**

the output prediction based on the offsets learned using HRNet2x. Comparing the methods side-by-side, SegFix performs around 0.1%–0.4% better than models trained with the SBCB framework. However, when combining the SBCB framework with FCN (as mentioned in Sec. 3.5.6), we achieve competitive performance and significantly outperform SegFix on two models.

It is important to consider that SegFix is an independent post-processing model, while our framework produces competitive results without any post-processing and additional parameters during inference. SegFix adds a post-processing module that requires separate training. Furthermore, SegFix is specifically designed to correct predictions around mask boundaries, which can be challenging for the base model to predict accurately. As a result, the base model might not actively learn boundary-aware features. In contrast, our training framework conditions the backbone to be boundary-aware by solving SBD, as demonstrated in Sec. 3.5.12. In other words, SegFix and our framework are complementary because boundary-aware predictions are easier for SegFix to correct. This is evident by the significant improvements achieved by using SBCB along with SegFix, as shown in the table.

Table 3.8: Comparison of DeepLabV3+ and GSCNN on Cityscapes, measuring mIoU, parameters, GFLOPs, and FPS.

Model		mIoU	Δ	Param. (M)	GFLOPs (G)	FPS
DeepLabV3+		79.5		60.2	506	12.3
	+SBCB (CASENet)	80.3	+0.8	60.2	506	12.3
	+SBCB (DDS)	80.6	+1.1	60.2	506	12.3
GSCNN		80.5		62.7	579	8.9
	+Canny	80.6	+0.1	62.8	579	8.4
	+SBD	80.0	−0.5	62.8	622	8.5
	+SBCB (CASENet)	**80.9**	**+0.4**	62.7	579	8.9

For inference, we remove the modules that are not necessary (e.g., semantic boundary detection head whose features are not used in the decoder). All inference evaluations were performed on a NVIDIA RTX3090 with Intel i9-9960X processor.

3.5.8 Comparisons with GSCNN

GSCNN [98] is a well-known semantic segmentation model that incorporates a binary boundary detection multi-task architecture with a dedicated boundary detection head, called the Shape Stream, branching out from the side layers, similar to the SBD heads in the SBCB framework. The main distinction is that GSCNN explicitly merges the features from the shape stream into the semantic segmentation head. GSCNN, based on ResNet-101 backbone, is a customized version of DeepLabV3+ that utilizes an ASPP module.

While it may be challenging to make a direct apples-to-apples comparison due to the different loss functions and the explicit feature merging in GSCNN, we aim to evaluate how effectively the SBCB framework enhances DeepLabV3+ in comparison to different configurations of GSCNN. Tab. 3.8 presents the results of our comparison. The baseline GSCNN is GSCNN without the image gradient (Canny Edge). "+Canny" is the original configuration with the image gradient. We also experimented with supervising the shape stream using the SBD task denoted by "SBD" and modified the shape stream by increasing the channels. Finally, we used the SBCB framework on GSCNN denoted by "+SBCB," which adds the SBD head on the backbone without any other modifications.

Comparing DeepLabV3+ with GSCNN, we observe a substantial improvement of +1.0% when using GSCNN. The SBD supervision on GSCNN results in a slightly lower improvement of +0.5%, indicating that boundary signals do have a positive impact on semantic segmentation. The reduction in improvements can be attributed to the Gated convolution kernel, which restricts features to a single-channel, leading to a degradation in

Table 3.9: Comparison of SBCB and Active Boundary Loss (ABL) on DeepLabV3.

Model		mIoU	Δ
		79.5	
DeepLabV3	+SBCB	80.3	+0.8
	+ABL	79.6	+0.1

representation capability.

However, the SBCB framework proves to be highly effective in improving DeepLabV3+. When utilizing CASENet and DDS as SBD heads, the SBCB framework achieves improvements of +0.8% and +1.1%, respectively. These improvements are competitive with the results obtained using the original GSCNN configuration, further emphasizing the potential of the SBCB framework.

The flexibility of the SBCB framework allows it to be easily applied to GSCNN as well, leading to an even higher improvement of +1.4%. This demonstrates the versatility and efficacy of the SBCB framework, which can enhance segmentation performance across different models.

3.5.9 Comparisons with Active Boundary Loss

In Tab. 3.9, we compare our SBCB framework with Active Boundary Loss (ABL) [108], a loss function designed to enhance boundary accuracy in semantic segmentation. For this comparison, we used 769×769 input resolution with ResNet-50 backbone with DeepLabV3 head and 80K iterations to match the experimental setup from [108]. The results indicate that the SBCB framework outperforms ABL by a significant margin of +0.7%, showcasing the benefits of multi-task learning with SBD as an auxiliary task. The SBCB framework encourages the backbone features to be boundary-aware, leading to improved segmentation performance compared to solely relying on a complex loss function like ABL.

3.5.10 Backbone Trick

In this section, we explore the use of the "backbone trick" which is a modification to the backbone architecture introduced to obtain better edge detection and semantic boundary detection (SBD) performance. The "backbone trick" involves modifying the strides and

Table 3.10: ResNet backbone stride and dilation configurations for different tasks.

Task	**Stem Stride**	**Strides**	**Dilations**	**Resolutions**
Original	2	(1, 2, 2, 2)	(1, 1, 1, 1)	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32})$
Segmentation	2	(1, 2, 1, 1)	(1, 1, 2, 4)	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$
Edge Det.	1	(1, 2, 2, 1)	(2, 2, 2, 4)	$(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$

Table 3.11: Results of the backbone trick (HED-style modification) on three datasets. "HED" denotes backbones with modified stride/dilation for higher-resolution features.

a Results for Cityscapes.

Head	mIoU	mF (ODS)	Param.	GFLOPs
DeepLabV3+	79.8	–	60.2	506
CASENet	–	68.6	42.5	417
DFF	–	70.0	42.8	455
DDS	–	76.3	243.3	2661
SBCB (CASENet)	**81.0**	75.1	60.2	508
SBCB (DFF)	80.8	75.4	60.5	545
SBCB (DDS)	80.8	**76.5**	261.0	2228

b Results for BDD100K.

Head	mIoU	mF (ODS)
DeepLabV3+	59.8	–
CASENet	–	56.6
DFF	–	58.1
DDS	–	60.1
SBCB (CASENet)	62.4	59.3
SBCB (DFF)	62.0	58.9
SBCB (DDS)	**63.5**	**60.5**

c Results for Synthia.

Head	mIoU	mF (ODS)
DeepLabV3+	77.0	–
CASENet	–	64.0
DFF	–	65.6
DDS	–	68.5
SBCB (CASENet)	78.0	67.5
SBCB (DFF)	77.8	**68.9**
SBCB (DDS)	**78.6**	68.4

dilations of the backbone stages to increase the output resolutions without changing the number of parameters, making it suitable for edge detection and SBD tasks.

In edge detection and SBD tasks, higher resolution feature maps are essential to accurately capture small edges and boundaries. Traditional backbones like ResNet, designed for image classification, produce smaller feature maps that may not be well-suited for edge detection. By applying the "backbone trick," we can retain the pre-trained weights while achieving higher resolution feature maps, improving edge detection and SBD

Table 3.12: Comparison of SBD models on Cityscapes using the instance-sensitive "thin" protocol. †: reported performance.

Method	**Backbone**	mF (ODS)
CASENet†	HED ResNet-101	68.1
SEAL†	HED ResNet-101	69.1
STEAL†	HED ResNet-101	69.7
DDS†	HED ResNet-101	73.8
CSEL[102]	HED ResNet-101	78.1
DeepLabV3+ + SBCB (CASENet)	ResNet-101	77.8
DeepLabV3+ + SBCB (CASENet)	HED ResNet-101	78.4
DeepLabV3+ + SBCB (DDS)	ResNet-101	**78.8**
DeepLabV3+ + SBCB (DDS)	HED ResNet-101	**78.8**

performance.

Similarly, in semantic segmentation, we modify the strides and dilations of the last two stages to maintain the final feature resolution to 1/8 of the input image size, which is commonly used for accurate segmentation. The configurations of the two modifications are shown in Tab. 3.10.

Tabs. 3.11a to 3.11c present results using the HED version of ResNet-101 (HED ResNet-101) on Cityscapes, BDD100K, and Synthia datasets, respectively. Compared to the normal segmentation ResNet-101 in Tab. 3.1, HED ResNet-101 generally achieves better performance for both single-task and SBCB framework models. The Synthia dataset, in particular, shows higher performance gains, benefiting from the higher-resolution feature maps that capture more detailed and precise ground truths.

While the "backbone trick" is commonly applied to ResNet-101, it can also be extended to other backbones, such as transformer backbones, as demonstrated in Sec. 3.6.7. By conditioning the backbones with SBD through the SBCB framework, we can achieve significant performance improvements without complex modeling, making it a practical and effective approach for enhancing edge detection and SBD tasks.

3.5.11 Does SBCB also improve SBD metrics?

Based on the previous ablation studies, it is clear that the SBCB framework improves the metrics for semantic segmentation. In addition to semantic segmentation, we also evaluated the performance of the models trained using the SBCB framework on semantic

Table 3.13: Boundary F-score comparison on Cityscapes with ResNet-101 backbone at different trimap widths.

Head	**SBCB**	12px	Δ	9px	Δ	5px	Δ	3px	Δ
PSPNet		80.9		79.6		75.7		70.2	
	✓	83.3	+2.4	82.1	+2.5	78.5	+2.8	73.3	+3.1
DeepLabV3		81.8		80.6		76.7		71.2	
	✓	83.4	+1.6	82.2	+1.6	78.7	+2.0	73.4	+2.2
DeepLabV3+		81.2		80.0		76.4		71.4	
	✓	83.0	+1.8	81.8	+1.8	78.5	+2.1	73.7	+2.3

boundary detection (SBD) tasks, as shown in Tab. 3.12.

The results demonstrate that models trained on the SBCB framework achieve significant improvements in SBD performance compared to state-of-the-art single-task methods. The improvements range from 5% to over 10%, showcasing the effectiveness of the SBCB framework in enhancing boundary detection. Moreover, when comparing our DeepLabV3+ model trained on the SBCB framework to the joint semantic segmentation and semantic boundary detection model CSEL, our method outperforms CSEL without explicitly utilizing the features learned in the segmentation head with feature fusion. This demonstrates that the SBCB framework, which is primarily designed for semantic segmentation, effectively improves SBD performance as well.

Overall, the SBCB framework's success can be attributed to its ability to condition the backbone for semantic segmentation tasks, resulting in improved performance for both semantic segmentation and semantic boundary detection without the need for complex modeling explicitly dedicated to boundary detection.

3.5.12 Does SBCB improve segmentation around boundaries?

In Tab. 3.13, we present the boundary F-scores for both baseline models and models trained on the SBCB framework. The results clearly indicate that the models trained with the SBCB framework consistently achieve higher boundary F-scores, particularly when the trimap widths are smaller. The improved boundary F-scores demonstrate the effectiveness of the SBCB framework in enhancing the model's ability to detect and delineate object boundaries accurately. We believe the SBCB framework enables the backbone to learn and preserve the boundary-aware features, which results in segmentation masks with better quality around the mask boundaries.

Table 3.14: ROM and RUM comparison on Cityscapes with ResNet-101 backbone. Lower values indicate better performance.

Head	**SBCB**	ROM ↓	Δ	RUM ↓	Δ
PSPNet		0.078		0.102	
	✓	0.061	−0.017	0.098	−0.004
DeepLabV3		0.072		0.104	
	✓	0.060	−0.012	0.100	−0.004
DeepLabV3+		0.080		0.094	
	✓	0.065	−0.015	0.086	−0.008

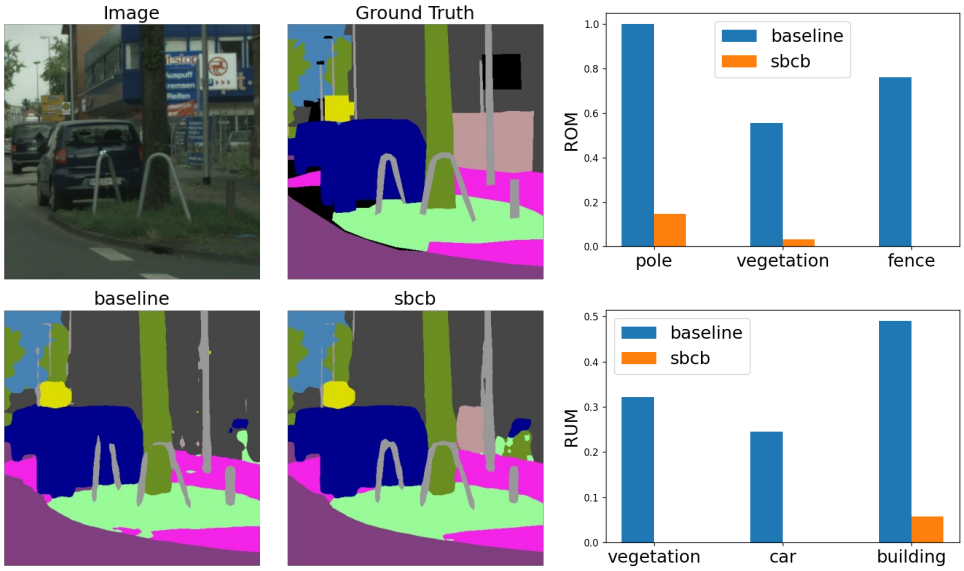


Figure 3.10: Qualitative ROM and RUM comparison between PSPNet baseline and SBCB. SBCB reduces over-segmentation for fences, vegetation, and poles, while alleviating under-segmentation in vegetation and cars.

3.5.13 Does SBCB improve over- and under-segmentation?

In this section, we evaluate the effects of the SBCB framework in terms of over- and under-segmentation using the recently proposed region-based over-segmentation measure (ROM) and region-based under-segmentation measure (RUM) [154]. The results are presented in Tab. 3.14, where lower ROM and RUM values indicate better segmentation quality, reflecting reduced over- and under-segmentation, respectively.

The table clearly shows that the models trained using the SBCB framework consistently exhibit improvements in both ROM and RUM metrics. This indicates that the SBCB

framework effectively mitigates over- and under-segmentation issues in the segmentation outputs. The semantic boundary conditioning in the SBCB framework reinforces strict distinction in object groupings, which helps to resolve unwanted partitioning and leads to an overall improvement in the segmentation quality.

For detailed per-category results of ROM and RUM, please refer to Sec. A.2. Furthermore, the qualitative analysis in Fig. 3.10 provides visual evidence of the improved over- and under-segmentation. For instance, the over-segmentation of the pole is notably reduced by the application of the SBCB framework.

While the improvements in under-segmentation may not be as easily distinguishable in qualitative comparisons, the quantitative evaluation using ROM and RUM metrics confirms the effectiveness of the SBCB framework in addressing both over- and under-segmentation issues in semantic segmentation tasks.

3.6 Experiments

In this section, we conduct a comprehensive evaluation of the proposed Semantic Boundary Conditioned Boosting (SBCB) approach by applying it to various architectures and datasets, aiming to assess its impact on semantic segmentation performance.

We begin by exploring the effectiveness of SBCB training across a wide range of backbone architectures and popular segmentation heads in Secs. 3.6.1 and 3.6.2. Next, we benchmark our method with the DeepLabV3+ architecture on the Cityscapes dataset in Sec. 3.6.3. We compare the results with state-of-the-art (SOTA) methods to demonstrate the superiority of our approach. In Sec. 3.6.4, we experiment on the challenging ADE20k dataset. Furthermore, we apply SBCB training on recent lightweight segmentation architectures in Secs. 3.6.5 and 3.6.6 to showcase the flexibility and effectiveness of the SBCB framework. Finally, we validate the compatibility of the SBCB training paradigm with modern backbones, ConvNeXt, and Segformer, in Sec. 3.6.7. This evaluation underscores the continued relevance and applicability of our approach in the evolving landscape of semantic segmentation.

In the appendix, we have also explored extending the SBCB architecture joint modeling—similar to GSCNN—in Sec. A.6.

Table 3.15: Effect of SBCB on different CNN backbones (Cityscapes validation).

Head	Backbone	SBCB	mIoU ↑	Δ	Fscore ↑	Δ	ROM ↓	Δ	RUM ↓	Δ
DenseASPP	ResNet-50		77.5		69.0		0.108		0.096	
		✓	78.3	+0.8	70.6	+1.6	0.100	−0.008	0.093	−0.003
DenseASPP	DenseNet-169		76.6		69.0		0.077		0.102	
		✓	78.2	+1.6	72.1	+3.1	0.072	−0.005	0.101	−0.001
ASPP	ResNeSt-101		79.5		72.3		0.079		0.102	
		✓	80.3	+0.8	75.2	+2.9	0.065	−0.014	0.094	−0.008
OCR	HR18		78.9		71.9		0.074		0.093	
		✓	79.7	+0.8	74.0	+2.1	0.066	−0.008	0.092	−0.001
OCR	HR48		80.7		74.4		0.073		0.090	
		✓	82.0	+1.3	77.7	+3.7	0.069	−0.004	0.083	−0.007
ASPP	MobileNetV2		73.9		66.2		0.074		0.100	
		✓	74.4	+0.5	68.3	+2.1	0.070	−0.004	0.095	−0.005
LRASPP	MobileNetV3		64.5		58.0		0.128		0.082	
		✓	67.5	+3.0	62.1	+4.1	0.115	−0.013	0.080	−0.002

3.6.1 Different Backbones

Tabs. 3.15 and 3.16 present the performance improvements achieved by employing the SBCB framework during the training of various backbones. Notably, we evaluate our approach on two datasets with different levels of annotation qualities, demonstrating the robustness and consistency of the SBCB framework.

Our findings in both tables reveal that the SBCB framework consistently leads to significant improvements in mIoU, even across different backbone architectures. Particularly, the Cityscapes evaluation showcases enhancements in F-score, ROM, and RUM metrics for every backbone, illustrating the effectiveness of our method.

Furthermore, we provide qualitative results of the SBCB framework on the Cityscapes dataset in Fig. 3.11, further substantiating the impact and practicality of our approach. These results collectively underscore the potential of SBCB in boosting semantic segmentation performance across diverse scenarios.

3.6.2 Different Heads

In Tabs. 3.17 and 3.18, we present the performance evaluation of models trained with the SBCB framework, utilizing different segmentation heads, while keeping the backbone fixed

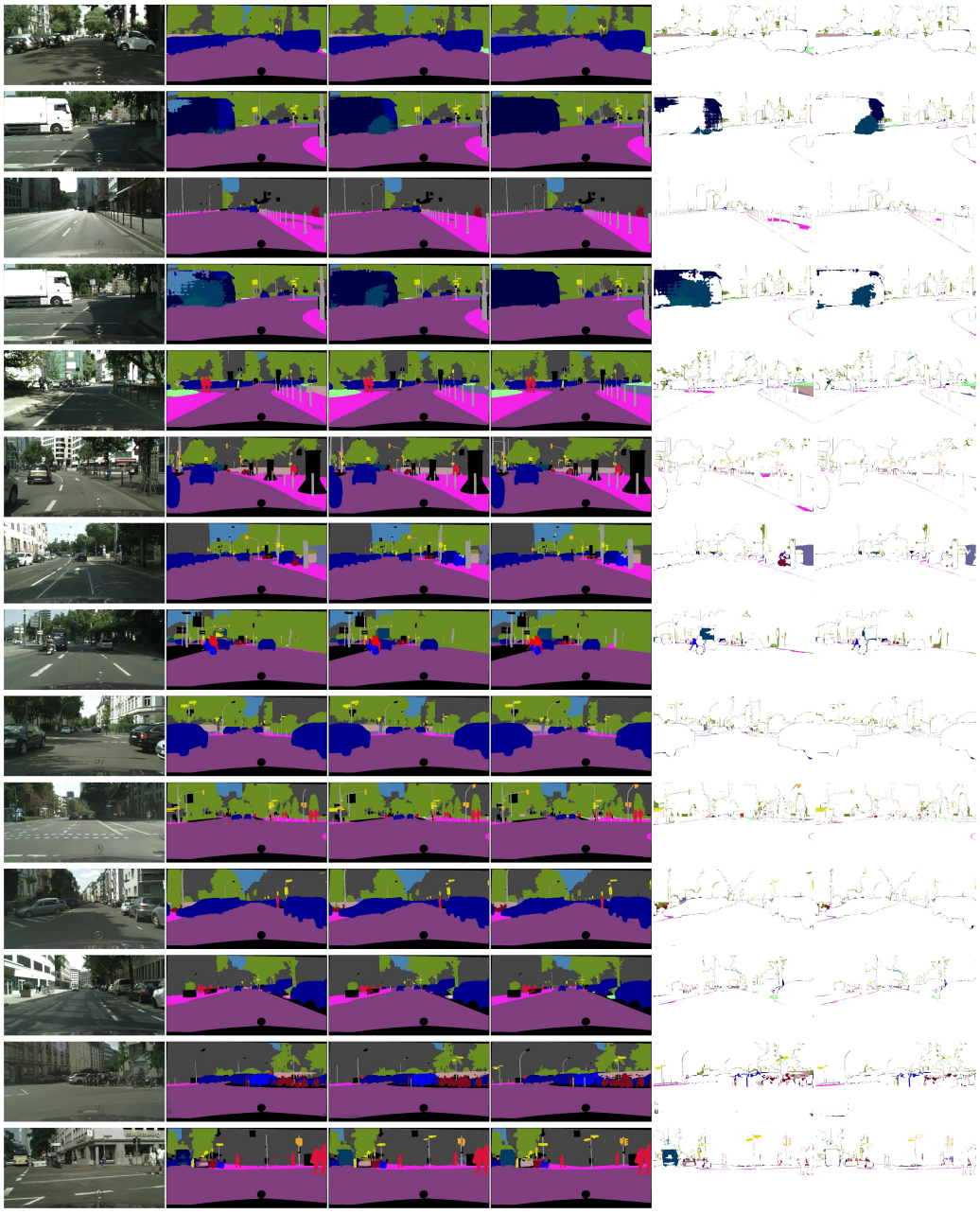


Figure 3.11: Segmentation masks and errors for different backbones. Columns: input, prediction without/with SBCB, GT, errors without/with SBCB. Two rows per backbone (ResNet-50 through MobileNetV3).

Table 3.16: Effect of SBCB on different CNN backbones (Synthia).

Head	**Backbone**	**SBCB**	mIoU ↑	**Δ**
DenseASPP	ResNet-50		69.6	
		✓	70.5	+0.9
DenseASPP	DenseNet-169		71.3	
		✓	72.0	+0.7
ASPP	ResNeSt-101		72.3	
		✓	73.8	+1.5
OCR	HR18		70.1	
		✓	70.9	+0.8
OCR	HR48		74.3	
		✓	76.0	+1.7
ASPP	MobileNetV2		65.3	
		✓	67.0	+1.7
LRASPP	MobileNetV3		60.8	
		✓	64.8	+4.0

at ResNet-101.

The results in these tables demonstrate the consistent improvement achieved by the SBCB framework in terms of mIoU and boundary F-score across various segmentation heads. Notably, the IoU metric is consistently enhanced for all the examined heads. ROM is improved for every segmentation head, while RUM is improved for every head except for OCR. However, OCR has the most gains in IoU and boundary F-score, leading us to believe that this is a trade-off in performances.

To complement our quantitative findings, we include qualitative results of the SBCB framework on the Cityscapes dataset in Fig. 3.12. These visual results further substantiate the efficacy of our proposed approach and showcase the learned features from the last stage of the backbone.

3.6.3 Cityscapes Benchmarks

Tab. 3.19 presents the performance of DeepLabV3+ trained using the SBCB framework, along with a comparison to other SOTA methods with and without boundary auxiliary training. Notably, our SBCB-empowered DeepLabV3+ surpasses other methods in performance while leveraging off-the-shelf segmentation head and backbone, without the need

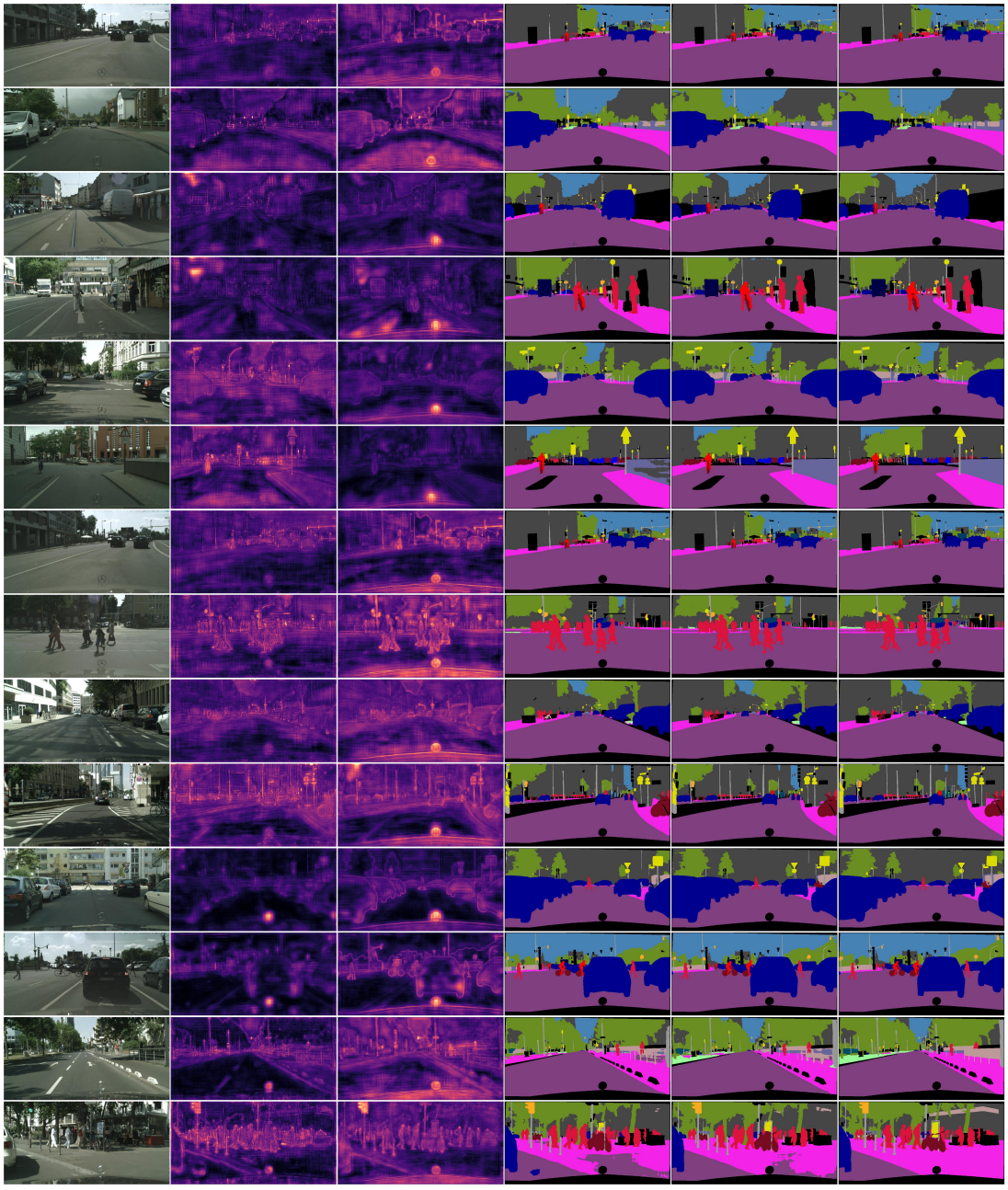


Figure 3.12: Backbone features and segmentation results for different heads. Columns: input, features without/with SBCB, prediction without/with SBCB, GT. Two rows per head (FCN through OCR).

Table 3.17: Effect of SBCB on different segmentation heads (Cityscapes, ResNet-101 backbone).

Head	**SBCB**	mIoU ↑	Δ	Fscore ↑	Δ	ROM ↓	Δ	RUM ↓	Δ
FCN		74.6		69.3		0.072		0.104	
	✓	76.3	+1.7	71.6	+2.3	0.058	−0.014	0.096	−0.008
PSPNet		77.6		70.2		0.078		0.102	
	✓	78.7	+1.1	73.3	+3.1	0.061	−0.017	0.098	−0.004
ANN		77.4		70.1		0.074		0.100	
	✓	79.0	+1.6	72.8	+2.7	0.059	−0.015	0.091	−0.009
GCNet		77.8		70.2		0.070		0.103	
	✓	78.9	+1.1	73.0	+2.8	0.058	−0.012	0.092	−0.011
ASPP		79.2		71.2		0.072		0.104	
	✓	79.9	+0.7	73.4	+2.2	0.060	−0.012	0.100	−0.004
DNLNet		78.7		71.2		0.070		0.101	
	✓	79.7	+1.0	73.6	+2.4	0.052	−0.018	0.093	−0.008
CCNet		79.2		71.9		0.068		0.102	
	✓	80.1	+0.9	73.9	+2.0	0.053	−0.015	0.089	−0.013
UPerNet		78.1		71.9		0.082		0.091	
	✓	78.9	+0.8	73.9	+2.0	0.068	−0.014	0.087	−0.004
OCR		78.2		70.6		0.071		0.096	
	✓	80.2	+2.0	74.4	+3.8	0.064	−0.007	0.100	+0.004

for explicit architecture redesign. This underscores the effectiveness of our approach in boosting semantic segmentation performance without significant modifications to the base model.

Furthermore, the SBCB-empowered DeepLabV3+ achieves competitive results compared to joint-task models, which are inherently designed to better incorporate boundary information into their architecture.

These findings highlight the remarkable capabilities of the SBCB framework in enhancing semantic segmentation, offering a compelling alternative to achieve state-of-the-art results without complex architectural changes.

We have also evaluated our DeepLabV3+ with SBCB on the Cityscapes test split in Sec. A.3, where it continues to demonstrate competitive performance against other SOTA methods.

Table 3.18: Effect of SBCB on different segmentation heads (Synthia, ResNet-101 backbone).

Head	**SBCB**	mIoU ↑	Δ
FCN		70.0	
	✓	70.9	+0.9
PSPNet		70.5	
	✓	71.7	+1.2
ANN		70.4	
	✓	71.8	+1.4
GCNet		70.8	
	✓	71.4	+0.6
ASPP		70.9	
	✓	71.9	+1.0
DNLNet		70.5	
	✓	71.9	+1.4
CCNet		70.8	
	✓	71.3	+0.5
UPerNet		72.4	
	✓	73.1	+0.7
OCR		69.7	
	✓	72.4	+2.7

3.6.4 Experiments on ADE20k

We trained DeepLabV3+ models using both ResNet-50 and ResNet-101 as backbones and carefully compared their performance against models trained using the SBCB framework on the challenging ADE20k dataset. The compelling results of these experiments are presented in Tab. 3.20, where it is evident that the SBCB framework yields notable improvements of over 0.5% compared to the base models.

3.6.5 BiSeNet

In our pursuit of broader applicability and performance gains, we extended the application of the SBCB framework to the Bilateral Segmentation Network (BiSeNet) V1 and V2, which are specialized models designed for real-time semantic segmentation [75, 76].

Both BiSeNet V1 and V2 architectures comprise a split backbone, consisting of the Detail Path (or Spatial Path) and the Semantic Path (or Context Path). The Detail Path

Table 3.19: Comparison with state-of-the-art on Cityscapes validation (fine annotations only, no coarse data or Mapillary pre-training).

Method	**Backbone**	mIoU ↑
Without boundary auxiliary		
PSPNet [59]	ResNet-101	78.8
DeepLabV3+ [62]	ResNet-101	78.8
CCNet [67]	ResNet-101	80.5
DANet [66]	ResNet-101	81.5
SegFix [105]	ResNet-101	81.5
With boundary auxiliary		
GSCNN [98]	ResNet-38	80.8
RPCNet [103]	ResNet-101	82.1
CSEL [102]	HED ResNet-101	83.7
BANet [101]	HED ResNet-101	82.5
With SBCB auxiliary (Ours)		
DeepLabV3+ + SBCB	ResNet-101	82.2
DeepLabV3+ + SBCB	HED ResNet-101	82.6

Table 3.20: SBCB results on ADE20K validation with ResNet backbones.

Head	**Backbone**	**Batch**	**SBCB**	mIoU ↑	**Δ**
PSPNet	50	8		39.9	
	50	8	✓	40.6	+0.7
	101	4		38.2	
	101	4	✓	38.7	+0.5
DeepLabV3+	50	8		41.5	
	50	8	✓	42.0	+0.5
	101	4		37.7	
	101	4	✓	38.2	+0.5

is a shallow CNN with a few stages, retaining large feature resolutions (four stages for BiSeNet V1 and three stages for BiSeNet V2). The Semantic Path, on the other hand, is a deeper CNN tailored to capture high-level semantics. While BiSeNet V1 adopts off-the-shelf architectures like ResNet-50 for the Semantic Path, BiSeNet V2 employs a customized six-stage ConvNet with FCN auxiliary heads for supervising features from the middle stages.

To incorporate the SBCB framework, we selected specific stages (sides) of the backbone to be supervised by the SBD head. Specifically, we chose three stages from the Detail Path as the Binary Sides for the SBD head, and the last stage of the Semantic Path

Table 3.21: SBCB results for BiSeNet and STDC on Cityscapes validation.

Model	SBCB	mIoU ↑	Δ	Fscore ↑	Δ
BiSeNetV1 R50		74.3		66.0	
	✓	75.4	+1.1	69.9	+3.9
BiSeNetV2		70.7		63.8	
	✓	71.6	+0.9	66.2	+2.4
STDC V1 FCN (+Detail Head)		73.7		66.5	
STDC V1 FCN	✓	75.4	+1.7	67.9	+1.4

as the Semantic Side. It is essential to highlight that our approach did not require any modifications to the original model. We simply added the SBD head by extracting the mid-features from the backbones. For more details, please refer to Sec. A.4.

The results obtained through the SBCB framework on BiSeNet (V1 and V2) are presented in Tab. 3.21. As anticipated, applying the SBCB framework led to improvements in both IoU and boundary F-score metrics, further affirming its effectiveness in enhancing models based on non-conventional architectures. This outcome underscores the versatility and potential performance gains offered by the SBCB framework, even for specialized models like BiSeNet V1 and V2, thereby contributing to the advancement of semantic segmentation research in real-time scenarios.

3.6.6 STDC

Like BiSeNet, the STDC network is efficient for real-time semantic segmentation [77]. However, the STDC network is a single branch network that replaces the Detail Path with the Detail Head that uses the features from the third stage to perform "detail guidance" only during the training phase. The Detail Head is supervised with "Detail GT," which is generated using a multi-scale Laplacian Convolution kernel in an on-the-fly manner similar to our method. The Detail GT contains spatial details like boundaries and corners.

In this section, we replace the Detail Head with the SBD head and train using the SBCB framework. We take the first four stages of the backbone for the Binary Sides and use the output of the FFM as the Semantic Side for the SBD head. Please see Sec. A.5 for more details.

The results are shown in Tab. 3.21, where we compare the original STDC with STDC that replaced the Detail Head with our SBD head. Remarkably, substantial improvements

Table 3.22: SBCB results on modern architectures (ConvNeXt, SegFormer) on Cityscapes validation.

Head	Backbone	SBCB	mIoU ↑	Δ	Fscore ↑	Δ
UPerNet	ConvNeXt-Base		81.8		74.4	
	ConvNeXt-Base	✓	82.0	+0.2	75.5	+1.1
	Mod ConvNeXt-Base	✓	82.2	+0.4	76.5	+2.1
SegFormer	MiT-B0		75.5		66.9	
	MiT-B0	✓	76.5	+1.0	68.1	+1.2
	Mod MiT-B0	✓	76.8	+1.3	69.7	+2.8
SegFormer	MiT-B2		80.9		73.2	
	MiT-B2	✓	81.1	+0.2	74.7	+1.5
	Mod MiT-B2	✓	81.6	+0.7	76.0	+2.8
SegFormer	MiT-B4		81.6		75.5	
	MiT-B4	✓	82.2	+0.6	76.7	+1.2

are observed when employing the SBD head as the auxiliary task, particularly in terms of IoU metrics. While the Detail Head aimed to enhance the segmentation quality around boundaries, our SBCB framework demonstrates higher improvements in the boundary F-score, further accentuating its efficacy in leveraging boundary information to enhance semantic segmentation.

3.6.7 ConvNeXt and SegFormer

In this section, we explore the applicability of the SBCB framework and the "backbone trick" on two contemporary architectures: ConvNeXt and SegFormer.

ConvNeXt represents a backbone architecture comprised of pure ConvNet components with design elements borrowed from vision Transformers (ViT) [166, 163]. On the other hand, SegFormer is an architecture designed for segmentation, featuring a ViT-based backbone called the Mix Transformer (MiT), along with a lightweight All-MLP segmentation head [164]. Notably, both architectures incorporate hierarchical feature extraction, rendering them compatible with the SBCB framework.

In Tab. 3.22, we present the results obtained by applying the SBCB framework to these modern architectures. Additionally, we assess the impact of integrating the "backbone trick" denoted by "Mod" in the backbones.

The results demonstrate that the SBCB framework remains effective in improving these state-of-the-art modern architectures, leading to consistent performance gains in

terms of both IoU and boundary F-score. This outcome further affirms the versatility of the SBCB approach and its ability to enhance the performance of contemporary models by leveraging boundary information, reinforcing its relevance in advancing the field of semantic segmentation.

3.7 Conclusion

This chapter introduced *SBCB*, a training framework showing that auxiliary supervision derivable from existing annotations—specifically, semantic boundaries—can systematically improve urban scene segmentation without additional labels or inference-time overhead.

Across multiple datasets and architectures, SBCB yields consistent gains: average improvements of 1.2% mIoU, 2.6% boundary F-score, and reductions in both over-segmentation and under-segmentation on the challenging Cityscapes benchmark. Just as importantly, SBCB directly addresses the boundary-precision issue highlighted in Sec. 1.2, where accuracy typically degrades near object contours. Improvements observed from lightweight mobile backbones to modern Vision Transformers suggest the effect is *architecture-agnostic within the range tested*, rather than tied to a particular model family.

Evidence for Boundary-Derived Auxiliary Supervision. SBCB provides affirmative evidence for the first research question in Sec. 1.3: auxiliary supervision extracted from existing segmentation masks can improve performance without new labeling effort. Boundaries are complementary—rather than redundant—to region labeling, delivering explicit signals where segmentation uncertainty is highest. Coupled with hierarchical feature utilization, this encourages representations that are both regionally accurate and boundary-precise.

Implications for Deployment. SBCB meets the deployment constraints in Sec. 1.2.3 by keeping inference cost unchanged: auxiliary components are used only during training and removed at test time. Unlike the fusion approaches reviewed in Chapter 2, SBCB improves deployed models without altering their inference pipeline, lowering adoption friction for latency-sensitive applications.

Limitations and Scope. The effectiveness of SBCB depends on annotation quality: imprecise or noisy masks (*e.g.* crowd-sourced settings) can propagate errors into derived boundaries. Our Cityscapes results (polygon-based masks) suggest some robustness to annotation style, but systematic evaluation on noisier datasets is left to future work. SBCB

currently applies uniform boundary supervision across classes, though some categories (*e.g.* pedestrians, vehicles) may demand higher precision; class-aware weighting could improve impact where it matters most. Training cost increases by roughly 10–20% due to the additional boundary detection task, which may be non-trivial under tight training budgets. Finally, while we observe clear benefits in urban driving scenes, improvements on ADE20K (Sec. 3.6.4) are more modest, indicating domain dependence when boundary delineation is not the dominant challenge.

Outlook: Semi-Supervised Learning. SBCB's conditioning of the backbone with multi-task objectives brought by boundary signals not only provides boundary-awareness in the downstream task, but also provides good regularization during training—by helping the model to avoid overfitting [91, 90]. This regularization is crucial in semi-supervised semantic segmentation as consistency regularization approaches have sought after effective methods of strong regularization for improved representation learning. Chapter 4 extends this idea with *BoundMatch*, which explores the use of multi-task learning with boundaries in a consistency regularization framework to provide strong regularization and improve the boundary-awareness.

Practical Guidelines from This Chapter. While our claims are limited to boundary-derived supervision in the settings evaluated, three practical takeaways emerged: (i) choose an auxiliary signal that is *complementary* to the primary task (boundaries target a known failure mode); (ii) prefer signals *derivable* from existing resources (*e.g.* annotations) to preserve scalability; (iii) design for *zero inference overhead* by confining auxiliary components to training. These guidelines inform the developments in Chapter 4 and also Chapter 5, where we move to limited-label regimes.

In summary, SBCB shows that segmentation masks contain underused, derivable structure—here, semantic boundaries—that can be turned into effective auxiliary supervision with no runtime cost. As accuracy demands rise under tight compute budgets, extracting more value from existing resources offers a practical path forward.

4

BoundMatch: Boundary Detection Applied to Semi-Supervised Segmentation

4.1 Introduction

Chapter 3 demonstrated that semantic boundaries derived from existing segmentation masks provide valuable auxiliary supervision in fully-supervised settings, achieving consistent improvements across diverse architectures through the SBCB framework. This success naturally raises a critical question: *How does auxiliary boundary supervision behave when labeled data is scarce?* In semi-supervised learning, where models must learn from limited annotations alongside abundant unlabeled data, the regularization provided by auxiliary tasks becomes even more crucial—yet new challenges emerge. Pseudo-labels at boundaries are inherently noisy, and the teacher-student frameworks common in semi-supervised learning require careful handling of multi-task objectives.

This chapter addresses the second research question posed in Sec. 1.3: *Can auxiliary tasks provide regularization in label-scarce scenarios?* We explore this through BoundMatch, a framework that extends the boundary supervision principle established in Chapter 3 to semi-supervised semantic segmentation (SS-SS). While SBCB showed that boundaries and segmentation are complementary in fully-supervised settings, semi-supervised learning

introduces unique challenges that require rethinking how these tasks interact.

The importance of auxiliary supervision is amplified in semi-supervised settings for two key reasons. First, when labeled data is limited, models are prone to overfitting on the small labeled set, and the additional constraints from boundary detection help regularize the learning process. Second, pseudo-labels generated for unlabeled data are least reliable at object boundaries—precisely where boundary detection can provide complementary geometric supervision. This multi-task perspective aligns with the noise-robust learning principles identified in recent SS-SS literature [125], where different "views" of the same data provide mutual regularization.

Semi-supervised semantic segmentation has emerged as a practical solution to the annotation bottleneck identified in Sec. 1.2.1, leveraging abundant unlabeled data to achieve strong performance with minimal labeled examples. Among various approaches, Consistency Regularization (CR) has proven particularly effective, typically implemented through teacher-student frameworks where a teacher model generates pseudo-labels for unlabeled data [119, 33]. However, existing CR methods focus primarily on pixel-wise classification without explicit boundary modeling, missing the opportunity to leverage the boundary-segmentation complementarity established in Chapter 3.

Recent attempts to incorporate boundary information in SS-SS have limitations. Some methods derive boundaries from segmentation pseudo-labels using edge operators [123, 115], propagating segmentation errors to boundary regions. Others like BoundaryMatch [157] add binary boundary detection but still derive boundary pseudo-labels from segmentation outputs, limiting the independence of supervision signals. These approaches fail to fully leverage the insight from Chapter 3: that semantic boundaries learned from hierarchical features are a complementary task that can boost segmentation performance.

Building on this understanding, we propose **BoundMatch**, which adapts the boundary auxiliary principle to the unique challenges of semi-supervised semantic segmentation. The framework introduces **Boundary Consistency Regularized Multi-Task Learning (BCRM)**, enforcing prediction agreement on both segmentation and boundaries between teacher and student models. To improve learning from potentially noisy pseudo-labels, we develop fusion modules—**Boundary-Semantic Fusion (BSF)** and **Spatial Gradient Fusion (SGF)**—that establish bidirectional information flow between tasks while maintaining computational efficiency. BSF helps the segmentation head leverage boundary cues for sharper delineation, while SGF refines boundary predictions using spatial gradients

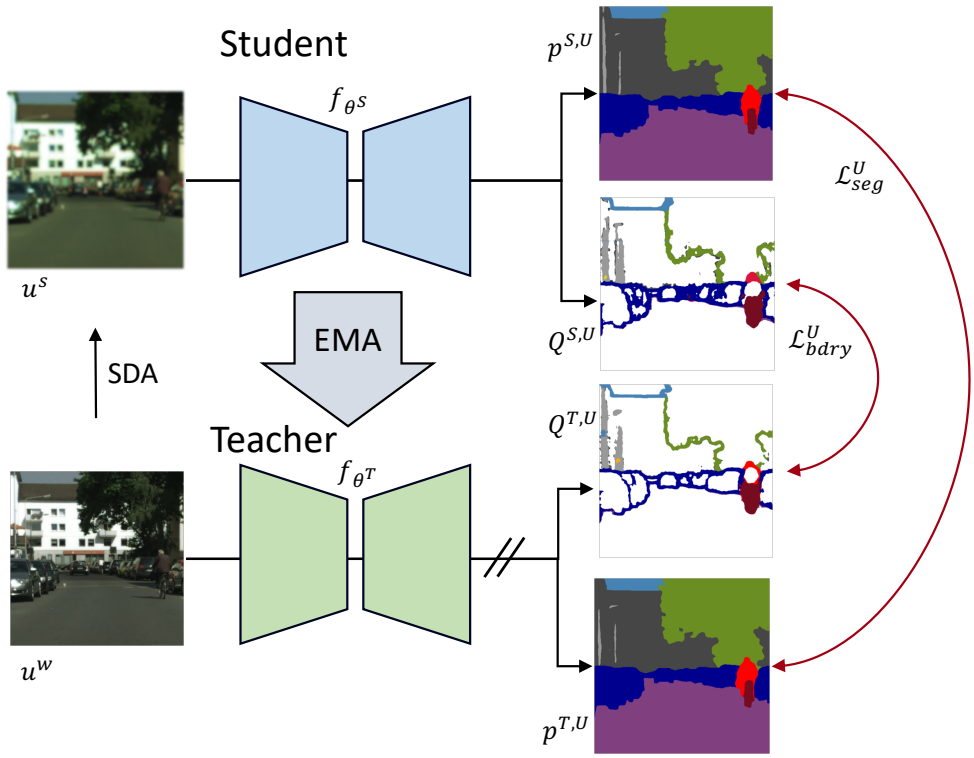


Figure 4.1: Overview of BoundMatch applying consistency regularization to both segmentation and boundary predictions. Fusion modules (BSF and SGF) enable bidirectional information flow between tasks.

from the segmentation mask, producing cleaner pseudo-labels for boundary CR.

Our experiments across diverse datasets demonstrate that the auxiliary supervision principle extends effectively to semi-supervised settings, with BoundMatch achieving improvements of 0.4–2.4% mIoU over strong baselines. The framework's modular design echoes SBCB's flexibility, allowing integration with different CR methods (UniMatch, PrevMatch) and architectures (CNNs, transformers, lightweight models). Notably, improvements in boundary-specific metrics (BIoU, BF1) confirm that the boundary-segmentation complementarity identified in Chapter 3 remains valuable even with noisy pseudo-labels.

The contributions of this chapter are:

- Extension of the boundary auxiliary supervision principle from fully-supervised (Chapter 3) to semi-supervised settings, validating its effectiveness under label scarcity
- Development of BCRM to improve consistency regularization where an independent boundary prediction task acts as strong regularization

- Design of fusion modules (BSF, SGF) that provide bidirectional information flow and refinement between segmentation and boundary tasks while maintaining efficiency
- Comprehensive evaluation showing that boundary-segmentation regularization improves performance across datasets, architectures, and semi-supervised methods
- Extensive evaluation on the effect of boundary auxiliary supervision on the quality of segmentation performance at boundaries using boundary-specific metrics (BIoU, BF1)
- Introduction of Harmonious Batch Normalization (HBN) to address training instabilities in EMA-based teacher-student frameworks, benefiting the broader SS-SS community

4.2 Related Work

This section focuses on semi-supervised semantic segmentation methods and the limited exploration of boundary information in this context. For comprehensive background on semantic segmentation architectures and boundary-aware methods in fully-supervised settings, we refer readers to Sec. 2.1 and Sec. 2.2.2.1.

4.2.1 Semi-Supervised Semantic Segmentation

Building on the segmentation foundations established in Sec. 2.1, semi-supervised methods aim to reduce annotation requirements by leveraging unlabeled data. The challenge is particularly acute given the annotation costs discussed in Sec. 1.2.1—a single Cityscapes image requires approximately 90 minutes of expert annotation [3].

Consistency Regularization (CR) has emerged as the dominant paradigm, operating on the smoothness assumption that semantically similar inputs should produce similar predictions [125]. The Mean Teacher framework [32], detailed in Sec. 2.3.3.1, provides the foundation for many modern approaches. CutMix-MT [119] integrated CutMix augmentation for stronger perturbations, while recent works like AugSeg [120] and UniMatch [33] explored increasingly sophisticated augmentation strategies.

Alternative approaches include co-training with multiple networks (CPS [138], Diverse Co-Training [136]) and hybrid methods combining consistency with contrastive learning (ReCo [127], U2PL [128]). Recent advances have adapted these techniques to transformer architectures [116, 167] and explored their application with foundation models [168].

Table 4.1: Evolution of boundary utilization in semi-supervised segmentation. BoundMatch learns boundaries independently through hierarchical features rather than deriving them from noisy segmentation outputs.

	SBCB 2023 (Chapter 3)	CFCG 2023 [123]	CW-BASS 2025 [115]	BoundaryMatch 2024 [157]	**BoundMatch (Ours)**
Training Paradigm	Fully-supervised	Semi-supervised	Semi-supervised	Semi-supervised	Semi-supervised
Boundary Type	Semantic	Binary	Binary	Binary	**Semantic**
Boundary Source	GT masks	Derived (Lap.)	Derived (Sobel)	Derived (Lap.)	**Learned**
Integration	Auxiliary head	Loss reweight	Loss reweight	Parallel MTL	**Fused MTL**
Consistency Reg.	–	–	–	✓	✓
Feature Stages	Hierarchical	–	–	Single	**Hierarchical**

4.2.2 Boundary Information in Semi-Supervised Settings

Despite the proven benefits of boundary supervision in fully-supervised settings (Chapter 3), its application to semi-supervised learning remains limited. Tab. 4.1 summarizes the evolution from SBCB's fully-supervised approach to various semi-supervised adaptations.

Early attempts derive boundary masks from segmentation pseudo-labels for loss reweighting: CFCG [123] uses Laplacian operators while CW-BASS [115] employs Sobel filters. These approaches inherit the noise present in pseudo-labels, potentially amplifying errors at boundaries where uncertainty is highest.

BoundaryMatch [157] represents the first explicit multi-task approach, adding binary boundary detection to UniMatch. However, it still derives boundary pseudo-labels from segmentation predictions—resulting in learning boundaries that are dependent on segmentation quality. Additionally, it uses only one stage of the backbone features, missing the hierarchical supervision that SBCB showed to be valuable.

Our BoundMatch explores a different combination of design choices: (1) Semantic boundaries rather than binary edges, providing class-specific information; (2) Learned boundary predictions from a teacher model rather than deriving them from pseudo-labels; (3) Bidirectional fusion between tasks (BSF and SGF) rather than parallel multi-task learning; (4) Hierarchical features from multiple backbone stages [103, 169] for improved regularization and multi-scale boundary reasoning. We expand SBCB, which focused on fully-supervised learning, to the semi-supervised domain by addressing the unique challenges of noisy pseudo-labels and task interaction. While MTL provides regularization in the backbone features, it is still susceptible to error propagation if pseudo-labels are poor.

This motivates the new modules—SGF and BSF—which provide bidirectional information flow. Boundary prediction is a relatively difficult task compared to semantic segmentation, as it requires precise localization of edges, thus SGF refines boundary predictions using spatial gradients from segmentation masks to improve pseudo-label quality and reduce confirmation bias.

4.3 Approach

In this section, we present BoundMatch, our multi-task framework that incorporates boundary detection into semi-supervised semantic segmentation (SS-SS). Our design integrates multiple noise-handling mechanisms identified in recent SS-SS literature [125]: EMA-based label refinement, strong data augmentation, multi-task learning with independent segmentation and boundary heads, and confidence-based sample selection. The boundary detection task, learned from hierarchical features rather than derived from segmentation outputs, provides strong regularization and can potentially improve segmentation accuracies in object boundaries.

We begin by outlining the notation and preliminaries for SS-SS, including the teacher-student paradigm, in Secs. 4.3.1 and 4.3.2. We then introduce SAMTH, our baseline method detailed in Sec. 4.3.3. Building upon this baseline, Sec. 4.3.4 describes our core Boundary Consistency Regularized Multi-Task Learning (BCRM) framework. Subsequently, Sec. 4.3.5 details the complementary boundary-aware modeling and refinement strategies: Boundary-Semantic Fusion (BSF) and Spatial Gradient Fusion (SGF). Finally, Sec. 4.3.6 synthesizes these components (SAMTH, BCRM, BSF, and SGF) into the complete BoundMatch framework.

4.3.1 Notation and Preliminaries

Let $\mathcal{D}^L = \{(x_i, y_i)\}_{i=1}^N$ be the labeled dataset, where x_i denotes the input image and $y_i \in \{0, 1, \ldots, C-1\}^{H \times W}$ represents the corresponding semantic segmentation map with C semantic classes and spatial dimensions $H \times W$. In addition, let $\mathcal{D}^U = \{u_j\}_{j=1}^M$ be an unlabeled dataset consisting of images u_j.

The goal of SS-SS is to learn a segmentation model $f_\theta(\cdot)$ parameterized by θ by

leveraging both $\mathcal{D}^L$ and $\mathcal{D}^U$. The overall loss is formulated as:

$$\mathcal{L} = \mathcal{L}^L + \lambda \mathcal{L}^U, \tag{4.1}$$

where $\mathcal{L}^L$ is the supervised loss over the labeled set and $\mathcal{L}^U$ is the unsupervised loss applied to the unlabeled set. Using the pixel-wise cross-entropy loss H, we represent the overall supervised loss as:

$$\mathcal{L}^L = \frac{1}{B^L} \sum_{b=1}^{B^L} \frac{1}{HW} \sum_{i,j} H\Big(y^L(b,i,j), p^{S,L}(b,i,j)\Big), \tag{4.2}$$

with B^L representing mini-batch size and $p^L = f_\theta(x)$ representing the semantic segmentation prediction for the network.

4.3.2 Teacher-Student Framework for Consistency Regularization

The teacher-student framework, as explained in Sec. 2.3.3.1, is a popular paradigm for SS-SS. It employs consistency regularization (CR) between the predictions of a teacher network and a student network for unlabeled samples. The core idea is to encourage the student network to produce consistent predictions for an unlabeled image under different perturbations. The teacher network, which is typically an exponential moving average (EMA) of the student network, provides more stable and reliable predictions, which are then used to guide the student's learning.

In our framework, the teacher's output from weakly augmented images is used to generate hard pseudo-labels that supervise the student's predictions under strong data augmentation (SDA). This process can be seen as a form of self-distillation, where the student learns from the teacher's predictions on unlabeled data.

Specifically, given a weakly augmented image u^w and its strongly augmented counterpart u^s, we first obtain the teacher's prediction $p^{T,w}$ and generate hard pseudo-labels $\hat{p}^T = \text{argmax}(p^{T,w})$. These pseudo-labels represent the teacher's most confident predictions for each pixel in the unlabeled image.

The unsupervised consistency loss is then defined as:

$$\mathcal{L}^U = R(p^{S,s}, p^{T,w}) = \frac{1}{B^U}\sum_{b=1}^{B^U}\frac{1}{HW}\sum_{i,j}\mathbf{1}\Big(\max\Big(p^{T,w}(b,i,j)\Big) \geq \tau\Big) \\ H\Big(\hat{p}^T(b,i,j), p^{S,s}(b,i,j)\Big), \tag{4.3}$$

where $\mathbf{1}(\cdot)$ is an indicator function, and τ is the confidence threshold to filter out noisy pseudo-labels. This loss function measures the discrepancy between the student's predictions on the strongly augmented unlabeled image and the teacher's hard pseudo-labels on the weakly augmented version. By minimizing this loss, the student is encouraged to learn from the teacher's predictions and become more robust to different augmentations.

The student's parameters are jointly updated using labeled set via $\mathcal{L}^L$. The teacher's parameters θ^T are updated using EMA of the student's parameters:

$$\theta^T \leftarrow \alpha\,\theta^T + (1-\alpha)\,\theta^S, \tag{4.4}$$

with decay rate α. EMA helps to stabilize the teacher's predictions by averaging the student's parameters over time, leading to more reliable pseudo-labels.

4.3.3 Strong Data Augmentation Mean-Teacher with Hard Pseudo-Labels (SAMTH)

Our baseline, SAMTH, builds upon the CutMix-MT framework [119] but distinguishes itself by using thresholded pseudo-labels (hard pseudo-labels) rather than Softmax probabilities (soft pseudo-labels), similar to recent methods such as AugSeg [120].

A key differentiator of SAMTH is its **Harmonious Batch Normalization (HBN)** update strategy, addressing potential instabilities in teacher-student training using Batch Normalization (BN) with EMA. Common alternatives risk issues:

- Forwarding only partial data through the teacher (*e.g.* CutMix-MT [119], U2PL [128]) can lead to biased BN statistics unsuitable for the teacher's full role.
- Copying or taking the EMA of the student's BN statistics (*e.g.* AugSeg [120], iMAS [121]) couples the teacher's normalization to the student's potentially noisy state and risks incompatibility between these borrowed BN statistics and the teacher's own EMA-updated weights.

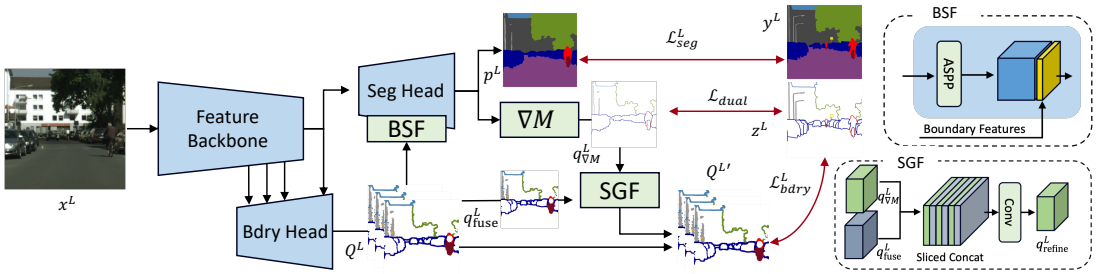


Figure 4.2: BoundMatch architecture for labeled samples. BSF integrates boundary cues into segmentation features, while SGF refines boundary predictions using the spatial gradient of the segmentation mask (∇M).

HBN aims to mitigate these concerns. We forward the complete batch (x, u^w, u^s) through both networks: $p^{S,L}, p^{S,w}, p^{S,s} = f_{\theta^S}(x, u^w, u^s)$ and $p^{T,L}, p^{T,w}, p^{T,s} = f_{\theta^T}(x, u^w, u^s)$. Crucially, the teacher's BN layers remain in `train` mode, updating statistics independently based on the full data stream passing through the teacher's own parameters. For every iteration, the teacher updates its BN statistics, while the rest of the model parameters (*i.e.* layer weights) are EMA-updated from the student's parameters. This ensures the teacher's BN statistics are *harmonious* with its specific weights and the full input distribution, decoupling its normalization from student fluctuations. This promotes more stable training dynamics and improves performance, as validated empirically across multiple baselines in our ablations (Sec. 4.4.5.8). For more complete details and pseudocode, please refer to Sec. A.7.

4.3.4 Boundary Consistency Regularized Multi-Task Learning (BCRM)

While many semantic segmentation approaches rely solely on segmentation supervision, recent works have demonstrated the benefit of multi-task learning (MTL) with a boundary detection task. BCRM takes this concept further by applying consistency regularization (CR) to both segmentation and boundary predictions.

4.3.4.1 Notation and Boundary Detection Head

We adopt a lightweight boundary detection head (SBCB) introduced in Chapter 3. The architecture leverages hierarchical features from the backbone (*e.g.* features from Stem,

layer1, layer2, and layer4 of a ResNet). Features from the earlier layers are processed to yield single-channel outputs, while the layer4 feature produces an output with C channels, where C is the number of classes. Specifically, the features from these layers are passed through a single 1×1 convolutional layer, respectively, each followed by BN and ReLU activation. These processed features are then bilinearly upsampled to $1/2$ of the input image size. Following SBCB, we use an additional 3×3 convolutional layer to reduce artifacts from upsampling. The outputs are then fused using a sliced concatenation and undergo a grouped convolution to form the final boundary prediction map of shape $C \times H \times W$. For more details on the boundary detection head architecture, please refer to Sec. A.8.

Let Q denote the set of boundary predictions (*e.g.* fused output q_{fuse} and the layer4 output q_{last}). For the boundary detection task, the ground-truth boundary is denoted by z and the boundary prediction q is a sigmoid-activated probability map in $[0, 1]$. We supervise these outputs using a reweighted pixel-wise binary cross-entropy loss:

$$H_{bdry}(z, q) = \beta\,(1 - z)\,\log(1 - q) + (1 - \beta)\,z\,\log(q), \tag{4.5}$$

where the weights are defined by $\beta = |z^+|/|z|, \quad 1-\beta = |z^-|/|z|$, with $|z^+|$ and $|z^-|$ representing the number of boundary and non-boundary pixels, respectively. This reweighting addresses the class imbalance between boundary and non-boundary pixels, ensuring that the model learns to detect boundaries.

For semantic (multi-label) boundaries, we use a reweighted multi-label cross-entropy loss over the C channels. The supervised boundary loss is then given by:

$$\mathcal{L}^L_{bdry} = \sum_{q^{S,L} \in Q^L} \frac{1}{B^L} \sum_{b=1}^{B^L} \frac{1}{CHW} \sum_{c,i,j} H_{bdry}\Big(z(b, c, i, j),\, q^{S,L}(b, c, i, j)\Big). \tag{4.6}$$

While our primary auxiliary task is semantic boundary detection, we also explore the use of binary boundaries in Sec. 4.4.5.3. We opt for semantic boundaries because they provide richer, class-specific edge information compared to generic binary edges. This detailed supervision can encourage the model to learn features that better distinguish between adjacent classes, potentially leading to improved feature representations in the backbone proven in Chapter 3.

4.3.4.2 MTL for SS-SS with Boundaries

For the labeled set, the boundary detection head is trained alongside the segmentation head. For unlabeled data, we perform consistency regularization analogous to segmentation as shown in Fig. 4.1. We generate boundary pseudo-labels from the teacher's weakly augmented predictions $q^{T,w}$, where $q^{T,w} = q^{T,w}_{\text{fuse}}$ represents the teacher's fused boundary output (as described in Sec. 4.3.4.1). These are converted to hard pseudo-labels using threshold τ_{bdry}: $\hat{q}^T = \mathbf{1}\Big(q^{T,w} \geq \tau_{bdry}\Big)$. The unsupervised boundary loss then enforces consistency between these pseudo-labels and the student's strongly augmented predictions:

$$\mathcal{L}^U_{bdry} = \sum_{q^{S,s} \in Q^U} \frac{1}{B^U} \sum_{b=1}^{B^U} \frac{1}{CHW} \sum_{c,i,j} H_{bdry}\Big(\hat{q}^T(b,c,i,j),\, q^{S,s}(b,c,i,j)\Big). \tag{4.7}$$

Note that when SGF is enabled (Sec. 4.3.5.2), we use the refined boundary prediction $q^{T,w}_{\text{refine}}$ instead of $q^{T,w}_{\text{fuse}}$ to generate higher-quality pseudo-labels, as the refinement process produces cleaner boundaries by incorporating spatial gradient information.

From a noise-robustness perspective, BCRM leverages multi-view learning principles where the segmentation and boundary detection heads serve as *two complementary views* of the same visual understanding task. Unlike methods that derive boundaries directly from segmentation outputs (*i.e.* BoundaryMatch [157] creates dependent pseudo-labels), our boundary head learns from hierarchical backbone features independently, providing a distinct supervisory signal. This independence is important: when one task branch generates erroneous pseudo-labels, the other branch can continue learning from its own independently-generated pseudo-labels, preventing error propagation between tasks. Following noise-robust learning principles identified in [125], such multi-view consistency with independent learning objectives aims to help prevent overfitting to task-specific noise patterns.

4.3.5 Boundary-Aware Modeling and Refinement

To further enhance the boundary quality and boost segmentation performance, we propose two complementary modules: **Boundary-Semantic Fusion (BSF)** and **Spatial Gradient Fusion (SGF)**. BSF integrates learned boundary cues into the segmentation decoder to improve object delineation, while SGF refines boundary predictions using spatial gradients from the segmentation mask to produce cleaner boundary pseudo-labels.

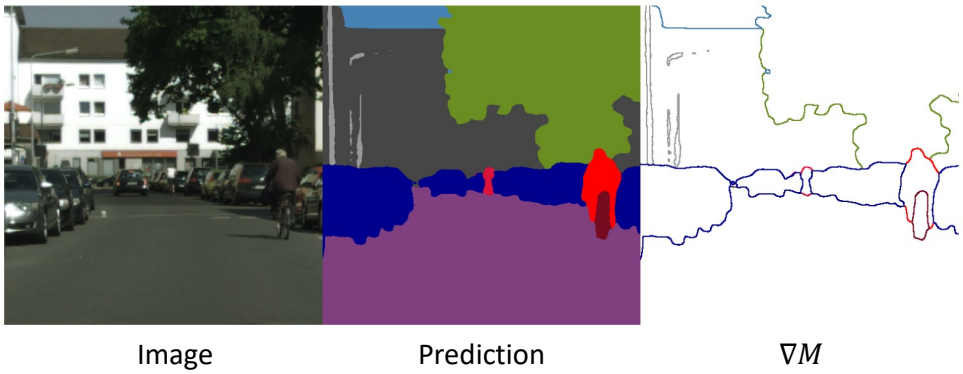


Figure 4.3: Visualization of the output of spatial gradient operator on segmentation prediction.

These simple fusion mechanisms establish bidirectional information flow between tasks as shown in Fig. 4.2.

4.3.5.1 Boundary-Semantic Fusion (BSF)

Inspired by GSCNN [98], BSF explicitly integrates boundary predictions into the segmentation head through concatenation-based fusion. Formally, let $F_{\text{ASPP}} \in \mathbf{R}^{(C_{\text{ASPP}} \times H/16 \times W/16)}$ denote the ASPP features and $q \in \mathbf{R}^{(C \times H/16 \times W/16)}$ denote the downsampled boundary features from the boundary prediction head which is detached from gradient computation. The BSF operation is defined as:

$$F_{\text{fused}} = \text{Conv}_{1\times1}([F_{\text{ASPP}}; q]) \tag{4.8}$$

where $[\cdot;\cdot]$ denotes concatenation along the channel dimension, and $\text{Conv}_{1\times1}$ is the existing bottleneck layer in the DeepLabV3+ decoder. This design choice preserves full information from both modalities without information loss through element-wise operations. We emphasize that BSF introduces no additional layers; we only expand the input channels of the existing bottleneck layer from C_{ASPP} to $C_{\text{ASPP}} + C$, maintaining architectural simplicity while enabling effective feature fusion. By incorporating boundary cues, BSF aims to provide the segmentation decoder with complementary edge information that may assist in object delineation.

4.3.5.2 Spatial Gradient Fusion (SGF)

While the boundary detection head learns from hierarchical features, it can benefit from geometric cues present in the segmentation predictions. We adopt the spatial gradient operator from RPCNet [103] to extract these cues. The spatial gradient of the semantic mask M is computed as:

$$\nabla M(i,j) = \left\| M(i,j) - \text{pool}\left(M(i,j)\right) \right\|, \tag{4.9}$$

where pool denotes adaptive average pooling with a 3×3 kernel. An example of the spatial gradient operator is shown in Fig. 4.3. The resulting gradient map, $q_{\nabla M}$, is concatenated with the boundary prediction q_{fuse} for each channel:

$$[q_{\text{fuse},0},\, q_{\nabla M,0},\, q_{\text{fuse},1},\, q_{\nabla M,1},\, \ldots,\, q_{\text{fuse},C-1},\, q_{\nabla M,C-1}], \tag{4.10}$$

forming a $2C$-channel activation map. This interleaving arrangement ensures that each paired boundary-gradient features for the same class are processed together. The map is then processed by a single C-grouped convolutional layer:

$$q_{\text{refine}} = \text{GroupConv}_C([q_{\text{fuse},0}, q_{\nabla M,0}, \ldots]) \tag{4.11}$$

where GroupConv_C denotes a grouped convolution with C-groups. This design choice follows established semantic boundary detection methods [82, 83] where per-class grouped convolutions enable class-specific boundary refinement. Each group independently refines its corresponding class boundary using the paired spatial gradient information, preserving class-specific edge patterns while maintaining computational efficiency through a single convolution operation. The refined boundaries are supervised using the same H_{bdry} loss with the labeled set.

Moreover, we introduce a duality loss on the labeled set to enforce consistency between the spatial gradient and the ground-truth boundaries:

$$\mathcal{L}_{dual} = \frac{1}{B^L} \sum_{b=1}^{B^L} \frac{1}{CHW} \sum_{c,i,j} \left| q^L_{\nabla M}(b,c,i,j) - z(b,c,i,j) \right|. \tag{4.12}$$

We apply this loss only on labeled samples because the ground-truth boundary annotations are reliable; applying it on unlabeled data, which relies on pseudo-labels, introduces instability.

4.3.6 BoundMatch

Finally, BCRM combined with BSF and SGF forms the **BoundMatch** framework. The labeled and unlabeled losses when we apply BoundMatch to SAMTH (SAMTH+BoundMatch) is

$$\mathcal{L}^L = \mathcal{L}^L_{seg} + \lambda_{bdry}\,\mathcal{L}^L_{bdry} + \mathcal{L}_{dual}, \tag{4.13}$$

$$\mathcal{L}^U = \lambda_{seg}\,\mathcal{L}^U_{seg} + \lambda_{bdry}\,\mathcal{L}^U_{bdry}, \tag{4.14}$$

and the total loss is

$$\mathcal{L} = \mathcal{L}^L + \lambda\,\mathcal{L}^U. \tag{4.15}$$

The design of BoundMatch emphasizes modularity and flexibility. The framework's core principles—boundary CR through independent task heads, bidirectional fusion between tasks, and hierarchical feature extraction—are architecture-agnostic. This modularity enables practitioners to adopt different configurations based on their requirements: full BoundMatch for maximum accuracy, BCRM+SGF for maintaining inference speed, or BCRM alone for minimal overhead. While demonstrated primarily with SAMTH, BoundMatch integrates naturally with other CR methods. For UniMatch [33], we incorporate boundary consistency alongside its existing feature-level and strong-view losses, using an EMA teacher for stable boundary pseudo-labels. For PrevMatch [124], we leverage its robust ensemble pseudo-labels for boundary consistency, circumventing single-iteration instabilities.

BoundMatch's approach differs from prior methods by learning boundaries independently through dedicated task heads rather than deriving them from segmentation outputs. The teacher model generates boundary pseudo-labels from its learned boundary head (potentially refined by SGF), providing supervision that is complementary to, rather than dependent on, segmentation predictions. This independence aims to provide noise robustness: when segmentation pseudo-labels are uncertain at object edges, the boundary task can provide additional supervision signals, and vice versa. The implementation of BSF and SGF relies on standard operations—feature concatenation and grouped convolutions—which facilitates integration with various segmentation architectures without requiring substantial modifications.

4.4 Experiments

In this section, we validate our proposed approach, BoundMatch, through comprehensive experiments. We first benchmark BoundMatch against recent state-of-the-art SS-SS methods on several benchmark datasets. Subsequently, we present detailed ablation studies to analyze the individual contributions of our core components (BCRM, BSF, SGF), evaluate the effect of the chosen boundary formulation (semantic vs. binary), and assess the impact of our Harmonious Batch Normalization (HBN) update strategy. Finally, we report the computational cost associated with BoundMatch and applications to more real-world settings.

4.4.1 Experimental Configuration

Here, we specify the experimental configuration used for evaluating BoundMatch. This includes the datasets utilized (covering urban scenes and academic benchmarks), the model architectures, and key implementation details. We follow the standard SS-SS evaluation protocols for each dataset, ensuring a fair comparison with existing methods following [33].

4.4.1.1 Datasets

We primarily evaluate our method on urban scene datasets, including Cityscapes, BDD100K, and SYNTHIA, while also considering Pascal VOC 2012 and ADE20K to assess performance in more diverse scenarios. In all semi-supervised learning protocols, the images not selected as labeled data form the unlabeled set for training.

Cityscapes dataset [3] is tailored for semantic understanding of urban street scenes. It comprises 2,975 high-resolution training images and 500 validation images, with annotations for 19 semantic categories. In our experiments, we follow the splits defined by UniMatch [33] and evaluate on label partitions corresponding to $1/16$, $1/8$, and $1/4$ of the full labeled training set.

BDD100K dataset [18], originally designed for multi-task learning in autonomous driving, contains 100,000 video frames. For our segmentation experiments, we utilize a subset of 8,000 images (1280×720 resolution), divided into 7,000 for training and 1,000 for validation. As BDD100K has not been widely used previously for SS-SS evaluation, we define labeled splits corresponding to $1/64$, $1/32$, and $1/16$ of the available training annotations.

SYNTHIA dataset [151] provides synthetic urban scenes generated via computer graphics. We utilize the SYNTHIA-RAND ("Rand") subset, containing 13,400 images (1280×760 resolution) with annotations aligned to the Cityscapes semantic categories. For our experiments, we use a split of 7,000 training images and 1,000 validation images, evaluating on labeled partitions corresponding to $1/64$, $1/32$, and $1/16$.

PASCAL VOC 2012 [152] is a standard object recognition benchmark consisting of 10,582 images annotated for 21 classes (20 object categories plus background). Following established SS-SS protocols, we considered two training settings: the "Classic" protocol uses only the high-quality subset of 1,464 images for the labeled pool, while the "Blender" protocol samples labeled images randomly from the entire dataset.

ADE20K dataset [26] offers diverse scenes with dense annotations covering 150 semantic categories. It includes 20,210 training images and 2,000 validation images. Following prior work, we evaluate on label partitions of $1/128$, $1/64$, and $1/32$ of the training set.

4.4.1.2 Implementation Details

Our experiments utilize ResNet-50 and ResNet-101 as encoder backbones across all datasets, coupled with a DeepLabV3+ segmentation head using an output stride of 16 for training efficiency. For generating semantic and binary boundary labels, we employ the on-the-fly (OTF) strategy from Sec. 3.3.3. This approach integrates boundary extraction into the data loading pipeline, guaranteeing consistent boundary widths despite random resizing augmentations.

Key hyperparameters are configured per dataset group:

Cityscapes, BDD100K, and SYNTHIA: We use a base learning rate of 0.01. Loss weights are set to $\lambda_{seg} = 1.0$ and $\lambda_{bdry} = 1.0$, with an overall unsupervised loss weight $\lambda = 1.0$. The EMA decay is $\alpha = 0.99$, and the confidence threshold for segmentation pseudo-labels is $\tau = 0$. τ_{bdry} is set to 0.5. Training uses a mini-batch size of 16 (8 labeled, 8 unlabeled) for 80,000 iterations. The input crop size is 801×801 for Cityscapes and 641×641 for BDD100K and SYNTHIA.

Pascal VOC: Hyperparameters are adjusted: base learning rate is 0.001 and boundary loss weight $\lambda_{bdry} = 0.1$ (other weights remain $\lambda_{seg} = 1.0, \lambda = 1.0$). EMA decay is increased to $\alpha = 0.999$, and the segmentation confidence threshold is raised to $\tau = 0.95$. τ_{bdry} is set to 0.5. We use a mini-batch size of 32 (16 labeled, 16 unlabeled) and train for 120,000

iterations with a 321×321 crop size.

ADE20K: We adopt the same loss weights ($\lambda_{seg}, \lambda_{bdry}, \lambda$), EMA decay ($\alpha$), and thresholds ($\tau$ and τ_{bdry}) as Pascal VOC. However, specific training parameters differ: the base learning rate is 0.01, mini-batch size is 16 (8 labeled, 8 unlabeled), training duration is 80,000 iterations, and the input crop size is 513×513.

Training employs the stochastic gradient descent (SGD) optimizer with a polynomial learning rate decay scheduler. Image augmentations follow UniMatch practices [33], combining weak transformations (resizing, random cropping, horizontal flipping) with strong augmentations (color jittering, grayscale conversion, Gaussian blurring, CutMix). To stabilize the unsupervised loss components, we apply a sigmoid ramp-up schedule: λ_{seg} increases from 0 to its target value over the first 15% of iterations, while λ_{bdry} ramps up linearly throughout the entire training process.

All experiments are implemented using PyTorch and the `mmsegmentation` library [165]. Training is conducted on dual-GPU setups, using either two NVIDIA RTX 3090 or two NVIDIA A6000 GPUs, depending on the memory footprint of the specific benchmark and model configuration.

We use the same experimental setup for all our ablation studies, in Sec. 4.4.5 unless otherwise specified.

4.4.1.3 Evaluation Metrics

Mean intersection over union (mIoU) is the primary evaluation metric for semantic segmentation, but is known to be insensitive to boundary-specific errors [153, 156]. To quantitatively assess boundary improvements, we adopt **Boundary IoU (BIoU)** and **Boundary F1 Score (BF1)**, which explicitly measure boundary alignment and detection quality, respectively.

Boundary IoU (BIoU): BIoU evaluates the overlap between predicted ($\hat{y}$) and ground-truth segmentation masks y restricted only to pixels lying within a narrow-band around object boundaries. Formally, BIoU is computed as

$$\text{BIoU} \;=\; \frac{1}{C}\sum_{c=0}^{C-1} \frac{\sum_n \sum_{(i,j)\in BD_k(y_n)} \left[\hat{y}_{n,c,i,j} \;\wedge\; y_{n,c,i,j}\right]}{\sum_n \sum_{(i,j)\in BD_k(y_n)} \left[\hat{y}_{n,c,i,j} \;\vee\; y_{n,c,i,j}\right]}, \tag{4.16}$$

where $BD_k(y) = y \oplus \text{MinPool}_k(y)$ denotes the binary boundary mask obtained by applying a $k \times k$ min-pooling (stride 1) to the ground-truth segmentation map y and taking the pixel-wise XOR ($\oplus$) with the original mask, and $\wedge$ and $\vee$ denote pixel-wise logical AND and OR, respectively [170]. Because it ignores interior pixels, BIoU directly measures boundary alignment and penalizes both over- and under-segmentation along object edges. It is therefore particularly sensitive to boundary precision but agnostic to interior region accuracy. We use a 5 pixel radius ($k = 11$) for the boundary mask in our experiments.

Boundary F1 Score (BF1): In contrast, BF1 captures both boundary precision (the fraction of predicted boundary pixels that lie near a true boundary) and recall (the fraction of true boundary pixels recovered by the prediction), combining them via the harmonic mean:

$$\text{BF1} \;=\; \frac{1}{C}\sum_{c=0}^{C-1}\frac{2\,\text{Prec}_c\,\text{Rec}_c}{\text{Prec}_c + \text{Rec}_c}, \tag{4.17}$$

where precision and recall are computed by matching predicted and ground-truth boundary pixels within a small tolerance radius [153]. Formally, precision and recall can be defined as:

$$\text{Prec}_c \;=\; \frac{\sum_{n,i,j}\left[\text{BD}_k(\hat{y}_n)_{c,i,j} \;\wedge\; \text{MaxPool}_k\big(\text{BD}_k(y_n)\big)_{c,i,j}\right]}{\sum_{n,i,j}\text{BD}_k(\hat{y}_n)_{c,i,j}}, \tag{4.18}$$

and

$$\text{Rec}_c \;=\; \frac{\sum_{n,i,j}\left[\text{MaxPool}_k\big(\text{BD}_k(\hat{y}_n)\big)_{c,i,j} \;\wedge\; \text{BD}_k(y_n)_{c,i,j}\right]}{\sum_{n,i,j}\text{BD}_k(y_n)_{c,i,j}}. \tag{4.19}$$

BF1 thus provides insight into the trade-off between missing fine boundary details (low recall) and producing spurious edges (low precision), while allowing for minor spatial misalignments. We use a 5 pixel tolerance radius ($k = 11$) for our experiments.

Complementary Insights: BIoU emphasizes exact boundary overlap, making it well suited for tasks requiring crisp, well-aligned boundaries. BF1, by explicitly balancing precision and recall under a spatial tolerance, is more robust to slight shifts but sensitive to fragmented or noisy boundary predictions. Reporting both metrics alongside mIoU therefore provides a holistic view of a model's segmentation performance, particularly its ability to delineate object boundaries accurately.

Table 4.2: Comparison with state-of-the-art methods on Cityscapes using DeepLabV3+ with ResNet-50/101. † denotes reproduced results. Results averaged over three runs.

		ResNet-50			ResNet-101		
		1/16	1/8	1/4	1/16	1/8	1/4
Supervised		63.3	70.2	73.1	66.3	72.8	75.0
U2PL [128]	[CVPR'22]	70.6	73.0	76.3	70.3	74.4	76.5
CVCC [171]	[CVPR'23]	74.9	76.4	77.3	-	-	-
iMAS [121]	[CVPR'23]	74.3	77.4	78.1	-	-	-
AugSeg [120]	[CVPR'23]	73.7	76.5	78.8	75.2	77.8	79.6
CFCG† [123]	[ICCV'23]	75.0	76.9	78.8	76.8	78.4	79.5
LogicDiag [130]	[ICCV'23]	-	-	-	76.8	78.9	**80.2**
CSS [172]	[ICCV'23]	-	-	-	74.0	76.9	77.9
Co-T. [136]	[ICCV'23]	-	76.3	77.1	75.0	77.3	78.7
DAW [173]	[NeurIPS'23]	75.2	77.5	79.1	76.6	78.4	79.4
CorrMatch [122]	[CVPR'24]	-	-	-	77.3	78.5	79.4
RankMatch [129]	[CVPR'24]	75.4	77.7	79.2	77.1	78.6	80.0
DDFP [131]	[CVPR'24]	-	-	-	77.1	78.2	79.9
BoundaryMatch [157]	[IS'24]	-	-	-	76.0	78.4	79.1
UCCL [132]	[ICASSP'25]	75.8	77.2	78.2	-	-	-
NRCR [125]	[NN'25]	-	77.0	77.9	-	78.2	79.5
DMSI [126]	[TMM'25]	75.9	78.0	79.2	77.0	**79.0**	79.8
CW-BASS [115]	[ArXiv'25]	75.0	77.2	78.4	-	-	-
SAMTH (ours)		75.1	77.3	78.9	75.5	77.9	79.7
SAMTH + BoundMatch (ours)		**76.5**	**78.1**	**79.3**	**77.9**	**79.0**	80.1
		+1.4	+0.8	+0.4	+2.4	+1.1	+0.4
UniMatch† [33]	[CVPR'23]	75.2	76.9	77.5	76.6	77.8	79.1
UniMatch + BoundMatch (ours)		76.0	77.6	78.2	77.4	78.4	79.7
		+0.8	+0.7	+0.7	+0.8	+0.6	+0.6
PrevMatch† [124]	[ArXiv'24]	75.7	77.6	78.6	77.4	78.9	79.9
PrevMatch + BoundMatch (ours)		76.3	78.0	79.1	77.8	**79.0**	**80.2**
		+0.6	+0.4	+0.5	+0.4	+0.1	+0.3

4.4.2 Comparisons with State-of-the-Art Methods

4.4.2.1 Cityscapes

Tab. 4.2 summarizes the quantitative results on the Cityscapes dataset across various labeled data splits (1/16, 1/8, 1/4) for both ResNet-50 and ResNet-101 backbones. We have also included boundary metrics in Tab. 4.7 in the Appendix. Compared with recent state-of-the-art methods, SAMTH+BoundMatch achieves competitive performances, outperforming most evaluation protocols. Notably, incorporating the boundary-aware components allows SAMTH+BoundMatch to consistently outperform the SAMTH baseline and achieves 1.4% and 2.4% improvements on the difficult 1/16 split with ResNet-50 and ResNet-101 respectively. We attribute this improvement to the additional regularization provided by the boundary multi-task learning objective, as strong regularization tech-

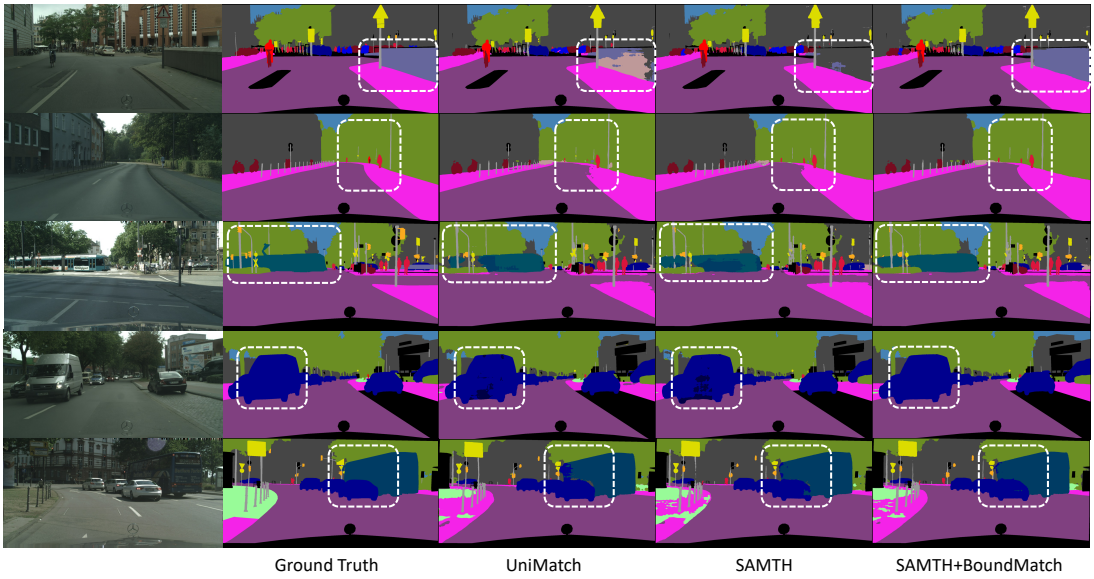


Figure 4.4: Qualitative results on Cityscapes ($^1/_{16}$ split) comparing UniMatch, SAMTH, and SAMTH+BoundMatch. Our method reduces segmentation errors at object boundaries.

niques are often particularly effective in low-supervision scenarios. The qualitative results presented in Fig. 4.4 further illustrate that SAMTH+BoundMatch produces more precise object boundaries and improves segmentation accuracy for challenging classes such as "wall", "pole", "train", and "bus". Our proposed simple baseline, SAMTH, also demonstrates strong performance, achieving competitive results against previous SOTA methods. Overall, these findings establish SAMTH as a strong baseline for urban scene SS-SS, while SAMTH+BoundMatch demonstrates competitive performance against state-of-the-art methods through its effective integration of boundary awareness.

As stated previously, BoundMatch can be integrated with other SS-SS methods, and we have shown results for UniMatch [33] and PrevMatch [124] in Tab. 4.2. For a fair comparison, we reproduced UniMatch and PrevMatch under the same experimental settings as our SAMTH baseline. Incorporating BoundMatch into these methods, we observe consistent performance gains across all splits and backbones. However, UniMatch and PrevMatch already leverage strong consistency regularization techniques which incurs multiple loss functions and may reduce the benefits of the boundary multi-task learning. In future works, we will explore approaches to better balance the regularization signals, but we believe that the results shown here proves the effectiveness of BoundMatch in

Table 4.3: Comparison of SAMTH + BoundMatch with UniMatch on three datasets using DeepLabV3+ (ResNet-50).

Method	BDD100K			SYNTHIA			ADE20K		
	1/64	1/32	1/16	1/64	1/32	1/16	1/128	1/64	1/32
Supervised	40.4	45.3	52.1	63.3	68.5	69.7	7.2	9.9	13.7
UniMatch† [33]	49.2	52.3	56.4	68.5	71.9	72.4	13.6	18.3	23.9
Ours	**52.4**	**53.9**	**57.8**	**69.9**	**73.0**	**74.8**	**15.6**	**19.6**	**25.3**

enhancing SS-SS methods.

Additionally, we reproduced CFCG [123] under our experimental settings to ensure a fair comparison, as the original benchmark partitions may differ from ours. Our results show that SAMTH+BoundMatch outperforms CFCG and the recent CW-BASS [115], highlighting the advantages of our approach. While CFCG and CW-BASS identify boundary regions as difficult areas and adjust loss weighting accordingly (using Laplacian and Sobel operators respectively), they do not aim to improve boundary quality directly. Our method takes a fundamentally different approach: we explicitly learn semantic boundaries through a dedicated auxiliary task and use these learned boundaries not just for identifying difficult regions, but for actively improving both segmentation and boundary predictions through bidirectional fusion modules. This creates a virtuous cycle where better boundaries lead to better segmentation and vice versa.

4.4.2.2 BDD100K, SYNTHIA, and ADE20K

To assess the generalizability of our approach, we extended the evaluation to additional datasets: the urban driving scene datasets BDD100K and SYNTHIA, as well as the challenging academic benchmark ADE20K, known for its diversity and difficulty in semantic segmentation. As shown in Tabs. 4.3 and 4.10, SAMTH+BoundMatch consistently improves over supervised baselines and UniMatch on these datasets. For instance, on BDD100K, our method improves the mean IoU by over 3% for the 1/64 split. These results demonstrate that our boundary-aware learning framework leverages unlabeled data across various domains, including challenging datasets with complex scenes.

We show the qualitative results on the BDD100K dataset in Fig. 4.5. Compared to UniMatch, our SAMTH+BoundMatch generally outputs more accurate segmentations. For example, in the first and last rows, the bus and truck are accurately segmented with BoundMatch. In the second and fifth rows, the road and buildings are accurately

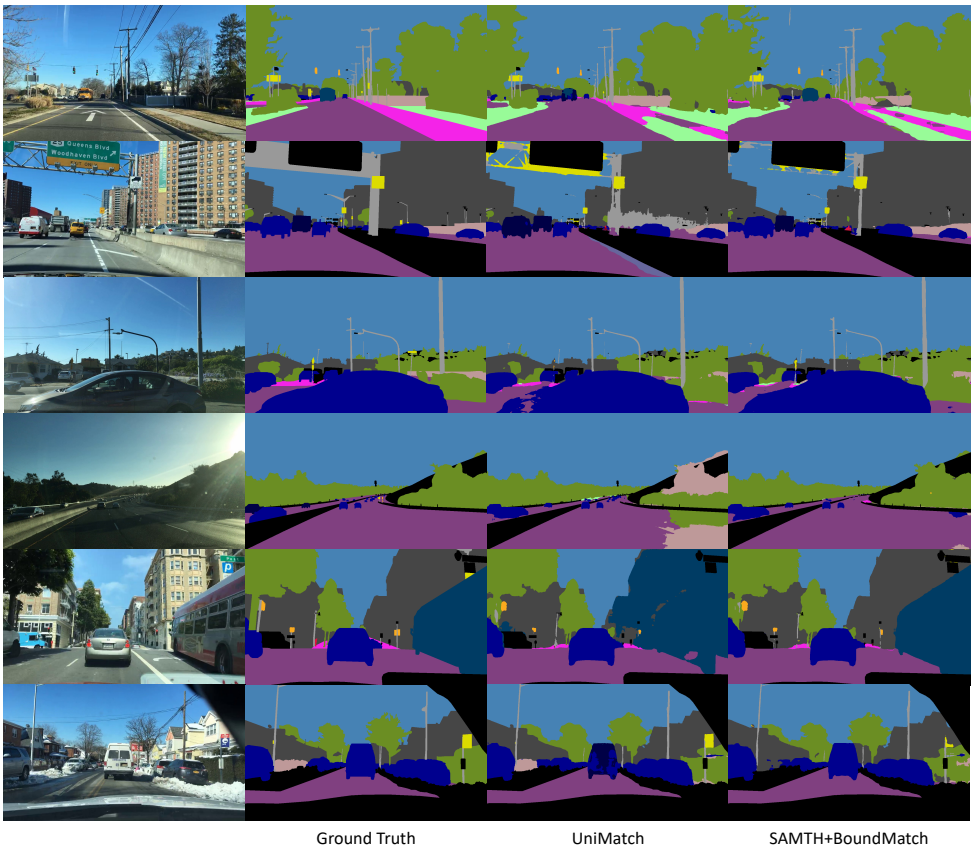


Figure 4.5: Qualitative results on BDD100K (1/64 split) comparing UniMatch and SAMTH +BoundMatch.

segmented with BoundMatch, where UniMatch fails to do so. The fourth row also shows BoundMatch's robustness against sun flares and shadows, being able to segment the road more accurately. However, there remains some challenges when the dataset's labels are ambiguous; such as the "sidewalk" region next to the road might be considered a "vegetation" in the first row.

4.4.2.3 Pascal VOC 2012

We further evaluate our method on the Pascal VOC 2012 dataset using both the *Classic* and *Blender* splits, acknowledging the challenges posed by this benchmark. It is important to note that Pascal VOC 2012 boundary annotations are known to be noisy, and boundary regions are typically treated as "ignore labels" during training and evaluation (see Fig. 4.6).

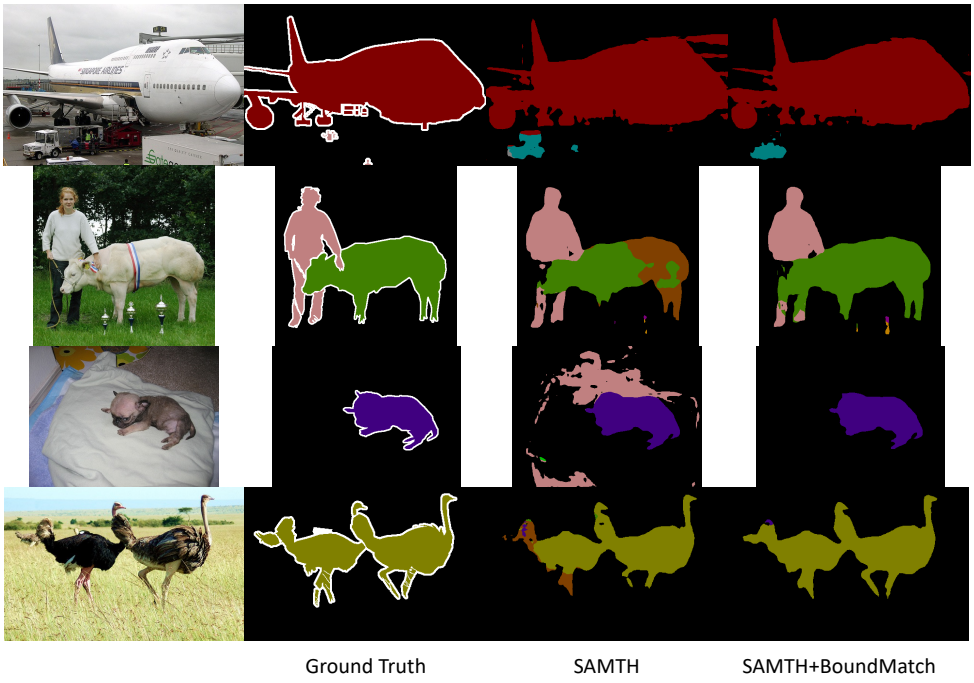


Figure 4.6: Qualitative results on Pascal VOC 2012 (*Classic* 92 split) comparing SAMTH and SAMTH+BoundMatch. White regions in ground truth are "ignore" regions.

Therefore, in this benchmark, we hypothesize that the gains from BoundMatch will be limited compared to the previous high-quality datasets. However, experiment results demonstrate that BoundMatch can still provide improvements in segmentation performance, especially in the low labeled data regime as shown in this section. We believe this is due to the strong regularization of using multi-task CR objectives with label-refinement strategy potentially improving noise robustness as theorized in [125].

Classic Split: Tabs. 4.4 and 4.8 presents the performance comparison on the Classic splits for both ResNet-50 and ResNet-101 backbones. While SAMTH+BoundMatch improves performance relative to earlier CR methods like AugSeg and UniMatch, it does not reach the state-of-the-art results achieved by the most recent approaches, such as PrevMatch and CW-BASS, on this dataset. We hypothesize this performance gap is partly attributable to the weak CR approach of SAMTH, making it less effective for this dataset. To further assess BoundMatch's boundary mechanism independently of our SAMTH baseline, we integrated it with UniMatch and PrevMatch again. When combined with these strong baselines, BoundMatch consistently yields mean IoU gains over the original methods,

Table 4.4: Comparison with recent state-of-the-art methods on the Pascal VOC 2012 dataset using the *Classic* splits. All methods are trained using DeepLabV3+ (ResNet-50/101).

Classic		ResNet-50			ResNet-101		
		92	183	366	92	183	366
Supervised		44.0	52.3	61.7	45.1	55.3	64.8
CVCC [171]	[CVPR'23]	-	-	-	70.2	74.4	77.4
iMAS [121]	[CVPR'23]	-	-	-	68.8	74.4	78.5
AugSeg [120]	[CVPR'23]	64.2	72.2	76.2	71.1	75.5	78.8
ESL [174]	[ICCV'23]	-	69.5	72.6	71.0	74.1	78.1
CSS [172]	[ICCV'23]	68.0	71.9	74.9	-	-	-
Co-T. [136]	[ICCV'23]	73.1	74.7	77.1	75.7	77.7	<u>80.1</u>
DAW [173]	[NeurIPS'23]	68.5	73.1	76.3	74.8	77.4	79.5
CorrMatch [122]	[CVPR'24]	-	-	-	76.4	78.5	79.4
RankMatch [129]	[CVPR'24]	71.6	74.6	76.7	75.5	77.6	79.8
BoundaryMatch [157]	[IS'24]	-	-	-	75.4	77.3	79.3
UCCL [132]	[ICASSP'25]	-	74.1	77.1	-	-	-
NRCR [125]	[NN'25]	-	-	-	<u>77.4</u>	**79.7**	**80.2**
DMSI [126]	[TMM'25]	-	-	-	76.5	77.4	79.7
CW-BASS [115]	[ArXiv'25]	72.8	**75.8**	76.2	-	-	-
SAMTH (ours)		70.7	72.1	76.1	73.2	76.4	78.5
SAMTH + BoundMatch (ours)		72.6	73.8	77.3	76.6	78.3	78.9
		+1.9	+1.7	+1.2	+3.4	+1.9	+0.4
UniMatch† [33]	[CVPR'23]	71.9	72.5	76.0	75.2	77.2	78.8
UniMatch + BoundMatch (ours)		<u>74.2</u>	**75.8**	76.9	76.0	78.2	78.9
		+2.3	+3.3	+0.9	+0.8	+1.0	+0.1
PrevMatch† [124]	[ArXiv'24]	73.4	<u>75.4</u>	<u>77.5</u>	77.0	78.5	79.6
PrevMatch + BoundMatch (ours)		**74.5**	**75.8**	**77.7**	**77.5**	<u>78.7</u>	79.7
		+1.1	+0.4	+0.2	+0.5	+0.2	+0.1

achieving results that are competitive with, or reach, current state-of-the-art performance (Tab. 4.4).

Blended Split: This split contains potentially even more noisy GT labels, particularly around object boundaries. Tabs. 4.5 and 4.9 shows trends similar to the Classic split. Again, the combinations of BoundMatch with UniMatch or PrevMatch achieve performance competitive with recent SOTA methods. Furthermore, BoundMatch outperforms CFCG [123] in this split, demonstrating its robustness even in the presence of noisy boundary annotations.

Collectively, these Pascal VOC results underscore the benefit of incorporating explicit boundary modeling in SS-SS frameworks. While BoundMatch does not consistently outperform the best methods on this dataset, it demonstrates its potential to enhance segmentation performance when integrated with strong SS-SS baselines attributed to its

Table 4.5: Comparison with recent state-of-the-art methods on the Pascal VOC 2012 dataset using the *Blender* splits. All methods are trained using DeepLabV3+ (ResNet-50).

Blender		1/16	1/8	1/4
Supervised		62.4	68.2	72.3
CVCC [171]	[CVPR'23]	74.5	76.1	76.4
iMAS [121]	[CVPR'23]	74.8	76.5	77.0
AugSeg [120]	[CVPR'23]	74.7	76.0	77.2
CFCG† [123]	[ICCV'23]	75.2	76.7	77.1
DAW [173]	[NeurIPS'23]	76.2	77.6	77.4
RankMatch [129]	[CVPR'24]	76.6	77.8	78.3
IpxMatch [175]	[IJCNN'24]	74.5	74.9	75.0
NRCR [125]	[NN'25]	76.6	**78.2**	78.7
DMSI [126]	[TMM'25]	76.3	76.9	77.2
SAMTH (ours)		73.5	75.8	76.9
SAMTH + BoundMatch (ours)		76.3	77.2	78.1
		+2.8	+1.4	+1.2
UniMatch† [33]	[CVPR'23]	76.0	76.9	76.7
UniMatch + BoundMatch (ours)		76.6	77.9	**78.9**
		+0.6	+1.0	+2.2
PrevMatch† [124]	[ArXiv'24]	75.8	77.0	77.7
PrevMatch + BoundMatch (ours)		**76.7**	77.8	78.8
		+0.9	+0.8	+1.1

Table 4.6: Comparison with recent state-of-the-art methods using DPT with DINOv2 backbones on the Cityscapes dataset.

Methods		**DINOv2-S**		**DINOv2-B**	
		1/16	1/8	1/16	1/8
Supervised		77.2	80.2	80.8	82.7
UniMatch-V2 [167]	[TPAMI'25]	80.6	81.9	83.6	84.3
SegKC [176]	[CoRR'25]	81.2	82.4	–	–
SAMTH (ours)		80.8	82.0	83.2	84.1
SAMTH + BoundMatch (ours)		**81.5**	**82.9**	**84.0**	**84.8**
		+0.7	+0.9	+0.8	+0.7

simple yet effective boundary-aware learning mechanism.

4.4.2.4 Recent Benchmarks with Transformer Backbones

While DeepLabV3+ with ResNet backbones has been the standard benchmark for SSSS methods, recent works have begun exploring vision transformers to leverage their superior representation learning capabilities [167]. Following this direction, we evaluate

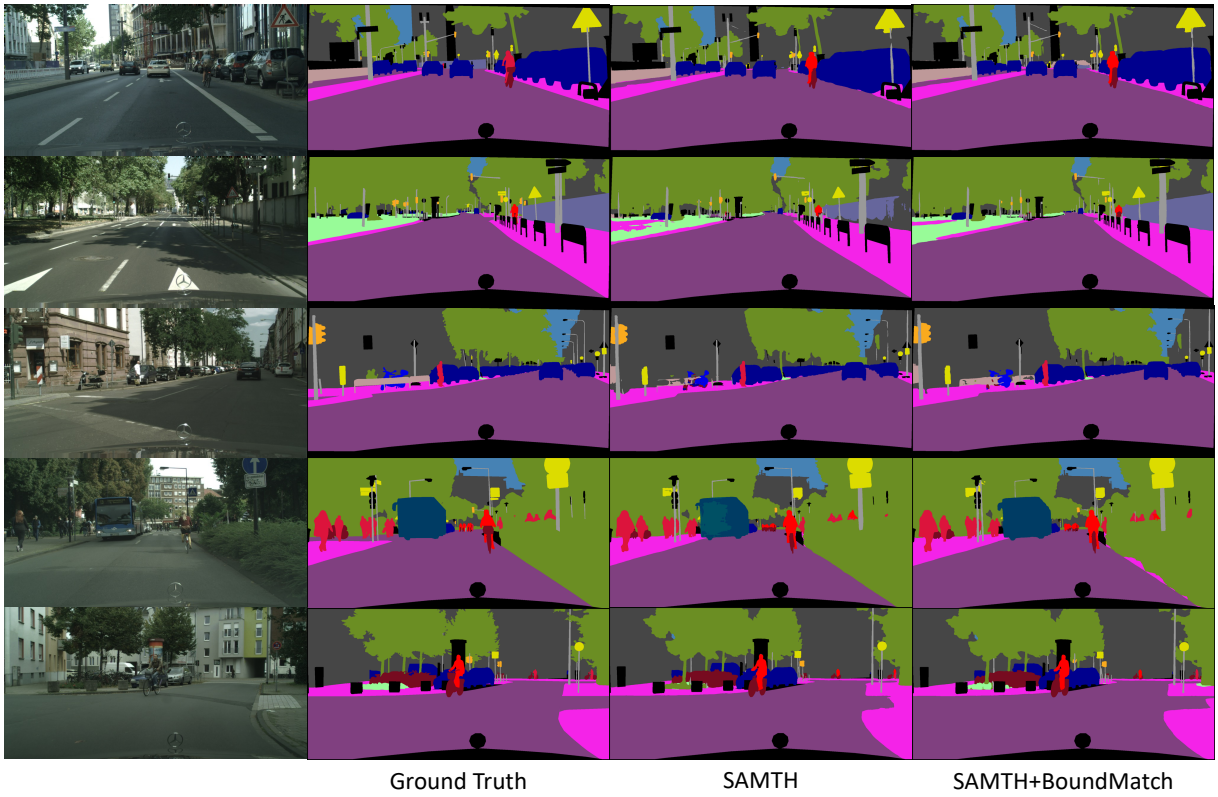


Figure 4.7: Qualitative results on Cityscapes (1/16 split) using DPT with DINOv2-S encoder.

BoundMatch using DPT [177] with DINOv2 [168] pretrained weights, a foundation model that has demonstrated strong performance across various vision tasks. Adapting BoundMatch to DPT requires minimal modifications, with implementation details provided in Appendix A.9.

As shown in Tab. 4.6, SAMTH+BoundMatch outperforms current state-of-the-art methods, UniMatch-V2 [167] and SegKC [176], across both DINOv2-S and DINOv2-B backbones on the challenging 1/16 and 1/8 splits. These results demonstrate that BoundMatch scales to modern foundation models while maintaining its boundary-aware advantages.

We show the qualitative results on the Cityscapes dataset using DINOv2-S pretrained ViT in Fig. 4.7. DINOv2-S is a strong foundation model that has been pre-trained on a large dataset, and it is known to perform well on various vision tasks. As shown in the samples, BoundMatch is able to improve the segmentation quality compared to SAMTH.

Table 4.7: Boundary Metrics for Cityscapes Benchmark.

	ResNet-50				ResNet-101			
	1/16		1/8		1/16		1/8	
	BIoU	BF1	BIoU	BF1	BIoU	BF1	BIoU	BF1
SAMTH	54.8	59.3	56.8	61.8	55.8	61.4	57.9	63.1
+ BoundMatch	56.6	63.2	57.8	64.7	58.3	65.2	59.4	66.7
UniMatch	55.3	60.7	56.6	63.5	56.9	63.3	57.8	65.1
+ BoundMatch	56.0	62.6	57.5	64.2	57.6	64.8	58.8	66.4
PrevMatch	55.3	60.7	57.0	63.3	57.5	63.7	58.8	65.6
+ BoundMatch	56.8	63.4	58.2	64.4	58.0	65.4	58.9	66.4

Table 4.8: Boundary Metrics for Pascal VOC Classic Split.

	ResNet-50				ResNet-101			
	92		183		92		183	
	BIoU	BF1	BIoU	BF1	BIoU	BF1	BIoU	BF1
SAMTH	65.2	55.0	65.6	53.4	70.7	59.5	70.5	59.6
+ BoundMatch	67.2	56.9	68.3	57.6	70.9	61.3	72.7	62.5
UniMatch	64.9	55.2	65.2	56.1	69.3	57.4	69.9	60.1
+ BoundMatch	68.2	57.7	68.6	57.5	70.2	59.0	72.1	61.7
PrevMatch	67.1	57.1	67.3	57.2	69.8	60.7	71.8	61.8
+ BoundMatch	69.0	58.0	68.7	57.4	71.0	61.2	72.4	61.8

Table 4.9: Boundary Metrics for Pascal VOC Blender Split.

	1/16		1/8	
	BIoU	BF1	BIoU	BF1
SAMTH	68.5	59.8	69.2	59.1
+ BoundMatch	68.8	60.1	70.0	59.9
UniMatch	66.6	57.3	68.0	57.2
+ BoundMatch	69.4	59.5	70.4	60.4
PrevMatch	68.4	57.7	69.3	58.2
+ BoundMatch	68.7	59.1	69.4	59.3

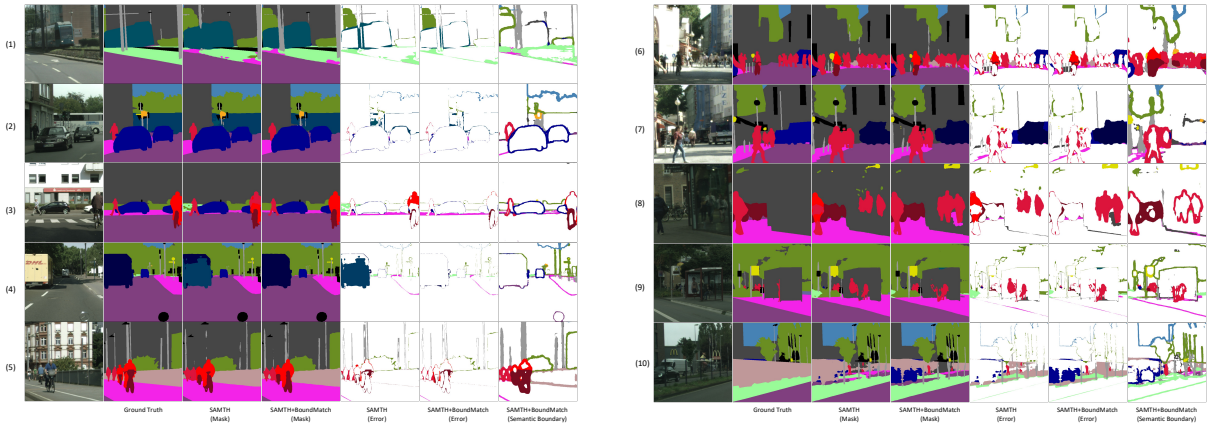
Table 4.10: Boundary Metrics for BDD100K, SYNTHIA, and ADE20K.

	BDD100K				SYNTHIA				ADE20K			
	1/64		1/32		1/64		1/32		1/128		1/64	
	BIoU	BF1	BIoU	BF1	BIoU	BF1	BIoU	BF1	BIoU	BF1	BIoU	BF1
Supervised	31.8	34.2	35.9	37.4	55.3	68.7	60.3	74.8	5.8	6.9	7.6	8.3
UniMatch	37.9	42.5	41.1	45.5	60.8	74.5	63.6	78.1	9.2	10.2	12.7	12.8
Ours	39.8	46.1	41.3	47.3	62.2	76.7	64.8	80.1	12.0	13.0	14.9	16.2

For example, the first row shows that BoundMatch is able to delineate the road/sidewalk and the car around their boundaries more accurately. Far away objects such as the wall and the thin poles are also better segmented with BoundMatch. Other samples also provide similar observations.

4.4.3 BIoU and BF1 Evaluation of Benchmark Results

In Tabs. 4.7 to 4.10, we show the boundary evaluation metrics (BIoU and BF1) for each of the datasets used in the main paper. While comprehensive boundary evaluation of all prior methods would be ideal, most existing SS-SS works do not provide open-source implementations or pretrained models, making it infeasible to fairly compute these specialized metrics retrospectively. We therefore present boundary metrics for the representative methods we could reliably reproduce (SAMTH, UniMatch, PrevMatch) alongside their BoundMatch-enhanced versions. Despite this limitation, the results clearly demonstrate that BoundMatch consistently improves boundary quality across different



(**a**) Improved Cases. (**b**) Failure Cases.

Figure 4.8: Additional qualitative comparisons on Cityscapes (1/16): (a) improved cases and (b) failure cases.

datasets and baseline models, validating our approach's effectiveness in enhancing boundary delineation.

4.4.4 Additional Qualitative Results on Cityscapes

In Fig. 4.8a, we present qualitative comparisons between SAMTH and SAMTH+BoundMatch on the Cityscapes dataset. For each example, we display the segmentation masks, prediction errors, and semantic boundary predictions (for SAMTH+BoundMatch only). BoundMatch reduces segmentation errors near object boundaries, as demonstrated in examples (1) through (5). Notably, BoundMatch better preserves object continuity: in (3) and (4), it accurately segments complete "terrain", "rider", and "truck" regions rather than producing fragmented results. Despite the inherent challenge of segmenting thin structures, BoundMatch successfully captures "poles" in examples (1) and (5), where the baseline typically struggles. The strong alignment between predicted boundaries and actual object edges indicates that the segmentation head leverages boundary information to generate more accurate masks.

In Fig. 4.8b, we show notable failure cases for SAMTH+BoundMatch on the Cityscapes dataset. We believe thin objects like "poles" show limited improvement for two main reasons: first, small objects create imbalanced supervision signals, and second, annotation inconsistencies lead to false positives. For instance, images (6) and (7) show the same scene

Table 4.11: Component analysis of our framework.

			ResNet-50						ResNet-101					
			1/16			1/8			1/16			1/8		
BCRM	**BSF**	**SGF**	IoU	BIoU	BF1	IoU	BIoU	BF1	IoU	BIoU	BF1	IoU	BIoU	BF1
–	–	–	75.1	54.8	59.3	77.3	56.8	61.8	75.5	55.8	61.4	77.9	57.9	63.1
✓	–	–	75.9	55.3	60.2	77.6	56.9	62.3	76.9	56.9	62.1	78.2	58.5	64.5
✓	✓	–	76.2	56.0	62.0	77.5	57.2	62.8	76.1	57.3	64.5	78.2	59.0	65.0
✓	–	✓	76.1	55.7	61.6	77.6	57.0	62.4	76.6	57.0	63.9	77.9	58.8	64.9
✓	✓	✓	**76.5**	**56.1**	**62.7**	**78.1**	**58.1**	**64.7**	**77.9**	**58.3**	**65.2**	**79.0**	**59.4**	**66.7**

from different samples, but poles are annotated only in (7), not in (6). This inconsistency causes false positives, particularly for SAMTH+BoundMatch, which aims to precisely segment boundaries. BoundMatch also struggles with challenging scenarios like reflections, as seen in (8). Additionally, transparent or semi-transparent structures pose difficulties: in (9) and (10), see-through "fences" and "buildings" challenge BoundMatch as it attempts to accurately segment partially occluded objects behind them.

4.4.5 Ablation Studies and Insights

4.4.5.1 Individual efficacy of the proposed components

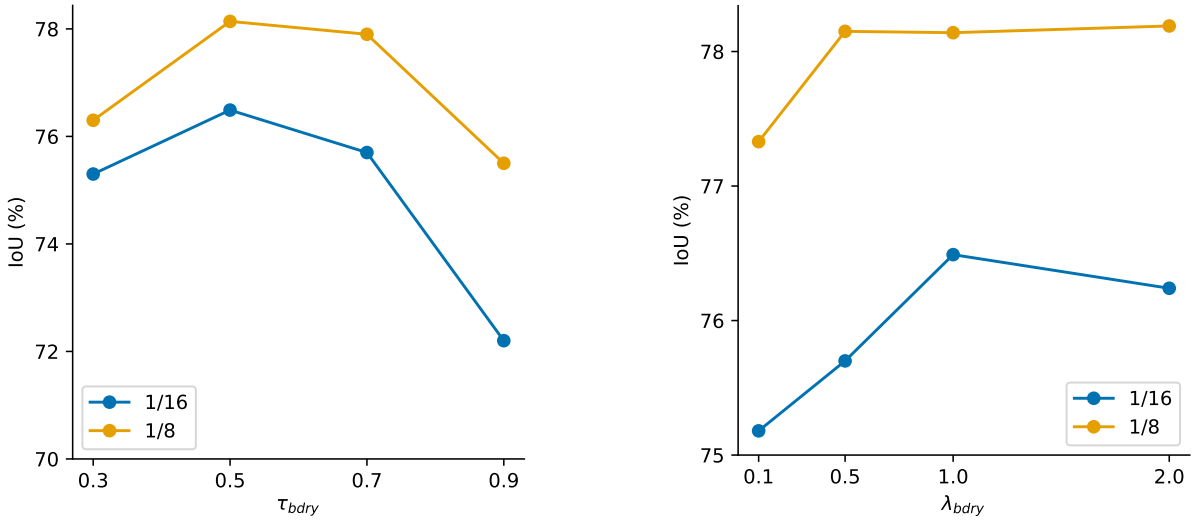
To quantify the individual contributions of each proposed component, we conduct systematic ablation experiments evaluating the Boundary Consistency Regularization Module (BCRM), Boundary-Semantic Fusion (BSF), and Spatial Gradient Fusion (SGF) modules. Tab. 4.11 reports mIoU, BIoU, and BF1 across multiple evaluation protocols.

The addition of our core BCRM framework alone results in notable gains in both segmentation (0.3–1.4%) and boundary quality metrics (0.1–1.1% for BIoU and 0.5–1.4% for BF1). Further incorporating the BSF module builds upon this, enhancing performance by integrating learned boundary cues into the segmentation head. BCRM with SGF has similar gains, indicating that refining boundary predictions using segmentation features is also beneficial. Finally, using both BSF and SGF modules provides complementary benefits; the strategy of refining boundary predictions using spatial gradients derived from the segmentation mask leads to the highest overall performance across all protocols (0.8–2.4% gains for mIoU).

In Tab. 4.12, we show the component analysis of our BoundMatch framework on Pascal VOC Classic 92 with ResNet-50 backbone. We can see that each component of

Table 4.12: Component analysis on Pascal VOC 2012 val set using ResNet-50.

			1/16			1/8		
BCRM	**BSF**	**SGF**	IoU	BIoU	BF1	IoU	BIoU	BF1
–	–	–	73.5	67.9	59.0	75.8	68.7	59.2
✓	–	–	75.6	68.3	59.2	76.8	69.1	59.3
✓	✓	–	75.9	68.1	59.3	76.5	69.3	60.0
✓	–	✓	75.8	68.0	59.2	76.6	68.8	59.8
✓	✓	✓	**76.3**	**68.9**	**59.8**	**77.2**	**69.7**	**60.3**



(**a**) τ_{bdry} (**b**) λ_{bdry}

Figure 4.9: Hyperparameter analysis: (a) boundary threshold τ_{bdry} and (b) boundary-loss weight λ_{bdry} on Cityscapes using ResNet-50.

BoundMatch contributes to the overall performance. The most contribution comes from the Boundary-aware Consistency Regularization Module (BCRM), which improves over 1.0% compared to the SAMTH baseline. Incremental, but gradual improvements can be seen when introducing BSF and SGF, and results in over 1.4–2.8% improvements.

4.4.5.2 Analysis of Hyperparameters

Having established the efficacy of individual components, we now examine the sensitivity to key hyperparameters. Fig. 4.9 presents the impact of two key hyperparameters introduced by BoundMatch: the boundary threshold τ_{bdry} and boundary loss weight λ_{bdry} on the Cityscapes dataset using ResNet-50 backbone. For the boundary threshold, we find that

Table 4.13: Binary vs. Multi-Label Boundaries. "Derived" uses boundaries from segmentation pseudo-labels; "Learned" uses predicted boundaries directly.

	1/16			1/8		
	IoU	BIoU	BF1	IoU	BIoU	BF1
SAMTH	75.1	54.8	59.3	77.3	56.8	61.8
+ Binary (Derived)	75.3	54.6	59.4	77.3	56.9	62.0
+ Binary (Learned)	75.9	55.4	60.1	77.6	57.3	62.5
+ Multi-Label (Derived)	75.8	55.4	60.8	77.4	57.1	62.5
+ Multi-Label (Learned)	**76.5**	**56.1**	**62.7**	**78.1**	**58.1**	**64.7**

$\tau_{bdry} = 0.5$ yields optimal performance across both data splits. This moderate threshold balances two factors: boundary predictions typically have low probability values due to the sparsity of boundary pixels, yet setting the threshold too high would exclude valuable boundary information from the consistency regularization process.

Regarding the boundary loss weight λ_{bdry}, optimal performance is achieved with values around $\lambda_{bdry} = 1.0$ for both splits. Interestingly, the 1/8 split shows greater robustness to variations in λ_{bdry} compared to the 1/16 split, suggesting that increased labeled data provides more stable training dynamics that are less sensitive to the precise weighting of the boundary loss.

4.4.5.3 Effect of boundary types

An important design decision in our method is what types of boundaries should be used: derived versus learned, and binary versus semantic (multi-label). As shown in Tab. 4.13, using learned multi-label boundaries obtains the highest improvements across both evaluation protocols, outperforming baseline with binary boundaries and derived boundaries from segmentation pseudo-labels. For 1/16, IoU improves by 1.4%, BIoU by 1.3%, and BF1 by 3.4%, indicating that learned class-aware boundary supervision provides benefits over the baseline approach.

Using derived boundaries shows minimal improvements over SAMTH compared to using learned boundaries for consistency regularization. We believe this is because generating boundaries (*e.g.* via Laplacian operators) from often noisy segmentation pseudo-labels propagates this noise to the boundary predictions, limiting their effectiveness. In contrast, learning boundaries directly from dedicated boundary predictions allows the model to focus on the geometric structures of the boundaries independent of the segmentation task,

Table 4.14: Instance-aware (IS) vs. Non-instance-aware (nonIS) boundaries for consistency regularization.

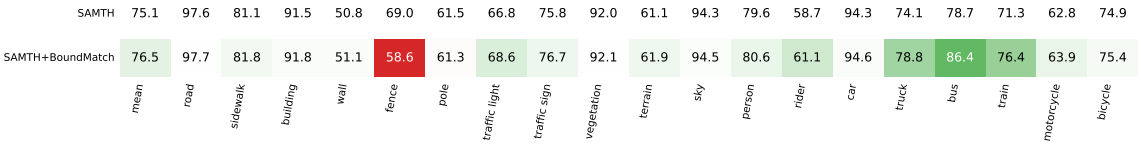
	1/16			1/8		
	IoU	BIoU	BF1	IoU	BIoU	BF1
SAMTH	75.1	54.8	59.3	77.3	56.8	61.8
+ Multi-Label (nonIS)	76.0	55.7	62.3	77.8	57.9	64.3
+ Multi-Label (IS)	76.5	56.1	62.7	78.1	58.1	64.7

and the ability to threshold boundaries by confidence (τ_{bdry}) allows the model to learn more robust boundary representations. Furthermore, our bidirectional fusion allows the boundary and segmentation heads to mutually benefit from each other, where SGF module enables learning with segmentation mask cues.

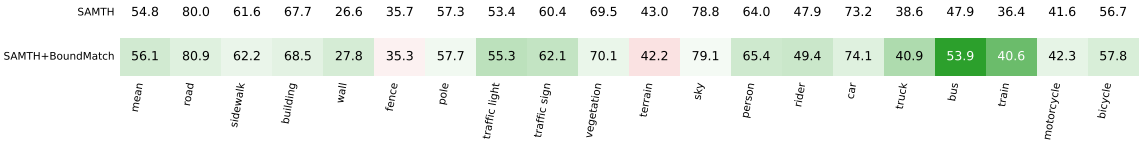
We compare instance-sensitive (IS) and non–instance-sensitive (non-IS; or instance-insensitive) boundary targets as the auxiliary task for consistency regularization. As shown in Tab. 4.14, IS boundaries yield consistently better results across metrics and evaluation protocols. We hypothesize that explicitly separating same-class instances makes the auxiliary task both harder and more distinct from semantic segmentation (which is not instance-sensitive), encouraging the network to learn sharper edge cues—especially in crowded scenes—and to avoid class-interior shortcuts. From a regularization standpoint, IS boundaries diversify the supervision signals across the two heads, reducing redundancy in their optimization dynamics; by contrast, non-IS boundaries largely mirror the semantics of the main task and therefore induce more similar (and less helpful) learning dynamics.

4.4.5.4 Per-Class Performance

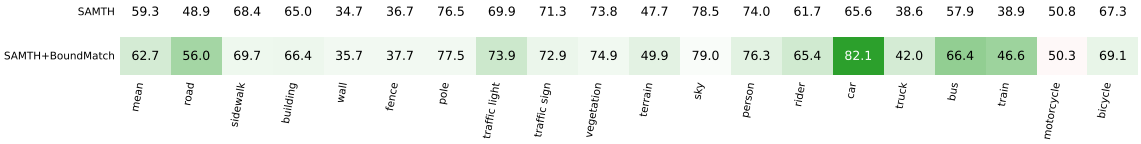
To understand how these boundary formulations affect different object categories, we analyze per-class performance. Here we present the per-class performance comparison between SAMTH and BoundMatch for Cityscapes 1/16 split in Figs. 4.10a to 4.10c for IoU, BIoU, and BF1 metrics respectively. For IoU, we observe that BoundMatch improves the performance on most of the classes, especially on vehicles (*e.g.* car, truck, bus, and train) and people (*i.e.* person and rider). While thin and small objects (*e.g.* traffic light and traffic sign) also see improvements, SAMTH+BoundMatch seems to have less quantitative effect on classes like pole, although we see qualitative improvements for poles in Fig. 4.8a This may be due to pixel imbalance issues, as poles occupy a very small fraction of the



(**a**) Per-class IoU results



(**b**) Per-class BIoU results



(**c**) Per-class BF1 results

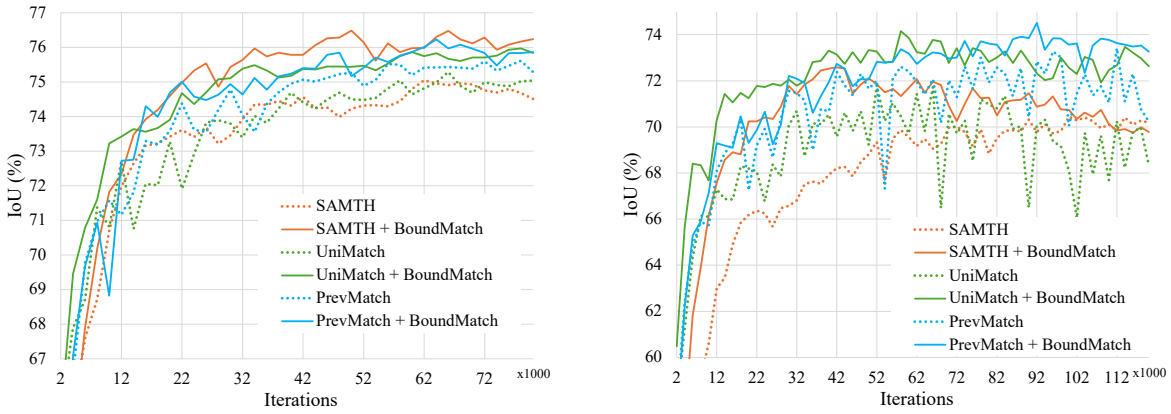
Figure 4.10: Per-class performance on Cityscapes (1/16, ResNet-50). Green: improvements; red: degradation.

image area, but it could also be caused by noisy annotations in the ground-truth which amplifies false positives as shown in Fig. 4.8b due to BoundMatch focusing more on the details. The fence category presents an interesting exception with decreased IoU but maintained boundary metrics, likely due to annotation ambiguities where fences are labeled as solid regions despite their sparse structure as shown in Fig. 4.14. BoundMatch improves BIoU and BF1 metrics for most object categories, though minor degradations are observed for certain classes with ambiguous boundaries (*e.g.* fence, terrain), which aligns with the challenges discussed in our limitations.

4.4.5.5 Boundary Evaluation

Fig. 4.11 presents quantitative and qualitative results comparing boundary predictions from BCRM alone versus the full BoundMatch framework (BCRM+BSF+SGF). Looking at the mean F-measure (MF) at optimal dataset scale (ODS), we observe that BoundMatch consistently outperforms BCRM alone across various experiment protocols, indicating that the SGF refinement module improves boundary quality. Qualitatively, BCRM alone

Figure 4.11: Semantic boundary prediction comparison. (a) MF (ODS) scores for BCRM vs. BoundMatch. (b) Qualitative comparison showing SGF produces sharper boundaries than BCRM alone.



(**a**) Cityscapes 1/16. (**b**) Pascal VOC Classic 92.

Figure 4.12: Pseudo-label accuracy (mIoU) vs. training iterations on (a) Cityscapes and (b) Pascal VOC Classic 92.

produces thicker boundaries with more artifacts, while the full BoundMatch framework generates sharper and cleaner boundaries. This improvement demonstrates the benefit of SGF's spatial gradient fusion in refining boundary predictions for more reliable pseudo-labels.

4.4.5.6 Pseudo-Label Accuracy

In Fig. 4.12, we plot the pseudo-label accuracy (IoU) on the validation set against training iterations for Cityscapes and Pascal VOC Classic 92. The training curves reveal BoundMatch's noise-robustness properties. On Cityscapes Fig. 4.12a, all methods (with and without BoundMatch) show monotonically increasing pseudo-label accuracy without

Table 4.15: Comparison between BoundMatch and BoundaryMatch. † denotes reproduced results. All models use DeepLabV3+ with ResNet-101.

Methods	VOC (92)	Cityscapes (1/16)		
	IoU	IoU	BIoU	BF1
BoundaryMatch	75.4	76.0	–	–
UniMatch	75.2	76.6	57.0	63.4
BoundaryMatch†	75.5	76.5	56.7	63.5
+ Multi-Label	75.5	76.7	57.1	63.6
UniMatch + BoundMatch (ours)	76.0	77.4	58.0	64.5
SAMTH + BoundMatch (ours)	**76.6**	**77.9**	**58.5**	**65.3**

late-stage degradation, indicating that consistency regularization prevents overfitting to early noisy pseudo-labels. On Pascal VOC Fig. 4.12b, where boundary annotations are often noisier, BoundMatch notably stabilizes the fluctuating pseudo-label accuracy observed in baseline methods, particularly for UniMatch and PrevMatch. SAMTH+BoundMatch shows improvements over SAMTH in terms of accuracy, but overfitting can be seen after 40K iterations, indicating the need for stronger regularizations which are equipped in UniMatch and PrevMatch.

This stabilization can be attributed to the multi-task learning framework: when segmentation pseudo-labels become noisy, the boundary task, learned from hierarchical features and refined through SGF, provides a strong regularization signal to deter the model in overfitting. This aligns with the noise-robust learning principles identified in [125], where combining multiple complementary views and regularization mechanisms proves more effective than single approaches.

4.4.5.7 Comparisons with BoundaryMatch

We compared BoundMatch against BoundaryMatch [157], a recent method that also utilizes boundary CR for SS-SS. For a fair comparison based on our experimental setup, we reimplemented BoundaryMatch and trained it using identical settings. Quantitative results on Pascal VOC and Cityscapes dataset using ResNet-101 backbone are presented in Tab. 4.15 which also includes the reported mIoU for BoundaryMatch.

BoundaryMatch achieves higher mIoU compared to UniMatch baseline in Pascal VOC, but does not seem to improve IoU on Cityscapes. With multi-label boundaries, the metrics do surpass UniMatch slightly. UniMatch+BoundMatch achieves 0.9% mIoU

Table 4.16: Effect of Harmonious Batch-Norm (HBN) on Pascal VOC (92 images) and Cityscapes (1/16) using ResNet-50.

Methods	VOC (92)		Cityscapes (1/16)	
	IoU	Δ	IoU	Δ
Mean-Teacher	51.7		66.1	
+ HBN	61.5	+9.8	71.7	+5.6
CutMix-MT	52.2		68.3	
+ HBN	68.6	+16.4	73.2	+4.9
ReCo	64.8		71.4	
+ HBN	67.5	+2.7	75.1	+3.7
U2PL	68.0		70.3	
+ HBN	69.1	+1.1	72.4	+2.1
iMAS	68.8		74.3	
+ HBN	69.7	+0.9	75.1	+0.8
AugSeg	71.1		73.7	
+ HBN	74.1	+3.0	75.1	+1.4
SAMTH (no HBN)	62.8		69.4	
SAMTH	70.7	+7.9	75.1	+5.7

and SAMTH+BoundMatch achieves an even higher 1.4% compared to BoundaryMatch on Cityscapes. Furthermore, boundary metrics (BIoU and BF1) also show notable improvements with BoundMatch, indicating that our method is more effective in capturing boundary details, while BoundaryMatch has similar boundary metrics to UniMatch.

BoundMatch's improved performance stems from four key technical advantages that address the limitations of simpler consistency approaches: (1) *semantic* boundary supervision, providing richer class-specific edge cues compared to binary boundaries; (2) the use of *learned* boundaries as pseudo-labels for consistency regularization; (3) boundary heads leveraging *hierarchical features* from the backbone with *deep supervision*, known to be effective for capturing multi-scale boundary details [82]; and (4) dedicated *fusion modules* (BSF, SGF) that integrate boundary information back into the segmentation process and refine boundary predictions. This suggests that simpler boundary consistency regularization might not fully exploit the potential of boundary cues, whereas our multi-faceted approach proves more effective in the SS-SS context.

4.4.5.8 Effect of Harmonious BN update strategy

To further examine the effect of our Harmonious Batch Normalization (HBN) update strategy, we compare different BN update approaches in Tab. 4.16. Methods like CutMix-

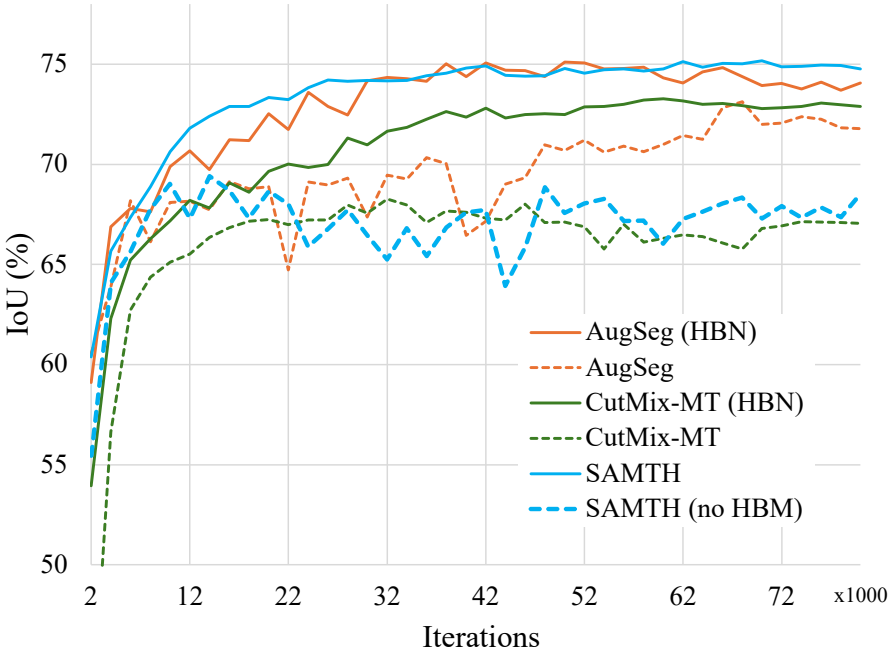


Figure 4.13: Training curves on Cityscapes (1/16, ResNet-50). Bold lines: with HBN; dotted lines: without HBN.

MT, ReCo, and U2PL also update the teacher's BN statistics via forward passes. However, unlike HBN (which processes the complete batch through the teacher specifically for BN updates, as described in Sec. 4.3.3), these methods often introduce discrepancies between the inputs used for the teacher's BN statistics and the student's training inputs. Other approaches, such as iMAS and AugSeg, update the teacher's BN buffers via EMA based on the student's statistics. In our experiments, incorporating HBN improves segmentation performance for most tested baseline methods, with mIoU gains ranging from 0.8–16.4%. For example, applying HBN to SAMTH results in a 5.7% absolute improvement in mean IoU on Cityscapes (from 69.4% to 75.1%). The training curves in Fig. 4.13 provide further evidence, indicating that not only does HBN result in higher final performance, but it also stabilizes the training process.

HBN is not the main focus of this work, but it is a simple yet effective strategy that can be applied to any SS-SS method that uses a teacher-student framework with EMA updates. It led SAMTH, a simple SDA method, to achieve competitive performance on the benchmarks.

Table 4.17: Computational cost comparison during training (time, memory) and inference (FLOPs, parameters, FPS).

Method	Accuracy (%)			Training cost		Inference cost		
	IoU	BIoU	BF1	Time	Mem	FLOPs	Params	FPS
UniMatch	75.3	55.1	61.0	1.21s	20.6GB	191G	40.5M	23.4
PrevMatch	75.7	55.3	60.7	1.56s	21.4GB	191G	40.5M	23.4
SAMTH	75.1	54.8	59.3	0.92s	9.0GB	191G	40.5M	23.4
+ BCRM	75.9	55.3	60.2	1.24s	13.2GB	191G	40.5M	23.4
+ BCRM + BSF	76.2	56.0	62.0	1.32s	13.5GB	192G	40.6M	19.0
+ BCRM + SGF	76.1	55.7	61.6	1.45s	15.0GB	191G	40.5M	23.4
+ BoundMatch	76.5	56.1	62.7	1.59s	16.5GB	192G	40.6M	19.0

4.4.5.9 Computational Cost

Tab. 4.17 analyzes the computational overhead of BoundMatch's individual components. BCRM adds 0.32s and 4.2GB for multi-scale boundary processing, BSF contributes minimal overhead (0.08s, 0.3GB), while SGF requires 0.21s and 1.8GB for gradient computations. The complete framework efficiently combines these modules with shared computations, resulting in a total training overhead of 0.67s and 7.5GB compared to SAMTH baseline, primarily due to additional boundary detection task and the fusion modules. Despite this increase, BoundMatch remains more memory-efficient than UniMatch (20.6GB) and has similar per-iteration time as PrevMatch (1.56s) while achieving accuracy gains. Optionally, if training costs are a concern, one can disable BSF and only use BCRM+SGF which still provides 1.0% mIoU improvement over SAMTH with only 0.53s and 6.0GB overhead.

At inference, BoundMatch introduces minimal overhead (1G FLOPs, 0.1M parameters), reducing FPS from 23.4 to 19.0 for 1.4% mIoU and up to 3.4% boundary metric improvements. BSF slightly increases FLOPs and reduces FPS due to additional convolutions required for feature fusion, thus if the trade-off is not desired, one can use the BCRM+SGF combination which has no impact on inference cost. BoundMatch's modular design allows users to change feature fusion modules entirely (*e.g.* use attention-based fusion), which may offer different trade-offs between accuracy and efficiency. Investigations into better feature fusion strategies that are computationally efficient are left for future work.

4.4.5.10 BoundMatch in other domains

In Tab. 4.18, we show the results on the ACDC dataset [178] which is a medical image segmentation dataset. We follow the same experimental setup as in [33] and use UNet

Table 4.18: Results on the ACDC dataset (Dice Similarity Coefficient) using UNet architecture.

	Cases		
	1	3	7
Supervised	28.5	41.5	62.5
CPS [138]	–	61.0	81.5
UniMatch [33]	85.4	88.9	89.9
UniMatch + BoundMatch	85.0	89.0	89.6

Table 4.19: Results on the LoveDA dataset.

	1/32			1/16		
	IoU	BIoU	BF1	IoU	BIoU	BF1
Supervised	45.2	27.0	24.8	45.7	29.1	25.3
UniMatch	48.9	28.7	28.3	48.8	29.4	29.7
SAMTH	49.8	29.7	29.8	50.1	30.7	29.6
SAMTH + BoundMatch	51.1	30.3	30.8	52.1	31.3	31.1

as the base model. To integrate BoundMatch, we took the hierarchical features from the encoder to the boundary detection head following Sec. A.8. We found that BoundMatch does not improve the performance of UniMatch on this dataset. This could be due to the fact that our architectures and losses for the introduced BoundMatch would need to be further optimized for medical images, which is out of the scope of this thesis.

In Tab. 4.19, we show the results on the LoveDA dataset [179] which is a remote sensing image segmentation dataset. We use DeepLabV3+ as the base model and produced results on 1/32 and 1/16 splits. We can see that BoundMatch consistently improves the performance of SAMTH and UniMatch on this dataset, providing potentials that BoundMatch can be generalized to other domains.

4.4.6 More Real-World SS-SS Setting

Inspired by the evaluation protocol in [167], we assess SS-SS in a realistic scenario: using all 2,975 Cityscapes training images as labeled data and the additional 19,997 "extra" images as truly unlabeled data. In this setting, the fully supervised baseline is already strong, making further gains challenging yet highly relevant for practical deployment. As shown in Tab. 4.20, SAMTH+BoundMatch achieves consistent improvements over both SAMTH and UniMatch, particularly in boundary-specific metrics (BIoU and BF1), demonstrating

Table 4.20: Real-world setting using all 2975 labeled Cityscapes images with 19997 unlabeled *extra* images (ResNet-50).

Method	IoU	BIoU	BF1
Supervised only	77.58	57.73	60.06
UniMatch	80.18	61.18	67.43
SAMTH	80.26	60.86	65.21
+ BoundMatch	80.83	62.24	69.02

Table 4.21: Lightweight architectures in the real-world setting (Tab. 4.20). "BoundMatch" refers to SAMTH+BoundMatch.

Method		**Accuracy (%)**			**Efficiency**		
		IoU	BIoU	BF1	FLOPs	Params	FPS
DLV3+ MV2	Baseline	73.4	53.9	56.5	89.7G	5.8M	36.0
	BoundMatch	78.2	59.0	64.3	93.7G	6.0M	31.8
AFFormer-Tiny	Baseline	76.7	57.5	65.2	7.3G	2.1M	89.1
	Mobile-Seed	78.0	58.3	65.1	10.0G	2.4M	70.2
	BoundMatch	80.1	61.1	68.2	9.2G	2.2M	78.0

its effectiveness at leveraging large unlabeled pools in real-world applications.

4.4.7 Application to lightweight models

We evaluate BoundMatch on lightweight architectures suitable for resource-constrained deployment, specifically DeepLabV3+ with MobileNet-V2 [78] and AFFormer-Tiny [180]. For AFFormer-Tiny, we compare against Mobile-Seed [104], a state-of-the-art boundary-aware lightweight segmentation method. More information about the experimental setup is provided in Appendix A.10.

As shown in Tab. 4.21, BoundMatch achieves accuracy improvements of 3.4–4.8% mIoU with moderate impact on runtime performance, maintaining above 30 FPS for real-time applications. For MobileNet-V2, we achieve 4.8% IoU and 7.8% boundary F1 gains with only 4.5% additional FLOPs, maintaining 31.8 FPS suitable for real-time applications. More notably, AFFormer-Tiny with BoundMatch reaches 80.1% IoU and 61.1% BIoU while sustaining 78 FPS, well above the 30 FPS threshold required for real-time video processing. With only 2.2M parameters and 9.2 GFLOPs, this configuration is particularly suited for mobile deployment where memory and power constraints are important considerations.

Compared to the specialized Mobile-Seed method, BoundMatch achieves superior

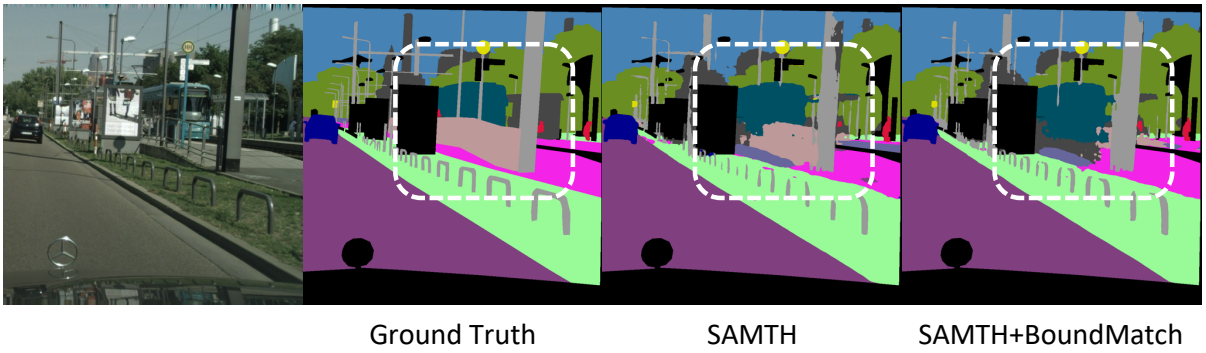


Figure 4.14: Failure case: BoundMatch incorrectly segments the train visible through the fence instead of the fence itself.

accuracy with 8% fewer FLOPs and 11% higher throughput, demonstrating that our multi-task learning framework with semi-supervised training provides a more efficient solution than dedicated boundary architectures. The modest computational overhead (approximately 26% additional FLOPs for AFFormer-Tiny) is justified by boundary quality improvements important for applications such as autonomous navigation, medical imaging on portable devices, and augmented reality systems.

These results confirm that BoundMatch scales to lightweight architectures without compromising deployment feasibility, making it practical for enhancing segmentation quality in resource-constrained environments where both accuracy and efficiency are important.

4.4.8 Limitations and Future Work

While BoundMatch demonstrates consistent improvements across benchmarks, several limitations merit discussion and guide future research:

Challenging structures: Classes with fine, partially transparent, or heavily occluded structures (*e.g.* fences, see-through facades) can degrade performance, likely due to annotation policies (filled silhouettes) and the model's tendency to favor occluded content behind meshes (Fig. 4.14 and Fig. 4.8b). Future work should explore attention-based modules for long-range context modeling and investigate alternative supervision strategies that better model transparency and occlusion.

Computational considerations: Training overhead increases (iteration time 0.92→1.59s, memory 9.0→16.5GB) primarily due to the boundary detection pipeline (*e.g.* boundary

head and the on-the-fly boundary generation). While our modular design allows efficiency-focused configurations (*e.g.* BCRM+SGF maintains baseline FPS), developing more efficient boundary modules through techniques like knowledge distillation or pruning could further reduce this overhead without sacrificing performance gains.

Domain generalization: BoundMatch shows strong performance across most evaluated domains including remote-sensing (Tab. 4.19), but does not necessarily improve on medical imaging (Tab. 4.18). The boundary detection components developed in this work have not been optimized for medical imaging settings, which may require different design considerations. Future work could investigate how to adapt boundary-aware learning to diverse application domains with their specific requirements and BoundMatch may benefit from dataset-specific optimizations.

Annotation quality dependencies: BoundMatch's performance is fundamentally limited by the quality of the underlying segmentation annotations. Errors or inconsistencies in segmentation masks propagate through both the training process and boundary generation via distance transforms. This dual dependency is particularly problematic in datasets with known annotation issues (*e.g.* Pascal VOC's ignore regions and Cityscapes' inconsistent annotations Fig. 4.8b) where both segmentation and derived boundaries inherit the same biases. Future work should explore self-supervised boundary discovery or leverage foundation models' priors to reduce dependence on precise annotations. Additionally, developing robust training strategies that explicitly account for annotation noise in both tasks could improve real-world deployment.

Modern architecture integration: As vision transformers and foundation models become prevalent, adapting BoundMatch's principles presents opportunities. Incorporating boundary-aware consistency into vision-language models could enhance their spatial understanding for tasks requiring precise localization. The multi-task consistency principle could extend to other complementary views (depth, surface normals) in semi-supervised settings, particularly as foundation models provide increasingly rich priors for such tasks [168].

4.5 Conclusion

This chapter extends the auxiliary-supervision idea of Chapter 3 to semi-supervised learning. With *BoundMatch*, boundary-derived signals remain effective under label scarcity

and pseudo-label noise, yielding consistent gains of 0.4–2.4% mIoU across datasets and label ratios, alongside improvements in boundary metrics (BIoU, BF1).

Relative to SBCB's clean-label setting, BoundMatch adapts the mechanism for semi-supervised learning: boundary-aware consistency and the BSF/SGF components manage noisy pseudo-labels while retaining the benefits of hierarchical, multi-scale boundary supervision. The modular design integrates with CNNs, lightweight backbones, and transformers, and the joint learning of segmentation and boundaries offers complementary views that help filter pseudo-label noise.

Practically, BoundMatch can be plugged into strong SSL baselines (*e.g.* UniMatch, PrevMatch) and improves both region and boundary quality. The main trade-offs are concentrated in training cost (about +73% time/memory) and diminishing returns as label coverage increases (*e.g.* +2.4% at 1/16 vs. +0.4% at 1/2). Unlike SBCB, some BoundMatch variants introduce a small inference overhead; where zero-overhead deployment is required, either *BCRM-only* or *BCRM+SGF* configuration preserves inference parity with typically smaller—but still reliable—gains. For a detailed discussion of limitations, see Section 4.4.8.

This chapter also sets up the next step in the thesis narrative. Chapter 5 (Chapter 5) moves from *within-task* auxiliary supervision under label scarcity to *cross-task* supervision: leveraging annotation-rich perspective-view resources to supervise annotation-scarce BEV segmentation. This progression also addresses a different scarcity regime—absence of target-domain labels in domain adaptation—and evaluates whether the same design instincts (complementary signals, derivable supervision, and deployment-mindedness) transfer across views.

5

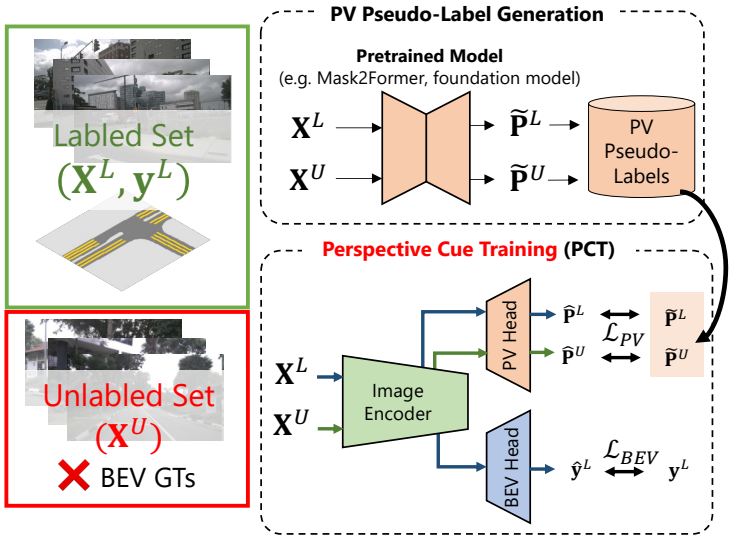
PCT: Perspective Cue Training for Multi-Camera BEV Segmentation

5.1 Introduction

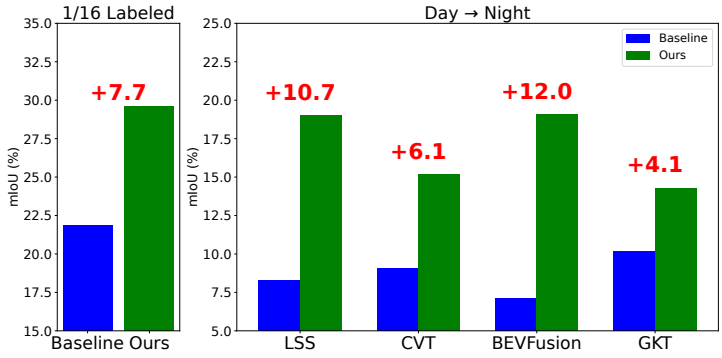
Chapters 3 and 4 demonstrated that auxiliary supervision with semantic boundaries is viable as it can be extracted from existing annotations and is effective even under label scarcity through consistency regularization. This chapter explores a different dimension of auxiliary supervision from existing resources: leveraging the knowledge embedded in pretrained models to address tasks where direct annotation is prohibitively expensive.

Bird's-eye-view (BEV) segmentation exemplifies this annotation challenge. Creating accurate BEV annotations requires cumbersome multi-camera calibration, 3D reconstruction, and consistent labeling across synchronized viewpoints—a process substantially more expensive than single-view annotation. Meanwhile, perspective view (PV) segmentation benefits from extensive datasets (Cityscapes, BDD100K) and mature pretrained models that represent significant community investment. This asymmetry between annotation costs presents an opportunity: *can pretrained PV models—themselves an existing resource—provide auxiliary supervision for annotation-scarce BEV tasks?*

We investigate this through the **Perspective Cue Training (PCT)** framework, which addresses the third research question posed in Sec. 1.3: whether auxiliary supervision can leverage pretrained models to enable learning when primary task annotations are



(**a**) Overview of the proposed PCT framework.



(**b**) Improvement across methods and tasks.

Figure 5.1: Overview of PCT framework. (a) PCT utilizes PV pseudo-labels to train multi-camera BEV segmentation models. (b) Relative improvements across methods and tasks.

scarce or absent. PCT demonstrates that readily available PV segmentation models (*e.g.* Mask2Former [74]) can generate pseudo-labels that serve as auxiliary supervision for BEV learning through multi-task training. Unlike the within-task auxiliary supervision of SBCB and BoundMatch, PCT bridges the geometric gap between perspective and bird's-eye views while transferring semantic knowledge.

The challenges in BEV segmentation extend beyond annotation scarcity. Domain shifts—where models trained in one environment fail in another due to weather, lighting, or geographic differences—compound the annotation problem. An autonomous vehicle

trained on daytime data from one city must adapt to nighttime conditions or different urban layouts without requiring new BEV annotations for each scenario. PCT addresses both semi-supervised learning (SSL), where few BEV annotations exist, and unsupervised domain adaptation (UDA)[1], where the target domain lacks BEV annotations entirely.

Furthermore, we introduce augmentation strategies tailored for multi-camera BEV learning: Camera Dropout (CamDrop) provides input-level perturbation by randomly masking camera views, while BEV Feature Dropout (BFD) applies feature-level perturbation in the BEV representation. These techniques, combined with PCT's auxiliary supervision from pretrained models, achieve improvements of up to 7.7% mIoU for SSL and 10.7% for UDA scenarios on the nuScenes dataset.

The contributions of this work are:

- **Extending auxiliary supervision to leverage pretrained models**: PCT demonstrates that pretrained models represent an underutilized existing resource that can provide effective supervision for annotation-scarce tasks, complementing the annotation-based auxiliary supervision explored in previous chapters.

- **Among the first systematic studies of SSL for multi-camera BEV segmentation**: We establish baselines and introduce techniques (CamDrop, BFD) that address the unique challenges of learning BEV representations from limited labels.

- **Camera-only UDA without additional modalities**: Our approach achieves competitive performance against methods requiring LiDAR supervision, demonstrating that auxiliary supervision from PV models can bridge significant domain gaps.

[1]While we classify our approach as unsupervised domain adaptation since no target domain annotations—from direct human intervention—are used during training, we acknowledge that the off-the-shelf models used for pseudo-label generation (*e.g.* Mask2Former) were trained on large-scale annotated datasets that may contain scenes with similar characteristics to the target domains. This indirect knowledge transfer through pretrained models is analogous to the common practice of using pretrained backbones (*e.g.* ImageNet) in UDA methods. However, we also acknowledge that a fine-grained distinction between the use of pretrained auxiliary task and purely unsupervised approach could exist for fairer comparisons in comparisons.

5.2 Related Work

Building on the comprehensive background of BEV segmentation methods established in Sec. 2.4, this section focuses on the specific challenges of learning with limited BEV labels and positions PCT within the broader context of auxiliary supervision approaches.

5.2.1 Learning with Limited BEV Labels

As discussed in Sec. 2.4.2, the scarcity of BEV annotations contrasts sharply with perspective view resources. While semi-supervised methods for 2D segmentation are well-established (Sec. 2.3), their application to BEV segmentation faces unique challenges: the geometric transformation between views, multi-camera fusion, and maintaining consistency across viewpoints.

SkyEye [148] explored self-supervised learning for monocular BEV segmentation using homography-based view synthesis, though limited to single cameras and planar scenes. Concurrent work [149] proposed rotation-based augmentation, but does not leverage the abundance of perspective view resources. To our knowledge, SSL for multi-camera BEV segmentation remains unexplored, motivating our systematic investigation.

For domain adaptation, DualCross [23] uses LiDAR teacher models for cross-modal knowledge distillation, while DA-BEV [150] applies query-based adversarial training in both image and BEV spaces. Both approaches require either additional sensor modalities or complex adversarial training. PCT differs fundamentally by leveraging auxiliary supervision from pretrained PV models—a resource that requires no additional sensors or constrains the BEV architecture.

5.2.2 Perspective View Models as Auxiliary Supervision

The relationship between PCT and prior auxiliary supervision approaches merits clarification. While SBCB (Chapter 3) extracts auxiliary signals from existing annotations and BoundMatch (Chapter 4) maintains them under label scarcity, PCT explores auxiliary supervision from a different existing resource: pretrained models.

X-Align [158] previously demonstrated that PV-BEV alignment can improve fully-supervised BEV segmentation. PCT extends this insight to the label-scarce regime, showing that pretrained PV models can provide effective supervision and regularization even when

BEV annotations are limited or absent. This aligns with the thesis's broader theme: existing resources—whether annotations or pretrained models—can provide valuable auxiliary supervision when properly utilized.

5.3 Approach

This work addresses the task of BEV segmentation of street-view scenes from multi-camera rigs, focusing on leveraging the abundance of unlabeled data to enhance model performance. We approach this through the paradigms of Semi-Supervised Learning (SSL) and Unsupervised Domain Adaptation (UDA), utilizing a combination of labeled and unlabeled datasets during training.

For coherent methodology, we unify the two tasks, where we have a labeled set $\mathbb{L}$ and an unlabeled set $\mathbb{U}$. For SSL, the labeled set $\mathbb{L}$ and the unlabeled set $\mathbb{U}$ are from the same domain, with the number of labeled samples being much smaller than the number of unlabeled samples, i.e., $|\mathbb{L}| \ll |\mathbb{U}|$. In UDA, the labeled set $\mathbb{L}$ is from the source domain, while the unlabeled set $\mathbb{U}$ is from the target domain, with a domain shift between the two. The labeled set contains a set of multi-camera data $\mathbf{X}^L$ and BEV ground truth (GT) $\mathbf{y}^L$ where $(\mathbf{X}^L, \mathbf{y}^L) \in \mathbb{L}$, while the unlabeled set only contains multi-camera data $\mathbf{X}^U \in \mathbb{U}$.

We define $\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^N = \{\mathbf{I}_i, \mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i\}_{i=1}^N$ as a batch of multi-view images, where $\mathbf{I}_i$ is the image, $\mathbf{K}_i$ is the intrinsic parameter, $\mathbf{R}_i$ and $\mathbf{t}_i$ are the extrinsic for the i-th camera in the batch, and N is the number of cameras on the vehicle. Following [139], we treat the BEV segmentation task as multi-label classification where each pixel can have multiple labels. Therefore, BEV GT is formalized as $\mathbf{y}^L \in \{0,1\}^{H \times W \times C}$, where H and W are the height and width of the BEV grid, and C is the number of classes.

Multi-camera BEV segmentation models typically comprise the following components: an Image Encoder, a 2D-to-BEV Module, and a BEV Encoder and Decoder, with the BEV Encoder sometimes omitted. Methods like CVT [145] have an additional BEV embedding as an input, but we omit these method-specific components in our discussion for brevity.

The processing flow of these modules is as follows:

$$\mathbf{F}_{image} = \text{ImageEncoder}(\mathbf{I}) \tag{5.1}$$

$$\mathbf{f}_{bev} = \text{2DtoBEV}(\mathbf{F}_{image}, \mathbf{K}, \mathbf{R}, \mathbf{t}) \tag{5.2}$$

$$\mathbf{f}'_{bev} = \text{BEVEncoder}(\mathbf{f}_{bev}) \tag{5.3}$$

$$\hat{\mathbf{y}} = \text{BEVDecoder}(\mathbf{f}'_{bev}), \tag{5.4}$$

where $\mathbf{F}_{image}$ is the image features obtained from multi-camera images, $\mathbf{f}_{bev}$ is the BEV feature, and $\hat{\mathbf{y}}$ is the output BEV prediction.

For brevity, we simplify our notation by treating $\mathbf{X} = \mathbf{I}$ in subsequent sections and assume that the correct camera parameters are provided to the 2DtoBEV module.

Our method is organized as follows:

- Sec. 5.3.1 introduces the Perspective Cue Training (PCT) framework, which leverages pseudo-labels of perspective view task to utilize unlabeled data for BEV segmentation.
- Sec. 5.3.2 presents Camera Dropout (CamDrop), a novel input perturbation technique for multi-camera BEV segmentation.
- Our training strategy with PCT for UDA is formalized in Sec. 5.3.3.
- Finally, Sec. 5.3.4 details our SSL approach, which incorporates BEV Feature Dropout (BFD) and a teacher-student network training strategy.

5.3.1 Perspective Cue Training (PCT) Framework

We propose the Perspective Cue Training (PCT) framework, which utilizes the PV pseudo-labels to train the BEV segmentation model in multi-task learning manner. This framework is only applied during the training stage of the target BEV segmentation model. In our work we first generate pseudo-labels of perspective images from all sets $\mathbf{X}^{L\cup U}$, including both labeled $\mathbf{X}^{L}$ and unlabeled images $\mathbf{X}^{U}$, where $\mathbf{X}^{L\cup U} \in (\mathbb{L} \cup \mathbb{U})$. We chose semantic segmentation for our main pseudo-labeling task because the pixel-wise classification task is similar to BEV segmentation. However, we have also experimented with relative depth estimation task, as shown in Fig. 5.2. We generally use Mask2Former [74] trained on BDD100k [18] as our default pseudo-label generator, but we explore the effects of other network architectures and training datasets in our ablation studies. Our pseudo-label

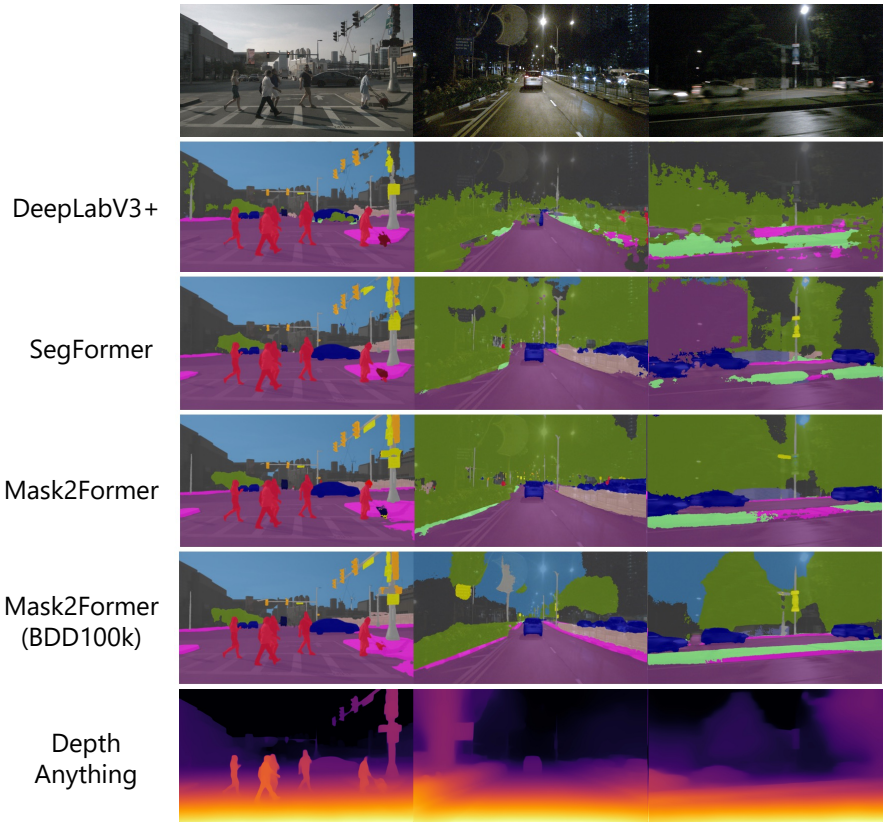


Figure 5.2: Visualization of pseudo-labels from different models on nuScenes. Mask2Former trained on BDD100k produces the cleanest results across domains including nighttime scenes.

generator PLGen obtains $\tilde{\mathbf{P}} = \{\text{PLGen}(\mathbf{x}_i) \mid \forall \mathbf{x}_i \in \mathbf{X}^{L \cup U}\}$, where $\tilde{\mathbf{P}}$ are one-hot encoded pseudo-labels.

PV task head is applied to the image encoder, and the entire BEV segmentation model is trained in a multi-task learning manner, as shown in Fig. 5.1. We utilize FPN with UPerNet [58] as our PV task head (PVHead) in the belief that utilizing all the hierarchical features of the image encoder is essential to condition the shared image encoder with PV cues. We can obtain PV predictions $\hat{\mathbf{P}}$ from the PV task head PVHead as follows:

$$\mathbf{F}^L_{image} = \text{ImageEncoder}(\mathbf{X}^L) \tag{5.5}$$

$$\mathbf{F}^U_{image} = \text{ImageEncoder}(\mathbf{X}^U) \tag{5.6}$$

$$\hat{\mathbf{P}} = \{\text{PVHead}(\mathbf{f}_i) \mid \forall \mathbf{f}_i \in [\mathbf{F}^L_{image}; \mathbf{F}^U_{image}]\}. \tag{5.7}$$

PV loss $\mathcal{L}_{PV}$ is computed using cross-entropy loss between the prediction $\hat{\mathbf{P}}$ and the

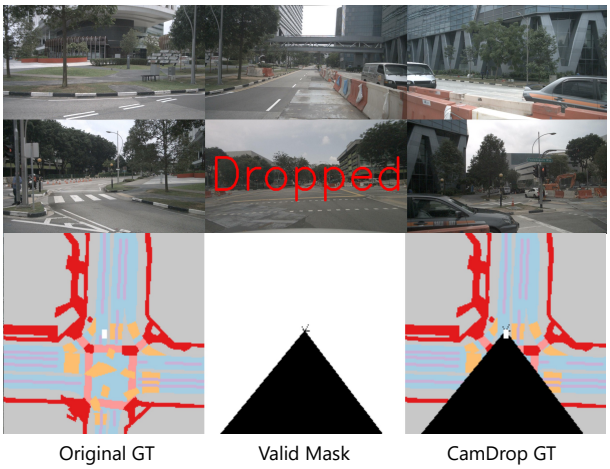


Figure 5.3: Camera Dropout (CamDrop) augmentation. Dropped camera views and their exclusively visible BEV regions are masked out.

pseudo-labels $\tilde{\mathbf{P}}$ as follows:

$$\mathcal{L}_{PV} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{CE}(\hat{\mathbf{p}}_i, \tilde{\mathbf{p}}_i), \tag{5.8}$$

where $\hat{\mathbf{p}}_i \in \hat{\mathbf{P}}$ and $\tilde{\mathbf{p}}_i \in \tilde{\mathbf{P}}$ are the i-th output probability map of the PV prediction and one-hot encoded pseudo-labels, respectively.

The final multi-task training loss is as follows:

$$\mathcal{L}_{PCT} = \mathcal{L}_{BEV} + \lambda_{PV}\mathcal{L}_{PV}, \tag{5.9}$$

where $\mathcal{L}_{BEV} = \mathrm{FocalLoss}(\hat{\mathbf{y}}^L, \mathbf{y}^L)$ and λ_{PV} is the weight for the PV loss. Note that $\hat{\mathbf{y}}^L$ is obtained from Eq. (5.4).

5.3.2 Camera Dropout (CamDrop) Augmentation

Applying traditional pixel-wise input augmentations to BEV segmentation is challenging due to its 3D nature. For instance, masking sections of the PV image, as in Cutout, would require corresponding masking in the BEV ground truth, which is not straightforward. To address this, we introduce Camera Dropout (CamDrop), a simple yet effective input perturbation inspired by Cutout [181]. CamDrop randomly drops cameras and masks the associated visible areas in the BEV ground truth, as illustrated in Fig. 5.3. This augmentation is efficient, as the horizontal viewport can be easily determined from the

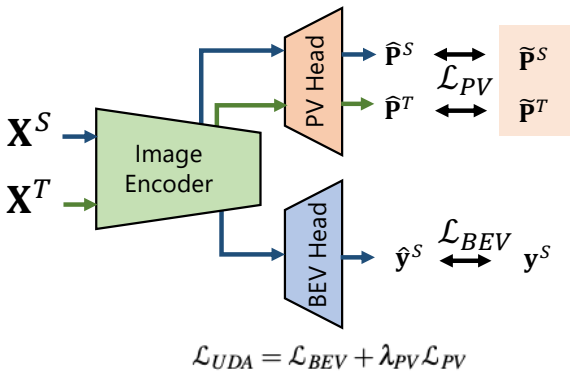


Figure 5.4: PCT training framework for UDA with joint training on labeled source and unlabeled target domains.

camera parameters. The perspective view image and the masked BEV area are labeled with ignore labels, ensuring that the dropped regions do not contribute to the loss. Importantly, we only mask regions exclusively visible from the dropped cameras, ensuring that the masked areas are not visible from the remaining cameras.

5.3.3 Training Method for UDA

The PCT framework provides a method of training on both domains using a shared image encoder (shown in Fig. 5.4), which motivates the model to learn domain-invariant features through PV tasks. The intuition can be explained using the theorem proposed in [182], which states that the upper-bound of the target domain error is composed of the source domain error and the domain discrepancy where the latter can be minimized through utilizing both domains with $\mathcal{L}_{PV}$.

For UDA, we formalize labeled set $\mathbb{L}$ as the source domain set and unlabeled set $\mathbb{U}$ as the target domain set. The loss for training UDA is as follows:

$$\mathcal{L}_{UDA} = \mathcal{L}_{BEV} + \lambda_{PV}\mathcal{L}_{PV}, \tag{5.10}$$

where we use the same loss as in Eq. (5.9).

Additionally, we utilize CamDrop to further enhance the model's robustness.

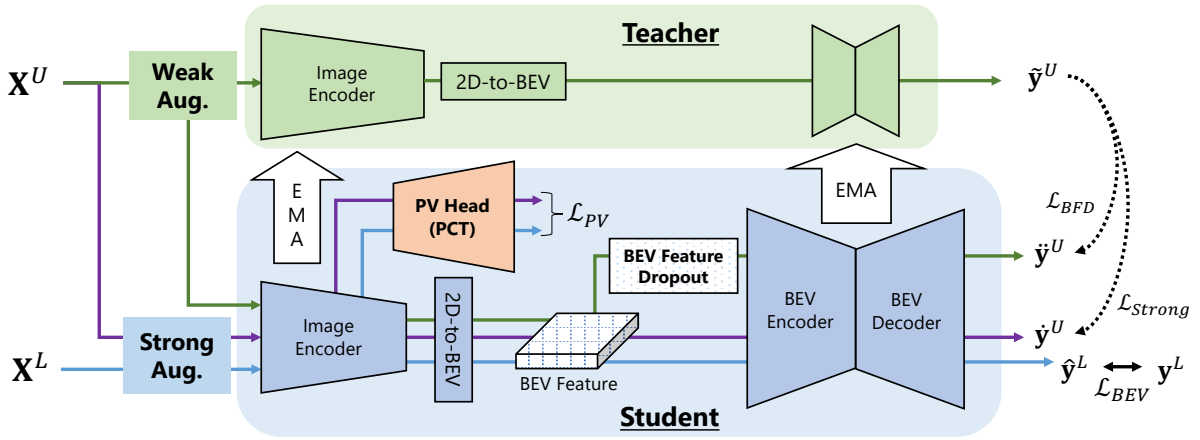


Figure 5.5: SSL training framework combining PCT, CamDrop, and BFD within a mean-teacher framework.

5.3.4 Training Method for SSL

For SSL, we introduce a teacher-student training framework following the Mean-Teacher (MT) framework [32], as shown in Fig. 5.5. In this framework, we have two identical models, the student and the teacher, where the teacher model is an exponential moving average (EMA) of the student model, computed as:

$$\theta'_{teacher} = \alpha\theta_{teacher} + (1-\alpha)\theta_{student}, \tag{5.11}$$

where θ is the model parameters and α is the momentum.

For the teacher network, a weakly augmented input is used, while a strongly augmented input is passed to the student network. The consistency loss is computed between the weakly augmented teacher's prediction and the strongly augmented student's prediction:

$$\tilde{\mathbf{y}}^U = \text{Teacher}(\text{WeakAug}(\mathbf{X}^U)) \tag{5.12}$$

$$\dot{\mathbf{y}}^U = \text{Student}(\text{StrongAug}(\mathbf{X}^U)) \tag{5.13}$$

$$\mathcal{L}_{Strong} = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i^U - \dot{y}_i^U)^2, \tag{5.14}$$

where we calculate the mean squared error for all $M = H \times W \times C$ pixels of the BEV grid. For WeakAug, we employ augmentations commonly used in BEV segmentation: random horizontal flip, rotation, scaling, and cropping. For StrongAug, we apply the same augmentations as WeakAug but with ColorJitter, GaussianBlur, and CamDrop.

We further propose a feature perturbation called BEV Feature Dropout (BFD) inspired by UniMatch [33]. We apply Dropout [183] of 50% to the BEV feature maps to obtain perturbed BEV features. The consistency loss is then computed between the weakly augmented teacher's predictions and the perturbed prediction:

$$\mathbf{f}_{BEV}^{U} = \text{2DtoBEV}(\text{ImageEncoder}(\text{WeakAug}(\mathbf{X}^{U}))) \tag{5.15}$$

$$\ddot{\mathbf{y}}^{U} = \text{BEVDecoder}(\text{BEVEncoder}(\text{BFD}(\mathbf{f}_{BEV}^{U}))) \tag{5.16}$$

$$\mathcal{L}_{BFD} = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i^{U} - \ddot{y}_i^{U})^2. \tag{5.17}$$

The final loss used for SSL is as follows:

$$\mathcal{L}_{SSL} = \mathcal{L}_{BEV} + \lambda_{PV}\mathcal{L}_{PV} + \lambda_{Strong}\mathcal{L}_{Strong} + \lambda_{BFD}\mathcal{L}_{BFD}, \tag{5.18}$$

where λ_{Strong} and λ_{BFD} are the weights for strong consistency loss and the BFD consistency loss, respectively.

We note that BFD's benefits are primarily realized through the MT framework, which empirically showed strong performance in SSL, but not for UDA.

5.4 Experiments

5.4.1 Experimental Setup

SSL Experimental Setup. We generate four SSL splits for 1/16, 1/8, 1/4, and 1/2 labeled data from the nuScenes dataset's [28] training set and treat the remaining labeled data as unlabeled data. Note that we divide the training scenes into labeled and unlabeled data, not individual sample frames. For testing, we evaluate on the validation scenes, which is the same for all SSL splits.

UDA Experimental Setup. We follow the splits introduced by DualCross [23] and use the following domain gaps: Day → Night, Dry → Rain, and Boston → Singapore for the nuScenes dataset. We added Singapore → Boston domain gap since Boston → Singapore contains mixed domain gaps, as explained in [23].

Common Experimental Setup. Following [139], we segment static categories, which includes the following: Drivable Area, Pedestrian Crossing, Walkway, Stop Line, Carpark

Area, and Divider. We measure the performance using mean Intersection over Union (mIoU).

Implementation Details. We base our BEV segmentation code base on [139] and mmsegmentation [165]. The hyperparameters are all consistent to ensure fair comparison across different methods. Unless explicitly stated, the crop size is 224×480, batch size is 32, total training iteration is $30k$, the optimizer is AdamW with a learning rate of 0.004 and weight decay of 0.01, and the learning rate is scheduled using OneCycle Learning Rate Scheduler. For all experiments, we present the metric of the final checkpoint. The baseline method of our work is LSS [143] with EfficientNet-b4 backbone. For loss weights in Eqs. (5.10) and (5.18), we set $\lambda_{PV} = 0.1$, $\lambda_{Strong} = 0.1$, and $\lambda_{BFD} = 0.5$. As stated in Sec. 5.3.4, the weak augmentations are random horizontal flip, rotation, scaling, and cropping, while the default strong augmentations are the same as the weak augmentations but with ColorJitter and GaussianBlur. For UDA, we use the strong augmentations without GaussianBlur. The baseline methods use strong augmentations by default unless explicitly stated. The momentum for the teacher model is set to $\alpha = 0.999$. Based on the original Mean-Teacher implementation, we utilize a sigmoid rampup function for the consistency loss weight, which starts at 0 and gradually increases to 1 over the first $9k$ iterations. We train and validate all of our experiments using 8 NVIDIA V100 GPUs.

Pseudo-label Generation. The nuScenes dataset does not have semantic segmentation annotations for the PV images. We utilized mmsegmentation [165] to generate PV pseudo-labels. We retrained Mask2Former with the same configuration as the Cityscapes dataset on datasets without pretrained weights. For relative depth estimation, we use the publicly available code provided by Depth Anything [184].

5.4.2 Semi-Supervised Learning Results

In our SSL benchmark, we compare our proposed method against the following baselines:

- **Sup. Only**: LSS supervised with only the labeled data
- **Mean-Teacher (MT)**: Mean-Teacher [32] adapted for BEV segmentation (more details in Sec. A.11.1)
- **UniMatch**: UniMatch [33] adapted for BEV segmentation with our proposed CamDrop and BFD (more details in Sec. A.11.2)

Table 5.1: Semi-supervised learning results on nuScenes. Results in mIoU (%).

Method	CamDrop	BFD	Labeled Ratio			
			1/16	1/8	1/4	1/2
Supervised Only			21.9	27.4	36.4	47.0
MT			23.7	29.8	37.5	47.9
	✓		25.5	31.4	38.6	47.5
		✓	25.2	31.8	39.3	49.1
	✓	✓	27.0	32.6	40.1	49.1
UniMatch	✓	✓	22.9	29.3	39.2	49.0
PCT			24.1	29.3	37.6	48.7
	✓		24.9	30.0	38.3	48.4
PCT+MT	✓		28.6	34.0	40.7	50.4
	✓	✓	**29.6**	**35.0**	**41.9**	**51.6**

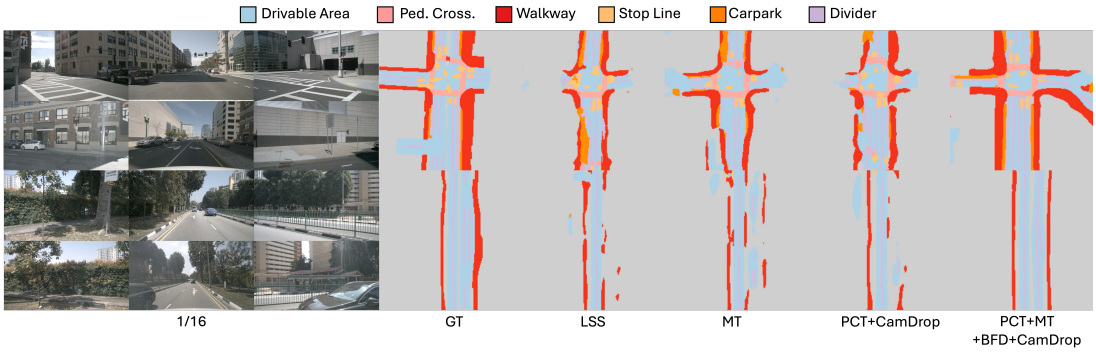


Figure 5.6: Qualitative SSL results on the 1/16 split.

All semi-supervised methods use the same LSS as the network architecture for fair comparison. Note that SSL for multi-camera BEV segmentation is under-explored and there are no existing methods for comparison. We also apply CamDrop and BFD to MT to show the effectiveness of our proposed input and feature perturbations.

In Tab. 5.1, we show the results of our SSL experiments on the nuScenes dataset. Compared to the supervised only method, our proposed PCT+MT method outperforms the baseline by a significant margin for all SSL splits. It can also be seen that PCT alone can perform better than supervised only baseline and can even perform competitively against SSL methods like MT and UniMatch. PCT, when combined with MT and strong perturbations (CamDrop and BFD), achieves the best performance for all SSL splits, improving on the difficult 1/16 split by a significant margin of 7.7%.

CamDrop and BFD have been shown to be effective in improving the performances of baseline and proposed method. MT with CamDrop improves MT on most splits, and MT with BFD improves MT on all splits. As shown in Fig. 5.6, MT with CamDrop and BFD improves the performance of MT for the difficult 1/16 split, especially for areas further away and partially occluded regions. The BEV segmentation predictions for PCT+MT with CamDrop and BFD, show improvements for dividers and smaller categories like "Stop Line."

5.4.3 Unsupervised Domain Adaptation Results

In our UDA benchmark, we compare our proposed method against the following baselines:

- **Baseline**: LSS supervised with only the source domain
- **DomainAdv**: Adversarial baseline used in [23], which adds domain classifiers to the Image Encoder and BEV Encoder and trained with GRL [185] (more details in Sec. A.11.3)
- **FDA+MT**: Fourier Domain Adaptation with Mean-Teacher [186] (more details in Sec. A.11.4)

All the UDA approaches use the same LSS as network architecture. We compare with DualCross [23] in Sec. 5.4.5 due to the drastically different experimental setup.

In Tab. 5.2, we show the results of our UDA experiments on the nuScenes dataset. The FDA+MT baseline improves upon the baseline LSS and DomainAdv in most domain gaps. Although DomainAdv works well in the difficult Day → Night and Boston → Singapore domain gaps, it fails to perform well on others. We believe this is due to the domain classifier not functioning correctly for the other domain gaps.

For all domain gaps, both of our proposed PCT and PCT+CamDrop significantly outperform baselines, especially for major categories like "Drivable Area" and "Walkway." As we show in Fig. 5.7, PCT+CamDrop generally has cleaner segmentation results than the baseline methods (LSS, DomainAdv, and FDA+MT). In Day → Night domain gap, PCT+CamDrop improves the segmentation of thin roads and dividers, showing the model's enhanced capability to segment distant areas. In Singapore → Boston domain gap, PCT+CamDrop improves the segmentation of "Pedestrian Crossing" and "dividers", and

Table 5.2: Unsupervised domain adaptation results on nuScenes across four domain gaps. Results in IoU (%).

Method	Drive.	Cross.	Walk.	Stop.	Car.	Div.	Mean
Day → Night							
Baseline	30.5	1.7	4.0	1.9	0.0	11.8	8.3
DomainAdv	47.1	16.1	10.7	5.7	0.0	11.2	15.1
FDA+MT	44.9	7.8	12.3	4.6	0.0	14.1	14.0
PCT	51.3	19.4	**16.1**	**7.6**	0.0	19.3	19.0
PCT+CamDrop	**52.5**	**19.8**	15.8	6.8	0.0	**20.6**	**19.2**
Dry → Rain							
Baseline	74.2	38.2	46.8	31.3	39.6	32.7	43.8
DomainAdv	72.0	39.8	42.0	33.7	38.9	33.6	43.3
FDA+MT	75.3	42.0	47.0	35.3	39.5	34.2	45.6
PCT	**78.3**	**45.2**	52.1	**37.6**	47.2	36.4	49.5
PCT+CamDrop	**78.3**	44.7	**52.6**	37.2	**48.7**	**37.3**	**49.8**
Singapore → Boston							
Baseline	39.5	3.1	12.8	4.1	1.0	12.1	12.1
DomainAdv	35.7	4.2	11.3	4.8	0.6	9.7	11.1
FDA+MT	41.8	6.5	14.5	7.1	1.1	10.8	13.6
PCT	47.0	8.0	19.3	6.3	0.7	13.7	15.8
PCT+CamDrop	**48.9**	**8.9**	**21.6**	**7.6**	**1.7**	**15.6**	**17.4**
Boston → Singapore							
Baseline	37.1	7.5	9.4	4.6	4.4	11.8	12.5
DomainAdv	40.0	8.3	11.7	4.8	2.2	11.6	13.1
FDA+MT	38.9	8.3	12.3	5.2	2.0	11.6	13.1
PCT	46.2	8.6	**14.2**	6.4	3.7	15.0	15.7
PCT+CamDrop	**47.2**	**9.0**	14.1	**7.7**	**4.8**	**16.4**	**16.5**

reduces the effect of placing "stop lines" on the opposite side of the road (due to reversed driving lanes). In Boston → Singapore domain gap, PCT+CamDrop improves the overall segmentation quality without inheriting issues of producing wide roads common in Boston.

5.4.4 Ablation Study

Effect of Pseudo-Label Quality. Tab. 5.3 evaluates pseudo-labels generated from different semantic segmentation architectures trained on the Cityscapes dataset. Higher-quality pseudo-labels result in better performance for UDA, but the difference is not as

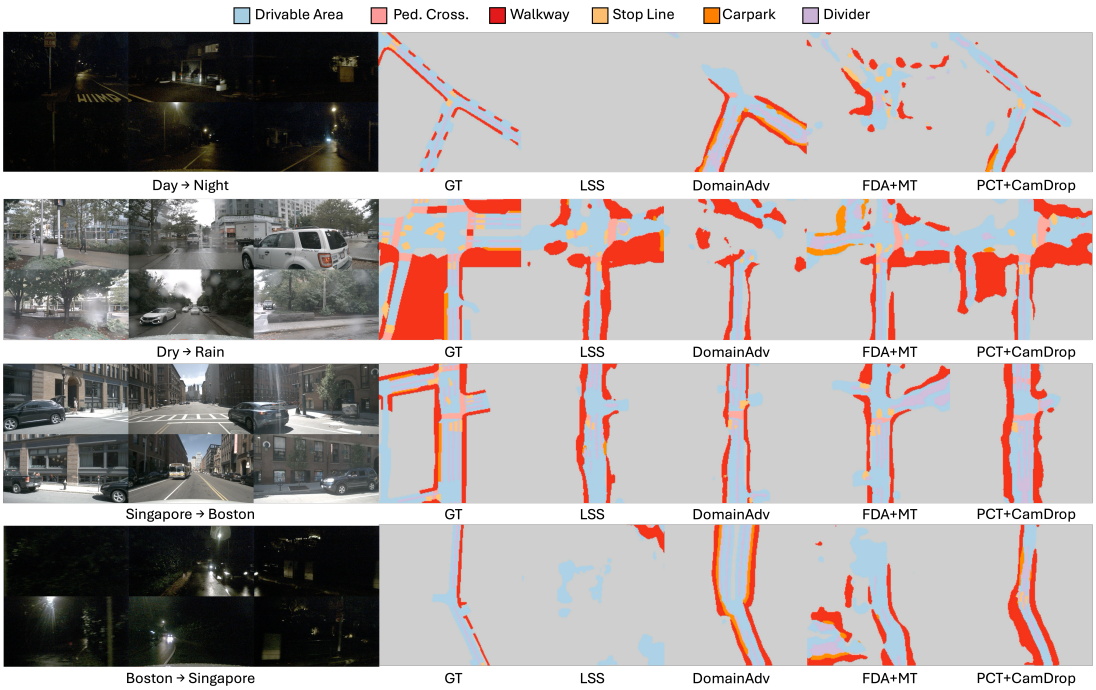


Figure 5.7: Qualitative UDA results on Day → Night.

Table 5.3: Effect of pseudo-label model quality. All models trained on Cityscapes.

Pseudo-Label Model	**Seg. Quality**	**SSL (1/16)**	**UDA (Day→Night)**
DeepLabV3+ [62]	79.5	23.3	11.7
SegFormer [164]	82.3	23.2	15.6
Mask2Former [74]	83.7	23.8	17.2

significant for SSL. Although not experimented with, carefully annotated PV images may further improve the capability of the PCT.

Effect of Pseudo-Label Dataset. In Tab. 5.4, we evaluated different datasets for training semantic segmentation models for generating pseudo-labels. Training with pseudo-labels generated from BDD100k, known for containing various domain variations, performed the best in SSL and UDA. While we hoped for joint datasets (*e.g.* Cityscapes+BDD100k) to further improve the performance, the results are similar to BDD100k alone. PCT with relative depth pseudo-labels does not perform as well for SSL, but performs comparable to semantic segmentation with BDD100k for UDA. With that being said, the performance disparities between the training datasets and tasks are relatively minor, and the choice of dataset and pseudo-labeling task is flexible for PCT.

Table 5.4: Effect of pseudo-label training dataset on PCT performance.

Pseudo-Label	Dataset	Performance	
		SSL 1/16	UDA Day→Night
Semantic Seg.	Cityscapes	23.8	17.2
	BDD100k	24.1	19.0
	Cityscapes+BDD100k	24.1	18.4
	BDD100k+Synthia	23.8	17.6
Relative Depth	Depth Anything	22.8	19.0

Table 5.5: Different crop sizes for training with PCT.

Crop Size	Performance	
	SSL 1/16	UDA Day→Night
128×352	20.2	16.9
224×480	24.1	19.0
360×720	24.3	19.8

Table 5.6: Effect of PCT on different BEV architectures.

BEV Architecture	PCT	Performance	
		SSL 1/16	UDA Day→Night
LSS [143]		21.9	8.3
	✓	**24.1**	**19.0**
CVT [145]		14.4	9.1
	✓	**16.3**	**15.2**
BEVFusion [139]		21.4	7.1
	✓	**23.0**	**19.1**
GKT [146]		15.5	10.2
	✓	**16.5**	**14.3**

Effect of Crop Size for PCT. In Tab. 5.5, we evaluated different crop sizes for training with PCT. In our PV Head, the resolution depends on the crop size, and the results show that the higher crop size results in better performance for both SSL and UDA. Lower crop size results in lower resolution and less context for the semantic segmentation model to learn to produce robust feature maps. In other words, PCT has the potential to perform better with higher resolution PV images.

Effect of PCT on Different BEV Architectures. We show the effect of PCT on different BEV architectures in Tab. 5.6. The image encoder and hyperparameters

Table 5.7: Effect of maximum cameras dropped in CamDrop.

# Max Drops	Performance	
	SSL (MT) 1/16	UDA (PCT) Day→Night
0	23.7	19.0
1	**25.5**	**19.2**
2	25.1	18.8
3	25.1	18.3
4	24.9	18.3
5	22.4	18.2

are consistent across different BEV architectures. Notably, all the BEV architectures benefit from PCT, and the performance improvements are consistent across different BEV architectures. The results show that our proposed PCT framework is flexible and boosts the base model performances. Additionally, PCT does not increase the network parameters in any way during inference time, as we can freely remove the PV Head.

Effect of maximum camera drops. Tab. 5.7 shows the effect of varying number of cameras to drop for CamDrop. For both UDA and SSL, dropping one camera results in significant performance gains compared to no drop, and further dropping results in worse returns. For methods which do not utilize MT, such as PCT, the CamDrop may be too strong when dropping more than one camera.

5.4.5 Comparisons with Other UDA BEV Methods

Here, we compare our method against DualCross [23], which utilizes cross-modality information with a LiDAR teacher. The experimental setup used for DualCross is drastically different, but we made our network and hyperparameters consistent with the ones used in DualCross for a fair comparison. More specifically, they use a modified LSS architecture with EfficientNet-b0. The results are shown in Tab. 5.8. DualCross uses a smaller crop size of 128×352, and our method can perform comparable to their method without needing multi-modal information. However, our method can perform superior to DualCross when utilizing a larger crop size of 224×480 since the PV Head greatly benefits from a larger crop size. DA-BEV [150] is another UDA method, but its implementation details are unclear (*e.g.* UDA splits), and their code is not publicly available at the time of writing.

Table 5.8: Comparison with DualCross. **C**: camera, **L**: LiDAR. Results in IoU (%).

Method	Crop Size	Modal	Road	Lane	Vehicle
Day → Night					
DualCross	128 × 352	C+L	51.8	16.9	17.0
PCT+CamDrop	128 × 352	C	49.5	17.8	18.3
	224 × 480	C	**56.3**	**21.2**	**22.3**
Dry → Rain					
DualCross	128 × 352	C+L	71.9	19.5	29.6
PCT+CamDrop	128 × 352	C	76.3	32.8	27.2
	224 × 480	C	**79.3**	**36.0**	**31.5**
Boston → Singapore					
DualCross	128 × 352	C+L	43.1	**15.6**	20.5
PCT+CamDrop	128 × 352	C	44.0	13.0	19.7
	224 × 480	C	**47.8**	**15.6**	**21.8**

Table 5.9: Semi-supervised learning results on Argoverse 2. Results in mIoU (%).

Method	1/16	1/8	1/4	1/2
Sup. Only	35.9	42.6	48.1	53.2
PCT+MT	**40.9**	**45.6**	**51.2**	**54.2**

Table 5.10: Unsupervised domain adaptation results on Argoverse 2 for city-to-city domain gaps. Results in IoU (%).

Method	Drivable.	Ped. Cross.	Divider	Mean
Palo Alto → Miami				
Baseline	50.1	9.5	17.7	25.8
PCT	**54.9**	**12.6**	**19.3**	**28.9**
Austin → Pittsburgh				
Baseline	47.2	7.3	27.2	27.2
PCT	**51.0**	**13.8**	**27.6**	**30.8**

5.4.6 Results on Argoverse 2 Dataset

To validate the generalizability of our approach beyond nuScenes, we evaluate PCT on the Argoverse 2 dataset [34], which presents different urban environments and sensor configurations.

Experimental Setup. Following the nuScenes setup, we create four SSL splits (1/16, 1/8, 1/4, 1/2) by dividing training scenes into labeled and unlabeled subsets. For UDA,

we select two city-to-city domain gaps: Palo Alto → Miami (representing west coast to southeast coastal environments) and Austin → Pittsburgh (representing southern to northeastern urban layouts). We segment three static categories available in Argoverse 2: Drivable Area, Pedestrian Crossing, and Divider. All other hyperparameters remain consistent with the nuScenes experiments.

Semi-Supervised Learning Results. Tab. 5.9 presents SSL results on Argoverse 2. For our method, we used the best performing PCT+MT with CamDrop and BFD configuration from nuScenes. Our complete method achieves consistent improvements across all label fractions, with gains ranging from 1.0% to 5.0% absolute mIoU. The improvements follow a similar pattern to nuScenes, where the most substantial gains occur under severe label scarcity (5.0% for 1/16 split) and diminish as more labels become available (1.0% for 1/2 split). The relative improvement at 1/16 (13.9%) is somewhat smaller than nuScenes (35.2%), which we attribute to Argoverse 2 dataset's higher baseline performance (35.9% vs. 21.9%), suggesting the supervised model already captures substantial scene structure with limited labels on this dataset.

Unsupervised Domain Adaptation Results. Tab. 5.10 shows UDA results for two city-to-city domain shifts. PCT achieves mean improvements of 3.1% and 3.6% for the two domain gaps, respectively. Notably, the improvements are more balanced across classes compared to nuScenes UDA results. For Palo Alto → Miami, the largest gain occurs in Drivable Area (+4.8%), while for Austin → Pittsburgh, Pedestrian Crossing shows the most substantial improvement (+6.5%). The modest gain for Divider in the Austin → Pittsburgh split (+0.4%) suggests this class may be similarly represented in both cities, requiring less adaptation. These city-to-city results support our earlier observation that PCT primarily addresses appearance and semantic variations rather than fundamental geometric differences, as urban layouts between American cities are relatively consistent.

The consistent performance improvements on Argoverse 2 demonstrate that PCT is effective across different datasets, camera rigs, and urban environments. Both SSL and UDA results maintain the key patterns observed on nuScenes: larger gains under severe label scarcity for SSL, and consistent improvements across domain gaps for UDA. This validates our approach on a second autonomous driving dataset without requiring dataset-specific tuning.

5.5 Conclusion

This chapter tackled annotation scarcity in BEV segmentation by leveraging *pretrained perspective-view (PV) models* as auxiliary teachers. Through *PCT*, we provided evidence for the third research question: existing model knowledge can be repurposed to supervise annotation-expensive BEV tasks. Empirically, PCT improves semi-supervised learning by up to 7.7% mIoU (at 1/16 labels) and unsupervised domain adaptation by up to 10.7% mIoU (day→night).

Validating Auxiliary Supervision from Pretrained Models. Unlike Chapters 3 and 4, which extract complementary cues (semantic boundaries) within the same view (image), PCT has validated that shared backbones with auxiliary task conditioning can provide helpful signals (semantic knowledge and/or regularization) across PV to the more complex BEV task. The magnitude of gains—especially in domain adaptation—suggests that auxiliary supervision is most impactful where direct supervision is weakest. We view this as an empirical trend across our studies rather than a universal rule.

Technical Insights and Contributions. The PV head operates only during training, so inference-time cost and pipelines remain unchanged. This modular, training-only design aligns with the deployment-minded philosophy established in earlier chapters. Furthermore, this chapter introduces BEV segmentation specific training techniques like *Camera Dropout (CamDrop)* to regularize multi-camera inputs and *BEV Feature Dropout (BFD)* to perturb BEV representations at the feature level. Both mechanisms add helpful regularization without modifying the underlying BEV architecture.

Limitations and Practical Considerations. PCT's effectiveness depends on the quality and breadth of the PV teacher; teachers trained on diverse datasets (*e.g.* BDD100K) tend to yield stronger supervision than those trained on narrower domains (*e.g.* Cityscapes-only). Maintaining both PV and BEV heads during training increases memory and compute, even though inference overhead is zero. Gains vary across BEV architectures (approximately 1.6–7.7%), indicating that some designs better absorb PV-derived signals. In domain adaptation, appearance shifts (day→night) benefit more than structural shifts (city→city), consistent with PCT primarily transferring semantic rather than geometric knowledge.

Position in the Thesis Narrative. Chapter 5 transitions from *within-task* auxiliary cues (boundaries) to *cross-task* supervision (PV task to BEV task) via pretrained PV teachers, retaining the same design instincts—complementarity to the primary task, derivability

from existing resources, and zero inference overhead—under a different scarcity regime. Chapter 6 summarizes the three frameworks and explores future directions.

6

Conclusion

6.1 Thesis Summary

This thesis investigated how auxiliary supervision derived from existing resources—semantic segmentation masks, pretrained models, and unlabeled data—can enhance urban scene understanding without requiring additional labeling efforts or prohibitive computational overhead. Through three complementary frameworks, this thesis demonstrated that valuable supervisory signals already exist within current resources and can be effectively extracted to improve segmentation performance.

SBCB (Chapter 3) established the foundational principle by showing that semantic boundaries, automatically derived from existing segmentation masks through on-the-fly generation, provide complementary supervision during training. Applied to fully-supervised settings, SBCB achieved consistent improvements averaging 1.2% mIoU and 2.6% boundary F-score across multiple datasets, with the critical advantage of zero inference overhead since the boundary detection head is removed after training. This work demonstrated that existing annotations contain derivable structural cues that enhance learning when properly utilized.

BoundMatch (Chapter 4) extended this principle to label-scarce scenarios, addressing the practical challenge of limited annotations in semi-supervised semantic segmentation. By incorporating boundary consistency regularization into existing semi-supervised learning frameworks, the method achieved improvements of 0.4–2.4% mIoU and significant

improvements in boundary metrics, with gains most pronounced under severe label scarcity. The framework's bidirectional fusion modules (BSF and SGF) enable information flow between tasks and provide additional regularization while maintaining efficiency. Orthogonally, this work introduced Harmonious Batch Normalization to stabilize training dynamics in teacher-student architectures—a contribution with broader applicability to semi-supervised methods.

PCT (Chapter 5) explored a different dimension of auxiliary supervision by leveraging pretrained perspective view models to supervise annotation-scarce BEV segmentation. This approach bridges the resource asymmetry between readily available PV models and expensive BEV annotations, achieving improvements of up to 7.7% mIoU in semi-supervised learning and 10.7% in domain adaptation. PCT demonstrates that pretrained models represent an underutilized existing resource that can provide effective cross-task supervision through multi-task learning.

These three studies converge on a central theme: existing resources—be they segmentation masks that can be reinterpreted as boundaries, knowledge embedded in pretrained models, or abundantly available unlabeled data—contain valuable auxiliary signals that can enhance the primary task. The effectiveness of this principle is most pronounced in label-scarce scenarios, where auxiliary supervision provides crucial regularization. Across all frameworks, this thesis maintained deployment efficiency through modular designs that allow auxiliary components to be removed or adjusted based on computational constraints.

6.2 Future Directions

While this thesis demonstrated the potential of auxiliary supervision from existing resources, several directions warrant further investigation to strengthen and extend these principles.

Expanding auxiliary task diversity. Although this thesis explored semantic boundaries and cross-task supervision, and experimented with depth pseudo-labels from foundation models in PCT, systematic investigation of other auxiliary signals remains promising. Surface normals, instance-level cues, and vision-language priors from modern foundation models could provide additional complementary supervision. The challenge lies in principled selection of auxiliary tasks that provide genuine complementarity rather than redundancy, potentially through information-theoretic analysis of task relationships.

Theoretical foundations. The empirical results of the studies consistently show improvements, yet formal understanding of when and why auxiliary supervision succeeds remains limited. Future work should explore the geometric and statistical relationships between task spaces, potentially leading to principled frameworks for auxiliary task selection and optimal multi-task architecture design.

Experimentation into Knowledge Distillation. While task cues were explored in the three frameworks, with the widespread success of large pretrained models (*i.e.* foundation models), there is an opportunity to investigate how knowledge distillation from these models can serve as auxiliary supervision, especially towards BEV perception tasks where annotations are scarce and expensive to obtain.

Training efficiency optimization. The computational overhead introduced by auxiliary supervision, particularly the on-the-fly boundary generation in SBCB and teacher-student dynamics in BoundMatch, presents opportunities for optimization. Efficient implementations through lower-level languages, hardware-aware designs, or distillation techniques could reduce training costs while maintaining performance gains. The boundary generation algorithm, currently implemented in Python, could particularly benefit from optimization in compiled languages.

Temporal and video understanding. Current frameworks process frames independently, missing temporal structure in urban scenes. Extending auxiliary supervision to leverage motion cues, temporal consistency, and predictive tasks could improve both accuracy and efficiency for video-based perception systems.

Generalization beyond urban scenes. While this thesis demonstrated BoundMatch's applicability to remote sensing (LoveDA dataset) and explored medical imaging (ACDC dataset) with mixed results, systematic investigation across diverse domains remains valuable. Each domain—medical imaging, satellite imagery, robotic perception—presents unique opportunities for auxiliary supervision that could reveal universal principles while identifying domain-specific requirements.

Efficient modifications. As demonstrated with lightweight architectures in BoundMatch, auxiliary supervision can enhance resource-constrained models. Future work should explore more efficient auxiliary supervision strategies tailored for mobile deployment, potentially through neural architecture search that jointly optimizes the trade-off between added computational overhead and performance.

This thesis established that auxiliary supervision from existing resources provides

a practical path toward more efficient urban scene understanding. By demonstrating consistent improvements across diverse settings while maintaining deployment efficiency, I hope to inspire continued research into extracting maximum value from the resources our community has already created. As perception systems become increasingly critical to urban infrastructure and the gap between annotation costs and data availability continues to widen, such approaches will become essential for scalable, robust scene understanding.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Yoshimitsu Aoki, for his continuous guidance and support throughout my doctoral studies. His insightful feedback, high standards, and long-term encouragement have been essential to shaping both the direction and the quality of this research.

I am also deeply grateful to all members of the laboratory for the technical support and collaborative environment that made this work possible. I particularly appreciate the efforts behind the laboratory's research infrastructure and the administrative support that enabled day-to-day progress.

I would like to acknowledge my co-authors and collaborators, as well as the collaborative research institutes that supported this work. In particular, I am grateful to Asahi Aero (now Aero Toyota) and SenseTime Japan, and especially to Takumi Iida and Yoshinori Konishi, for their valuable discussions, collaboration, and support.

I sincerely thank the members of my dissertation review committee— Professor Kentaro Yoshioka, Professor Mariko Isogawa, Professor Masaaki Ikehara, and Professor Kei Noda— for their time, careful reading of my thesis, and constructive comments that helped improve this manuscript.

This work was supported in part by JSPS KAKENHI funding and by the JEES Mitsui Corporation Science and Technology Scholarship. I am grateful for this financial support, which enabled me to focus on my research and academic development.

Finally, I would like to thank my family for their unwavering support. I am especially grateful to my parents for their continued encouragement and understanding, and to my fiancée, Jiani Liu, for her patience, kindness, and support throughout this journey.

References

[1] Partnership for Analytics Research in Traffic Safety. Market penetration of advanced driver assistance systems (adas). Technical report, MITRE Corporation, 2024. Available at: https://www.mitre.org/sites/default/files/2024-09/PR-24-2614-PARTS-Market-Penetration-Advanced-Driver-Assistance-Systems.pdf.

[2] Fan Zhang, Bolei Zhou, Liu Liu, Yu Liu, Helene H Fung, Hui Lin, and Carlo Ratti. Urban visual intelligence: Studying cities with artificial vision. *Annals of the American Association of Geographers*, 114(5):1039–1071, 2024.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[4] Mohamed R Ibrahim, James Haworth, and Tao Cheng. Understanding cities with machine eyes: A review of deep computer vision in urban analytics. *Cities*, 96: 102481, 2020.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[6] Tianfei Zhou, Fei Zhang, Boyu Chang, Wenguan Wang, Ye Yuan, Ender Konukoglu, and Daniel Cremers. Image segmentation in foundation model era: A survey. *ArXiv*, abs/2408.12957, 2024.

[7] Luiz G. Galvao, Maysam F. Abbod, Tatiana Kalganova, Vasile Palade, and Md. Nazmul Huda. Pedestrian and vehicle detection in autonomous vehicle perception systems—a review. *MDPI Sensors*, 21, 2021.

[8] Sanjeda Akter, Ibne Farabi Shihab, and Anuj Sharma. Image segmentation with large language models: A survey with perspectives for intelligent transportation systems. *ArXiv*, abs/2506.14096, 2025.

[9] Siyuan Zhou, Duc Van Le, and Rui Tan. Ecseg: Edge-cloud switched image segmentation for autonomous vehicles. *21st Annual IEEE International Conference on Sensing, Communication , and Networking (SECON)*, pages 1–9, 2024.

[10] Jeyoen Kim, Takumi Soma, Tetsuya Manabe, and Aya Kojima. Image segmentation-based bicycle riding side identification method. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E106.A(5):775–783, 2023.

[11] SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Surface Vehicle Recommended Practice J3016_202104, SAE International, Warrendale, PA, April 2021. Revised 2021-04, Superseding J3016 JUN2018.

[12] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3): 1341–1360, 2021.

[13] Subaru Corporation. The technology that makes subaru different: Enjoyment and peace of mind. `https://www.subaru.co.jp/en/difference/technology/`, January 2020. Based on presentations at the Subaru Technology Briefing held on January 20, 2020. Accessed: 2026-01-03.

[14] Lucas Dal'Col, Miguel Oliveira, and Vítor Santos. Joint perception and prediction for autonomous driving: A survey. *ArXiv*, abs/2412.14088, 2024.

[15] International Organization for Standardization. ISO 21448:2022 road vehicles — safety of the intended functionality, 2022.

[16] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, and Xinge Zhu. Vision-Centric BEV Perception: A Survey . *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46 (12):10978–10997, 2024. ISSN 1939-3539.

[17] Siqi Fan, Zhe Wang, Xiao Huo, Yan Wang, and Jingjing Liu. Calibration-free bev representation for infrastructure perception. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9008–9013, 2023.

[18] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642, 2018.

[19] DXC Technology. Ensuring Effective Autonomous Vehicle Data Ingestion. `https://dxc.com/us/en/insights/perspectives/blogs/ensuring-effective-autonomous-vehicle-data-ingestion`, 2025. Accessed: 26 Sept. 2025.

[20] Jiaxin Zhang, Shiyuan Chen, Haoran Yin, Ruohong Mei, Xuan Liu, Cong Yang, Qian Zhang, and Wei Sui. A vision-centric approach for static map element annotation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 15861–15867, 2024.

[21] Shaoyu Chen, Yunchi Zhang, Bencheng Liao, Jiafeng Xie, Tianheng Cheng, Wei Sui, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene. *ArXiv*, abs/2304.09807, 2023.

[22] Tzofi Klinghoffer, Jonah Philion, Wenzheng Chen, Or Litany, Zan Gojcic, Jungseock Joo, Ramesh Raskar, Sanja Fidler, and José M. Álvarez. Towards viewpoint robustness in bird's eye view segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8481–8490, 2023.

[23] Yunze Man, Liangyan Gui, and Yu-Xiong Wang. Dualcross: Cross-modality cross-domain adaptation for monocular bev perception. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10910–10917, 2023.

[24] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, 2017.

[25] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1067–10676, 2018.

[26] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017.

[27] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, C. Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9296–9306, 2019.

[28] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yuxin Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020.

[29] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Žagar, and Alois Knoll. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. *IEEE Transactions on Intelligent Vehicles*, 9:7138–7164, 2024.

[30] Anurag Das, Yongqin Xian, Yang He, Zeynep Akata, and Bernt Schiele. Urban scene semantic segmentation with low-cost coarse annotation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5967–5976, 2022.

[31] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision (IJCV)*, 77:157–173, 2008.

[32] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[33] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[34] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *ArXiv*, abs/2301.00493, 2023.

[35] Eric Zimmermann, Justin Szeto, and Frederic Ratle. An empirical study of uncertainty in polygon annotation and the impact of quality assurance. In *Work-in-Progress and Demonstrations track, AAAI HCOMP*, 2023.

[36] Moshe Kimhi, Eden Grad, Lion Halika, and Chaim Baskin. Noisy annotations in semantic segmentation. *ArXiv*, abs/2406.10891, 2024.

[37] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11067–11075, 2019.

[38] Bowen Cheng, Ross B. Girshick, Piotr Doll'ar, Alexander C. Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15329–15337, 2021.

[39] Ruilin Yu, Cheng Wang, Yuxin Zhang, and Fuming Zhao. Decomposition and quantification of SOTIF requirements for perception systems of autonomous vehicles. *ArXiv*, abs/2501.10097, 2025.

[40] Dazhou Guo, Ligeng Zhu, Yuhang Lu, Hongkai Yu, and Song Wang. Small object sensitive segmentation of urban street scene with spatial adjacency between object classes. *IEEE Transactions on Image Processing*, 28:2643–2653, 2019.

[41] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: Detecting small road hazards for self-driving vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1099–1106, 2016.

[42] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[43] Sohyun Lee, Nam-Won Kim, Sungyeon Kim, and Suha Kwak. Frest: Feature restoration for semantic segmentation under multiple adverse conditions. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2024.

[44] Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021.

[45] Fabian Oboril, Cornelius Buerkle, Alon Sussmann, Simcha Bitton, and Simone Fabris. MTBF model for AVs – from perception errors to vehicle-level failures. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2022.

[46] Ming Yang, Shige Wang, Joshua Bakita, Thanh Vu, F. Donelson Smith, James H. Anderson, and Jan-Michael Frahm. Re-thinking cnn frameworks for time-sensitive autonomous-driving applications: Addressing an industrial challenge. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 305–317, 2019.

[47] Yujia Luo. *Time Constraints and Fault Tolerance in Autonomous Driving Systems*. Technical report, University of California, Berkeley, 2019. URL `https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-39.pdf`.

[48] NVIDIA Corporation. Nvidia autonomous vehicles safety report. Technical report, NVIDIA Corporation, 2025. URL `https://images.nvidia.com/aem-dam/en-zz/Solutions/auto-self-driving-safety-report.pdf`. Accessed: 2025-09-28.

[49] Ruba Islayem, Fatima Alhosani, Raghad Hashem, Afra Alzaabi, and Mahmoud Meribout. Hardware accelerators for autonomous cars: A review. *ArXiv*, abs/2405.00062, 2024.

[50] Smitha Rajagopal. Edge computing-smart cities: Optimizing data processing & resource management in urban environments. *Journal of Information Systems Engineering and Management*, 2025.

[51] Adam Zewe. Computers that power self-driving cars could be a huge driver of global carbon emissions. *MIT News*, 2023. Available at: https://news.mit.edu/2023/autonomous-vehicles-carbon-emissions-0113.

[52] Shervin Minaee, Yuri Boykov, Fatih Murat Porikli, Antonio J. Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44:3523–3542, 2020.

[53] Pascal Getreuer. Chan-vese segmentation. *Image Processing On Line*, 2:214–224, 2012.

[54] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision (IJCV)*, 1:321–331, 2004.

[55] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

[56] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, volume 39, pages 2481–2495, 2015.

[57] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[58] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2018.

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017.

[60] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin P. Murphy, and Alan Loddon Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40:834–848, 2018.

[61] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 2017.

[62] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2018.

[63] Xiaolong Wang, Ross B. Girshick, Abhinav Kumar Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[64] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. In *British Machine Vision Conference (BMVC)*, 2018.

[65] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *IEEE European Conference on Computer Vision (ECCV)*, 2018.

[66] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3141–3149, 2019.

[67] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, Humphrey Shi, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 603–612, 2019.

[68] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 593–602, 2019.

[69] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Global context networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[70] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[71] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *IEEE European Conference on Computer Vision (ECCV)*, 2020.

[72] Robin Strudel, Ricardo Garcia Pinel, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7242–7252, 2021.

[73] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[74] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2021.

[75] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *IEEE European Conference on Computer Vision (ECCV)*, 2018.

[76] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision (IJCV)*, 129:3051 – 3068, 2020.

[77] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9711–9720, 2021.

[78] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[79] Andrew G. Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019.

[80] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision (IJCV)*, 125:3–18, 2015.

[81] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IEEE International conference on computer vision (ICCV)*, pages 991–998, 2011.

[82] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1761–1770, 2017.

[83] Yuan Hu, Yunpeng Chen, Xiang Li, and Jiashi Feng. Dynamic feature fusion for semantic edge detection. *International Journal of Computer Vision (IJCV)*, 2019.

[84] Yun Liu, Ming-Ming Cheng, Jiawang Bian, Le Zhang, Peng-Tao Jiang, and Yang Cao. Semantic edge detection with diverse deep supervision. *International Journal of Computer Vision (IJCV)*, 130:179–198, 2022.

[85] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34:5586–5609, 2017.

[86] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[87] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5454–5463, 2017.

[88] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14636–14645, 2020.

[89] Ishan Misra, Abhinav Shrivastava, Abhinav Kumar Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, 2016.

[90] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *ArXiv*, abs/1706.05098, 2017.

[91] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018.

[92] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning (ICML)*, 2017.

[93] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[94] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.

[95] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *IEEE European Conference on Computer Vision (ECCV)*, 2018.

[96] Lukas Liebel and Marco Körner. Auxiliary tasks in multi-task learning. *ArXiv*, abs/1805.06334, 2018.

[97] Amir Zamir, Alexander Sax, Bokui (William) Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3712–3722, 2018.

[98] Towaki Takikawa, David Acuna, V. Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5228–5237, 2019.

[99] Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yunhai Tong. Improving semantic segmentation via decoupled body and edge supervision. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[100] Jiacong Xu, Zixiang Xiong, and S. Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[101] Quan Zhou, Yong Qiang, Yuwei Mo, Xiaofu Wu, and Longin Jan Latecki. Banet: Boundary-assistant encoder-decoder network for semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 23:25259–25270, 2022.

[102] Zhiding Yu, Rui Huang, Wonmin Byeon, Sifei Liu, Guilin Liu, Thomas Breuel, Anima Anandkumar, and Jan Kautz. Coupled segmentation and edge learning via dynamic graph propagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[103] Mingmin Zhen, Jinglu Wang, Lei Zhou, Shiwei Li, Tianwei Shen, Jiaxiang Shang, Tian Fang, and Quan Long. Joint semantic segmentation and boundary detection

using iterative pyramid contexts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13663–13672, 2020.

[104] Youqi Liao, Shuhao Kang, Jianping Li, Yang Liu, Yun Liu, Zhen Dong, Bisheng Yang, and Xieyuanli Chen. Mobile-seed: Joint semantic segmentation and boundary detection for mobile robots. *IEEE Robotics and Automation Letters*, 2024.

[105] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[106] Peng Zhou, Brian L. Price, Scott D. Cohen, Gregg Wilensky, and Larry S. Davis. Deepstrip: High-resolution boundary refinement. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10555–10564, 2020.

[107] Hao Hao Tan, Sitong Wu, and Jimin Pi. Semantic diffusion network for semantic segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[108] Chi Wang, Yunke Zhang, Miaomiao Cui, Jinlin Liu, Peiran Ren, Yin Yang, Xuansong Xie, Xiansheng Hua, Hujun Bao, and Weiwei Xu. Active boundary loss for semantic segmentation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2022.

[109] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Murat Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5897–5907, 2021.

[110] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30: 88–97, 2009.

[111] Adrian Pel'aez-Vegas, Pablo Mesejo, and Julián Luengo. A survey on semi-supervised semantic segmentation. *ArXiv*, abs/2302.09899, 2023.

[112] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *British Machine Vision Conference (BMVC)*, 2018.

[113] Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson W. H. Lau. Guided collaborative training for pixel-wise semi-supervised learning. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[114] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-trainingwork better for semi-supervised semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4258–4267, 2022.

[115] Ebenezer Tarubinga and Jenifer Kalafatovich Espinoza. Confidence-weighted boundary-aware learning for semi-supervised semantic segmentation. *ArXiv*, abs/2502.15152, 2025.

[116] Haonan Wang, Qixiang Zhang, Yi Li, and Xiaomeng Li. Allspark: Reborn labeled features from unlabeled in transformer for semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3627–3636, 2024.

[117] Iñigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana Cristina Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8199–8208, 2021.

[118] Xiangyu Zhao, Raviteja Vemulapalli, P. A. Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. Contrastive learning for label efficient semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10603–10613, 2021.

[119] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supevised semantic segmentation needs strong, varied perturbations. In *British Machine Vision Conference (BMVC)*, 2020.

[120] Zhen Zhao, Lihe Yang, Sifan Long, Jimin Pi, Luping Zhou, and Jingdong Wang. Augmentation matters: A simple-yet-effective approach to semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11350–11359, 2022.

[121] Zhen Zhao, Sifan Long, Jimin Pi, Jingdong Wang, and Luping Zhou. Instance-specific and model-adaptive supervision for semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23705–23714, 2022.

[122] Bo Sun, Yuqi Yang, Le Zhang, Ming-Ming Cheng, and Qibin Hou. Corrmatch: Label propagation via correlation matching for semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3107, 2023.

[123] Shuo Li, Yue He, Weiming Zhang, Wei Zhang, Xiao Tan, Junyu Han, Errui Ding, and Jingdong Wang. Cfcg: Semi-supervised semantic segmentation via cross-fusion and contour guidance supervision. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16302–16312, 2023.

[124] Wooseok Shin, Hyun Joon Park, Jin Sob Kim, and Sung Won Han. Revisiting and maximizing temporal knowledge in semi-supervised semantic segmentation. *ArXiv*, abs/2405.20610, 2024.

[125] Haikuan Zhang, Haitao Li, Xiufeng Zhang, Guanyu Yang, Atao Li, Weisheng Du, Shanshan Xue, and Chi Liu. Noise-robust consistency regularization for semi-supervised semantic segmentation. *Neural networks : the official journal of the International Neural Network Society*, 184:107041, 2024.

[126] Qiankun Ma, Ziyao Zhang, Pengchong Qiao, Yu Wang, Rongrong Ji, Chang Liu, and Jie Chen. Dual-level masked semantic inference for semi-supervised semantic segmentation. *IEEE Transactions on Multimedia*, 2025.

[127] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J. Davison. Bootstrapping semantic segmentation with regional contrast. In *International Conference on Learning Representations (ICLR)*, 2021.

[128] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4238–4247, 2022.

[129] Huayu Mai, Rui Sun, Tianzhu Zhang, and Feng Wu. Rankmatch: Exploring the better consistency regularization for semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3391–3401, 2024.

[130] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Logic-induced diagnostic reasoning for semi-supervised semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16151–16162, 2023.

[131] Xiaoyang Wang, Huihui Bai, Limin Yu, Yao Zhao, and Jimin Xiao. Towards the uncharted: Density-descending feature perturbation for semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3303–3312, 2024.

[132] Jianjian Yin, Yi Chen, Zhichao Zheng, Junsheng Zhou, and Yanhui Gu. Uncertainty-participation context consistency learning for semi-supervised semantic segmentation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024.

[133] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. ISBN 9780262033589.

[134] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1368–1377, 2020.

[135] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12671–12681, 2020.

[136] Yijiang Li, Xinjiang Wang, Lihe Yang, Litong Feng, Wayne Zhang, and Ying Gao. Diverse cotraining makes strong semi-supervised segmentor. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16009–16021, 2023.

[137] Jaemin Na, Jung-Woo Ha, HyungJin Chang, Dongyoon Han, and Wonjun Hwang. Switching temporary teachers for semi-supervised semantic segmentation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[138] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2613–2622, 2021.

[139] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781, 2022.

[140] Hanspeter A. Mallot, Heinrich H. Bülthoff, J. Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64:177–185, 2004.

[141] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11135–11144, 2020.

[142] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5:4867–4873, 2019.

[143] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[144] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jian-Yuan Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI Conference on Artificial Intelligence*, 2022.

[145] Brady Zhou and Philipp Krahenbuhl. Cross-view transformers for real-time map-view semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[146] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer. *ArXiv*, abs/2206.04584, 2022.

[147] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *ArXiv*, abs/2203.17270, 2022.

[148] Nikhil Gosala, Kürsat Petek, Paulo L. J. Drews-Jr, Wolfram Burgard, and Abhinav Valada. Skyeye: Self-supervised bird's-eye-view semantic mapping using monocular frontal view images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14901–14910, 2023.

[149] Junyu Zhu, Lina Liu, Yu Tang, Feng Wen, Wanlong Li, and Yong Liu. Semi-supervised learning for visual bird's eye view semantic segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9079–9085, 2023.

[150] Kai Jiang, Jiaxing Huang, Weiying Xie, Yunsong Li, Ling Shao, and Shijian Lu. Dabev: Unsupervised domain adaptation for bird's eye view perception. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2024.

[151] Germán Ros, Laura Sellart, Joanna Materzynska, David V ázquez, and Antonio M. López. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[152] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2012 (voc2012) results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[153] Gabriela Csurka, Diane Larlus, and Florent Perronnin. What is a good evaluation measure for semantic segmentation? In *British Machine Vision Conference (BMVC)*, 2013.

[154] Yuxiang Zhang, Sachin Mehta, and Anat Caspi. Rethinking semantic segmentation evaluation for explainability and model selection. *ArXiv*, abs/2101.08418, 2021.

[155] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation

methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016.

[156] Hao Li, Chenxin Tao, Xizhou Zhu, Xiaogang Wang, Gao Huang, and Jifeng Dai. Auto seg-loss: Searching metric surrogates for semantic segmentation. In *International Conference on Learning Representations (ICLR)*, 2020.

[157] Yitong Li, Changlun Zhang, and Hengyou Wang. Boundaries matters: A novel multi-branch semi-supervised semantic segmentation method. *IEEE Intelligent Systems*, 2024.

[158] Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3287–3297, 2023.

[159] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[160] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. doi: 10.23915/distill.00003. URL `http://distill.pub/2016/deconv-checkerboard`.

[161] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *ACM SIGACT Theory of Computing*, 8:415–428, 2012.

[162] Zhiding Yu, Weiyang Liu, Yang Zou, Chen Feng, Srikumar Ramalingam, B. V. K. Vijaya Kumar, and Jan Kautz. Simultaneous edge alignment and learning. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 2018.

[163] Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022.

[164] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jos é Manuel Álvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with

transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[165] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

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

[167] Lihe Yang, Zhen Zhao, and Hengshuang Zhao. Unimatch v2: Pushing the limit of semi-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025.

[168] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Herv é Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023.

[169] Haruya Ishikawa and Yoshimitsu Aoki. Boosting semantic segmentation by conditioning the backbone with semantic boundaries. *MDPI Sensors*, 23, 2023.

[170] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision (IJCV)*, 82:302–324, 2008.

[171] Zicheng Wang, Zhen Zhao, Luping Zhou, Dong Xu, Xiaoxia Xing, and Xiangyu Kong. Conflict-based cross-view consistency for semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19585–19595, 2023.

[172] Changqi Wang, Haoyu Xie, Yuhui Yuan, Chong Fu, and Xiangyu Yue. Space engage: Collaborative space supervision for contrastive-based semi-supervised semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 931–942, 2023.

[173] Rui Sun, Huayu Mai, Tianzhu Zhang, and Feng Wu. Daw: Exploring the better weighting function for semi-supervised semantic segmentation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[174] Jie Ma, Chuan Wang, Yang Liu, Liang Lin, and Guanbin Li. Enhanced soft label for semi-supervised semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1185–1195, 2023.

[175] Kebin Wu, Wenbin Li, and Xiaofei Xiao. Ipixmatch: Boost semi-supervised semantic segmentation with inter-pixel relation. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2024.

[176] Thuan Than, Nhat-Anh Nguyen-Dang, Dung Nguyen, Salwa K. Al Khatib, Ahmed Elhagry, Hai Phan, Yihui He, Zhiqiang Shen, Marios Savvides, and Dang Huynh. Knowledge consultation for semi-supervised semantic segmentation. *CoRR*, abs/2503.10693, 2025.

[177] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12159–12168, 2021.

[178] Olivier Bernard, Alain Lalande, Clément Zotti, Fréd éric Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel A. González Ballester, Gerard Sanromá, Sandy Napel, Steffen Erhard Petersen, Georgios Tziritas, Elias Grinias, Mahendra Khened, Varghese Alex Kollerathu, Ganapathy Krishnamurthi, Marc-Michel Rohé, Xavier Pennec, Maxime Sermesant, Fabian Isensee, Paul F. Jäger, Klaus Hermann Maier-Hein, Peter M. Full, Ivo Wolf, Sandy Engelhardt, Christian F. Baumgartner, Lisa M. Koch, Jelmer M. Wolterink, Ivana I*vs.* gum, Yeonggul Jang, Yoonmi Hong, Jay Patravali, Shubham Jain, Olivier Humbert, and Pierre-Marc Jodoin. Deep learning techniques for automatic mri cardiac multi-

structures segmentation and diagnosis: Is the problem solved? *IEEE Transactions on Medical Imaging*, 37:2514–2525, 2018.

[179] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[180] B. Dong, Pichao Wang, and Fan Wang. Head-free lightweight semantic segmentation with linear transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

[181] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *ArXiv*, abs/1708.04552, 2017.

[182] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando C Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

[183] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15:1929–1958, 2014.

[184] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10371–10381, 2024.

[185] Yaroslav Ganin, E. Ustinova, Hana Ajakan, Pascal Germain, H. Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. In *Journal of machine learning research*, 2015.

[186] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[187] Yanwei Pang, Yazhao Li, Jianbing Shen, and Ling Shao. Towards bridging semantic gap to improve semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4229–4238, 2019.

A.1 On-the-fly ground truth generation (OTFGT) algorithm

In this section, we provide a simple Python code for generating semantic boundaries from semantic segmentation masks using the OTFGT module, as shown in Code A.1.

The `mask2bdry` function is responsible for generating boundaries from binary masks. We chose to use OpenCV's `DistanceTransform` instead of `scipy`'s `distance_transform_edt` function because we observed that the former provides the same boundaries with a faster speed, around 40× faster for our setup. The `mask2sbd` function utilizes `mask2bdry` to generate semantic boundaries from semantic segmentation masks. For instance-sensitive boundaries, you can add another `mask2bdry` in the for-loop for instance segmentation masks and then obtain a category-specific boundary by using a bitwise OR operation of the two boundaries.

The OTFGT module is parallelizable since each for-loop is independent. However, we did not add multi-processing to the module since PyTorch already uses multi-processing for the dataloader, and adding extra multi-processing could lead to unnecessary overhead and slow down the dataloader.

Regarding the widely used offline preprocessing script introduced in [82], it is written in MATLAB and is not compatible with online preprocessing used in Python libraries like PyTorch. The MATLAB code uses a two-step method of first generating the entire binary boundaries and then extracting the semantic boundaries based on the candidate edges. The code uses a circular neighborhood approach, which requires looping every pixel.

To make a fair comparison, we translated the MATLAB code to Python and modified it to be compatible with online preprocessing. Our preprocessing pipeline for a single 1024×2048 image takes around 160ms, while the algorithm used in [82] takes around 1070ms. This demonstrates the superiority of the OTFGT, making it more suitable for online preprocessing in libraries like PyTorch.

Finally, we believe the OTFGT module can still be further optimized by using lower-level languages like C and wrapping the function with Cython, potentially leading to even better performance.

Code A.1: The algorithm for the on-the-fly ground-truth (OTFGT) module. The `mask2bdry` function takes in a binary segmentation mask and produces a boundary. The function uses `ignore_mask` to ignore unnecessary boundaries like boundaries near the frame of the image. The `mask2sbd` function takes in a segmentation mask and produces a semantic boundary detection ground-truth.

```
import cv2
import numpy as np

def mask2bdry(m, ignore_mask, radius):
    """Convert binary mask to boundaries.

    Args:
      m (np.ndarray): 2D binary mask
      ignore_mask (np.ndarray): 2D binary mask
      radius (int): boundary thickness

    Returns:
      bdry (np.ndarray): 2D boundary
    """
    inner = cv2.distanceTransform(((m + ignore_mask) > 0).astype(np.uint8), cv2.DIST_L2)
    outer = cv2.distanceTransform(((1.0 - m) > 0).astype(np.uint8), cv2.DIST_L2)
    dist = outer + inner

    dist[dist > radius] = 0
    bdry = (dist > 0).astype(np.uint8)
    return bdry

def mask2sbd(mask, ignore_indices=[], radius=2):
    """Convert Segmentation Mask to Semantic Boundaries.

    Args:
      mask (np.ndarray): segmentation mask
      ignore_indicies (List[int]): list of indices to ignore
      radius (int): boundary thickness

    Returns:
      bdrys (np.ndarray): 3D array containing boundaries
    """
    assert mask.ndim == 3
    num_labels, h, w = mask.shape

    # make ignore mask
    ignore_mask = np.zeros((h, w), dtype=np.uint8)
    for i in ignore_indices:
        ignore_mask += mask[i]

    bdrys = np.zeros_like(mask)
    for label in range(num_labels):
        m = mask[label]

        if label in ignore_indices:
            continue

        # if there are no class labels in the mask
        if not np.count_nonzero(m):
            continue

        bdrys[label] = mask2bdry(m, ignore_mask, radius)

    return bdrys
```

A.2 ROM and RUM

In Tabs. A.1a and A.1b, we present the region-based over-segmentation measure (ROM) and region-based under-segmentation measure (RUM) for the Cityscapes validation split, respectively.

For all models, the over-segmentation for categories such as "pole," "vegetation," "building," and "sidewalk"' improves significantly, with around -0.04 in ROM. This indicates that the segmentation quality around the boundaries of these categories has been notably enhanced by the SBCB framework. There are no specific categories where the metric degrades across all models in terms of over-segmentation. This suggests that the SBCB framework's improvements in boundary-aware features do not lead to a degradation in the overall segmentation quality for any particular category.

On the other hand, for the RUM metric, there seems to be a more visible trend of trade-offs. While there are drastic improvements for the "car"' category across all the models, the rest of the models do not exhibit a consistent trend. This suggests that the SBCB framework's effectiveness in reducing under-segmentation varies depending on the specific category.

A.3 SBCB performance on Cityscapes Benchmark

We also benchmarked DeepLabV3+ trained with the SBCB framework on the Cityscapes *test* split. We further fine-tune the model trained for the validation split (Tab. 3.19) for an additional 40k iterations using the training and validation split, following the approach in [102].

Tab. A.2 displays the performance of DeepLabV3+ trained using the SBCB framework and provides a comparison with other SOTA models on the Cityscapes Benchmark. While our approach did not surpass the performance of SOTA multi-task methods, DeepLabV3+ trained with the SBCB framework demonstrated competitive performance and was able to match the results of some SOTA models. These results underscore the effectiveness of the SBCB framework in enhancing the performance of DeepLabV3+ on the challenging Cityscapes dataset, positioning it as a compelling alternative in the landscape of semantic segmentation methods. We believe the lower performance of SBCB mainly stems from improper hyperparameter tuning for the Cityscapes benchmark and the learning dynamics

Table A.1: Per-category ROM and RUM for the Cityscapes validation split.

a Per-category ROM for the Cityscapes validation split.

Method	SBCB	ROM	road	swalk	build.	wall	fence	pole	tlight	sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike
PSPNet		0.078	0.054	0.197	0.218	0.054	0.066	0.315	0.004	0.017	0.176	0.063	0.079	0.038	0.031	0.037	0.018	0.019	0.013	0.014	0.075
	✓	0.061	0.035	0.125	0.145	0.04	0.056	0.29	0.01	0.014	0.165	0.058	0.061	0.029	0.03	0.013	0.017	0.009	0.006	0.007	0.042
DeepLabV3		0.072	0.066	0.133	0.194	0.045	0.066	0.344	0.004	0.014	0.173	0.061	0.07	0.049	0.025	0.034	0.011	0.012	0.007	0.006	0.059
	✓	0.06	0.07	0.112	0.155	0.034	0.041	0.303	0.005	0.018	0.128	0.051	0.062	0.033	0.023	0.015	0.012	0.006	0.007	0.008	0.05
DeepLabV3+		0.08	0.074	0.158	0.206	0.065	0.066	0.367	0.01	0.023	0.195	0.073	0.061	0.045	0.031	0.05	0.009	0.007	0.005	0.008	0.071
	✓	0.065	0.073	0.132	0.161	0.048	0.053	0.309	0.005	0.019	0.159	0.066	0.054	0.037	0.03	0.01	0.011	0.01	0.006	0.011	0.05

b Per-category RUM for the Cityscapes validation split.

Method	SBCB	RUM	road	swalk	build.	wall	fence	pole	tlight	sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike
PSPNet		0.102	0.095	0.136	0.45	0.009	0.019	0.19	0.042	0.095	0.334	0.022	0.102	0.093	0.069	0.151	0.002	0.006	0.009	0.011	0.108
	✓	0.098	0.079	0.123	0.399	0.01	0.015	0.188	0.056	0.1	0.343	0.026	0.121	0.096	0.078	0.094	0.001	0.004	0	0.012	0.108
DeepLabV3		0.104	0.079	0.157	0.436	0.016	0.031	0.179	0.047	0.096	0.328	0.023	0.126	0.093	0.063	0.162	0.003	0.005	0.002	0.01	0.113
	✓	0.1	0.1	0.147	0.429	0.011	0.026	0.192	0.05	0.104	0.3	0.03	0.116	0.107	0.074	0.078	0.002	0.004	0.001	0.009	0.117
DeepLabV3+		0.094	0.05	0.142	0.396	0.013	0.025	0.156	0.048	0.089	0.302	0.027	0.093	0.098	0.062	0.143	0.001	0.006	0.002	0.01	0.115
	✓	0.086	0.087	0.147	0.358	0.013	0.02	0.156	0.045	0.072	0.262	0.029	0.104	0.101	0.078	0.061	0.001	0.002	0	0.01	0.094

Table A.2: Comparison of our method and state-of-the-art approaches on the Cityscapes *test* split. All methods are trained using only fine annotations, without additional coarse data or Mapillary Vistas pre-training.

Method	**Backbone**	mIoU ↑
Without boundary auxiliary		
PSPNet [59]	ResNet-101	78.4
PSANet [65]	ResNet-101	80.1
SeENet [187]	ResNet-101	81.2
ANNNet [68]	ResNet-101	81.3
CCNet [67]	ResNet-101	81.4
DANet [66]	ResNet-101	81.5
With boundary auxiliary		
RPCNet [103]	ResNet-101	81.8
CSEL [102]	HED ResNet-101	82.1
With SBCB auxiliary (Ours)		
DeepLabV3+ + SBCB	ResNet-101	81.4
DeepLabV3+ + SBCB	HED ResNet-101	81.0

changing due to the additional multi-task learning objective. Further investigation into how we can perform better fine-tuning with SBCB could potentially lead to improved performance. While not outperforming all SOTA models, our approach exhibits valuable competitiveness, as our DeepLabV3+ model performs just as well as the prior non-boundary auxiliary methods.

A.4 BiSeNet + SBCB

In Fig. A.1(a), we show a detailed architecture diagram showing which features of the BiSeNet backbone are used in the SBD head. In both BiSeNet V1 and V2, the architecture is composed of a Context Path and a Spatial Path. We use the three stages of the spatial path for the earlier Side Layers of the SBD head. We used the last feature of the Aggregation Layer for the last Side Layer.

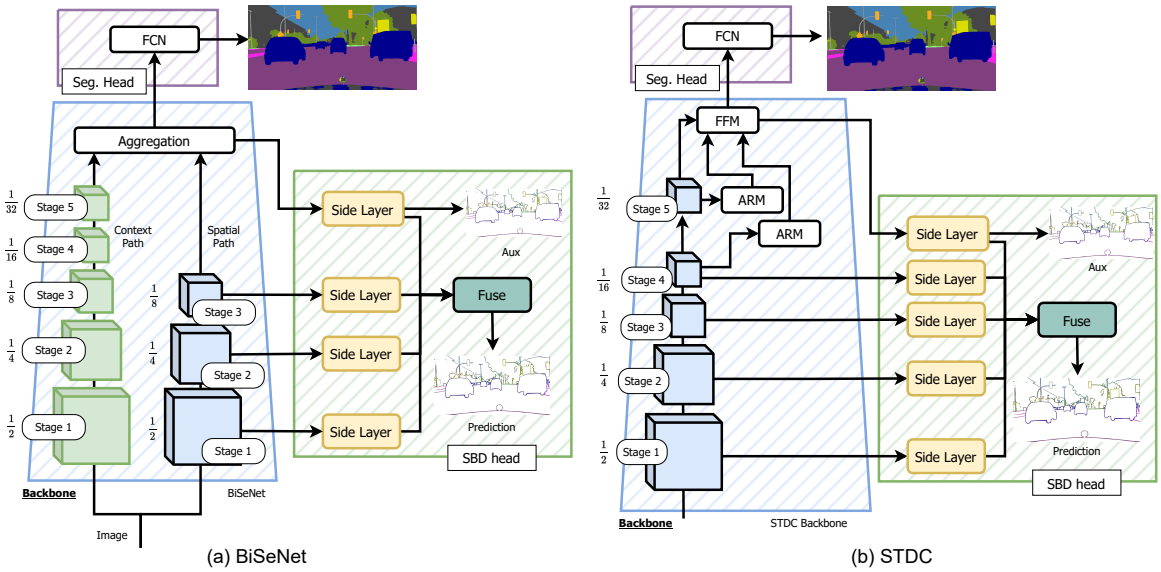


Figure A.1: We show how we applied the SBCB framework for BiSeNet and STDC in (a) and (b) respectively.

A.5 STDC + SBCB

In Fig. A.1(b), we show a detailed architecture diagram showing how we applied the SBCB framework to the STDC architecture. The architecture is more reminiscent of a ResNet-like hierarchical backbone, but the original STDC applies a Detail Head, which uses the features of the third stage. Instead, we remove the Detail Head and instead add an SBD head by using the first four stages for the binary side layer and the final output of the FFM as the input to the semantic side layer.

A.6 Feature Fusion with SBCB

In addition to our primary focus on the Semantic Boundary Conditioned Boosting (SBCB) framework, we also explored the explicit utilization of features obtained from the semantic boundary head through two feature fusion techniques. These techniques aim to further enhance segmentation performance by leveraging the knowledge learned in the SBD head.

Channel-Merge. The first technique, known as Channel-Merge, involves straightforward channel concatenation with a few convolutional kernels to facilitate feature fusion, as depicted in Fig. A.2a. In this method, we take the features before upsampling from the

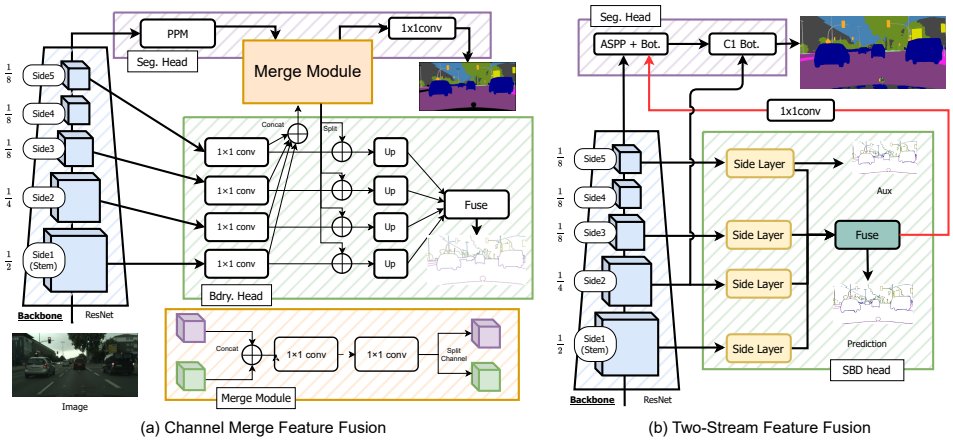


Figure A.2: In (a), we show how to apply the Channel-Merge module for explicit feature fusion based on the SBCB framework. In (b) we show how to apply the two-stream approach for explicit feature fusion modeled after the GSCNN architecture.

Side Layers of the SBD head. Each feature is resized and concatenated into a single tensor, which is further combined with the features obtained from the segmentation head (e.g. Pyramid Pooling Module (PPM)). To integrate the features effectively in the channel direction, we employ two 1×1 convolutional kernels. It is worth noting that the number of convolutions can be adjusted according to specific requirements.

Two-Stream Merge. The second technique, known as Two-Stream Merge, establishes a direct connection between the features learned in the SBD head and the segmentation head, achieved by employing a 1×1 convolutional kernel, as illustrated in Fig. A.2b. This approach is inspired by the GSCNN architecture, wherein we treat the SBD head as the Shape Stream, and the fusion mechanism mirrors that of the GSCNN.

While these feature fusion techniques are not the primary focus of this chapter, they serve as valuable supplementary approaches to leverage the knowledge acquired in the SBD head for improved segmentation performance. By explicitly incorporating boundary-related information through these fusion methods, we seek to further enhance the segmentation quality and offer additional insights into the potential benefits of integrating boundary-aware features into the segmentation process.

Tabs. A.3 and A.4 presents the results of two baseline architectures, where we applied the SBCB framework and feature fusion methods. The evaluation was performed on two datasets: Cityscapes and Synthia.

While we initially expected the Two-Stream architecture to exhibit superior perfor-

Table A.3: Comparison of feature fusion methods with baseline methods on the Cityscapes validation split.

Model		mIoU ↑	Δ	Fscore ↑	Δ
PSPNet		77.6		70.2	
	+SBCB	78.7	+1.1	**73.3**	+3.1
	Two-Stream Merge	78.7	+1.0	73.0	+2.8
	Channel-Merge	**79.1**	+1.5	73.2	+3.0
DeepLabV3+		79.5		71.4	
	+SBCB	80.2	+0.7	73.7	+2.3
	Two-Stream Merge	**80.5**	+1.0	73.6	+2.2
	Channel-Merge	**80.5**	+1.0	**74.5**	+3.1

Table A.4: Comparison of feature fusion methods with baseline methods on the Synthia dataset.

Model		mIoU ↑	Δ	Fscore ↑	Δ
PSPNet		70.5		63.7	
	+SBCB	71.7	+1.2	65.9	+2.2
	Two-Stream Merge	71.3	+0.8	65.5	+1.8
	Channel-Merge	**72.5**	+2.0	**67.2**	+3.5
DeepLabV3+		72.4		67.2	
	+SBCB	73.5	+1.1	69.1	+1.9
	Two-Stream Merge	73.8	+1.4	69.2	+2.0
	Channel-Merge	**74.0**	+1.6	**69.9**	+2.7

mance, we observed that while it performed well on certain datasets, it was outperformed by the SBCB framework on some occasions. Surprisingly, the Channel-Merge architecture achieved the best IoU metrics across all models and datasets, and it also displayed the best boundary F-scores in most cases.

Comparing the SBCB framework with the Channel-Merge approach against the baseline model, we noticed a significant improvement in segmentation performance when using the SBCB framework. This highlights the substantial contributions of the SBCB framework, primarily driven by the representational capabilities of the backbone, while the feature fusion methods yielded smaller improvements on top of the improvements obtained from the SBCB framework. This may also highlight the importance of the boundary-aware feature representations learned in the backbone.

Furthermore, we observed that the Channel-Merge method proved particularly advantageous when training ground truth masks were precise. The Synthia dataset, which

provides generated segmentation masks, benefits more from the feature fusion approach due to its cleaner annotations, whereas the Cityscapes dataset naturally contains noisier boundaries attributed to human annotators.

It is important to note that feature fusion methods introduce a dependency of the segmentation head on the SBD head, which in turn increases computational costs. This dependency also introduces the complexity of designing the fusion methods, which must be carefully tuned to avoid instability. For example, Channel-Merge might be better than Two-Stream Merge due to the Merge Module mixing various representations obtained earlier rather than a single representation obtained at the end of the SBD head.

The SBCB framework remains instrumental in consistently enhancing existing segmentation models without modifications to the original architecture. The insights gleaned from the SBD heads hold promise for inspiring novel joint architectures, such as Channel-Merge and Two-Stream Merge.

A.7 Harmonious Batch Normalization (HBN)

Algorithm 1: One training iteration with HBN.

Input : Student weights θ^S, teacher weights θ^T, mini-batch (x, u^w, u^s), EMA decay α, BN momentum ρ

Output : Updated θ^S, θ^T

1. **Teacher forward pass** (`train` mode): $p^{T,L}, p^{T,w}, p^{T,s} \leftarrow f_{\theta^T}(x, u^w, u^s)$; update teacher BN buffers with ρ.
2. **Generate hard pseudo-labels**: $\hat{p}^T \leftarrow \operatorname{argmax} p_c^{T,w}$ (apply confidence τ if needed).
3. **Student forward pass** (`train` mode, twin-view): $p^{S,L}, p^{S,w}, p^{S,s} \leftarrow f_{\theta^S}(x, u^w, u^s)$; also update student BN buffers with ρ.
4. **Loss computation & student update**: compute $\mathcal{L}^L$ on $(p^{S,L}, y)$ and $\mathcal{L}^U$ on $(p^{S,s}, \hat{p}^T)$; update θ^S.
5. **EMA weight update (teacher)**: $\theta^T \leftarrow \alpha\,\theta^T + (1-\alpha)\,\theta^S$.

In teacher-student consistency training the teacher weights θ^T are updated as an exponential moving average (EMA) of the student weights θ^S. Copying the student's batch normalization (BN) running statistics, common in prior work, makes those statistics *incompatible* with θ^T, because the teacher's parameter trajectory and input distribution differ from those of the student. HBN fixes this by

- **Teacher recalibration:** computing and accumulating BN statistics *inside the teacher's own forward pass* on every iteration, yielding self-consistent normalization.

- **Student twin-view:** forwarding the weakly augmented unlabeled images u^w through the *student* as well. u^w is normally not used in the student forward pass and only used to generate pseudo-labels with the teacher, but this simple step aligns the stochastic input distribution seen by both networks, further reducing BN mismatch.

A mini-batch $\mathcal{B} = \mathcal{B}^L \cup \mathcal{B}^{U,w} \cup \mathcal{B}^{U,s}$ contains N images: labeled $x \in \mathcal{B}^L$, weakly augmented unlabeled $u^w \in \mathcal{B}^{U,w}$, and strongly augmented counterparts $u^s \in \mathcal{B}^{U,s}$. For a BN layer ℓ in the *teacher* network f_{θ^T}, let the activation tensor be $h^{(\ell)} \in \mathbb{R}^{N \times C_\ell \times H_\ell \times W_\ell}$.

During the teacher forward pass (`train` mode) we compute

$$\mu_{\mathcal{B}}^{(\ell)} = \tfrac{1}{N} \sum_{i=1}^{N} h_i^{(\ell)}, \qquad (\sigma_{\mathcal{B}}^{(\ell)})^2 = \tfrac{1}{N} \sum_{i=1}^{N} \Big(h_i^{(\ell)} - \mu_{\mathcal{B}}^{(\ell)}\Big)^2. \tag{A.1}$$

The teacher's running statistics are updated with BN momentum ρ:

$$\tilde{\mu}^{T,(\ell)} \leftarrow (1-\rho)\, \tilde{\mu}^{T,(\ell)} + \rho\, \mu_{\mathcal{B}}^{(\ell)}, \tag{A.2}$$

$$(\tilde{\sigma}^{\,T,(\ell)})^2 \leftarrow (1-\rho)\, (\tilde{\sigma}^{\,T,(\ell)})^2 + \rho\, (\sigma_{\mathcal{B}}^{(\ell)})^2. \tag{A.3}$$

Normalization and affine transformation then proceed as usual. Algorithm 1 shows one training iteration with HBN.

Note that HBN *can not* be applied to SS-SS algorithms that do not use a teacher-student framework with EMA updates.

A.8 Boundary Detection Head Design

In Fig. A.3, we illustrate the architecture of the boundary detection head used in our BoundMatch framework. We follow CASENet [82] and use a multi-scale boundary detection head that consists of four "Side Layers". Each "Side Layer" consists of a 1×1 convolution up sampling to 1/2 of the input image size, followed by a 3×3 convolution similar to the one introduced in SBCB (Chapter 3). For more details on the boundary detection head, please refer to the original paper [82].

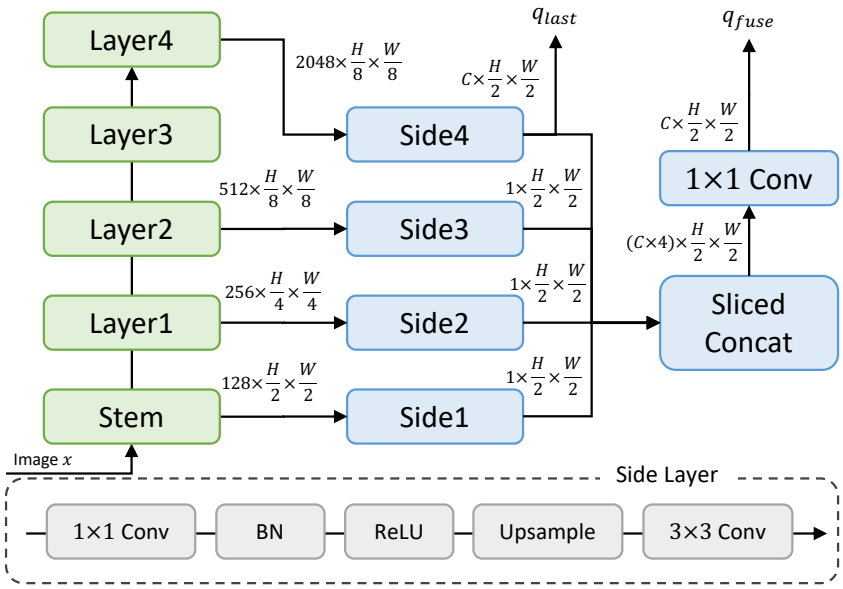


Figure A.3: Architecture of the boundary detection head. The boundary head consists of four "Side Layers" which consists of a 1×1 convolution up sampling to $1/2$ of the input image size, and a 3×3 convolution. The outputs are then fused together with a sliced concatenation operation followed by a 1×1 convolution to produce the final boundary prediction.

A.9 DINOv2 with BoundMatch

In Fig. A.4, we illustrate the architecture of DPT with boundary detection head used in our BoundMatch framework. Following the design of DPT, we use the reassembled feature maps used for the Fusion modules as the hierarchical features for the boundary detection head. For more details on DPT, please refer to the original paper [177].

For training under semi-supervised setting with BoundMatch, we use the same hyper-parameters introduced in [167]. Other hyperparameters such as λ_{bdry} and τ_{bdry} remain the same as those used in SAMTH.

A.10 Lightweight Models with BoundMatch

For DeepLabV3+ with MobileNet-V2 [78], we attach the boundary detection head to the output of the backbone in the same manner as DeepLabV3+ with ResNet backbones. The hyperparameters used for training are identical to the setup introduced in Sec. 4.4.1.

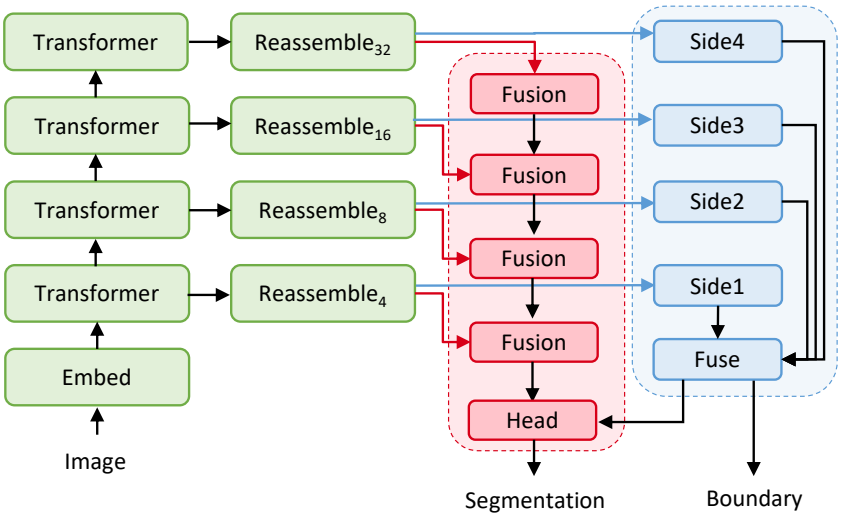


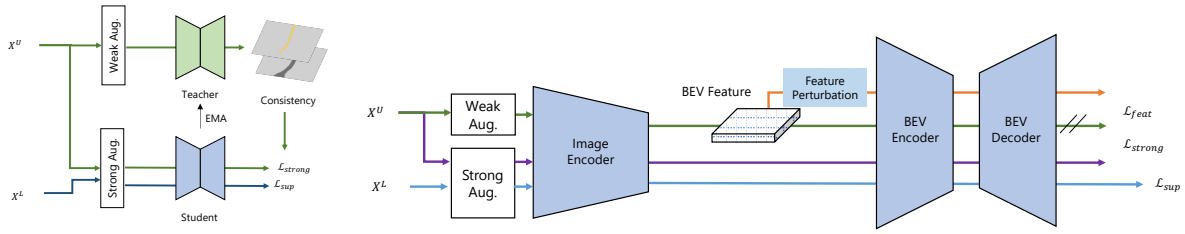
Figure A.4: Architecture of DPT with boundary detection head used for BoundMatch framework.

For integrating BoundMatch to AFFormer-Tiny [180], we follow Mobile-Seed [104] and use four features from the backbone (Blocks 1, 3, 5, and 6). The input to the decoder assumes input channel size of 216 and the feature channel size before the classification layer is 96, which is identical to Mobile-Seed and AFFormer. For feature fusion with BSF, we use one convolutional layer which takes in concatenated features from the backbone and boundary detection head to obtain the final feature. The training hyperparamters are identical to the one used in AFFormer [180].

A.11 Additional BEV Segmentation Baselines

A.11.1 Mean-Teacher Baseline for BEV Segmentation

Fig. A.5a shows the mean-teacher [32] applied to BEV segmentation task. The baseline approach employs a teacher-student framework for semi-supervised learning with consistency regularization. Unlabeled data (X^U) is fed through weak augmentation to the teacher model, while both unlabeled and labeled data (X^L) undergo strong augmentation before being processed by the student model. The student model is trained using two loss components: a supervised loss ($\mathcal{L}_{sup}$) computed on the labeled data and a consistency loss that enforces agreement between the teacher's predictions on weakly-augmented inputs and



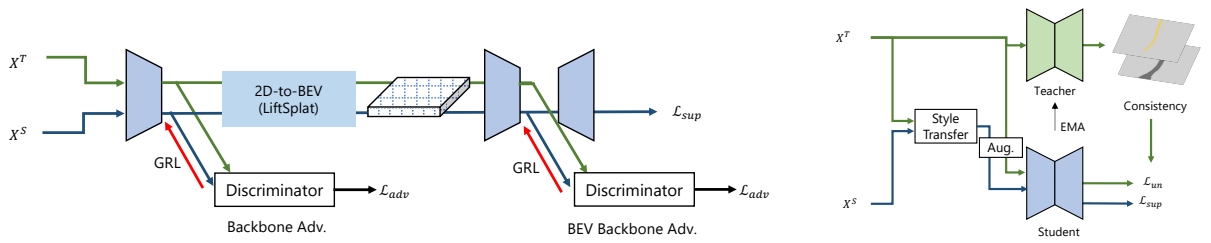
(**a**) Mean-Teacher Baseline. (**b**) UniMatch-like (UniPerb) Baseline.

Figure A.5: Overall figure caption describing both images.

the student's predictions on strongly-augmented versions of the same inputs. The teacher model parameters are not trained directly but are instead updated as an exponential moving average (EMA) of the student model's parameters, providing stable pseudo-labels for the unlabeled data. This asymmetric augmentation strategy—weak augmentation for generating pseudo-labels and strong augmentation for training—encourages the student to learn robust representations that are invariant to strong perturbations while maintaining consistent predictions.

A.11.2 UniMatch-like Baseline for BEV Segmentation

Fig. A.5b shows the UniMatch-like (UniPerb) [33] baseline for the BEV segmentation task. This method adapts the UniMatch perturbation strategy to BEV segmentation by introducing feature-level perturbations in the BEV representation space along with a single strong image augmentation. Unlabeled data (X^U) undergoes both weak and strong image augmentations before being processed by the image encoder to produce BEV features. We apply feature perturbation directly to these BEV features, which are then fed into the BEV encoder and decoder to generate predictions. The model is trained using three loss components: a supervised loss ($\mathcal{L}_{sup}$) on labeled data, an unsupervised consistency loss ($\mathcal{L}_{unsup}$) that enforces agreement between predictions from weakly-augmented and feature-perturbed inputs, and a strong augmentation loss ($\mathcal{L}_{strong}$) on the strongly-augmented branch. Unlike the dual-stream image augmentation approach in the full UniMatch framework, we focus on the feature perturbation mechanism to introduce variability, which is particularly well-suited for BEV segmentation where feature-space perturbations can capture view-specific uncertainties inherent in the perspective-to-BEV transformation.



(**a**) DomainAdv Baseline. (**b**) FDA Baseline.

Figure A.6: Overall figure caption describing both images.

A.11.3 DomainAdv Baseline for BEV Segmentation

Fig. A.6a shows the DomainAdv baseline [23] for the BEV segmentation task. This method employs multi-level adversarial domain adaptation for BEV segmentation, applying domain alignment at two critical stages of the processing pipeline. Source domain data (X^S) with labels is processed through a backbone encoder, undergoes 2D-to-BEV transformation using LiftSplat (LSS), and produces final predictions via a BEV backbone. To achieve domain adaptation, the framework incorporates adversarial training at two levels: first at the 2D backbone features before the view transformation, and second at the BEV features after the transformation. Each adversarial branch utilizes a Gradient Reversal Layer (GRL) [185] connected to a domain discriminator, which attempts to distinguish between source and target domain features while the GRL encourages the feature extractors to learn domain-invariant representations. The model is optimized using a supervised loss ($\mathcal{L}_{sup}$) on the labeled source data and adversarial losses ($\mathcal{L}_{adv}$) at both the 2D and BEV feature levels. This dual-level adversarial strategy is designed to address domain shift at multiple stages: the 2D adversarial branch aligns image-level features before view transformation, while the BEV adversarial branch ensures domain invariance in the transformed bird's-eye-view representation space, where domain-specific artifacts from the perspective transformation may emerge.

A.11.4 FDA Baseline for BEV Segmentation

Fig. A.6b shows the FDA [186] applied to BEV segmentation task. This method combines domain adaptation with semi-supervised learning through a teacher-student framework that leverages style transfer for cross-domain alignment. The approach processes two types

of data: unlabeled target domain data (X^T) and labeled source domain data (X^S). Target domain data is fed directly to the teacher model without modification, while both source and target domain data undergo style transfer followed by augmentation before being processed by the student model. The style transfer module serves as the domain adaptation mechanism, transforming source domain images to match the visual characteristics of the target domain, thereby reducing the domain gap at the input level. The student model is trained using a supervised loss ($\mathcal{L}_{sup}$) on the style-transferred and augmented source data, and an unsupervised consistency loss ($\mathcal{L}_{un}$) that enforces agreement between the teacher's predictions on original target domain images and the student's predictions on their style-transferred and augmented versions. The teacher model parameters are updated via exponential moving average (EMA) of the student weights, providing stable pseudo-labels. This architecture effectively addresses both the domain shift problem through style transfer and the limited labeled data problem through consistency regularization, enabling the model to leverage labeled source domain data while adapting to the unlabeled target domain.