FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Introducing Domain Knowledge to Scene Parsing in Autonomous Driving

Rafael Valente Cristino

Mestrado em Engenharia Informática e Computação

Supervisor: Ricardo Pereira de Magalhães Cruz
Co-Supervisor: Jaime dos Santos Cardoso

July 28, 2023

Introducing Domain Knowledge to Scene Parsing in Autonomous Driving

Rafael Valente Cristino

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: António Augusto de Sousa

External Examiner: André Ribeiro da Silva de Almeida Marçal

Supervisor: Ricardo Pereira de Magalhães Cruz

July 28, 2023

Resumo

A área da condução autónoma tem visto um aumento em investimento e investigação à medida que as empresas trabalham para alcançar a automação total da condução. Uma parte importante de tal sistema é a unidade de perceção, que se centra na segmentação do ambiente em redor do carro e é normalmente implementada com redes neuronais convolucionais. No entanto, as abordagens existentes baseadas em redes neuronais contêm limitações, como a sua capacidade de generalização – a rede falha em fazer os ajustes apropriados ao analisar uma situação que não apareceu no conjunto de dados usado para a treinar. Uma possível explicação é que a rede não tem conhecimento do domínio intrínseco da tarefa.

Esta dissertação procura melhorar as capacidades de generalização das redes neuronais de segmentação introduzindo conhecimento do domínio ordinal através de funções de custo aumentadas que penalizam a rede quando as restrições ordinais são quebradas. Duas categorias de funções de custo para segmentação ordinal foram estudadas: (**1**) intra-píxel, onde cada píxel é tratado individualmente, com a promoção de unimodalidade na sua distribuição probabilística; e (**2**) espacial, onde cada píxel é considerado no contexto da sua vizinhança e a superfície de contacto entre classes não ordinalmente adjacentes é minimizada.

Para avaliar o impacto dos métodos em domínios de condução autónoma, os modelos foram treinados com o conjunto de dados BDD100K e testados em dois cenários principais: (**1**) no conjunto de dados BDD100K; e (**2**) num domínio fora de distribuição, através do conjunto de dados Cityscapes, de forma a avaliar a sua capacidade de generalização. Foram obtidos resultados promissores – os métodos ordinais alcançaram melhorias máximas no coeficiente de Dice com um valor absoluto de 1.5% (4% em termos relativos) no conjunto de dados BDD100K e um valor absoluto de 5.3% (15.7% em termos relativos) no domínio fora de distribuição. Estes resultados indicam o potencial benefício de incorporar consistência ordinal de forma a melhorar as capacidades de aprendizagem e generalização dos modelos de segmentação semântica para condução autónoma.

Palavras-Chave: Segmentação ordinal. Segmentação semântica. Conhecimento de domínio. Rede neuronal profunda. Condução autónoma. Análise de cena. Aprendizagem profunda.

Abstract

Autonomous driving has seen a surge in investment and research as companies work to achieve full driving automation. One major part of such a system is the perception unit, which centers around scene parsing, commonly implemented with convolutional neural networks. However, existing neural network approaches contain limitations, such as their generalization ability – the networks fail to make appropriate adjustments when parsing situations outside their training domain. An explanation is that the neural network does not intrinsically have domain knowledge of the task.

This dissertation tackles the lack of generalization ability in semantic segmentation neural networks by introducing ordinal domain knowledge through augmented loss functions that penalize the network when ordinal constraints are broken. Two categories of loss functions for ordinal segmentation were studied: (**1**) pixel-wise, where each pixel is treated individually by promoting unimodality in its probability distribution; and (**2**) spatial, where each pixel is considered in the context of its neighborhood and the contact surface between non-ordinally adjacent classes is minimized.

To evaluate the impact of the methods in autonomous driving domains, the models were trained with the BDD100K dataset and tested in two main scenarios: (**1**) on the BDD100K dataset; and (**2**) in an out-of-distribution domain, through the Cityscapes dataset, to evaluate their generalization ability. Promising results were obtained – the ordinal methods achieved maximum improvements in the Dice coefficient with an absolute value of 1.5% (4% in relative terms) on the BDD100K dataset and an absolute value of 5.3% (15.7% in relative terms) in the out-of-distribution domain. These findings indicate the potential of incorporating ordinal consistency to enhance the learning capabilities and generalizability of autonomous driving semantic segmentation models.

Keywords: Ordinal segmentation. Semantic segmentation. Domain knowledge. Deep neural network. Autonomous driving. Scene parsing. Deep learning.

Acknowledgements

I am extremely grateful to my supervisor, Ricardo Cruz, for all the guidance and feedback during the project execution, from the project proposal specification to the development of the solution and final document writing, and to my co-supervisor, Prof. Jaime Cardoso, for all the support, insightful discussions, and the opportunity to conduct my work as part of project THEIA.

Special thanks to Bacalhaus, for all the study, work, and fun sessions, which have made the last 5 years truly memorable.

Lastly, I express my deepest appreciation to all of my family and friends, without whom this journey would not have been possible.

Rafael Cristino

This thesis was partially supported by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 047264; Funding Reference: POCI-01-0247-FEDER-047264].

Contents

List of Figures

List of Tables

Abbreviations

CNN(s)	Convolutional Neural Network(s)
CS	Contact Surface Metric
CSDT	Contact Surface Distance Transform Loss
CSNP	Contact Surface Neighbor Pixels Loss
DNN(s)	Deep Neural Network(s)
FCN	Fully Convolutional Network
OOD	Out of Distribution
SSL	Semi-Supervised Learning
UP	Unimodal Percentage Metric

Chapter 1

Introduction

The present chapter introduces the dissertation topic. Section 1.1 goes over the autonomous driving context. Section 1.2 motivates the problem to be researched. Section 1.3 defines the objectives for the work to be carried out. Section 1.4 provides an overview of the dissertation document structure.

1.1 Context

The latest global road safety reports of the World Health Organization show that every year, approximately 1.3 million people lose their lives due to road accidents. Over half of this number is among vulnerable road users – pedestrians, cyclists, and motorcyclists. Furthermore, the same reports show that road accidents are the leading cause of death for children and young adults aged 5 to 29. [1, 2]

Other studies report that around 94% of road accidents are a result of driver error, e.g., internal and external distractions, driving too fast for conditions, driving too fast for the curve, misjudgment of gaps or the speed of others, overcompensation, lack of sleep, and more. [3]

If an automated alternative replaced the human driver, many unnecessary deaths could potentially be prevented, given the high percentage of road accidents due to human error. This is one of the main motivations behind autonomous vehicles – a vision that has seen a surge in investment and research from both academia and industry. Beyond safety, autonomous vehicles can potentially improve road congestion, parking, and travel comfort. [4]

The Society of Automotive Engineers (SAE) provides a taxonomy with detailed definitions for six levels of driving automation, ranging from no driving automation (level 0) to full driving automation (level 5) [5]. A suitable visualization of these levels can be consulted in Figure 1.1. Currently, there are various level 2 driving automation certified vehicle offerings, such as vehicles from Tesla, Hyundai, Ford, and more[1]. At this level the driver must remain alert and is required

[1] https://www.jdpower.com/cars/shopping-guides/levels-of-autonomous-driving-explained.

to actively supervise the technology. Recently, Honda became the world's first automaker to offer a level 3 autonomous driving certified vehicle[2]. In contrast, Mercedes-Benz became the first automaker to have a level 3 autonomous driving system certified anywhere in the United States[3]. The jump from level 2 to level 3 is significant, as starting from level 3, the driver is no longer required to monitor the environment. However, the car will ask for the driver's intervention if some problem or a more complex situation arises.

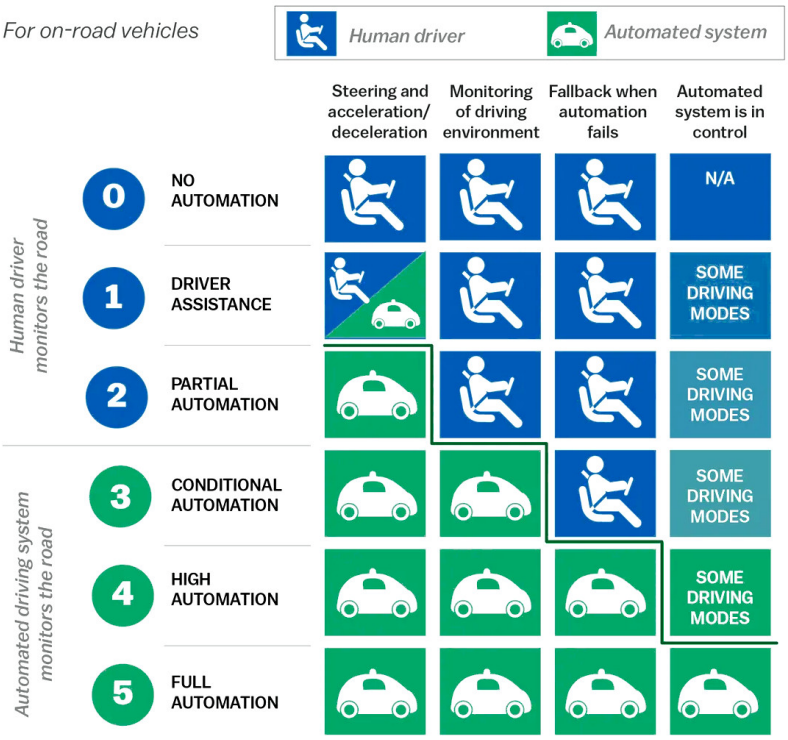


Figure 1.1: SAE levels of driving automation[4] [5].

A crucial part of such a system is the perception sub-system, which provides the car with information about the surrounding environment. This information retrieval process centers around scene parsing – semantically segmenting the environment around the car – and object detection, using the data from its sensors, normally a combination of RGB cameras and LiDAR [6]. When it comes to semantic segmentation, the main focus of this dissertation, the current state-of-the-art approaches generally use deep neural networks with an encoder-decoder architecture [7, 8, 9].

The work of this thesis is supported and executed as part of the project THEIA – Automated Perception Driving (POCI-01-0247-FEDER-047264), a partnership between the University of Porto and Bosch Portugal, which has as its main purpose the research and development of intelligent perception algorithms for autonomous vehicles.

[2] https://www.autox.com/news/car-news/worlds-first-certified-level-3-autonomous-car-to-hit-streets-of-japan-109099/.

[3] https://www.freethink.com/hard-tech/drive-pilot.

[4] Adapted from Vox, https://www.vox.com/2016/9/19/12966680/department-of-transportation-automated-vehicles.

1.2 Motivation

Existing scene parsing approaches contain limitations. One of the issues is their lack of generalization ability, which means that the network fails to make appropriate predictions when parsing a situation that did not occur in the dataset used in its training [10]. An explanation for this reality is that perhaps the neural network model does not have the necessary intrinsic domain knowledge of the task – it failed to accurately infer high-level relations from the data used to train it. For example, it does not know that the lane markings only make sense inside the lane, that both the sky and lane are contiguous, etc.

One extreme instance showcasing the lack of generalization described in the previous paragraph is the existence of adversarial examples. Adversarial examples are 'natural images with visually imperceptible perturbations added' [11, p. 1]. When the original and the adversarial example are fed into the neural network, they result in completely different outputs. An example is given in Figure 1.2, where adding noise to an initially correctly classified image results in a completely different prediction, even though the modified image looks the same as the original when seen by a human.

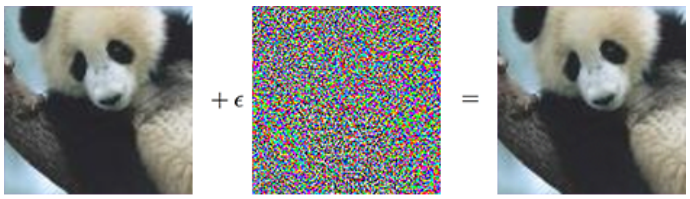


Figure 1.2: Classifier fooled by an adversarial example[5].

Deep neural networks are like black boxes, i.e., humans have no control over the representations they learn, leading to the characteristics described above. This is a problem for autonomous driving, an area where these mechanisms are safety-critical[6]. By helping the network learn higher-level concepts, these risks can possibly be mitigated and its reliability improved.

1.3 Objectives

As motivated by the previous section, this dissertation proposes to address the black-box nature and lack of generalization in deep neural networks. It seeks to infuse the resulting model with the appropriate autonomous driving knowledge to represent higher-level concepts by introducing domain knowledge during training.

[5] https://openai.com/blog/adversarial-example-research/

[6] https://www.businessinsider.com/tesla-stops-tunnel-pileup-accidents-driver-says-fsd-enabled-video-2023-1

The work was divided into two stages. Stage one was characterized as an exploratory phase centered around the literature review. The objectives were to:

1. Study how domain knowledge is introduced into deep neural networks;
2. Explore different applications of domain knowledge introduction to various problem domains;
3. Explore ordinal problems and methods as an example of domain knowledge.

Stage two was characterized as a development phase centered around conceiving and evaluating novel scene parsing methods for autonomous driving. The objectives were to:

1. Adapt ordinal segmentation to an autonomous driving context;
2. Propose novel ordinal segmentation methods and metrics, focusing on spatial characteristics;
3. Evaluate the results with appropriate performance baselines based on current state-of-the-art approaches, focusing on the generalization ability of the resulting models, i.e., evaluation through domain shift and dataset scale variation.

1.4 Document Structure

This document is composed of six chapters:

- Chapter 1, Introduction – introduces the dissertation topic, delineating the context, motivation, and objectives behind the work;
- Chapter 2, Background Knowledge – introduces concepts that are essential to the understanding of the work carried out;
- Chapter 3, State of the Art – reviews the current state-of-the-art literature on the introduction of domain knowledge, ordinal problems, and autonomous driving datasets;
- Chapter 4, Introducing Domain Knowledge to Scene Parsing in Autonomous Driving – delineates the proposal of the adaptation of ordinal segmentation to autonomous driving and novel ordinal segmentation methods and metrics;
- Chapter 5, Results – displays and analyses the experimental results, evaluating them with respect to the defined baselines;
- Chapter 6, Conclusions – reviews the work carried out and concludes the document by suggesting ideas that could be explored in the future.

Chapter 2

Background Knowledge

The present chapter approaches deep learning topics essential for the complete understanding of the work carried out. Section 2.1 describes the most common architectures, loss, and metrics for semantic segmentation with deep neural networks. Section 2.2 defines regularization in the context of deep learning.

2.1 Semantic Segmentation with Deep Neural Networks

Semantic segmentation is the task of attributing a semantic label to each of the pixels in an image, resulting in a segmentation map (Figure 2.1).

Figure 2.1: Semantic segmentation applied to autonomous driving [12].

Most state-of-the-art DNN semantic segmentation approaches employ an encoder-decoder architecture. This type of architecture is based on downsampling (done by the encoder) followed by upsampling (done by the decoder). The downsampling step results in an internal representation of the image contents, which is used by the decoder to construct the segmentation map.

2.1.1 Architectures

The following paragraphs describe three of the most widely used encoder-decoder semantic segmentation architectures.

Fully Convolutional Network (FCN) As the name suggests, FCN (Figure 2.2) is an exclusively convolutional DNN. This is achieved by converting a classification CNN, through the replacement of its dense layers with convolutional layers, and then appending a 1×1 convolution for predicting scores for segmentation classes. This allows for variable image input sizes. Variants of FCN add links from lower layers to higher ones, which help the network retain location information and provide more refined predictions. This architecture uses a single-layer decoder. [7]

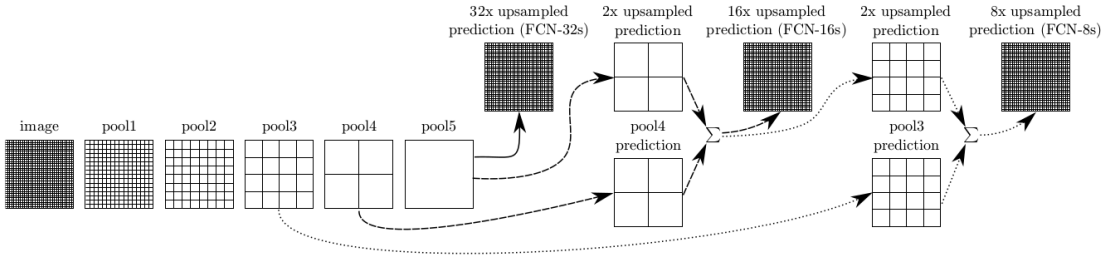


Figure 2.2: FCN architecture [7].

SegNet The SegNet (Figure 2.3) architecture is very similar in concept to the U-Net. However, in SegNet, only the pooling indices are transferred from the encoder to the decoder, and not the entire feature map, which results in less memory usage [9].

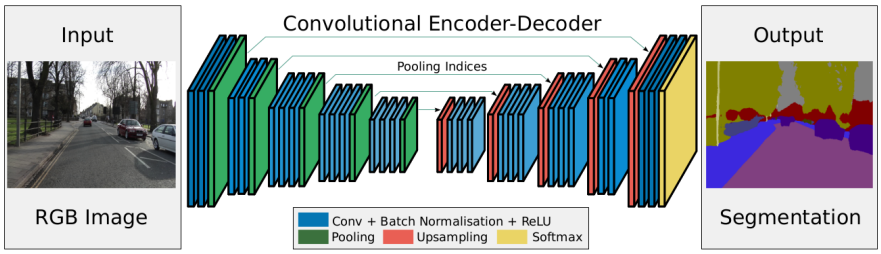


Figure 2.3: SegNet architecture [9].

U-Net The U-Net (Figure 2.4) builds upon FCN, by adding multiple upsampling layers with learnable filters (resulting in a symmetric encoder-decoder), as well as skip connections between analogous steps of the encoder to the decoder in order for location information to be preserved [8, 13]. This was the architecture used in this dissertation work.

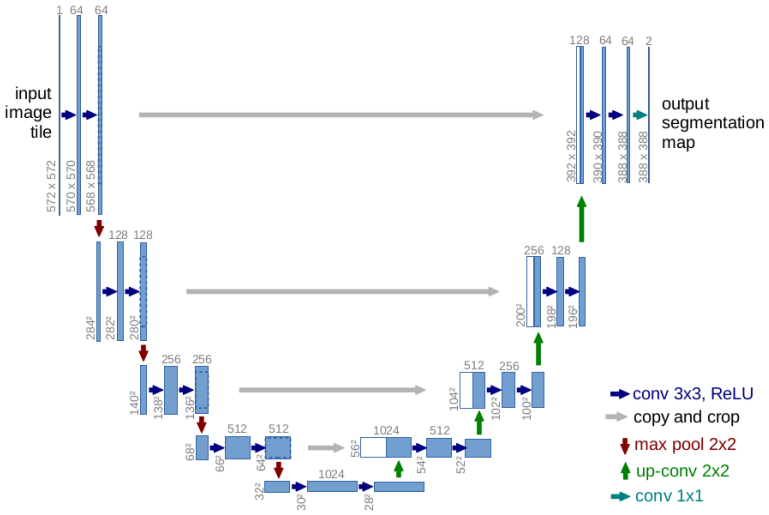


Figure 2.4: U-Net architecture [8].

2.1.2 Loss Function

Cross entropy is one of the most commonly used loss functions for image classification and segmentation problems. This was the main loss function used in this dissertation work. Defining cross entropy for a semantic segmentation problem,

$$\text{CE}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) = -\frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{k=1}^{K} \mathbb{1}(y_{n,i,j} = k) \log(\hat{p}_{n,k,i,j}), \tag{2.1}$$

where $\hat{p}$ is the model output as probabilities, in shape (N,K,H,W), where N is the batch size, K is the number of classes, and (H,W) are, respectively, the height and width of each segmentation mask; y is the ground truth segmentation map, in shape (N,H,W), where each value $y_{n,i,j}$ corresponds to the ground truth class $k \in [1..K]$ of the pixel at position (i,j) of observation n; and $\mathbb{1}(x)$ is the indicator function of x.

It is clear that cross-entropy maximizes the probability of the ground truth class for each pixel in the observation, ignoring the prediction for the other classes. This is a potential area where new loss functions can improve, by restricting the probabilities of the non-ground truth class according to domain knowledge on the task.

2.1.3 Metrics

The Dice coefficient is one of the most popular sample set similarity statistics used for image segmentation,

$$\text{Dice}(A,B) = \frac{2|A \cap B|}{|A| + |B|}, \tag{2.2}$$

where A and B are two sample sets. When using Boolean data, as with image segmentation, the coefficient can be calculated as:

$$\text{Dice} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{2.3}$$

where TP, FP, and FN are the amount of true positive, false positive, and false negative pixels for the given sample and ground truth segmentation masks. Other metrics exist that are variations on Dice, like the Jaccard index, or intersection over union,

$$\text{Jaccard}(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.4}$$

2.2 Regularization for Deep Learning

Regularization can be defined as 'any supplementary technique that aims at making the model generalize better, i.e., produce better results on the test set' [14, p. 1]. Taxonomies of existing regularization methods have been proposed. One such taxonomy splits methods into regularization via [14]:

- **data** – the DNN learns from data, therefore, regularization via data can be employed by applying some transformation to the training data, e.g., feature extraction, pre-processing, data augmentation, etc;
- **network architecture** – 'a network architecture [...] can be selected to have certain properties or match certain assumptions in order to have a regularizing effect' [14, p. 6];
- **error function** – the choice of the loss function can have a regularizing effect, e.g., cross-entropy, mean squared error;
- **regularization term** – by adding a regularization term 'independent of the targets' [14, p. 9] to the loss function, one can 'encode other properties of the desired model, to provide inductive bias (i.e., assumptions about the mapping other than the consistency of outputs with targets)' [14, p. 9]. Equation 2.5 is an example of a loss function with the added regularization term, where L_{targets} could be the cross-entropy loss function, for example. This term can be introduced along with a meta parameter, λ, which controls its influence on the overall loss;

$$L(\boldsymbol{y}, \hat{\boldsymbol{p}}) = L_{\text{targets}}(\boldsymbol{y}, \hat{\boldsymbol{p}}) + \lambda L_{\text{regularization\_term}}(\hat{\boldsymbol{p}}) \tag{2.5}$$

- **optimization** – the optimization process can also be a source of regularization, e.g., with the choice of algorithm (SGD, Adam, etc) or other practices (dropout, weight decay, etc).

Chapter 3

State of the Art

The present chapter reviews the current literature on the topics of the introduction of domain knowledge to deep neural networks and autonomous driving. Section 3.1 defines the concept of the introduction of domain knowledge to deep neural networks. Section 3.2 showcases various examples of how domain knowledge is introduced to deep neural networks. Section 3.3 explores the introduction of domain knowledge through ordinality, in the context of ordinal problems. Section 3.4 discusses state-of-the-art autonomous driving datasets.

3.1 Introduction of Domain Knowledge to Deep Neural Networks

The introduction of domain knowledge to deep neural networks is centered around providing the neural network with knowledge of the problem domain that it would otherwise not be able to infer or would hardly infer from the training process.

Deep neural networks learn from data, and learning just from looking at the dataset may result in incorrect assumptions. This is similar to what would happen if a person were to learn to play a game just by looking at the history of plays and game states, without any other context (i.e., the game rules). Therefore, by introducing domain knowledge, one can guide the DNN in its understanding of the task and help bridge the relationship between the data and its context.

The research on this task centers mostly around how to precisely encode this knowledge and provide it in a way that positively influences the network's results [15, 16]. Several ways of categorizing domain knowledge have been proposed:

- The authors of [15] divide the introduction of domain knowledge to DNNs into three categories, differentiated by where in the network the knowledge is injected, and consequently how it is encoded: 'through changes to the input, the loss function, and the architecture of DNNs' [15, p. 1]. These categories map nicely to three of the categories in the regularization taxonomy analyzed in Section 2.2, respectively, regularization via data, regularization term, and architecture;

- The authors of [16] worked on the representation of domain knowledge for Deep Neural Networks, dividing it into two categories: (**1**) as logical constraints, including propositional logic and first-order logic; and (**2**) as numerical constraints, including the loss function (e.g., the addition of loss terms), constraints on weights (e.g., transfer learning, priors), and regularization (this article does not seem to adopt a broad definition of regularization as the one shown in Section 2.2, referring mostly to loss regularization).

Taking the above information and the knowledge acquired in Section 2.2 into account, the following definition can be produced:

Definition 1. Introduction of Domain Knowledge to Deep Neural Networks is the usage of regularization techniques to include some domain knowledge in the model resulting from the DNN's training.

Many works achieve state-of-the-art results in specific deep learning tasks by applying regularization techniques to the neural network to reflect some domain knowledge of the task at hand. In many cases, that fact is not explicitly acknowledged in the text. We can extend these affirmations to say that every deep learning work incorporates domain knowledge.

Therefore, it is important to establish a distinction between **low-level** and **high-level** domain knowledge:

- **Low-level domain knowledge** – is the kind that is used in every deep learning work, regardless of the specific high-level domain of its application – e.g., the usage of a CNN for an image classification task, the optimization algorithm used, the usage of dropout [17], the activation function (e.g., mapping the value of an artificial neuron to an output between 0 and 1), etc. Considering the regularization taxonomy by [14], this type of domain knowledge would be more characteristic of the error function and optimization regularizations;
- **High-level domain knowledge** – is more intentional and stems from the specific application of the neural network – e.g., the rules of a game [18], the regions of a face most relevant to emotion recognition [19], the ordinal arrangement of classes [20] (e.g., lane markings only make sense inside a road lane), the inherent constraints between two related tasks [21], etc. Considering the regularization taxonomy by [14], this type of domain knowledge would be more characteristic of the data, regularization term, and architecture regularizations – these are also the three domain knowledge introduction categories proposed by [15].

This document more closely explores high-level domain knowledge applications and ideas. However, in this State of the Art chapter, some examples of low-level domain knowledge are also provided.

3.2 Domain Knowledge Introduction Techniques

Considering the works introduced in Section 3.1, and the regularization taxonomy shown in Section 2.2, a categorization of domain knowledge introduction techniques can be provided: (1) augmented loss function; (2) architecture; and (3) input data. Figure 3.1 shows a visualization of the categorization in the context of a neural network.

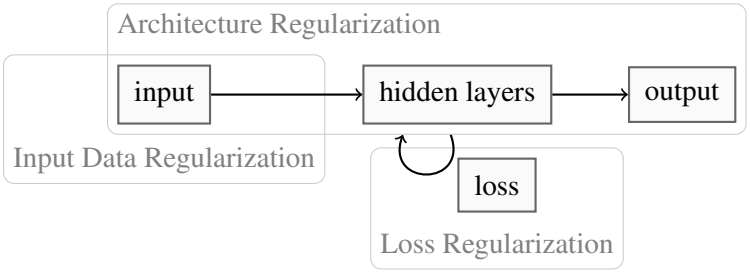


Figure 3.1: Categorization of domain knowledge introduction [14, 15, 16].

The following subsections will describe each of these categories in detail, while providing examples of their application in existing publications or real-world problem domains.

3.2.1 Augmented Loss Function

The loss in DNNs corresponds to the cost of the current record prediction, i.e., how much it strayed from the ground truth, and it is the goal of the learning process to minimize it. For each prediction the loss, after its calculation, is backpropagated through the network, updating its weights and biases.

By augmenting the loss function through the addition of a regularization term, one can control what the network learns – the goal being to not only learn about the correspondence between input and output, $L_{\text{ground\_truth}}$, but also about the rules of why that correspondence exists, i.e., domain knowledge, that can help the network generalize to previously unseen inputs. This is done by defining the regularization term, $L_{\text{domain\_knowledge}}$, as the error between the network's prediction and some piece of domain knowledge. That term may or may not require the ground truth labels and is usually introduced along with a meta parameter, λ, that controls its influence on the loss function. The following is an example of the structure of an augmented loss function, L,

$$L(\boldsymbol{y}, \hat{\boldsymbol{p}}) = L_{\text{ground\_truth}}(\boldsymbol{y}, \hat{\boldsymbol{p}}) + \lambda L_{\text{domain\_knowledge}}(\boldsymbol{y}, \hat{\boldsymbol{p}}) \tag{3.1}$$

One category of problems with an abundance of domain knowledge is rule-based problems. These types of problems have declarative solutions, and there is expert knowledge about their constraints that could be injected into a DNN – helping it learn the rules behind finding suitable

outputs for the given inputs beyond simply learning mappings from inputs to outputs. DNN approaches to solving these types of problems have been explored. One such approach leverages Semantic Based Regularization (SBR) through the addition of a regularizing term to the training loss, which penalizes predictions that violate the problem's constraints [18], resulting in the loss,

$$L(x,y) = L_{\text{cross\_entropy}}(y, f(x)) + \lambda L_{\text{SBR}}(x), \tag{3.2}$$

where x is the input sample, y is the ground truth, f is the network function, and L_{SBR} is the regularization term,

$$L_{\text{SBR}}(x) = (C(x) - f(x))^2, \tag{3.3}$$

where C may encode multiple constraints over the network input.

Constraint problems are an example of problems with a very exact and defined domain knowledge – their constraints. Contrastingly, emotion recognition can benefit from including a more abstract type of domain knowledge, using L1 and total variation regularization loss regularization terms [19]. The resulting loss is as follows:

$$L(\boldsymbol{y}, \hat{\boldsymbol{p}}) = L_{\text{classification}}(\boldsymbol{y}, \hat{\boldsymbol{p}}) + \lambda L_{\text{facial\_parts}}(\hat{\boldsymbol{p}}), \tag{3.4}$$

where $L_{\text{classification}}$ is the classification loss that corresponds to the categorical cross-entropy loss and trains the model to predict the labels, and $L_{\text{facial\_parts}}$ is the domain knowledge injecting regularization term,

$$L_{\text{facial\_parts}}(\hat{\boldsymbol{p}}) = L_{\text{sparsity}}(\hat{\boldsymbol{p}}) + \gamma L_{\text{contiguity}}(\hat{\boldsymbol{p}}), \tag{3.5}$$

where L_{sparsity} is the L1 regularization loss, $L_{\text{contiguity}}$ is the total variation regularization loss, and γ is controls the relative weight of these two terms. These regularization terms were inferred from domain knowledge characteristic from emotion recognition tasks: the sparsity term from 'just small and disjoint facial regions are relevant for the recognition task' [19, p. 6] and the contiguity term from the knowledge that the 'activations of $\hat{x}$ [should] be smooth and spatially localized' [19, p. 6].

3.2.2 Architecture

All DNNs are inherently restricted by their architecture. CNNs, for example, are a specialized architecture often used for DNNs dealing with bi-dimensional visual data, where the domain knowledge dictates that convolutions are operations capable of extracting meaningful information and providing adequate properties. They are therefore restricted by the use of convolutions, their parameters, and various other architectural characteristics.

Parameter Sharing 'Parameter sharing has enabled CNNs to dramatically lower the number of unique model parameters and significantly increase network sizes without requiring a correspond-

ing increase in training data. It remains one of the best examples of how to effectively incorporate domain knowledge into the network architecture.' [22, p. 251]

Multi-task A multi-task architecture allows a DNN to output predictions for more than one task. If the tasks are sufficiently related, something that comes from the domain knowledge of the tasks, then the shared representation learned by the network for both tasks will probably be more generalizable than if a dedicated DNN handled each task. During training, the DNN is forced to abstract higher-level concepts to converge. [23, 24, 25]

Ordinal Output A DNN's architecture also restricts the range of outputs it can provide. For example, the output layer of the DNN can be forced to provide ordinal output [26]. The case of ordinal problem domains will be explored in detail in Section 3.3.1.

3.2.3 Input Data

The data fed to a DNN sets the stage for the rest of the learning process. Therefore, it must be adequately pre-processed and encoded. One such example is the usage of data augmentation, i.e., augmenting the data to cover more possible scenarios (e.g., in image-related tasks: image rotation and flip, brightness variation, warping, etc). Another example, in an ordinal classification domain, is to encode the ordinal target as a cumulative distribution [27] – something that will be explored in detail in Section 3.3.1.

The case of audio analysis illustrates the use of domain knowledge in the pre-processing and data encoding steps. Before being fed to a CNN, stereo audio is commonly converted to mono, and the pulse-code modulated audio data (time domain representation) is encoded as a mel-transformed spectrogram (time-frequency domain representation) [28]. This is part of audio analysis domain knowledge since spectrograms are classically used in audio analysis tasks due to enabling better feature extraction. Despite this, new forms of input data regularization can be developed for every domain. The performance of DNNs for audio scene classification has shown to improve by employing another time-frequency representation, Constant-Q-Transform, motivated by the domain knowledge that 'the human auditory system is approximately "constant" in most of the audible frequency range, and also the fundamental frequencies of the tones in Western music are geometrically spaced along the standard 12-tone scale' [29, p. 2].

3.2.4 Case Study: Domain Knowledge applied to Lane Estimation

The introduction of domain knowledge to DNNs normally employs various techniques without being limited to one. In this section, a case study of domain knowledge applied to lane estimation is reviewed.

The authors of [21] propose a domain knowledge-infused multi-task learning framework that provides a solution to road lane estimation. The proposed model has two direct outputs: (1) lane segmentation and (2) lane boundaries. Additionally, two more indirect outputs are generated: (3) from the (direct) output of lane segmentation, the corresponding (indirect) lane boundaries are calculated, and (4) from the (direct) output of lane boundaries, the corresponding (indirect) lane segmentation is calculated. These indirect outputs are also used to train the DNN, serving as a way to introduce the inherent geometric constraints between the two tasks, something that has been ignored by previous approaches [21]. They encode these constraints by adding two regularising terms to the loss calculation, depicted in Figure 3.2:

1. **Boundary-aware Loss** – the loss of the (indirect) lane boundaries calculated from the (direct) lane segmentation prediction when compared with the lane boundary ground truth;
2. **Area-aware Loss** – the loss of the (indirect) lane segmentation calculated from the (direct) lane boundary prediction when compared with the lane segmentation ground truth.

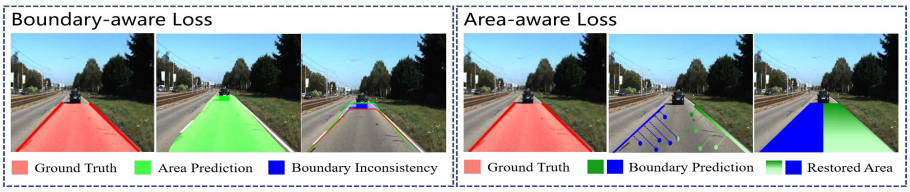


Figure 3.2: Loss for geometrically constrained lane estimation [21].

The authors argue that using multi-task architectures produces better segmentation and that the indirect outputs help make it more consistent. The addition of these loss regularization terms helps the neural network learn the relationship between the lane boundaries and the lane itself (i.e., the lane boundaries are the contour of the lane).

This approach complements the augmented loss function with a custom multi-task architecture, shown in Figure 3.3. As lane segmentation and boundary prediction tasks are strongly related, the model can benefit from a shared representation materialized as a shared encoder component. Then, each of the outputs is assembled by an individual decoder. Furthermore, the individual decoders are connected by a link encoder that allows them to share information.

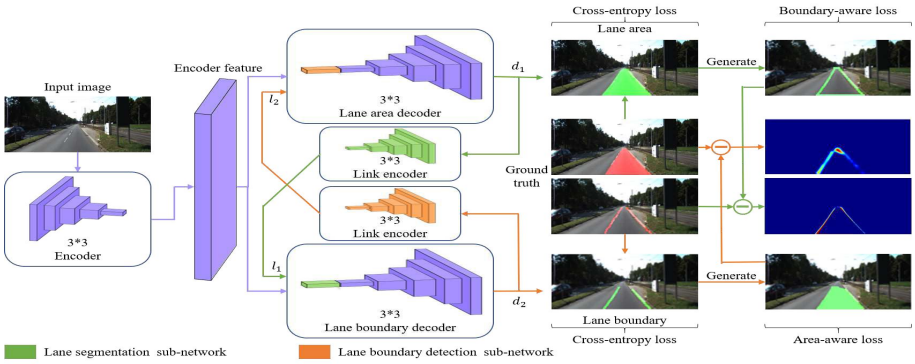


Figure 3.3: Multi-task architecture for geometrically constrained lane estimation [21].

3.3 Ordinal Problems

There is a variety of research works that seek to imbue DNNs with ordinal domain knowledge when it comes to ordinal problem domains. This section reviews the current literature on DNN solutions to two categories of ordinal problems – ordinal classification and ordinal segmentation.

3.3.1 Ordinal Classification

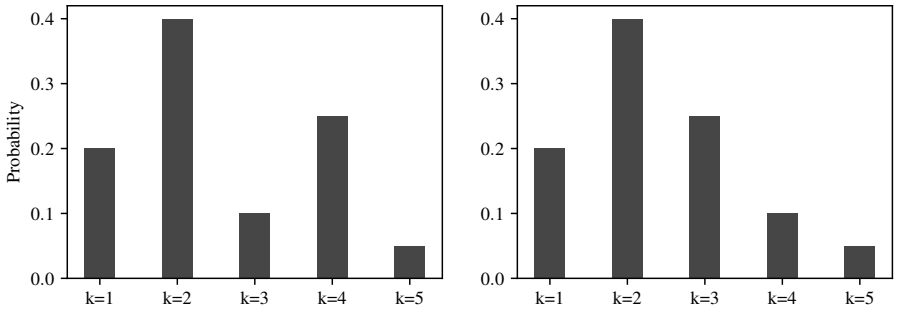
Ordinal classification is the task of classifying an image as one of K ordered classes, where $C_1 \prec C_2 \prec \ldots \prec C_K$, as opposed to nominal classes in the case of classic classification. The following subsections explore the ordinal encoding and unimodality-promoting methods.

Ordinal Encoding An architectural way of introducing ordinality using classical machine learning methods is training $K-1$ binary classifiers, $\{D_2, D_3, \ldots, D_K\}$, where each classifier, D_k, distinguishes between classes $C_{<k}$ and $C_{\geq k}$ [30]. The results from each classifier are then aggregated, resulting in the ensemble's output, i.e., the class prediction.

When using DNNs, this ordinal encoding can be achieved by having multiple outputs in one single neural network instead of multiple classifier networks. Each output corresponds to one of the classifiers and makes the same binary decision. Another way involves regularizing only the input data by encoding the ordinal distribution in the ground truth labels [27]. Defining $k^\star$ as the ground truth class for a given sample, this input data encoding, as opposed to the generic one-hot encoding, which encodes each class as $\mathbb{1}(k = k^\star)$, encodes each class as $\mathbb{1}(k < k^\star)$.

Unimodality The promotion of unimodality in the distribution of the model output probabilities has achieved good results in ordinal classification tasks [26, 31, 32]. This can be advantageous in ordinal problems because the model should be more uncertain between ordinally adjacent classes. Figure 3.4 shows the difference between multimodal and unimodal distributions.

To promote unimodal output probability distributions, various approaches involve architectural restrictions, which restrict the network output and use binomial or Poisson probability distributions in order to convert it to class probabilities [26, 31]. The CO2 augmented loss function



(a) multimodal distribution (b) unimodal distribution

Figure 3.4: Example of possible multimodal and unimodal output probability distributions for a given pixel.

allows unimodality to be achieved without any architectural changes, doing so by penalizing the network for each non-unimodal output inconsistency [32],

$$\mathrm{CO2}(y_n, \hat{\boldsymbol{p}}_n) = \mathrm{CE}(y_n, \hat{\boldsymbol{p}}_n) + \lambda \mathrm{O2}(y_n, \hat{\boldsymbol{y}}_n), \tag{3.6}$$

where O2 is the regularization term,

$$\begin{aligned} \mathrm{O2}(y_n, \hat{\boldsymbol{p}}_n) = &\sum_{k=1}^{K-1} \mathbb{1}(k \geq y_n)\, \mathrm{ReLU}(\delta + \hat{p}_{n,k+1} - \hat{p}_{n,k}) \\ &+ \sum_{k=1}^{K-1} \mathbb{1}(k \leq y_n)\, \mathrm{ReLU}(\delta + \hat{p}_{n,k} - \hat{p}_{n,k+1}), \end{aligned} \tag{3.7}$$

where δ is an imposed margin, assuring that the difference between consecutive probabilities is at least δ, and ReLU is defined as $\mathrm{ReLU}(x) = \max(0, x)$.

3.3.2 Ordinal Segmentation

Ordinal segmentation is the task of segmenting an image such that there is an explicit order between the output classes. For example, in an ordinal segmentation problem with three distinct classes, $C \in [1..3]$, there may be defined an ordering such that $C_1 \supset C_2 \supset C_3$, therefore, an area segmented as C_1 can only possibly have a direct boundary with areas segmented as C_2, whereas C_2 can have boundaries both with C_1 and C_3 (Figure 3.5). Three methodologies that can be used for ordinal segmentation were identified in the literature: ordinal encoding, pixel-wise consistency, and parameter sharing and decision boundary parallelism [20].

Ordinal Encoding The ordinal encoding from ordinal classification can be adapted for segmentation problems by similarly encoding the ground truth masks at a pixel level [20]. Defining $k^\star$ as the ground truth class for a given pixel, each class C_k of the same pixel will be encoded as

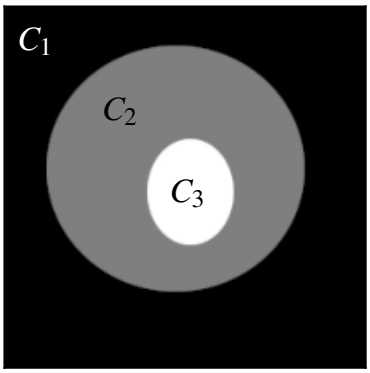
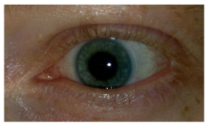


Figure 3.5: Example of a segmentation mask for an ordinal problem with three distinct classes, such that $C \in [1..3]$ and $C_1 \supset C_2 \supset C_3$.

$\mathbb{1}(k < k^\star)$. Figure 3.6 shows an example of the resulting ordinal representation of the ground truth masks for an ordinal segmentation problem.



(a) image

(b) nominal masks

(c) ordinal masks

Figure 3.6: Example of ground-truth masks using the ordinal and nominal representation [20].

Pixel-Wise Consistency Using ordinal encoding does not guarantee that the output probabilities are monotonous, i.e., the probability of ordinal class k, P_k, may be less than P_{k+1}. The consistency of the output class probabilities can be achieved by using,

$$P(C_{k+1}^+) = P(C_{k+1}^+ | C_k^+) P(C_k^+), \tag{3.8}$$

where $P(C_{k+1}^+ | C_k^+)$ is the $(k+1)$-th output of the network and $P(C_k^+)$ is the corrected probability of class k [20].

Decision Boundary Parallelism The previous methods do not hold spatial consistency constraints, i.e., two adjacent pixels can be predicted as non-ordinally adjacent classes. Spatial consistency can be promoted by removing the intersection between the decision hyperplanes, i.e., by considering a model with common slope coefficients and individual bias terms for each of the outputs [20].

3.3.2.1 Datasets

Various biomedical datasets appropriate for ordinal segmentation, i.e., where there is a clear ordering between classes, were identified from the literature [20]. Table 3.1 displays an overview of those datasets that could be obtained, and Figure 3.7 shows, for each of them, a sample image and the corresponding segmentation mask.

Dataset	# Images	# Classes
Breast Aesthetics [33]	120	4
Cervix-MobileODT [34]	1480	5
Mobbio [35]	1817	4
Teeth-ISBI [36]	40	5
Teeth-UCV [37]	100	4

Table 3.1: A selection of appropriate biomedical datasets for ordinal segmentation.

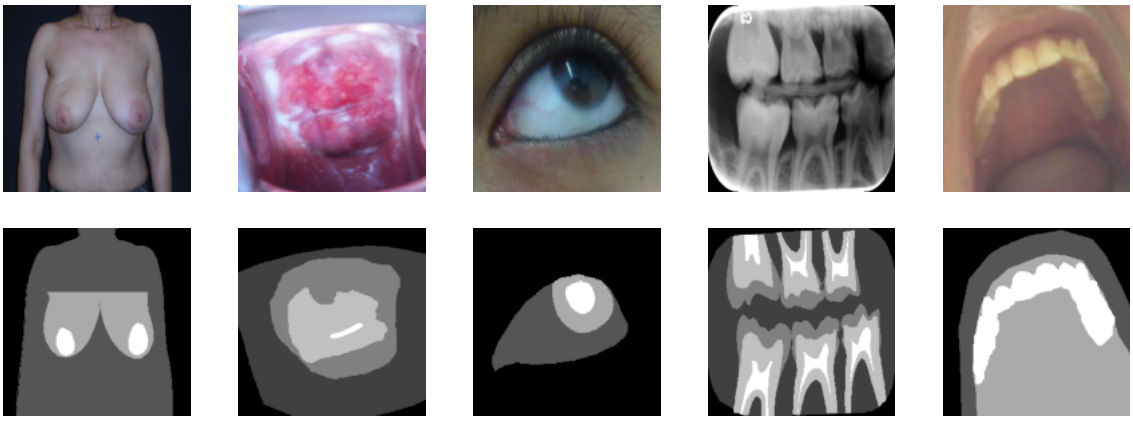


Figure 3.7: Image and mask samples for each of the biomedical datasets in Table 3.1, in the same order.

3.4 Datasets for Autonomous Driving

As autonomous driving is an active area of research, there are multiple datasets with driving scenes that offer semantic segmentation labels, such as: BDD100K [38], Cityscapes [12], KITTI [39],

nuScenes [40], Waymo [41], and others. For the purpose of the work carried out in this dissertation, the BDD100K and Cityscapes datasets will be introduced in greater detail.

BDD100K The BDD100K dataset is a multi-task, large-scale, and diverse dataset, obtained in a crowd-sourcing manner. Its images are split into two sets, each supporting a different subset of tasks: (1) 100K images – 100 000 images with labels for the object detection, drivable area, and lane marking tasks, with a train/validation/test split of 70 000/10 000/20 000 images, and (2) 10K images – 10 000 images with labels for the semantic segmentation, instance segmentation, and panoptic segmentation tasks, with a train/validation/test split of 7 000/1 000/2 000 images. The 10K dataset is not a subset of the 100K, but there is considerable overlap. Figure 3.8 shows an example of an annotated image from the dataset.

Figure 3.8: Annotated driving scene from the BDD100K dataset [38].

Cityscapes The Cityscapes dataset is a large-scale and diverse dataset, with scenes obtained from 50 different cities. It provides 5 000 finely annotated images, with a train/validation/test split of 1 975/500/1 525. Figure 3.9 shows an example of an annotated image from the dataset.

Figure 3.9: Annotated driving scene from the Cityscapes dataset [12].

Chapter 4

Introducing Domain Knowledge to Scene Parsing in Autonomous Driving

The present chapter details the research work that was carried out. Section 4.1 briefly introduces and bridges the research topic from the introduction of domain knowledge to ordinal segmentation. Section 4.2 describes the autonomous driving semantic segmentation problem and how it can be transposed to an ordinal domain. Section 4.3 proposes novel ordinal segmentation metrics and spatial losses, including their generalization to hierarchies with arbitrary ordinal relations.

4.1 Introduction

Current autonomous driving DNN-based semantic segmentation solutions suffer from a lack of generalization ability. This is a critical issue for the safe real-world application of these algorithms, which have the potential to save thousands of lives. Could introducing domain knowledge to DNNs improve their reliability?

Various ideas for domain knowledge that could be introduced were analyzed, resulting in the choice of ordinal segmentation. Through ordinal segmentation, the network can benefit from the following knowledge:

1. The ordinal relation between different objects, e.g., the lane marks and the lane itself – the lane marks are inside the lane;
2. The model should not output high probabilities to pairs of object classes that are not similar, e.g., indecision between car/truck is understandable but not between car/person;
3. The relative placement of objects, e.g., the sky or the sidewalk.

Consequently, the central research question for the work carried out during this dissertation is: could ordinal segmentation improve the generalization ability and reliability of DNNs in an autonomous driving domain?

The next section explores how autonomous driving can be thought of as an ordinal problem.

4.2 Semantic Segmentation in Autonomous Driving as an Ordinal Problem

When analyzing an autonomous driving scene (e.g., Figure 4.1), we can, a priori, derive that, usually:

- The vehicles will be on the road or in parking spaces;
- The drivable area will be on the road;
- The ego lane will be in the drivable area;
- The sidewalk will be on either side of the road;
- The pedestrians will either be on the sidewalk or the road;
- The remainder of the environment surrounds the road.

Figure 4.1: Driving scene from the BDD100K dataset [38].

There will always be a variety of scenarios where these affirmations will either be wrong or not enough to describe what is happening accurately. However, helping the DNN infer this knowledge from the training could improve its reliability.

Let us describe the concrete application of ordinal segmentation to the autonomous driving dataset BDD100K [38]. Table 4.1 contains the classes for the semantic segmentation and drivable area tasks. Each of these tasks uses a different variant of the dataset: semantic segmentation uses the 10K (10 000 images), and drivable area uses the 100K (100 000 images). These two datasets can be intersected and originate a dataset that supports both tasks simultaneously – BDDIntersected – which contains 2 976 annotated images.

Ordinal relations must be derived from the classes in the dataset to obtain an ordinal segmentation task. Tables 4.2, 4.3 and 4.4 introduce, respectively, the *reduced*, *wroadagents* and *wroadagents_nodrivable* ordinal segmentation mask setups, including the ordinal relationship between classes in the form of trees. For mask setups *wroadagents* and *wroadagents_nodrivable* (Tables 4.3 and 4.4), some abstract classes can also be derived, i.e., classes that are the grouping

Index	Class Name
0	road
1	sidewalk
2	building
3	wall
4	fence
5	pole
6	traffic light
7	traffic sign
8	vegetation
9	terrain
10	sky
11	person
12	rider
13	car
14	truck
15	bus
16	train
17	motorcycle
18	bicycle
255	unknown

Index	Class Name
0	direct
1	alternative
2	background

Table 4.1: BDD100K classes for the semantic segmentation (left) and drivable area (right) tasks. On the left, *unknown* means any object that was not annotated. On the right, *direct* means the ego lane, and *alternative* means the remaining lanes. [38]

of objects with similar characteristics and whose masks are the union of their children's. In each table, abstract classes are demarked with an **A** and have their class name in *italic*. Figures 4.2, 4.3 and 4.4 show, respectively, the *reduced*, *wroadagents* and *wroadagents_nodrivable* ordinal segmentation mask setups for the autonomous driving scene in Figure 4.1.

4.3 Ordinal Segmentation

In this section, novel ordinal segmentation methods will be proposed. The proposal is split into pixel-wise and spatial methods, including appropriate evaluation metrics.

Sections 4.3.1 and 4.3.2 consider hierarchies with a single ordinal relation path, i.e., $C_1 \supset C_{...} \supset C_K$, where K is the total number of classes. Arbitrary tree ordinal relations (hierarchies) are discussed in Section 4.3.3.

4.3.1 Pixel-Wise Ordinal Segmentation

Pixel-wise ordinal segmentation methods encompass those methods that act on a pixel level, i.e., they impose restrictions on the pixel taking into account its own characteristics and disregarding

Index and Ordinal Relation	**Class Name**	**Corresponding Classes** (Semantic Segmentation)	**Corresponding Classes** (Drivable Area)
1	unknown	unknown	-
└ 2	environment	every class not directly represented	-
└ 3	road	road	-
└ 4	sidewalk	sidewalk	-
└ 5	road agents	person, rider, motorcycle, bicycle, car, truck, bus, train	-
└ 6	drivable area	-	alternative
└ 7	ego lane	-	direct

Table 4.2: *reduced* ordinal segmentation mask setup for the BDD100K dataset.

Figure 4.2: *reduced* mask for the BDD100K driving scene in Figure 4.1.

the context of the neighboring pixels. Such methods include the ordinal pixel encoding and pixel-wise consistency methods discussed in the state-of-the-art analysis [20].

In an ordinal segmentation problem, for a given pixel, the network should conceptually be more uncertain about the classes that are ordinally closer to the output class – that either precede or succeed it. Therefore, actively promoting the output of unimodal pixel probability distributions, as seen in Section 3.3.1 with ordinal classification, could help the network in the ordinal segmentation task. The following sections propose the adaptation of the unimodal pixel percentage metric and CO2 augmented loss function introduced in Section 3.3.1 from classification to ordinal segmentation. The adaptation is straightforward – treat each pixel in the output map as its own classification problem.

Index and Ordinal Relation	**Class Name**	**Corresponding Classes** (Semantic Segmentation)	**Corresponding Classes** (Drivable Area)
1	unknown	unknown	-
└ 2	environment	every class not directly represented	-
└ 3	road	road	-
└ 4	sidewalk	sidewalk	-
└ **5A**	*road agents*	-	-
└ **6A**	*human*	-	-
└ 7	person	person	-
└ 8	rider	rider	-
└ **9A**	*two wheels*	-	-
└ 10	motorcycle	motorcycle	-
└ 11	bicycle	bicycle	-
└ **12A**	*others*	-	-
└ 13	car	car	-
└ 14	truck	truck	-
└ 15	bus	bus	-
└ 16	train	train	-
└ 17	drivable area	-	alternative
└ 18	ego lane	-	direct

Table 4.3: *wroadagents* ordinal segmentation mask setup for the BDD100K dataset. To refer to the version of this mask without abstract classes, *wroadagents_noabstract* can be used.

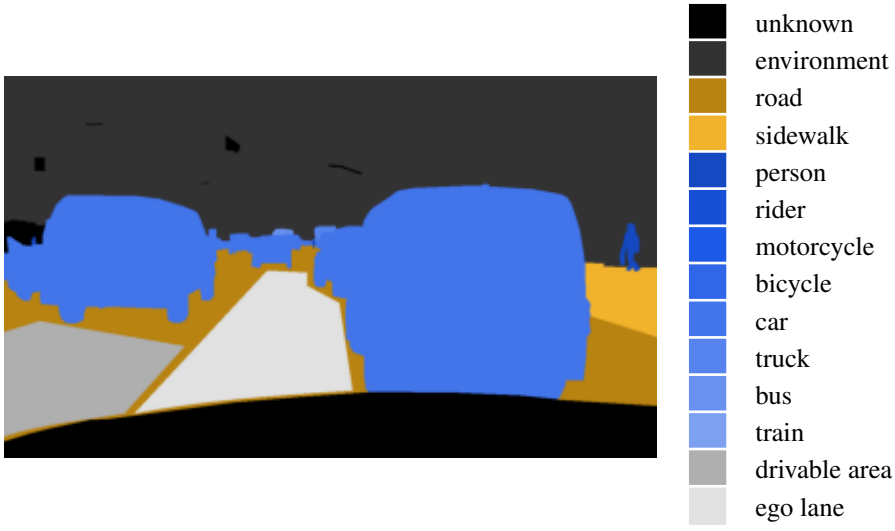


Figure 4.3: *wroadagents* mask for the BDD100K driving scene in Figure 4.1.

Index and Ordinal Relation	**Class Name**	**Corresponding Classes** (Semantic Segmentation)
1	unknown	unknown
└ 2	environment	every class not directly represented
└ 3	road	road
├ 4	sidewalk	sidewalk
└ 5**A**	*road agents*	-
├ 6**A**	*human*	-
├ 7	person	person
└ 8	rider	rider
└ 9**A**	*two wheels*	-
├ 10	motorcycle	motorcycle
└ 11	bicycle	bicycle
└ 12**A**	*others*	-
├ 13	car	car
├ 14	truck	truck
├ 15	bus	bus
└ 16	train	train

Table 4.4: *wroadagents_nodrivable* ordinal segmentation mask setup for the BDD100K dataset. To refer to the version of this mask without abstract classes, *wroadagents_nodrivable_noabstract* can be used.

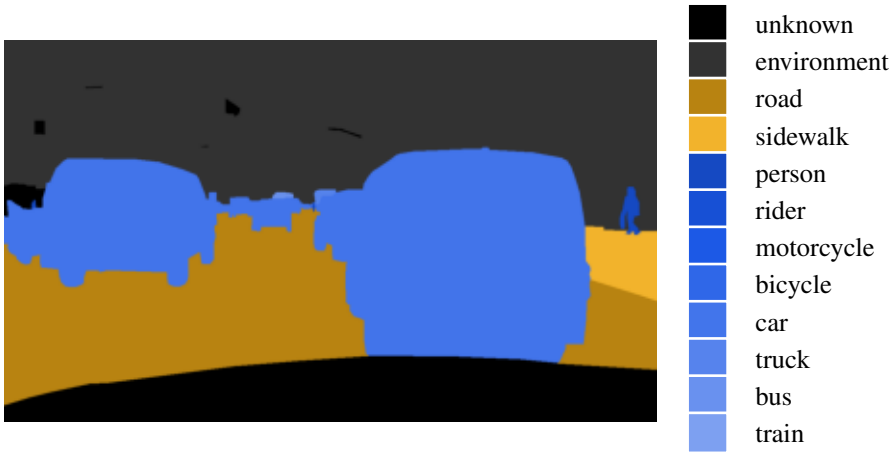


Figure 4.4: *wroadagents_nodrivable* mask for the BDD100K driving scene in Figure 4.1.

4.3.1.1 Percentage of Unimodal Pixels Metric

To evaluate the effect of the unimodality-enforcing method, a straightforward metric is proposed: the percentage of Unimodal Pixels (UP) in the model output,

$$\mathrm{UP}(\hat{\boldsymbol{p}}_n) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}\left(\left[\sum_{k=1}^{K-2} \mathbb{1}(\mathrm{diff}(\hat{\boldsymbol{p}}_{n,:,i,j}, k) - \mathrm{diff}(\hat{\boldsymbol{p}}_{n,:,i,j}, k+1) \neq 0) \right] \leq 1 \right), \tag{4.1}$$

where $\mathrm{diff}(\boldsymbol{P}, k)$ is the sign of the first-order difference between the probability of two consecutive classes,

$$\mathrm{diff}(\boldsymbol{P}, k) = \mathrm{sgn}\,(P_k - P_{k+1}), \tag{4.2}$$

where $\boldsymbol{P}$ is a probability vector with shape K, and k is the class for which to calculate the difference.

In Equation 4.1, the indicator function highlighted in blue is equal to 1 when there is at most a single change in the sign of the first-order difference in the current pixel's class probability distribution and 0 otherwise, i.e., when the current pixel's probability output is unimodal. Taking the example from Figure 3.4, the calculation of the indicator function for pixel (a) results in 0, while the calculation for pixel (b) results in 1. Averaging the sum of the results of that function over every pixel over the total number of pixels results in the UP metric.

As this metric is not itself differentiable, a different loss function must be devised to promote unimodal pixel probability distributions during training.

4.3.1.2 CO2 Augmented Loss Function for Segmentation

The adaptation for segmentation of the CO2 augmented loss function, a state-of-the-art unimodality-promoting loss for ordinal classification, is proposed,

$$\mathrm{CO2}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) = \mathrm{CE}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) + \lambda \mathrm{O2}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n), \tag{4.3}$$

where O2 is the regularization term,

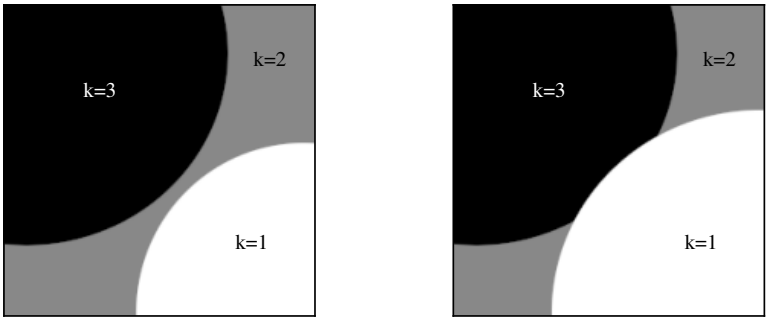
$$\begin{aligned} \mathrm{O2}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \Bigg[& \sum_{k=1}^{K-1} \mathbb{1}(k \geq y_{n,i,j}) \,\mathrm{ReLU}(\delta + \hat{p}_{n,k+1,i,j} - \hat{p}_{n,k,i,j}) \\ & + \sum_{k=1}^{K-1} \mathbb{1}(k \leq y_{n,i,j}) \,\mathrm{ReLU}(\delta + \hat{p}_{n,k,i,j} - \hat{p}_{n,k+1,i,j}) \Bigg], \end{aligned} \tag{4.4}$$

where δ is an imposed margin, assuring that the difference between consecutive probabilities is at least δ [32] and ReLU is defined as $\mathrm{ReLU}(x) = \max(0, x)$. This equation is similar to the one shown for classification in Section 3.3.1.

4.3.2 Spatial Ordinal Segmentation

Spatial ordinal segmentation methods, as opposed to pixel-wise, consider the spatial nature of a segmentation problem, leveraging the context of each given pixel's neighborhood in the applied constraints. Such methods include the decision boundary parallelism method [20] discussed in the state-of-the-art analysis, Section 3.3.2.

As detailed in Section 3.3.2, in an ordinal segmentation problem, an area segmented as C_k can only possibly have a direct boundary with areas segmented as C_{k+1} and C_{k-1}. Therefore, the spatial methods work to minimize the contact surface between the segmented areas of non-ordinally adjacent classes. However, in a 2D projection of the real world, the absolute minimization of these contact surfaces may not be the best solution since occlusions and different perspectives may originate legitimate contact between non-ordinally adjacent classes. A relaxed application of these methods will therefore result in the best outcome. Figure 4.5 shows a hypothetical example of how the spatial ordinal constraints can be broken in the model output.



(a) ground truth mask (b) hypothetical non-constrained model output

Figure 4.5: Hypothetical example of how the contact surface ordinal constraints can be broken in the model output.

The following sections introduce a metric for the contact surface between non-ordinally adjacent classes and three variations of augmented loss functions that seek to minimize it. These losses require an ordinal problem with $K \geq 3$ because no spatial constraints are broken with less than three classes.

4.3.2.1 Contact Surface Metric

The percentage of ordinally invalid inter-class jumps between adjacent pixels was chosen as a metric for the contact surface between the masks of non-ordinally adjacent classes. Ordinally valid jumps are considered to be jumps between classes whose ordinal distance equals 1. If the ordinal distance between the classes of adjacent pixels exceeds 1, then that is an ordinally invalid jump. This requires that each pixel and its immediate neighborhood be examined during calculation.

Defining the Contact Surface (CS) metric,

$$\mathrm{CS}(\hat{\boldsymbol{y}}_n) = \frac{1}{2}\left[\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbb{1}(\mathrm{dx}(\hat{\boldsymbol{y}}_n)_{i,j} > 1)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbb{1}(\mathrm{dx}(\hat{\boldsymbol{y}}_n)_{i,j} > 0)} + \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbb{1}(\mathrm{dy}(\hat{\boldsymbol{y}}_n)_{i,j} > 1)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbb{1}(\mathrm{dy}(\hat{\boldsymbol{y}}_n)_{i,j} > 0)}\right], \tag{4.5}$$

where $\hat{\boldsymbol{y}} = \mathrm{argmax}_{k=1}^{K}(\hat{\boldsymbol{p}})$ with shape (N,H,W), where each value equals the index of the respective pixel's predicted class, $k \in [1..K]$, which corresponds to its ordinal ordering index, and dx and dy are the ordinal index variation, i.e., ordinal distance, from the current pixel (i,j) to the neighborhood, respectively, through the x and y axis,

$$\mathrm{dx}(\hat{\boldsymbol{y}}_n)_{i,j} = |\hat{y}_{n,i,j} - \hat{y}_{n,i,j+1}| \tag{4.6}$$

$$\mathrm{dy}(\hat{\boldsymbol{y}}_n)_{i,j} = |\hat{y}_{n,i,j} - \hat{y}_{n,i+1,j}| \tag{4.7}$$

As this metric is not itself differentiable, a different loss function must be devised to promote spatially consistent segmentation mask outputs during training.

4.3.2.2 Contact Surface Loss Using Neighbor Pixels

One approach to minimize the contact surface metric is penalizing the prediction of two adjacent pixels of non-ordinally adjacent classes. An approach similar to the contact surface metric can be followed. However, instead of the ordinal distance between the classes of adjacent pixels, a differential indicator depicting how wrong the prediction is must be obtained. This indicator can be calculated by multiplying the output probabilities of non-ordinally adjacent classes with an offset of one pixel in each direction – if the network predicts two adjacent pixels, then it will have a high value that must be minimized at that location. Figure 4.6 showcases an example of the calculation of part of the loss between two non-ordinally adjacent classes, $k = 0$ and $k = 3$. Each of the values in the output probability map of $k = 0$ (a) will be multiplied by the values of the immediately adjacent pixels (to the right and bottom) in the output probability map of class $k = 3$ (b). For example, the multiplications of 0.8 by 0 (the probability for $k = 3$ of the pixel to its right) and 0.9 (the probability for $k = 3$ of the pixel to its bottom) will be added, resulting in the local loss value of 0.72, seen in matrix (c).

This principle results in the augmented loss function, $\mathrm{L}_{\mathrm{CSNP}}$,

$$\mathrm{L}_{\mathrm{CSNP}}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) = \mathrm{CE}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) + \lambda \mathrm{CSNP}(\hat{\boldsymbol{p}}_n), \tag{4.8}$$

where λ is a meta parameter that controls the influence of the regularization term, and CSNP is

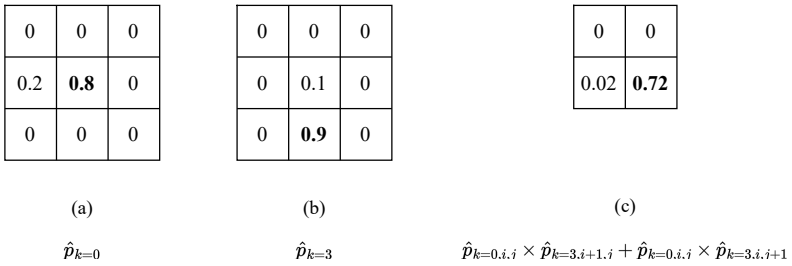


Figure 4.6: Visualization of the calculation of the CSNP loss between two non-ordinally adjacent classes. Matrixes (a) and (b) are the probability maps for classes $k = 0$ and $k = 3$ of a given sample. These two matrixes show the output of two adjacent pixels with high probability, highlighted in **bold**. As this breaks the spatial ordinal constraints, it results in a high value in the resulting loss, depicted in **bold** in matrix (c).

the CSNP (Contact Surface Neighbor Pixels) term,

$$\begin{aligned} \text{CSNP}(\hat{\boldsymbol{p}}_n) = \frac{1}{K^2 - 3K + 2} \sum_{k_1,k_2=1}^{K} \Bigg[\mathbb{1}(|k_2 - k_1| \geq 2) \frac{|k_2 - k_1| - 1}{H \times W} \\ \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{1}{2} \left(\hat{p}_{n,k_1,i,j} \times \hat{p}_{n,k_2,i+1,j} + \hat{p}_{n,k_1,i,j} \times \hat{p}_{n,k_2,i,j+1} \right) \Bigg], \end{aligned} \tag{4.9}$$

where the portion in blue equals the number of pairs $(k_1, k_2) \in [1..K]$ such that $|k_2 - k_1| \geq 2$. Moreover, the portion in red is the ordinal distance between k_2 and k_1, which is used as a weight for the loss calculated between the two classes, such that the loss between more ordinally distant classes has a higher impact in the overall loss. A PyTorch implementation of this loss can be consulted in Appendix A.1.

4.3.2.3 Contact Surface Loss Using the Distance Transform

Another approach is leveraging the distance transform, an image map where each pixel represents that pixel's distance, calculated with the customizable distance function d, to the nearest non-zero pixel in the target image. Defining the distance transform (DT) of the output probability map of class k,

$$\text{DT}(\hat{\boldsymbol{p}}_{n,k})_{p_1} = \min_{p_2 : \hat{p}_{n,k,p_2} \geq \delta} d(p_1, p_2), \tag{4.10}$$

where $p = (i, j)$ and δ is the threshold parameter that selects the high-confidence pixels, allowing the distance transform to be calculated for a high-certainty version of the output segmentation mask.

By calculating the approximated differentiable distance transform [42] of each output class probability map, the approximate distance between each pair of non-ordinally adjacent class masks in the output can be obtained, and the model trained to maximize it. This distance can be obtained by calculating the average of the non-zero values after multiplying the class probabilities map with the opposing class's distance transform. Figure 4.7 illustrates the calculation of part of the loss

between two non-ordinally adjacent classes, $k = 0$ and $k = 3$. For example, in the figure, to obtain the approximate distance between the masks of the two classes, the probability map of class $k = 0$ is multiplied by the distance transform of the output map for $k = 3$. The resulting values will be maximized during training, causing the model to predict the masks of non-ordinally adjacent classes increasingly further apart.

0.1	0	0	0
0	0	0	0
0	0	**0.6**	**0.8**
0	**0.55**	**0.9**	**0.9**

(a)

$\hat{p}_{k=0}$

0.9	**0.8**	0	0
0.8	0	0	0
0	0	0	0
0	0	0	0.1

(b)

$\hat{p}_{k=3}$

0	0	1	2
0	1	2	3
1	2	3	4
2	3	4	5

(c)

$\mathrm{DT}(\hat{p}_{k=3}), \delta = 0.5$

0	0	0	0
0	0	0	0
0	0	1.8	3.2
0	1.65	3.6	4.5

(d)

$\hat{p}_{k=0} \times \mathrm{DT}(\hat{p}_{k=3}), \delta = 0.5$

Figure 4.7: Visualization of the calculation of the CSDT loss between two non-ordinally adjacent classes. Matrixes (a) and (b) are the probability maps for classes $k = 0$ and $k = 3$ of a given sample. These two matrixes show the pixels with high probability, using the threshold $\delta = 0.5$, highlighted in **bold**. Matrix (c) is the distance transform for the thresholded probability map of class $k = 3$ from matrix (b) – each cell corresponds to that cell's Manhattan distance to the closest pixel in the output probability map with a probability greater than 0.5. The approximated distance from the mask of class 0 to the mask of class 3, i.e., the value that the network should maximize, is the average of the non-zero values in matrix (d).

This principle results in the augmented loss function, $\mathrm{L}_{\mathrm{CSDT}}$,

$$\mathrm{L}_{\mathrm{CSDT}}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) = \mathrm{CE}(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n), + \lambda \mathrm{CSDT}(\hat{\boldsymbol{p}}_n), \tag{4.11}$$

where λ is a meta parameter that controls the influence of the regularization term and CSDT is the CSDT (Contact Surface Distance Transform) term,

$$\begin{aligned} \mathrm{CSDT}(\hat{\boldsymbol{p}}_n) = \frac{-1}{\frac{K^2-3K+2}{2}} \sum_{k_1,k_2=1}^{K} \Big[& \mathbb{1}(k_2 - k_1 \geq 2)(|k_2 - k_1| - 1) \\ & \Big(\hat{\boldsymbol{p}}_{n,k_1} \times \mathrm{DT}(\hat{\boldsymbol{p}}_{n,k_2}) + \hat{\boldsymbol{p}}_{n,k_2} \times \mathrm{DT}(\hat{\boldsymbol{p}}_{n,k_1}) \Big) \Big], \end{aligned} \tag{4.12}$$

where the portion in blue is the number of pairs $(k_1, k_2) \in [1..K]$ such that $k_2 - k_1 \geq 2$. Moreover, as with the CSNP loss, the ordinal distance between k_2 and k_1, in red, is used as a weight for the loss calculated between the two classes, such that the loss between more ordinally distant classes has a higher impact in the overall loss.

At this stage, the CSDT term maximizes the distance between the masks of the pairs of non-ordinally adjacent classes indefinitely. This is problematic because this may cause exploding distances, drawing the masks away from each other and possibly completely deviating from the ground truth. This can be solved by limiting the distance transform to a maximum distance, γ. This way, the loss only penalizes masks closer to each other than the γ value. This results in the update distance transform,

$$\mathrm{DT}(\hat{\boldsymbol{p}}_{n,k})^{\gamma}_{p_1} = \min(\mathrm{DT}(\hat{\boldsymbol{p}}_{n,k})_{p_1}, \gamma), \tag{4.13}$$

where γ is the maximum distance at which the contact surface loss term is applied, and in the updated CSDT2 term,

$$\begin{aligned}\mathrm{CSDT2}(\hat{\boldsymbol{p}}) = \frac{-2}{K^2 - 3K + 2} \sum_{k_1,k_2=1}^{K} \Big[& \mathbb{1}(k_2 - k_1 \geq 2)(|k_2 - k_1| - 1) \\ & \left(\hat{\boldsymbol{p}}_{k_1} \times \mathrm{DT}(\hat{\boldsymbol{p}}_{n,k_2})^{\gamma} + \hat{\boldsymbol{p}}_{k_2} \times \mathrm{DT}(\hat{\boldsymbol{p}}_{n,k_1})^{\gamma}\right)\Big]\end{aligned} \tag{4.14}$$

A PyTorch implementation of this loss can be consulted in Appendix A.2.

4.3.2.4 Semi-Supervised Learning

As the proposed spatial losses are unsupervised, i.e., they do not need ground truth labels to be calculated, they can be used with unlabeled data, which allows them to be used for semi-supervised learning. Being able to use unlabeled images is advantageous in the case of image segmentation, where labeled segmentation masks require large amounts of effort to be made. With semi-supervised learning, each training epoch can process a mixture of labeled and unlabeled samples, maximizing the neural network's learning.

4.3.3 Adaptation of Ordinal Segmentation to Arbitrary Hierarchies

Only hierarchies with a single ordinal relation path have been considered for now. The proposed methods must be adapted to apply ordinal segmentation to autonomous driving, a domain with arbitrary ordinal relation paths.

4.3.3.1 Ordinality Trees

Firstly, an arbitrary hierarchy needs to provide the methods with the ordinal relations that motivate the ordinal constraints. These relations can be encoded through trees, as was shown in Section 4.2. Therefore, the ordinality tree of an arbitrary hierarchy is the tree encoding of the ordinal relations in that hierarchy.

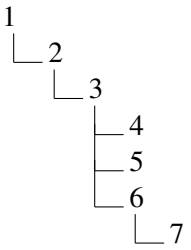


Figure 4.8: Ordinality tree for the *reduced* mask setup of the BDD100K autonomous driving dataset (shown in more detail in Table 4.2).

Figure 4.8 shows the ordinality tree for the *reduced* mask setup of the BDD100K autonomous driving dataset. From this tree, the following ordinal relation paths can be derived:

- $C_1 \supset C_2 \supset C_3 \supset C_4$
- $C_1 \supset C_2 \supset C_3 \supset C_5$
- $C_1 \supset C_2 \supset C_3 \supset C_6 \supset C_7$

In this example, the constraints that should be upheld are:

$$C_1 \supset C_2 \supset C_3 \quad \wedge \quad C_3 \supset C_4 \quad \wedge \quad C_3 \supset C_5 \quad \wedge \quad C_3 \supset C_6 \supset C_7 \tag{4.15}$$

The simplified versions of the proposed ordinal segmentation metrics and losses assumed that each class index corresponds to that class's ordinal index, i.e., its position in the ordinal relation path. In arbitrary hierarchies, however, the class's ordinal index corresponds to its level in the ordinality tree, and there may be multiple classes at the same level. Furthermore, no ordinal constraint can be applied between two classes that are not part of the same ordinal relation path because there is no information about the relation between classes in this situation.

Therefore, the adaptation of the proposed methods focuses on selecting the relevant ordinal relation paths and applied constraints. Let us define $\mathrm{ORP}(\hat{\boldsymbol{p}}_n, k)$ as the set of the ordinal relation paths that include class k, where each path is encoded as an ordered probability vector of each class's corresponding output probability and $\mathrm{OC}(\hat{\boldsymbol{p}}_n, k)$ as the set of the ordinal constraints resulting from the paths including k, each constraint encoded in the same way as each path. Computationally, these functions can be implemented using a tree traversal algorithm, e.g., depth-first search. Returning to the *reduced* mask example,

$$\mathrm{ORP}(\hat{\boldsymbol{p}})_{reduced} = \{[\hat{p}_{k=1}, \hat{p}_{k=2}, \hat{p}_{k=3}, \hat{p}_{k=4}], [\hat{p}_{k=1}, \hat{p}_{k=2}, \hat{p}_{k=3}, \hat{p}_{k=5}], [\hat{p}_{k=1}, \hat{p}_{k=2}, \hat{p}_{k=3}, \hat{p}_{k=6}, \hat{p}_{k=7}]\} \tag{4.16}$$

$$\mathrm{OC}(\hat{\boldsymbol{p}})_{reduced} = \{[\hat{p}_{k=1}, \hat{p}_{k=2}, \hat{p}_{k=3}], [\hat{p}_{k=3}, \hat{p}_{k=4}], [\hat{p}_{k=3}, \hat{p}_{k=5}], [\hat{p}_{k=3}, \hat{p}_{k=6}, \hat{p}_{k=7}]\} \tag{4.17}$$

$$\text{ORP}(\hat{\boldsymbol{p}}, k=5)_{reduced} = \{[\hat{p}_{k=1}, \hat{p}_{k=2}, \hat{p}_{k=3}, \hat{p}_{k=5}]\} \tag{4.18}$$

$$\text{OC}(\hat{\boldsymbol{p}}, k=5)_{reduced} = \{[\hat{p}_{k=1}, \hat{p}_{k=2}, \hat{p}_{k=3}, \hat{p}_{k=5}]\} \tag{4.19}$$

Furthermore, let us define the auxiliary functions:

- $\text{oidx}(k)$ – retrieves the ordinal index of class k. For example, in the tree from Figure 4.8, $\text{oidx}(6) = 4$;
- $\text{shareorp}(k_1, k_2)$ – equals 1 if k_1 and k_2 share at least an ordinal relation path, and 0 otherwise. For example, in the tree from Figure 4.8, $\text{shareorp}(7,5) = 0$ and $\text{shareorp}(2,4) = 1$.

4.3.3.2 Pixel-Wise Ordinal Segmentation

Ordinal Encoding Defining $k^\star$ as the ground truth class for a given pixel, each class C_k of the same pixel will be encoded as $\mathbb{1}(\text{shareorp}(k,k^\star) \wedge \text{oidx}(k) < \text{oidx}(k^\star))$.

Pixel-Wise Consistency In arbitrary hierarchies, the consistency of the output class probabilities can be achieved by using,

$$P(C^+_{k_1}) = \sum_{k_2=1}^{K} \mathbb{1}(\text{shareorp}(k_1,k_2) \wedge \text{oidx}(k_1) - \text{oidx}(k_2) = 1)\, P(C^+_{k_1}|C^+_{k_2}) P(C^+_{k_2}), \tag{4.20}$$

where $P(C^+_{k_1}|C^+_{k_2})$ is the corresponding output of the network and $P(C^+_{k_1})$ is the corrected probability of class k_1.

Percentage of Unimodal Pixels Metric A given pixel's probability distribution is considered unimodal if it is unimodal with respect to every ordinal relation path pertaining to that pixel's output class.

$$\text{UP}'(\hat{\boldsymbol{p}}_n, \hat{\boldsymbol{y}}_n) = \frac{1}{H \times W} \sum_{k=1}^{K} \sum_{i=1}^{H} \sum_{j=1}^{W} \left[\mathbb{1}(k = \hat{y}_{n,i,j}) \prod_{\boldsymbol{P} \in \text{ORP}(\hat{\boldsymbol{p}}_{n,:,i,j}, k)} \mathbb{1}\left(\left[\sum_{n=1}^{|\boldsymbol{P}|-2} \mathbb{1}(\text{diff}(\boldsymbol{P}, n) - \text{diff}(\boldsymbol{P}, n+1) \neq 0) \right] \leq 1 \right) \right] \tag{4.21}$$

CO2 Augmented Loss Function for Segmentation The O2′ term should have its ordinal term calculated for each ordinal constraint to uphold while using the class ordinal index to evaluate the ordinal relation between classes instead of the original class index.

$$\mathrm{O2}'(\boldsymbol{y}_n, \hat{\boldsymbol{p}}_n) = \frac{1}{H \times W} \sum_{k=1}^{K} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(\mathbb{1}(k = y_{n,i,j}) \sum_{\boldsymbol{P} \in OC(\hat{\boldsymbol{p}}_{n,:,i,j}, k)} \left[\sum_{n=1}^{|\boldsymbol{P}|-1} \mathbb{1}(\mathrm{oidx}(P_n) \geq \mathrm{oidx}(k)) \, \mathrm{ReLU}(\delta + P_{n+1} - P_n) \right. \right.$$
$$\left. \left. + \sum_{n=1}^{|\boldsymbol{P}|-1} \mathbb{1}(\mathrm{oidx}(P_n) \leq \mathrm{oidx}(k)) \, \mathrm{ReLU}(\delta + P_n - P_{n+1}) \right] \right) \tag{4.22}$$

4.3.3.3 Spatial Ordinal Segmentation

Contact Surface Metric The contact surface metric is the percentage of ordinally valid jumps between ordinally related classes, i.e., classes that share at least an ordinal relation path. The ordinal distance between classes is calculated using their ordinal index.

$$\mathrm{CS}_{\mathrm{dx}}(\hat{\boldsymbol{y}}_n)'_{i,j} = \mathbb{1}\left(\mathrm{shareorp}(\hat{y}_{n,i,j}, \hat{y}_{n,i,j+1})\right) \left|\mathrm{oidx}(\hat{y}_{n,i,j}) - \mathrm{oidx}(\hat{y}_{n,i,j+1})\right| \tag{4.23}$$

$$\mathrm{CS}_{\mathrm{dy}}(\hat{\boldsymbol{y}}_n)'_{i,j} = \mathbb{1}\left(\mathrm{shareorp}(\hat{y}_{n,i,j}, \hat{y}_{n,i+1,j})\right) \left|\mathrm{oidx}(\hat{y}_{n,i,j}) - \mathrm{oidx}(\hat{y}_{n,i+1,j})\right| \tag{4.24}$$

$$\mathrm{CS}(\hat{\boldsymbol{y}}_n)' = \frac{1}{2} \left[\frac{\sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}(\mathrm{CS}_{\mathrm{dx}}(\hat{\boldsymbol{y}}_n)'_{i,j} > 1)}{\sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}(\mathrm{CS}_{\mathrm{dx}}(\hat{\boldsymbol{y}}_n)'_{i,j} > 0)} + \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}(\mathrm{CS}_{\mathrm{dy}}(\hat{\boldsymbol{y}}_n)'_{i,j} > 1)}{\sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}(\mathrm{CS}_{\mathrm{dy}}(\hat{\boldsymbol{y}}_n)'_{i,j} > 0)} \right] \tag{4.25}$$

Contact Surface Loss Using Neighbor Pixels The CSNP$'$ term is calculated between the pairs of classes that share at least an ordinal relation path and uses each class's ordinal index in order to calculate ordinal distances instead of the original class index.

$$\mathrm{CSNP}'(\hat{\boldsymbol{p}}_n) = \frac{1}{\sum_{\boldsymbol{P} \in \mathrm{OC}(\hat{\boldsymbol{p}}_n)} |\boldsymbol{P}|^2 - 3|\boldsymbol{P}| + 2} \sum_{k_1, k_2 = 1}^{K} \left[\mathbb{1}(|\mathrm{oidx}(k_2) - \mathrm{oidx}(k_1)| \geq 2 \wedge \mathrm{shareorp}(k_1, k_2)) \right.$$
$$\left. \frac{|\mathrm{oidx}(k_2) - \mathrm{oidx}(k_1)| - 1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{1}{2} (\hat{p}_{n,k_1,i,j} \times \hat{p}_{n,k_2,i+1,j} + \hat{p}_{n,k_1,i,j} \times \hat{p}_{n,k_2,i,j+1}) \right] \tag{4.26}$$

Contact Surface Loss Using the Distance Transform Like the CSNP$'$ term, the CSDT$'$ is calculated between the pairs of classes that share at least an ordinal relation path and uses each class's ordinal index in order to calculate ordinal distances instead of the original class index.

$$\mathrm{CSDT2}'(\hat{\boldsymbol{p}}_n) = \frac{-2}{\sum_{\boldsymbol{P}\in \mathrm{OC}(\hat{\boldsymbol{p}}_n)} |\boldsymbol{P}|^2 - 3|\boldsymbol{P}| + 2} \sum_{k_1,k_2=1}^{K} \Big[\mathbb{1}(\mathrm{oidx}(k_2) - \mathrm{oidx}(k_1) \geq 2 \wedge \mathrm{shareorp}(k_1,k_2)) \\ (|\mathrm{oidx}(k_2) - \mathrm{oidx}(k_1)| - 1) \Big(\hat{\boldsymbol{p}}_{n,k_1} \times \mathrm{DT}(\hat{\boldsymbol{p}}_{n,k_2})^{\gamma} + \hat{\boldsymbol{p}}_{n,k_2} \times \mathrm{DT}(\hat{\boldsymbol{p}}_{n,k_1})^{\gamma} \Big) \Big] \tag{4.27}$$

Chapter 5

Results

The present chapter shows and analyzes the results obtained by the proposed methods. Section 5.1 describes the experimental setup used throughout the research work. Section 5.2 shows and analyzes the experimental results for the biomedical datasets. Section 5.3 shows and analyzes the experimental results for the autonomous driving datasets, including the dataset scale variation and semi-supervised learning experiments.

5.1 Experimental Setup

The performance of the proposed methods was validated on five real-life biomedical datasets, described in Section 3.3.2.1, and one autonomous driving dataset, BDD100K [38], with out-of-distribution testing on the Cityscapes [12] dataset, described in Section 3.4. For the BDD100K training, two configurations of the dataset were considered: (1) BDD10K, which was described in Section 3.4, and (2) BDDIntersected, which was described in Section 4.2.

Throughout this chapter, all of the results were obtained using the UNet architecture [8] with four groups of convolution blocks (each consisting of two convolution and one pooling layers) for each of the encoder and decoder portions[1]. All datasets were normalized with a mean of 0 and a standard deviation of 1 after data augmentation, consisting of random rotation, random horizontal flips, random crops, and random brightness and contrast. The networks were optimized for a maximum of 200 epochs, in the case of the biomedical datasets and a maximum of 100 epochs, in the case of the autonomous driving datasets, using the Adam [43] optimizer with a learning rate of 1e-4 and a batch-size of 16. Early stopping was used with a patience of 15 epochs. After a train-test split of 80-20%, a 5-fold training strategy consisting of 4 training folds and one validation fold was applied for each training dataset. For each fold, the best-performing model on the validation dataset was selected. An NVIDIA Tensor Core A100 GPU with 40GB of RAM and an NVIDIA RTX A2000 with 12GB of RAM were used to train the networks.

[1] An open-source PyTorch implementation of the UNet architecture was used, https://github.com/milesial/Pytorch-UNet.

In this chapter, the metrics to be focused on are: (**1**) the Dice coefficient, which evaluates the methods with respect to the ground truth labels; (**2**) the contact surface, which evaluates the methods with respect to the spatial ordinal constraints; and (**3**) the percentage of unimodal pixels metrics, which evaluates the methods with respect to the pixel-wise ordinal constraints. Additional results with different metrics, such as the Jaccard index and the mean absolute error, can be found in Appendix B. The models trained with the cross-entropy loss and with the state-of-the-art ordinal segmentation methods by Fernandes et al. [20], described in Section 3.3.2, were chosen as the performance comparison baselines.

The methods to be evaluated had their parameterization, including the range of regularization term weights (λ), empirically determined. These methods are:

- The semantic segmentation adaptation of the **CO2** augmented loss function, with the imposed margin $\delta = 0.05$, as recommended by the authors, Section 4.3.1.2;
- The proposed ordinal spatial losses, **CSNP**, Section 4.3.2.2, and **CSDT2**, with the distance transform threshold $\delta = 0.5$ and the maximum regularization distance $\gamma = 10$, Section 4.3.2.3;
- The mix of pixel-wise and spatial methods, through the **CO2 + CSNP** loss function, which includes both the CO2 and CSNP regularization terms.

To evaluate the statistical significance of the different results for each method and λ value combination, Welch's t-test was applied. For each dataset, the method with the best average of the results for each fold was selected (it can be the minimum or maximum, depending on the metric). Then, the best average was tested against the results of the other methods on the same dataset, using a significance level of 10% (one-sided). The t-test results can be seen in the tables referenced throughout this chapter, where the values in **bold** represent those values whose difference from the best mean was not statistically significant.

5.2 Experimental Results for the Biomedical Datasets

Tables 5.1, 5.2 and 5.3 display, respectively, the Dice coefficient, contact surface, and percentage of unimodal pixels metrics for the results from each of the models trained with the five biomedical datasets. Figures 5.3, 5.4 and 5.5 show the same results in a comparison plot. Each plot uses dynamic y axis view limits to improve the visibility of the results.

Baselines The baseline ordinal segmentation methods generally do not perform as well as the cross-entropy loss regarding the Dice coefficient, Table 5.1. The pixel-wise consistency method is the closest, as it results, at a maximum, in an absolute 0.7% (0.9% in relative terms) drop in Dice performance and barely surpasses the cross-entropy in the Breast Aesthetics dataset. Regarding ordinal metrics, Tables 5.2 and 5.3, the methods do not seem to impact the contact surface. Still, they result in a higher percentage of unimodal pixel predictions, especially the pixel-wise consistency and decision boundary parallelism methods.

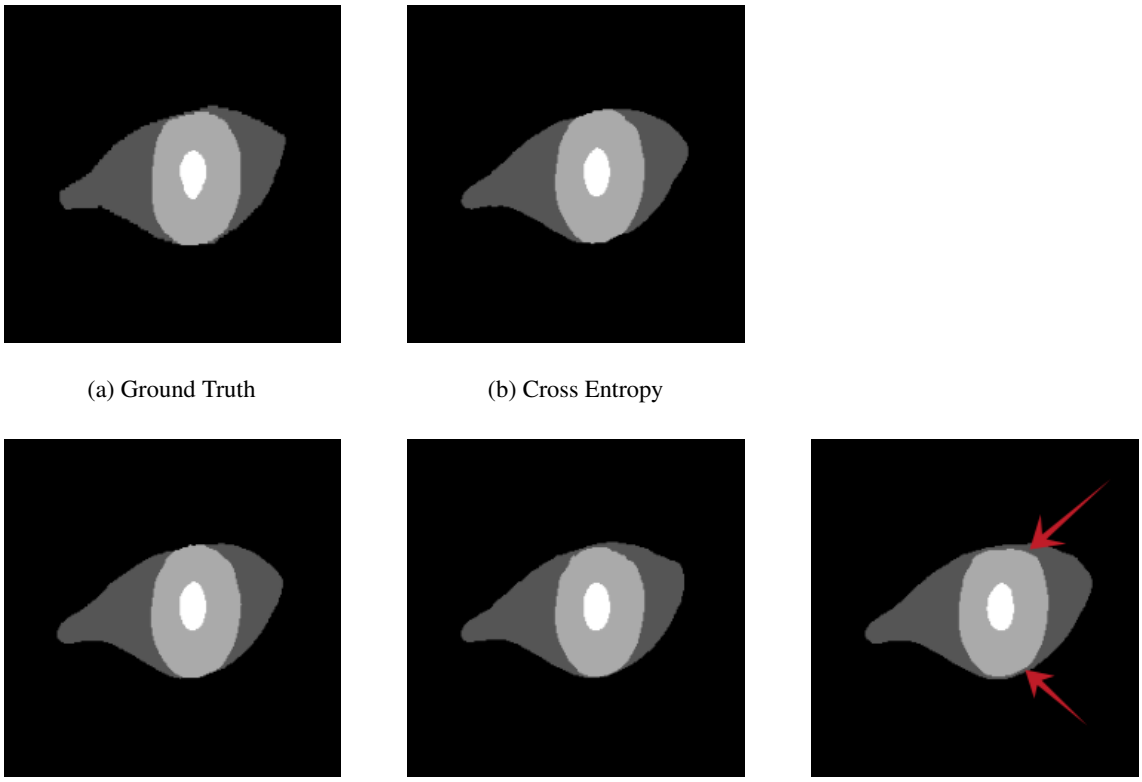
Fernandes et al. report absolute gains in the Dice coefficient, compared with their baseline, of up to 30% [20]. However, this is due to the usage of the sigmoid activation function in their baseline, which is not appropriate for multi-class segmentation problems but was used due to also being used with the ordinal methods. The decision boundary parallelism shows particularly weak results due to expected optimization difficulties in determining the additional bias terms. However, they seem to improve the contact surface metric, Table 5.2, for the cases where they optimize well, such as with the Cervix-MobileODT and Mobbio datasets.

Pixel-Wise Method Regarding the Dice coefficient, Figure 5.3, for every dataset, there are λ values where the CO2 loss either comes very close or surpasses cross-entropy performance, having maximum absolute gains of 0.6% (0.8% in relative terms). As expected, because the CO2 method inherently promotes ordinal pixel probability distributions, CO2 excels in the percentage of unimodal pixels metric, Figure 5.5, nearing 100% for all datasets at high λ values. Due to the specificities of the Cervix-MobileODT and Mobbio datasets, the λ value required to achieve relevant improvements to this metric is much larger than for the others. Therefore, these were the only datasets trained with $\lambda = 1000$ and $\lambda = 10000$. Additionally, this pixel-wise method improves ordinal spatial consistency, as can be seen by the reduction in the contact surface metric for high λ values, Figure 5.4, bringing it to values close to 0%. A highlight is that this method can improve ordinal constraint consistency without too high a penalization in the Dice coefficient metric.

Spatial Methods For the λ values of 0.1 and 1, the CSDT2 loss achieves better Dice coefficient values (with the exception of the Mobbio dataset) while resulting in a lower contact surface when compared with the CSNP loss, Figures 5.3 and 5.4. However, for higher λ values, the CSNP loss achieves a much lower contact surface percentage without much variation in the Dice coefficient. Therefore, the choice between these two losses for spatial ordinal segmentation is application dependent, but the CSNP loss should achieve more stable Dice coefficient results with a higher degree of regularization. Surprisingly, promoting spatially consistent output masks results in a higher percentage of unimodal pixels (with the exception of the Cervix-MobileODT and Mobbio datasets, but in this case, the results may not be indicative since the absolute variation in the metric values is not significative, being less than 1%), Figure 5.5. As the promotion of the pixel-wise constraints also resulted in improvements in the contact surface metric, this suggests that the spatial and pixel-wise constraints are complementary in ordinal problem domains.

Figures 5.1 and 5.2 show, respectively, sample model inference masks for the CSNP and CSDT2, where it is possible to discern the impact of those losses at high λ values. In both cases, it is possible to see that at the highest λ value, both losses manage to include a sclera border between the iris and the background, making it so that there are no infractions of the ordinal spatial constraints, i.e., there is no contact surface between non-ordinally adjacent classes. The CSDT2 loss, however, manages to achieve a thicker border, which is controlled by the γ parameter. This parameter controls the maximum distance between non-ordinally adjacent classes at which the regularization is applied.

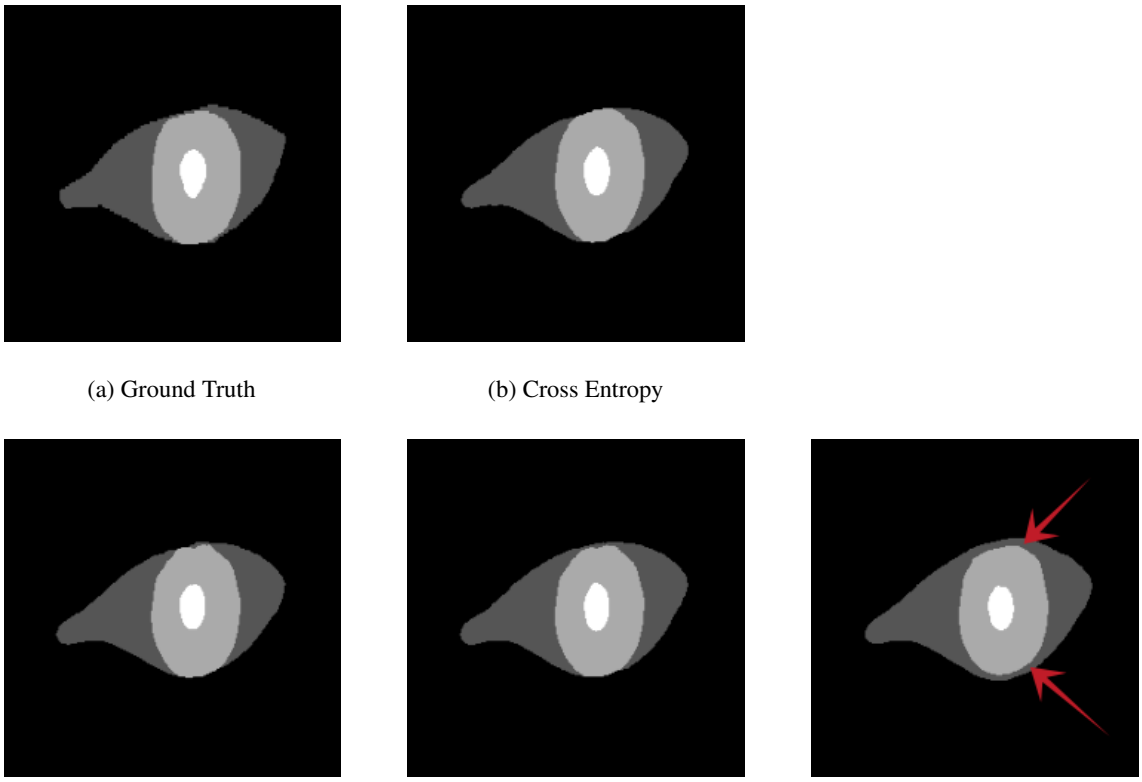
Mixture of Pixel-Wise and Spatial Methods The combination of the CO2 and CSNP losses does not seem to result in a significant difference in performance. Regarding the Dice coefficient, the mixture of the two methods appears more stable with the variation of the λ value. In the ordinal metrics, the junction of the two is a compromise, not being as good as either but offering some balance.



(a) Ground Truth (b) Cross Entropy

(c) CSNP, $\lambda = 1.0$ (d) CSNP, $\lambda = 10.0$ (e) CSNP, $\lambda = 100.0$

Figure 5.1: Sample model inference outputs for the CSNP loss with the Mobbio dataset compared with the ground truth and cross-entropy. Segmentation mask (e) is an example of excessive regularization, where the model includes a sclera border (dark gray) between the background (black) and the iris (light gray), which does not exist in the ground truth.



(a) Ground Truth (b) Cross Entropy

(c) CSDT2, $\lambda = 0.1$ (d) CSDT2, $\lambda = 1.0$ (e) CSDT2, $\lambda = 10.0$

Figure 5.2: Sample model inference outputs for the CSDT2 loss with the Mobbio dataset compared with the ground truth and cross-entropy. Segmentation mask (e) is an example of excessive regularization, where the model includes a sclera border (dark gray) between the background (black) and the iris (light gray), which does not exist in the ground truth.

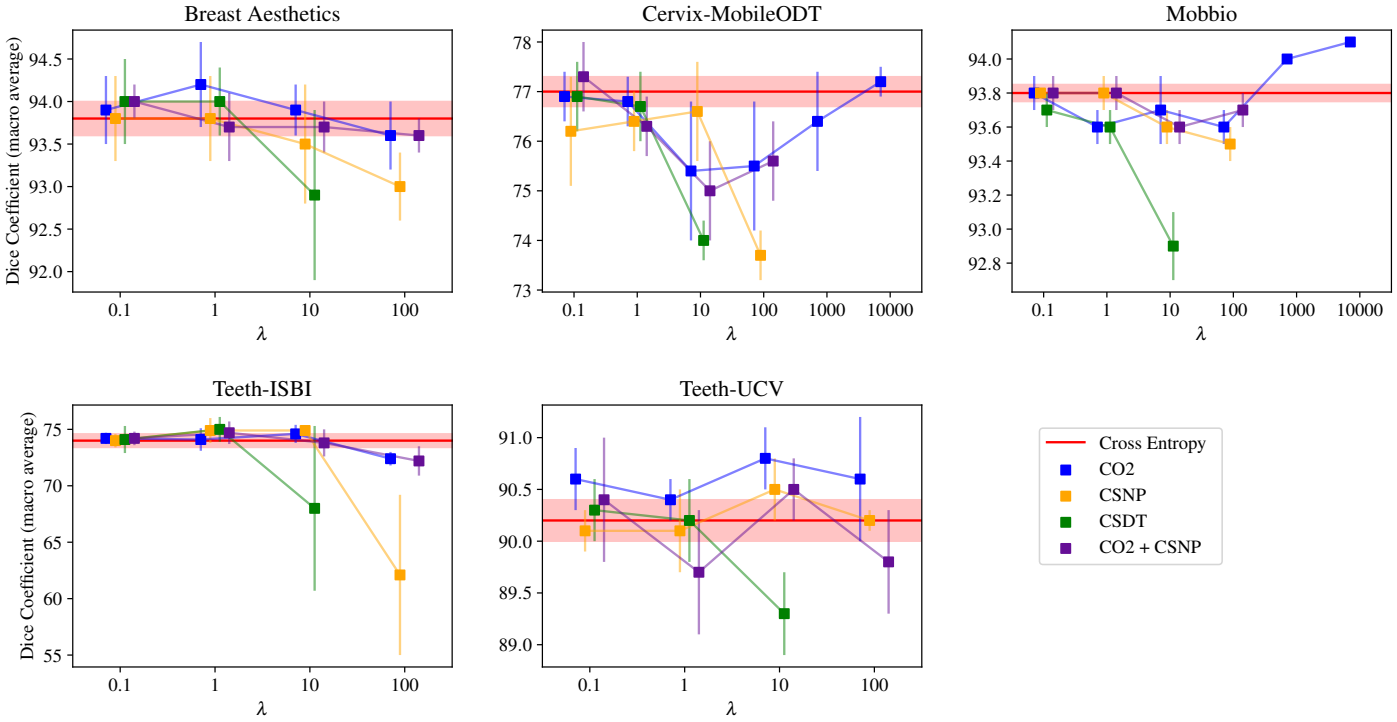


Figure 5.3: Dice coefficient (macro average) results for the biomedical datasets (higher is better).

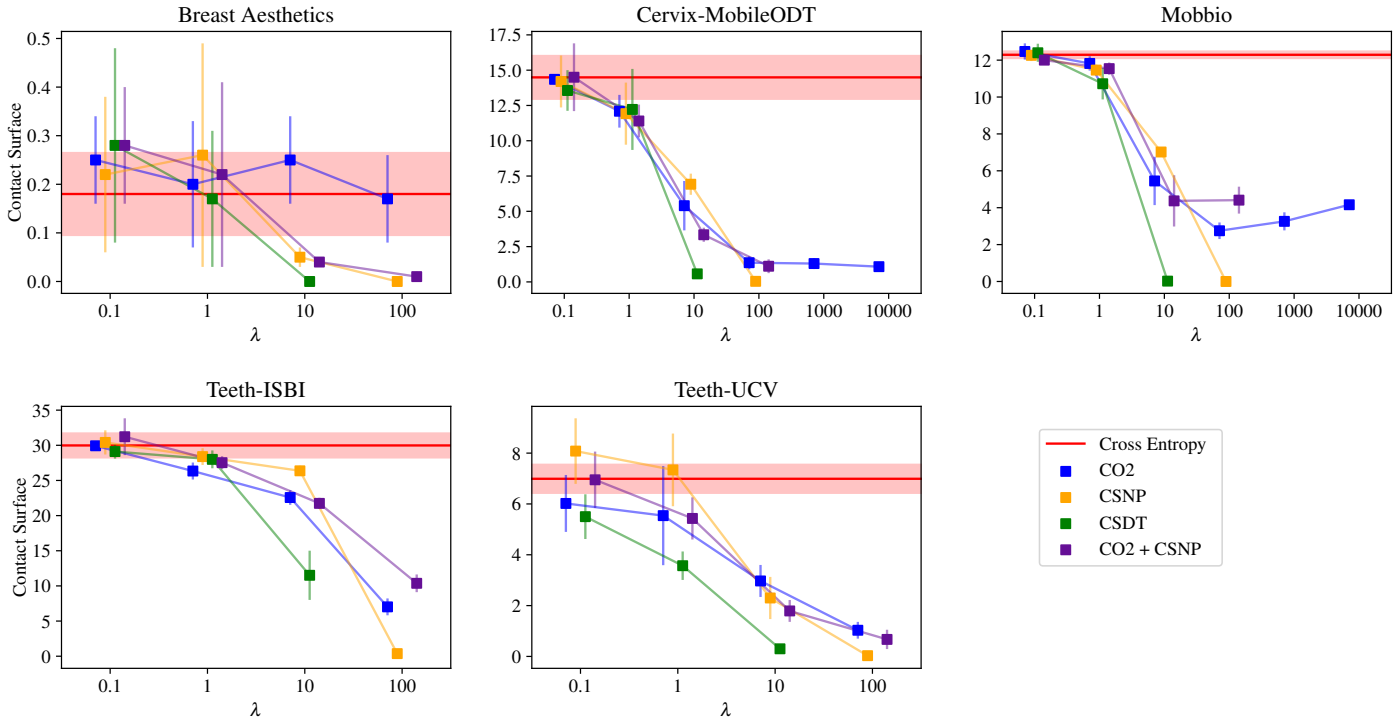


Figure 5.4: Contact surface results for the biomedical datasets (lower is better).

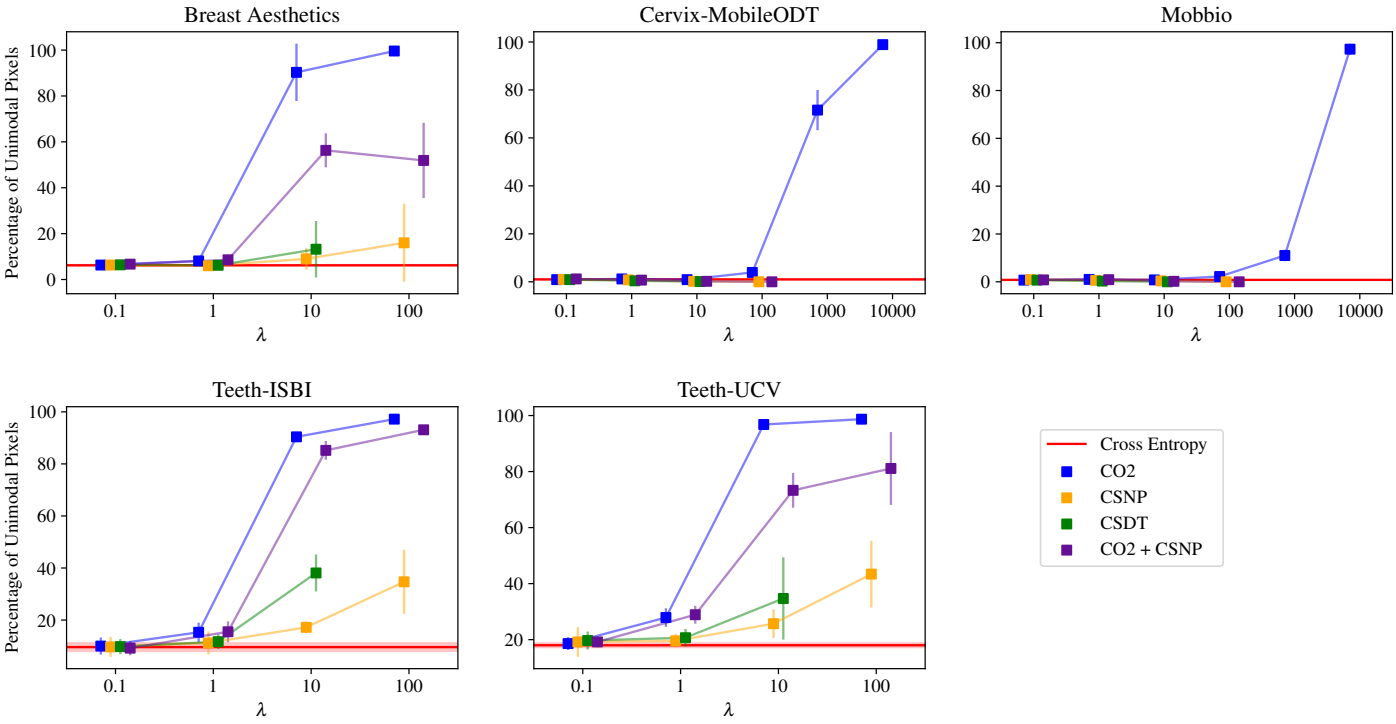


Figure 5.5: Percentage of unimodal pixels results for the biomedical datasets (higher is better).

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
Breast Aesthetics	**93.8 ± 0.4**	82.4 ± 4.3	**93.9 ± 0.5**	17.0 ± 1.1
Cervix-MobileODT	**77.0 ± 0.6**	75.4 ± 0.8	76.3 ± 0.6	61.8 ± 1.2
Mobbio	93.8 ± 0.1	93.5 ± 0.2	93.7 ± 0.1	93.3 ± 0.1
Teeth-ISBI	**74.0 ± 1.2**	14.0 ± 7.4	73.7 ± 0.7	15.6 ± 0.9
Teeth-UCV	90.2 ± 0.4	66.8 ± 5.7	89.7 ± 0.5	32.5 ± 0.1

	CO2					
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$	$\lambda = 1000.0$	$\lambda = 10000.0$
Breast Aesthetics	**93.9 ± 0.4**	**94.2 ± 0.5**	**93.9 ± 0.3**	**93.6 ± 0.4**		
Cervix-MobileODT	**76.9 ± 0.5**	**76.8 ± 0.5**	75.4 ± 1.4	75.5 ± 1.3	**76.4 ± 1.0**	**77.2 ± 0.3**
Mobbio	93.8 ± 0.1	93.6 ± 0.1	93.7 ± 0.2	93.6 ± 0.1	94.0 ± 0.0	**94.1 ± 0.0**
Teeth-ISBI	**74.2 ± 0.5**	**74.1 ± 1.0**	**74.6 ± 0.8**	72.4 ± 0.6		
Teeth-UCV	**90.6 ± 0.3**	90.4 ± 0.2	**90.8 ± 0.3**	**90.6 ± 0.6**		

	CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	**93.8 ± 0.5**	**93.8 ± 0.5**	93.5 ± 0.7	93.0 ± 0.4
Cervix-MobileODT	76.2 ± 1.1	76.4 ± 0.6	**76.6 ± 1.0**	73.7 ± 0.5
Mobbio	93.8 ± 0.0	93.8 ± 0.1	93.6 ± 0.1	93.5 ± 0.1
Teeth-ISBI	**74.0 ± 0.6**	**74.9 ± 1.1**	**74.9 ± 0.4**	62.1 ± 7.1
Teeth-UCV	90.1 ± 0.2	90.1 ± 0.4	**90.5 ± 0.3**	90.2 ± 0.1

	CSDT2		
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$
Breast Aesthetics	**94.0 ± 0.5**	**94.0 ± 0.4**	92.9 ± 1.0
Cervix-MobileODT	**76.9 ± 0.7**	**76.7 ± 0.7**	74.0 ± 0.4
Mobbio	93.7 ± 0.1	93.6 ± 0.1	92.9 ± 0.2
Teeth-ISBI	**74.1 ± 1.2**	**75.0 ± 1.1**	**68.0 ± 7.3**
Teeth-UCV	90.3 ± 0.3	90.2 ± 0.4	89.3 ± 0.4

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	**94.0 ± 0.2**	**93.7 ± 0.4**	**93.7 ± 0.3**	93.6 ± 0.2
Cervix-MobileODT	**77.3 ± 0.7**	76.3 ± 0.6	75.0 ± 1.0	75.6 ± 0.8
Mobbio	93.8 ± 0.1	93.8 ± 0.1	93.6 ± 0.1	93.7 ± 0.1
Teeth-ISBI	**74.2 ± 0.6**	**74.7 ± 1.0**	**73.8 ± 1.2**	72.2 ± 1.3
Teeth-UCV	**90.4 ± 0.6**	89.7 ± 0.6	**90.5 ± 0.3**	89.8 ± 0.5

Table 5.1: Dice coefficient (macro average) results for the biomedical datasets (higher is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
Breast Aesthetics	0.18 ± 0.17	1.78 ± 1.80	0.25 ± 0.14	89.32 ± 12.82
Cervix-MobileODT	14.49 ± 3.08	15.44 ± 1.21	15.77 ± 1.25	7.27 ± 1.05
Mobbio	12.29 ± 0.39	11.95 ± 0.63	12.31 ± 0.44	$\mathbf{0.00 \pm 0.00}$
Teeth-ISBI	29.98 ± 3.50	85.28 ± 9.71	29.86 ± 1.16	100.00 ± 0.00
Teeth-UCV	6.99 ± 1.13	64.29 ± 12.47	9.71 ± 1.14	100.00 ± 0.00

	CO2					
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$	$\lambda = 1000.0$	$\lambda = 10000.0$
Breast Aesthetics	0.25 ± 0.09	0.20 ± 0.13	0.25 ± 0.09	0.17 ± 0.09		
Cervix-MobileODT	14.35 ± 0.35	12.09 ± 1.16	5.40 ± 1.75	1.36 ± 0.47	1.30 ± 0.39	1.08 ± 0.29
Mobbio	12.47 ± 0.44	11.82 ± 0.39	5.45 ± 1.31	2.75 ± 0.45	3.26 ± 0.49	4.16 ± 0.19
Teeth-ISBI	29.93 ± 0.55	26.34 ± 1.21	22.57 ± 1.04	7.02 ± 1.20		
Teeth-UCV	6.02 ± 1.12	5.54 ± 1.95	2.97 ± 0.63	1.03 ± 0.33		

	CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	0.22 ± 0.16	0.26 ± 0.23	0.05 ± 0.02	$\mathbf{0.00 \pm 0.00}$
Cervix-MobileODT	14.20 ± 1.84	11.92 ± 2.20	6.92 ± 0.75	$\mathbf{0.04 \pm 0.01}$
Mobbio	12.26 ± 0.31	11.46 ± 0.30	7.02 ± 0.25	0.00 ± 0.00
Teeth-ISBI	30.41 ± 1.72	28.39 ± 1.19	26.37 ± 0.73	$\mathbf{0.37 \pm 0.40}$
Teeth-UCV	8.08 ± 1.29	7.34 ± 1.43	2.30 ± 0.83	$\mathbf{0.03 \pm 0.03}$

	CSDT2		
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$
Breast Aesthetics	0.28 ± 0.20	0.17 ± 0.14	$\mathbf{0.00 \pm 0.00}$
Cervix-MobileODT	13.56 ± 1.44	12.22 ± 2.87	0.57 ± 0.28
Mobbio	12.40 ± 0.49	10.72 ± 0.85	0.02 ± 0.01
Teeth-ISBI	29.11 ± 1.01	28.00 ± 1.27	11.51 ± 3.50
Teeth-UCV	5.50 ± 0.88	3.57 ± 0.56	0.30 ± 0.13

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	0.28 ± 0.12	0.22 ± 0.19	0.04 ± 0.01	$\mathbf{0.01 \pm 0.01}$
Cervix-MobileODT	14.50 ± 2.40	11.40 ± 1.16	3.34 ± 0.50	1.11 ± 0.49
Mobbio	12.00 ± 0.18	11.54 ± 0.33	4.37 ± 1.39	4.41 ± 0.73
Teeth-ISBI	31.22 ± 2.62	27.53 ± 0.92	21.74 ± 0.80	10.37 ± 1.26
Teeth-UCV	6.95 ± 1.11	5.43 ± 0.83	1.79 ± 0.43	0.67 ± 0.38

Table 5.2: Contact surface results for the biomedical datasets (lower is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
Breast Aesthetics	6.2 ± 0.5	3.7 ± 0.6	38.7 ± 19.2	55.9 ± 1.4
Cervix-MobileODT	1.0 ± 0.2	1.1 ± 0.4	1.1 ± 0.3	18.3 ± 9.0
Mobbio	0.8 ± 0.1	0.7 ± 0.1	0.8 ± 0.1	5.2 ± 5.7
Teeth-ISBI	9.6 ± 3.2	0.0 ± 0.0	7.5 ± 3.6	0.1 ± 0.1
Teeth-UCV	18.0 ± 1.6	8.0 ± 3.3	16.0 ± 4.2	3.5 ± 0.1

	CO2					
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$	$\lambda = 1000.0$	$\lambda = 10000.0$
Breast Aesthetics	6.3 ± 0.5	8.1 ± 2.0	$\mathbf{90.3 \pm 12.5}$	$\mathbf{99.6 \pm 0.2}$		
Cervix-MobileODT	0.9 ± 0.4	1.2 ± 0.6	0.9 ± 0.3	3.9 ± 1.2	71.6 ± 8.4	$\mathbf{98.9 \pm 0.4}$
Mobbio	0.7 ± 0.1	1.0 ± 0.2	0.8 ± 0.1	2.2 ± 0.5	11.0 ± 2.4	$\mathbf{97.3 \pm 2.3}$
Teeth-ISBI	10.0 ± 3.3	15.3 ± 3.7	90.4 ± 0.6	$\mathbf{97.2 \pm 0.2}$		
Teeth-UCV	18.6 ± 2.3	27.9 ± 3.3	96.8 ± 0.9	$\mathbf{98.7 \pm 0.2}$		

	CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	6.3 ± 0.7	6.0 ± 0.6	9.0 ± 4.7	16.0 ± 17.0
Cervix-MobileODT	1.0 ± 0.2	0.8 ± 0.4	0.2 ± 0.1	0.0 ± 0.0
Mobbio	0.9 ± 0.2	0.6 ± 0.1	0.4 ± 0.1	0.0 ± 0.0
Teeth-ISBI	9.7 ± 3.8	11.2 ± 4.4	17.2 ± 2.2	34.7 ± 12.3
Teeth-UCV	19.2 ± 5.3	19.6 ± 2.4	25.7 ± 5.1	43.4 ± 11.9

	CSDT2		
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$
Breast Aesthetics	6.4 ± 0.6	6.2 ± 1.2	13.2 ± 12.3
Cervix-MobileODT	0.9 ± 0.2	0.4 ± 0.1	0.1 ± 0.0
Mobbio	0.7 ± 0.2	0.3 ± 0.1	0.0 ± 0.0
Teeth-ISBI	9.8 ± 2.9	11.7 ± 2.8	38.1 ± 7.1
Teeth-UCV	19.7 ± 3.2	20.7 ± 3.1	34.7 ± 14.7

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	6.7 ± 0.7	8.6 ± 2.0	56.3 ± 7.4	51.9 ± 16.4
Cervix-MobileODT	1.2 ± 0.2	0.7 ± 0.1	0.2 ± 0.1	0.0 ± 0.0
Mobbio	0.8 ± 0.1	0.9 ± 0.0	0.2 ± 0.0	0.0 ± 0.0
Teeth-ISBI	9.2 ± 2.7	15.5 ± 4.0	85.2 ± 3.6	93.1 ± 1.4
Teeth-UCV	19.2 ± 1.9	28.9 ± 3.2	73.3 ± 6.2	81.1 ± 13.0

Table 5.3: Percentage of unimodal pixels results for the biomedical datasets (higher is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

5.3 Experimental Results for the Autonomous Driving Datasets

Tables 5.6, 5.7 and 5.8 display, respectively, the Dice coefficient, contact surface, and percentage of unimodal pixels metrics for the results from each of the models trained with the BDD100K dataset. Figures 5.6, 5.7 and 5.8 show the same results in a comparison plot. Each plot uses dynamic *y* axis view limits to improve the visibility of the results.

Out-of-Distribution Testing (OOD) To evaluate whether the inclusion of the ordinal domain knowledge improved the neural network's generalization ability, the models trained on the BDD100K dataset were subsequently tested on the Cityscapes dataset. This allows an evaluation that is closer to a real-world scenario.

The Cityscapes dataset was chosen because its annotations share similarities with the BDD100K's. The model with which out-of-distribution testing was performed was trained with the *wroadagents_nodrivable* mask setup from Table 4.4. Table 5.4 shows how the corresponding mask setup was achieved with the Cityscapes dataset.

Index and Ordinal Relation	**Class Name**	**Corresponding Classes** (Cityscapes)
1	unknown	unknown
└ 2	environment	every class not directly represented
└ 3	road	road, parking, rail track
├ 4	sidewalk	sidewalk
└ 5**A**	*road agents*	-
├ 6**A**	*human*	-
├ 7	person	person
└ 8	rider	rider
└ 9**A**	*two wheels*	-
├ 10	motorcycle	motorcycle
└ 11	bicycle	bicycle
└ 12**A**	*others*	-
├ 13	car	car
├ 14	truck	truck, caravan, trailer
├ 15	bus	bus
└ 16	train	train

Table 5.4: *wroadagents_nodrivable* ordinal segmentation mask setup for the Cityscapes dataset.

Abstract and Non-Abstract Mask Setups Section 4.2 introduced the *reduced* (Table 4.2), *wroadagents* (Table 4.3), and *wroadagents_nodrivable* (Table 4.4) BDD100K mask setups, of which the last two were introduced with abstract classes. The abstract classes help group subjects with similar characteristics, with the rationale that it may help the network be more undecided between similar classes, e.g., more undecided between a bike and a motorcycle than between

a car and a bike. However, due to the nature of the proposed methods, these masks with abstract classes only can be used with the ordinal segmentation methods proposed by Fernandes et al. [20], described in Section 3.3.2. For usage with the proposed methods, the versions of the *wroadagents* family of masks with no abstract classes will be used – *wroadagents_noabstract* and *wroadagents_nodrivable_noabstract*. Table 5.5 summarizes the mask setups to be evaluated in this section.

	K	**Abstract Classes**	**Datasets**			**Tasks**	
	-	-	BDDIntersected	BDD10K	Cityscapes	Semantic Segmentation	Drivable Area
reduced	7		✓			✓	✓
wroadagents_noabstract	14		✓			✓	✓
wroadagents_nodrivable_noabstract	12			✓	✓	✓	
wroadagents	18	✓	✓			✓	✓
wroadagents_nodrivable	16	✓		✓	✓	✓	

Table 5.5: Summary of the autonomous driving mask setups introduced in Section 4.2. The first three mask setups can be used with any proposed method. The last two can only be used with the semantic segmentation methods proposed by Fernandes et al. [20].

Baselines The baseline ordinal segmentation methods generally do not perform as well as the cross-entropy loss regarding the Dice coefficient, Table 5.6. They get the closest to the cross-entropy with the BDDIntersected dataset and the *reduced* and *wroadagents_noabstract* masks. In the other BDD10K and Cityscapes (OOD) datasets, these ordinal methods achieve, in absolute terms, 1% to 2% lower Dice coefficient results. The baseline ordinal segmentation methods do not meaningfully influence the ordinal metrics, keeping the contact surface results, Table 5.7, at the same levels and slightly improving the percentage of unimodal pixels results, Table 5.8, with the *wroadagents* family of target masks. Using mask setups with abstract masks did not result in meaningful variations in the Dice coefficient results but worsened the ordinal metrics performance.

Pixel-Wise Method The CO2 loss function results in multiple models surpassing the cross-entropy baseline's Dice coefficient when using the *wroadagents* family of masks at multiple λ values, with a maximum absolute gain, when testing with BDD10K, of 1.5% (4% in relative terms), Figure 5.6. With the *reduced* mask, it gets close but does not improve on the results of cross-entropy. The Dice results for CO2 are stable until $\lambda = 10$, which means that the ordinal constraints can be applied without hurting the performance of the resulting models, but have a steep drop at $\lambda = 100$. When testing in an out-of-distribution scenario with Cityscapes, the CO2 loss achieves relevant results, resulting in a maximum absolute gain of 4.2% (11.5% in relative terms) at $\lambda = 10$, which means that using this loss helps the model generalize better to previously unseen scenarios. It is also interesting that at $\lambda = 1$ the Dice results with the BDD10K dataset

are better than at Dice $\lambda = 10$, but with Cityscapes, the opposite happens – the model resulting from $\lambda = 10$ generalizes better but also has more difficulties when seeing images from the dataset it was trained with. Figure 5.9 shows a comparison between the model output of the cross-entropy and CO2 ($\lambda = 10$) losses in out-of-distribution inference. Regarding ordinal constraints, higher λ values result in a lower contact surface between non-ordinally adjacent classes, Figure 5.7 and, as expected, a higher percentage of unimodal pixel probability distributions, Figure 5.8.

Spatial Methods The spatial losses generally slightly underperform when compared with cross-entropy, but some λ values result in better Dice coefficient values, Figure 5.6. The CSNP loss has more stable λ values, while the CSDT2 loss results in steep decreases starting at $\lambda = 100$. Neither of the losses results in significant improvements in an out-of-distribution testing scenario with Cityscapes. As expected, the usage of either loss results in a lower contact surface between non-ordinally adjacent classes, Figure 5.7, and a higher percentage of unimodal pixel probability distributions, Figure 5.8, at higher λ values.

Mixture of Pixel-Wise and Spatial Methods The combination of the CO2 and CSNP losses does not seem to result in a significant difference in performance. In the ordinal metrics, joining the two results in better adherence to the ordinal constraints – generally, the contact surface values are lower, and the percentage of unimodal pixels values are higher than either of the separate losses, with exceptions at some λ values.

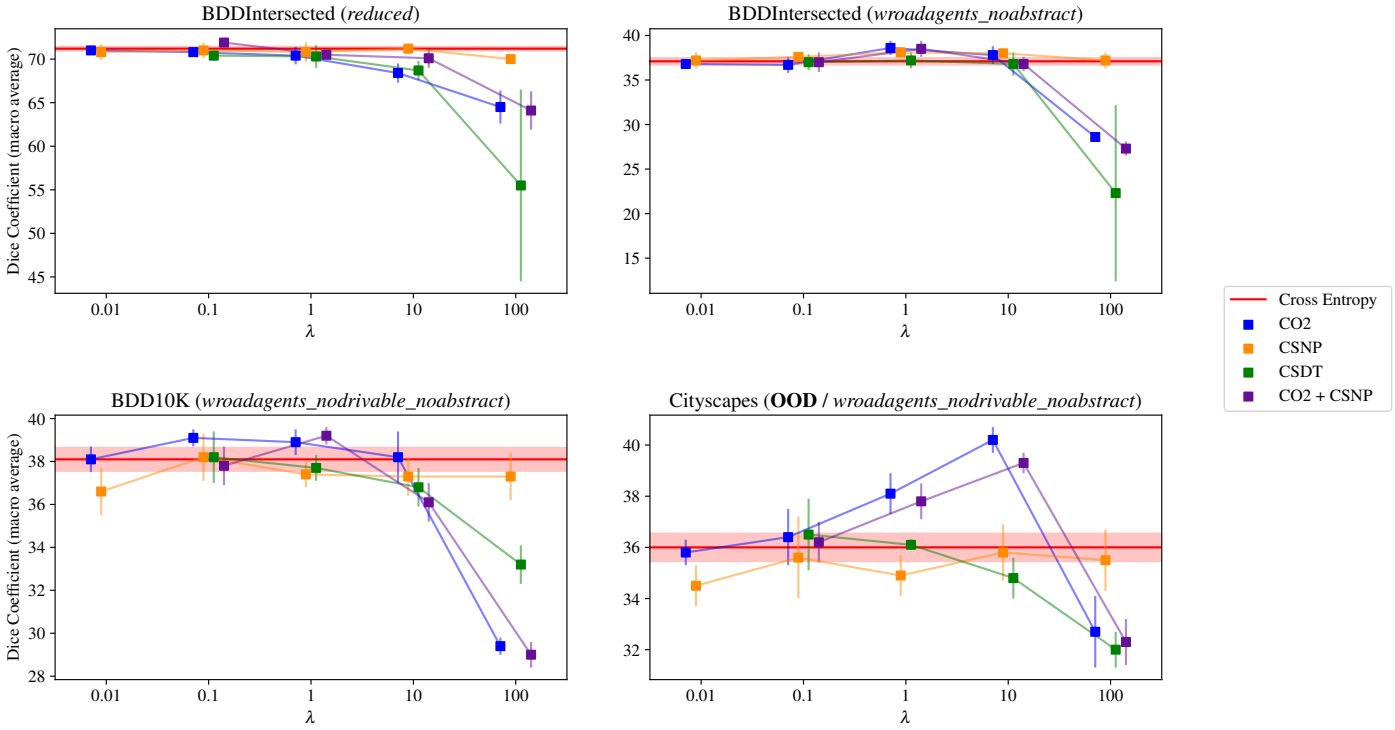


Figure 5.6: Dice coefficient (macro average) results for the autonomous driving datasets (higher is better).

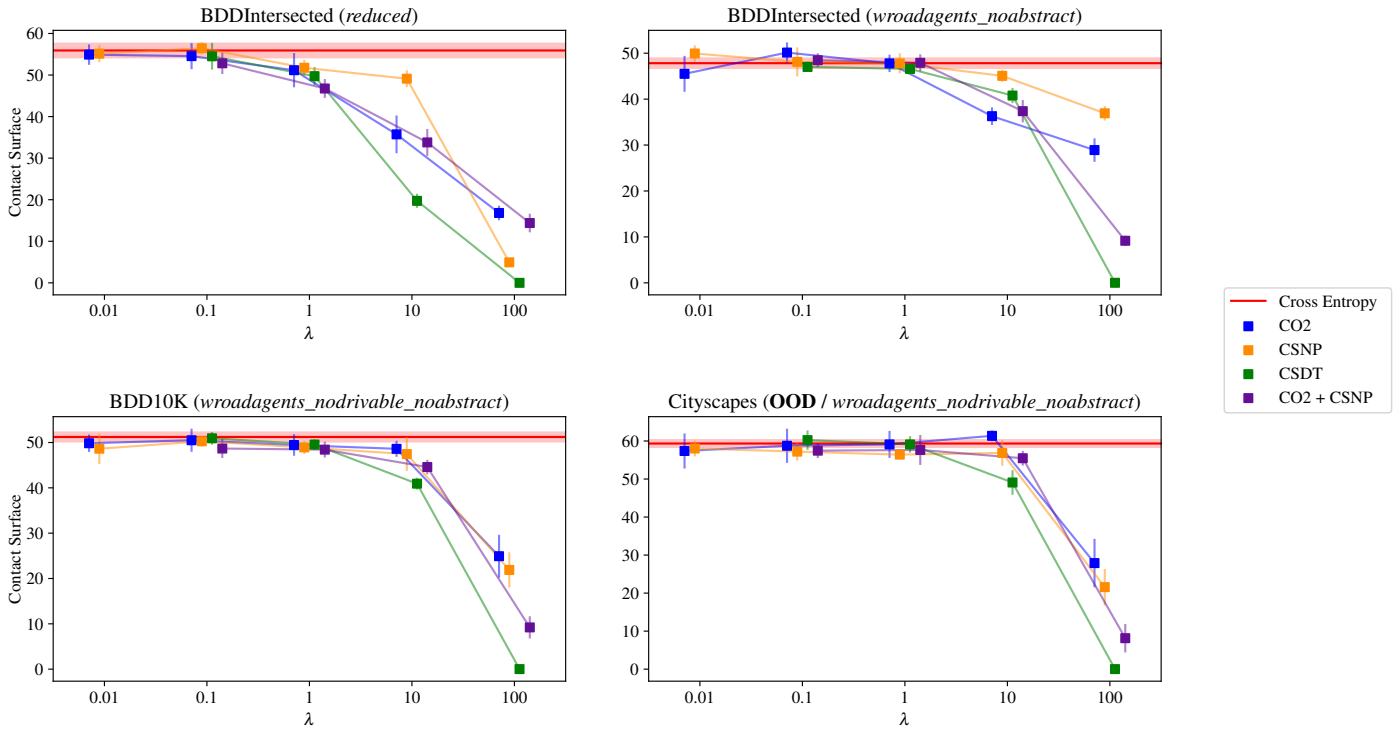


Figure 5.7: Contact surface results for the autonomous driving datasets (lower is better).

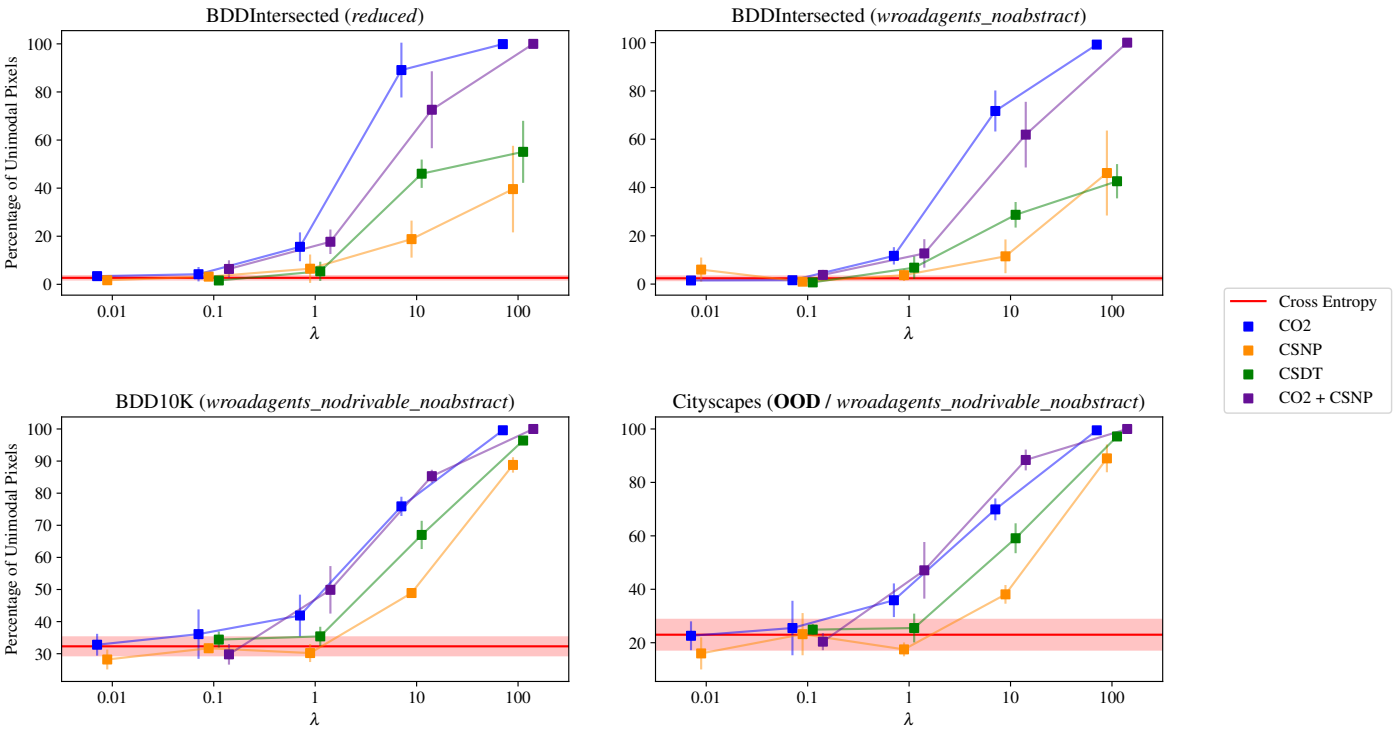


Figure 5.8: Percentage of unimodal pixels results for the autonomous driving datasets (higher is better).

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
BDDIntersected (*reduced*)	71.2 ± 0.5	$\mathbf{71.2 \pm 1.0}$	$\mathbf{71.5 \pm 0.5}$	$\mathbf{71.6 \pm 0.4}$
BDDIntersected (*wroadagents_noabstract*)	37.1 ± 0.8	36.2 ± 0.6	37.1 ± 0.9	36.6 ± 1.1
BDD10K (*wroadagents_nodrivable_noabstract*)	38.1 ± 1.1	36.2 ± 0.4	35.3 ± 0.8	36.9 ± 0.2
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	36.0 ± 1.1	34.6 ± 0.3	34.2 ± 1.0	35.1 ± 0.4
BDDIntersected (*wroadagents*)		$\mathbf{36.5 \pm 0.6}$	35.7 ± 0.4	35.7 ± 0.6
BDD10K (*wroadagents_nodrivable*)		35.9 ± 0.4	$\mathbf{36.7 \pm 1.2}$	$\mathbf{37.1 \pm 0.6}$
Cityscapes (**OOD** / *wroadagents_nodrivable*)		$\mathbf{34.5 \pm 0.3}$	$\mathbf{34.5 \pm 0.8}$	$\mathbf{34.7 \pm 0.7}$

	CO2				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	71.0 ± 0.3	70.8 ± 0.5	70.4 ± 1.0	68.4 ± 1.1	64.5 ± 1.9
BDDIntersected (*wroadagents_noabstract*)	36.8 ± 0.5	36.7 ± 0.9	$\mathbf{38.6 \pm 0.8}$	$\mathbf{37.8 \pm 1.0}$	28.6 ± 0.3
BDD10K (*wroadagents_nodrivable_noabstract*)	38.1 ± 0.6	$\mathbf{39.1 \pm 0.4}$	$\mathbf{38.9 \pm 0.6}$	$\mathbf{38.2 \pm 1.2}$	29.4 ± 0.4
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	35.8 ± 0.5	36.4 ± 1.1	38.1 ± 0.8	$\mathbf{40.2 \pm 0.5}$	32.7 ± 1.4

	CSNP				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	70.8 ± 0.9	71.0 ± 0.9	70.8 ± 1.1	71.2 ± 0.6	70.0 ± 0.4
BDDIntersected (*wroadagents_noabstract*)	37.2 ± 0.9	37.6 ± 0.4	$\mathbf{38.1 \pm 0.5}$	$\mathbf{38.0 \pm 0.4}$	37.2 ± 0.9
BDD10K (*wroadagents_nodrivable_noabstract*)	36.6 ± 1.1	$\mathbf{38.2 \pm 1.1}$	37.4 ± 0.6	37.3 ± 0.9	37.3 ± 1.1
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	34.5 ± 0.8	35.6 ± 1.6	34.9 ± 0.8	35.8 ± 1.1	35.5 ± 1.2

	CSDT2			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	70.4 ± 0.6	70.3 ± 1.3	68.7 ± 1.1	55.5 ± 11.0
BDDIntersected (*wroadagents_noabstract*)	37.0 ± 0.9	37.2 ± 0.9	36.8 ± 1.3	22.3 ± 9.9
BDD10K (*wroadagents_nodrivable_noabstract*)	$\mathbf{38.2 \pm 1.2}$	37.7 ± 0.6	36.8 ± 0.9	33.2 ± 0.9
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	36.5 ± 1.4	36.1 ± 0.2	34.8 ± 0.8	32.0 ± 0.7

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	$\mathbf{71.9 \pm 0.2}$	70.5 ± 0.5	70.1 ± 1.1	64.1 ± 2.2
BDDIntersected (*wroadagents_noabstract*)	37.0 ± 1.1	$\mathbf{38.5 \pm 0.9}$	36.8 ± 0.8	27.3 ± 0.8
BDD10K (*wroadagents_nodrivable_noabstract*)	37.8 ± 0.9	$\mathbf{39.2 \pm 0.4}$	36.1 ± 0.9	29.0 ± 0.6
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	36.2 ± 0.8	37.8 ± 0.7	39.3 ± 0.4	32.3 ± 0.9

Table 5.6: Dice coefficient (macro average) results for the autonomous driving datasets (higher is better). The smaller-sized results in the first table fragment are for the mask setups with abstract classes, which are only able to be used with the ordinal segmentation methods by Fernandes et al. [20]. The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
BDDIntersected (*reduced*)	55.91 ± 3.46	51.55 ± 3.61	54.49 ± 0.72	50.61 ± 2.12
BDDIntersected (*wroadagents_noabstract*)	47.83 ± 2.22	46.05 ± 2.32	45.81 ± 2.03	46.83 ± 1.39
BDD10K (*wroadagents_nodrivable_noabstract*)	51.22 ± 2.13	47.97 ± 2.41	49.77 ± 1.90	48.18 ± 1.58
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	59.34 ± 1.96	56.77 ± 2.10	57.81 ± 2.40	58.02 ± 2.38
BDDIntersected (*wroadagents*)		**53.71 ± 3.60**	**55.97 ± 3.56**	**54.72 ± 2.77**
BDD10K (*wroadagents_nodrivable*)		**58.18 ± 2.04**	**60.54 ± 2.60**	62.70 ± 1.51
Cityscapes (**OOD** / *wroadagents_nodrivable*)		**66.39 ± 1.55**	**65.95 ± 4.17**	70.79 ± 2.12

	CO2				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	54.91 ± 2.45	54.55 ± 3.14	51.16 ± 4.11	35.73 ± 4.54	16.81 ± 1.75
BDDIntersected (*wroadagents_noabstract*)	45.51 ± 3.91	50.14 ± 2.22	47.79 ± 1.90	36.29 ± 1.92	28.91 ± 2.55
BDD10K (*wroadagents_nodrivable_noabstract*)	49.84 ± 1.92	50.50 ± 2.55	49.45 ± 2.37	48.61 ± 1.78	24.92 ± 4.72
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	57.38 ± 4.62	58.75 ± 4.49	59.11 ± 3.55	61.36 ± 1.40	27.89 ± 6.37

	CSNP				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	55.14 ± 2.05	56.45 ± 1.41	51.75 ± 1.90	49.12 ± 2.03	4.93 ± 0.90
BDDIntersected (*wroadagents_noabstract*)	49.94 ± 1.77	48.11 ± 3.16	47.78 ± 2.18	45.07 ± 1.29	36.92 ± 1.59
BDD10K (*wroadagents_nodrivable_noabstract*)	48.63 ± 3.37	50.27 ± 1.16	48.92 ± 1.46	47.45 ± 3.72	21.89 ± 3.90
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	58.03 ± 2.08	57.21 ± 2.33	56.42 ± 1.23	56.87 ± 3.40	21.57 ± 4.79

	CSDT2			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	54.51 ± 3.22	49.73 ± 2.19	19.73 ± 1.70	**0.00 ± 0.00**
BDDIntersected (*wroadagents_noabstract*)	47.01 ± 1.10	46.58 ± 1.10	40.78 ± 1.66	**0.01 ± 0.01**
BDD10K (*wroadagents_nodrivable_noabstract*)	50.93 ± 1.41	49.55 ± 1.16	40.92 ± 1.30	**0.01 ± 0.01**
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	60.23 ± 2.55	59.11 ± 2.16	49.09 ± 3.23	**0.01 ± 0.01**

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	52.82 ± 2.59	46.76 ± 2.27	33.79 ± 3.24	14.39 ± 2.24
BDDIntersected (*wroadagents_noabstract*)	48.47 ± 1.51	47.91 ± 1.86	37.38 ± 2.43	9.16 ± 1.11
BDD10K (*wroadagents_nodrivable_noabstract*)	48.67 ± 2.09	48.43 ± 1.74	44.59 ± 1.59	9.21 ± 2.45
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	57.46 ± 1.96	57.65 ± 3.94	55.47 ± 1.94	8.14 ± 3.76

Table 5.7: Contact surface results for the autonomous driving datasets (lower is better). The smaller-sized results in the first table fragment are for the mask setups with abstract classes, which are only able to be used with the ordinal segmentation methods by Fernandes et al. [20]. The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
BDDIntersected (*reduced*)	2.7 ± 1.7	0.5 ± 0.2	1.4 ± 0.4	1.2 ± 0.4
BDDIntersected (*wroadagents_noabstract*)	2.4 ± 1.8	0.4 ± 0.2	10.2 ± 5.2	2.5 ± 1.1
BDD10K (*wroadagents_nodrivable_noabstract*)	32.3 ± 5.8	33.0 ± 6.3	35.7 ± 4.6	28.7 ± 3.4
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	23.0 ± 11.4	27.2 ± 6.9	28.7 ± 7.8	20.2 ± 5.0
BDDIntersected (*wroadagents*)		0.1 ± 0.1	$\mathbf{0.3 \pm 0.1}$	0.2 ± 0.1
BDD10K (*wroadagents_nodrivable*)		0.0 ± 0.0	$\mathbf{0.2 \pm 0.1}$	$\mathbf{0.1 \pm 0.1}$
Cityscapes (**OOD** / *wroadagents_nodrivable*)		0.1 ± 0.0	$\mathbf{0.6 \pm 0.2}$	$\mathbf{0.5 \pm 0.3}$

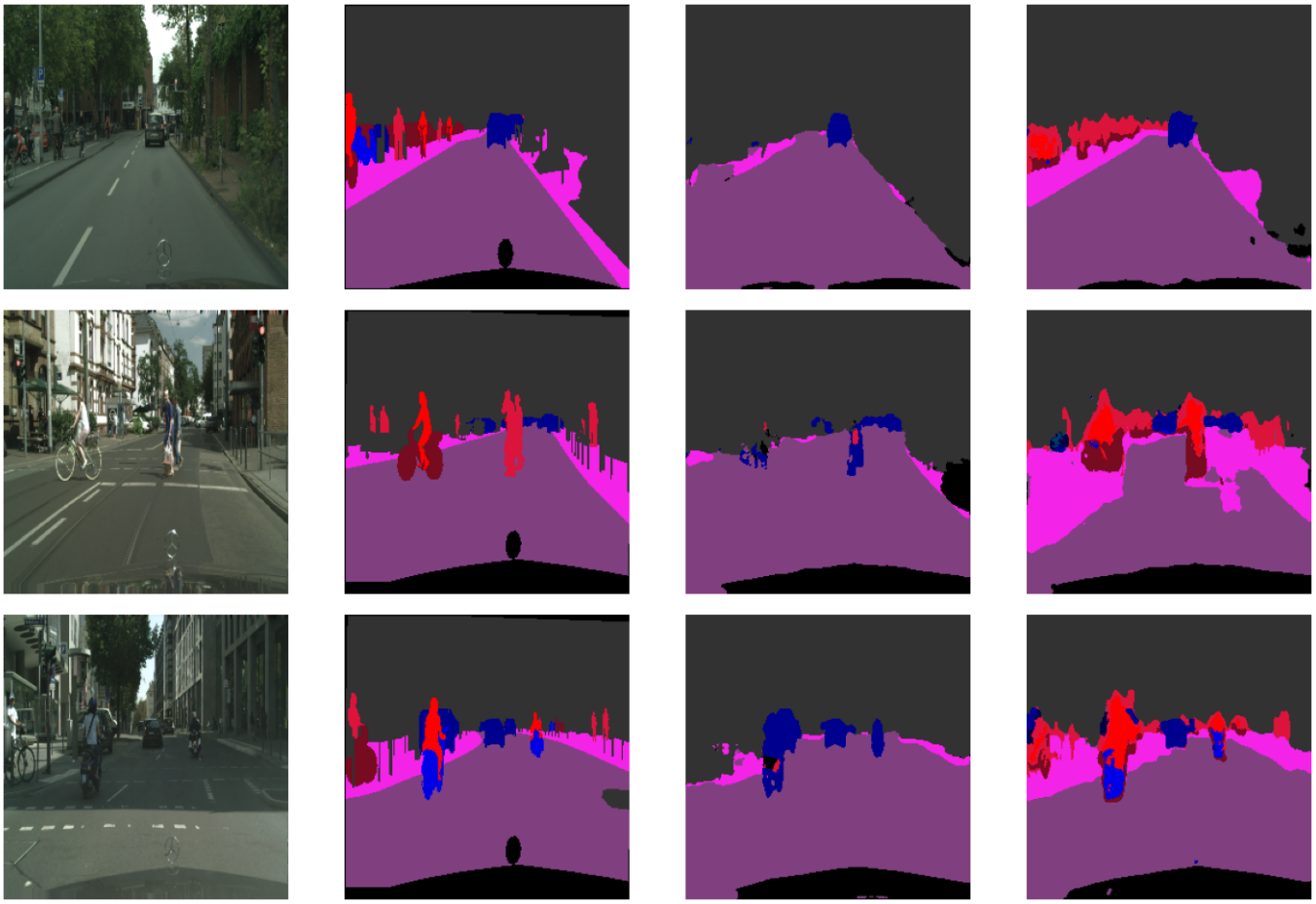
	CO2				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	3.4 ± 1.8	4.2 ± 3.0	15.6 ± 6.0	$\mathbf{89.1 \pm 11.4}$	99.9 ± 0.0
BDDIntersected (*wroadagents_noabstract*)	1.5 ± 0.5	1.6 ± 0.7	11.7 ± 3.6	71.7 ± 8.5	99.2 ± 0.1
BDD10K (*wroadagents_nodrivable_noabstract*)	32.8 ± 3.4	36.1 ± 7.7	41.9 ± 6.5	75.9 ± 3.0	99.6 ± 0.1
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	22.6 ± 5.4	25.5 ± 10.2	35.9 ± 6.3	69.9 ± 4.1	99.5 ± 0.2

	CSNP				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	1.7 ± 1.1	3.2 ± 1.8	6.5 ± 5.9	18.8 ± 7.7	39.6 ± 18.0
BDDIntersected (*wroadagents_noabstract*)	6.0 ± 5.0	1.0 ± 0.5	3.7 ± 2.4	11.5 ± 7.0	46.0 ± 17.6
BDD10K (*wroadagents_nodrivable_noabstract*)	28.2 ± 3.1	31.7 ± 1.5	30.2 ± 2.8	48.9 ± 1.3	88.8 ± 2.4
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	16.0 ± 6.0	23.2 ± 7.9	17.5 ± 2.6	38.1 ± 3.5	89.0 ± 5.2

	CSDT2			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	1.6 ± 1.1	5.4 ± 4.0	46.0 ± 5.9	55.1 ± 12.9
BDDIntersected (*wroadagents_noabstract*)	0.7 ± 0.3	6.8 ± 4.7	28.7 ± 5.3	42.6 ± 7.1
BDD10K (*wroadagents_nodrivable_noabstract*)	34.4 ± 2.5	35.4 ± 3.0	67.0 ± 4.4	96.4 ± 0.5
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	24.9 ± 2.1	25.5 ± 5.4	59.1 ± 5.6	97.2 ± 0.6

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	6.4 ± 3.6	17.7 ± 5.1	72.6 ± 16.0	$\mathbf{100.0 \pm 0.0}$
BDDIntersected (*wroadagents_noabstract*)	3.8 ± 2.1	12.7 ± 5.9	61.9 ± 13.6	$\mathbf{100.0 \pm 0.0}$
BDD10K (*wroadagents_nodrivable_noabstract*)	29.8 ± 3.2	49.9 ± 7.4	85.3 ± 2.0	$\mathbf{100.0 \pm 0.0}$
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	20.4 ± 3.2	47.1 ± 10.6	88.4 ± 3.9	$\mathbf{100.0 \pm 0.0}$

Table 5.8: Percentage of unimodal pixels results for the autonomous driving datasets (higher is better). The smaller-sized results in the first table fragment are for the mask setups with abstract classes, which are only able to be used with the ordinal segmentation methods by Fernandes et al. [20]. The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.



(a) Input image (b) Ground truth (c) Cross-entropy prediction (d) CO2 ($\lambda = 10.0$) prediction

Figure 5.9: Comparison of the influence of cross-entropy and CO2 ($\lambda = 10.0$) losses on model output in out-of-distribution inference. It can be seen that the CO2 loss output (d) more accurately identifies pedestrians, two-wheel vehicles, and riders when compared with the cross-entropy loss output (c).

5.3.1 Generalization through Dataset Scale Variance

In order to evaluate whether the inclusion of domain knowledge during the training of the models would help the network learn better with scarce data, an experiment that consisted of varying the scale of the dataset used to train the models was performed. In order to also evaluate the results with the Cityscapes out-of-distribution domain, the BDD10K mask used to train was *wroadagents_nodrivable_noabstract*. For each method, the best-performing lambda in the out-of-distribution scenario was selected, the rationale being that those models are the best at generalizing to unseen scenarios, which is useful when training with low amounts of data. For the CO2 and CSNP losses, $\lambda = 10.0$ was used, and for the CSDT2 loss, $\lambda = 0.1$ was used.

Figures 5.10, 5.11 and 5.12 show, respectively, the comparison plots for the Dice coefficient, contact surface, and percentage of unimodal pixels metrics for the results from each of the models trained with varying BDD10K dataset scales. Each plot uses dynamic *y* axis view limits to improve the visibility of the results.

When testing with the BDD10K dataset, the CO2 and CSNP methods achieve better Dice coefficient results at scales 0.25 and 0.5 when compared with the cross-entropy baseline, Figure 5.10, which suggests that, indeed, these losses help the network learn better when data is scarce. Especially when using the CO2 loss, resulting in absolute gains of 1.2% (5.7% in relative terms) in the Dice coefficient performance at scale 0.25 over cross-entropy. CO2 continues beating cross-entropy Dice performance through scales 0.1 and 0.05.

Cross-entropy generalizes better than the spatial methods when testing with the Cityscapes dataset in an out-of-distribution scenario for lower scales. However, the CO2 loss continues to generalize better than the cross-entropy throughout all scales, achieving a maximum absolute gain of 5.3% (15.7% in relative terms) in the Dice coefficient at scale 0.75. Still, it can be seen that the generalization ability of CO2 decreases faster than its own performance on BDD10K – training with fewer data has a higher impact on the model's generalization ability.

Regarding the ordinal metrics, the scale variance does not significantly affect the variation of the contact surface metric, Figure 5.11. In the percentage of unimodal pixels metric, Figure 5.12, the CO2 loss better maintains its performance at lower scale levels than the other methods.

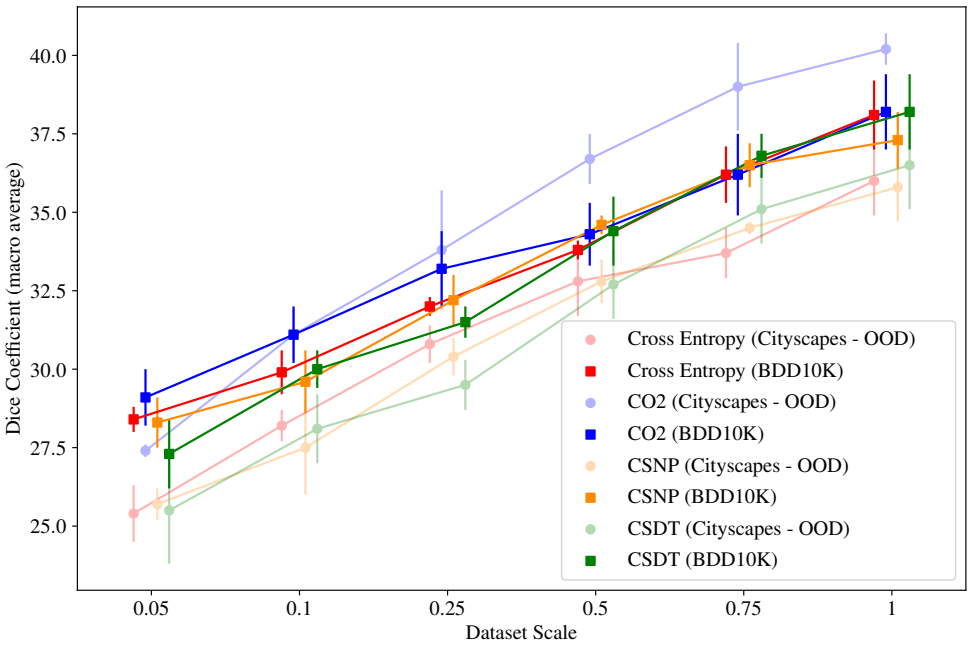


Figure 5.10: Dice coefficient (macro average) results for the autonomous driving datasets scale variation experiments (higher is better).

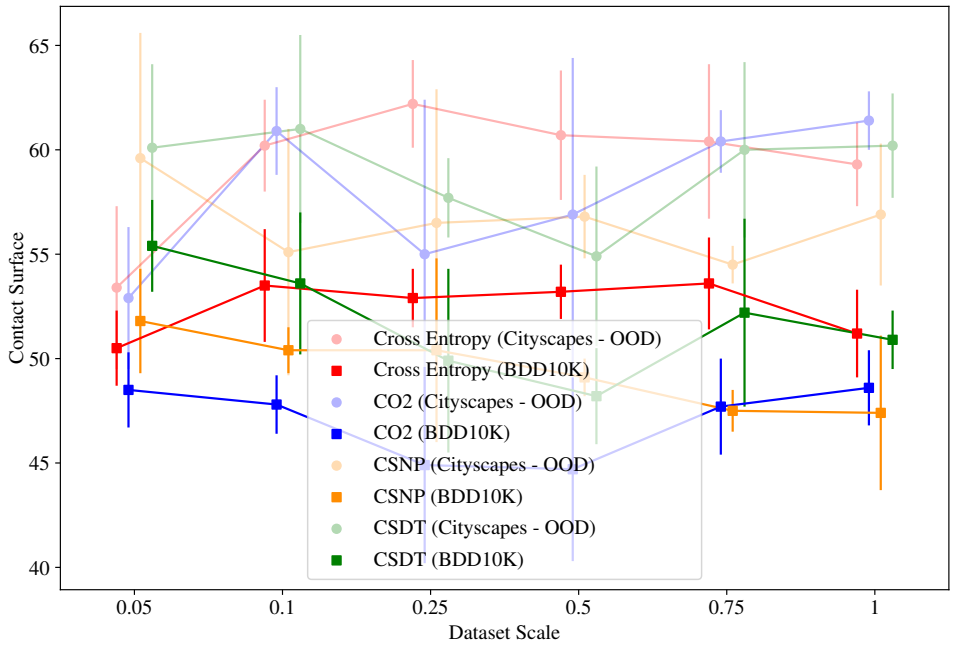


Figure 5.11: Contact surface results for the autonomous driving datasets scale variation experiments (lower is better).

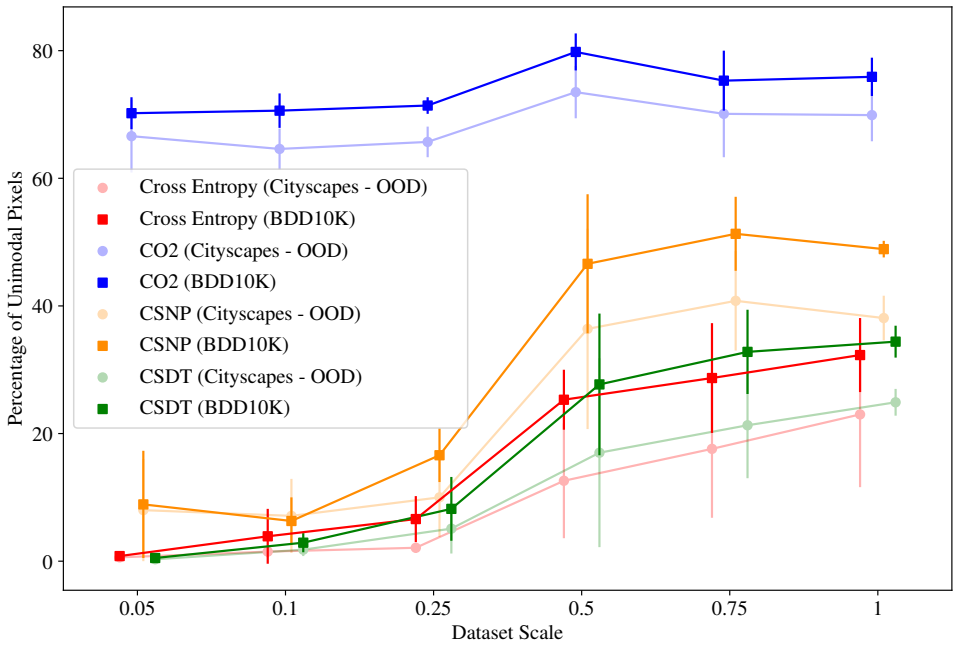


Figure 5.12: Percentage of unimodal pixels results for the autonomous driving datasets scale variation experiments (higher is better).

5.3.2 Semi-Supervised Learning

In order to evaluate whether using the unsupervised CSNP and CSDT2 losses with a mixture of labeled and unlabeled samples would result in additional inference performance gains, a semi-supervised learning (SSL) experiment was conducted. BDD100K is a good dataset for this experiment because it contains 10K images annotated for semantic segmentation and 100K without semantic segmentation labels, as described in Section 3.4. These unlabeled images can potentially improve the network's learning by being used to train it with unsupervised losses. To execute this experiment, a new modified version of the BDD100K dataset was created. This version of the dataset comprises 8 000 labeled images and around 1 300 unlabeled images. The 8 000 labeled images are used to compose the test and validation folds to evaluate the network mainly on the ground truth. This results in a training split of 5 120 labeled + 1 300 unlabeled images, a validation split of 1 280 labeled images, and a test split of 1 600 labeled images.

Tables 5.9, 5.10 and 5.11 display, respectively, the Dice coefficient, contact surface, and percentage of unimodal pixels metrics for the results from each of the models trained with semi-supervised learning on the custom dataset with varying λ values. Figures 5.13, 5.14 and 5.15 show the same results in a comparison plot. Each plot uses dynamic y axis view limits to improve the visibility of the results.

Overall, the unsupervised usage of the spatial ordinal segmentation losses did not result in improvements to the Dice coefficient metric results, Figure 5.13, decreasing it by around 2 to 4%

in absolute terms for low λ values and with a steep decrease starting at $\lambda = 10$. There were no improvements over the supervised learning results regarding the ordinal metrics.

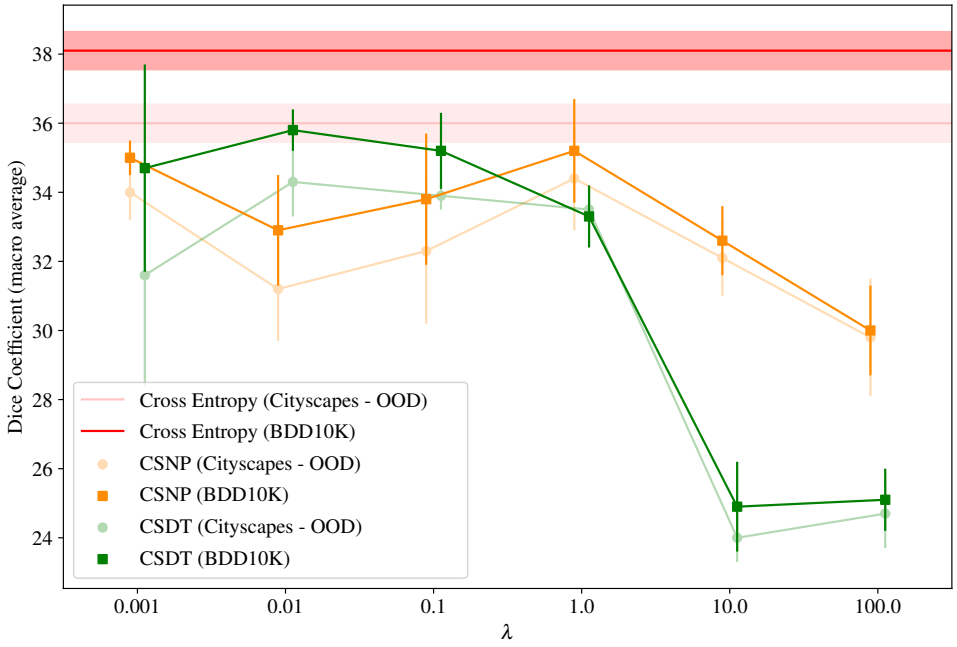


Figure 5.13: Dice coefficient (macro average) results for the autonomous driving datasets semi-supervised learning experiments (higher is better).

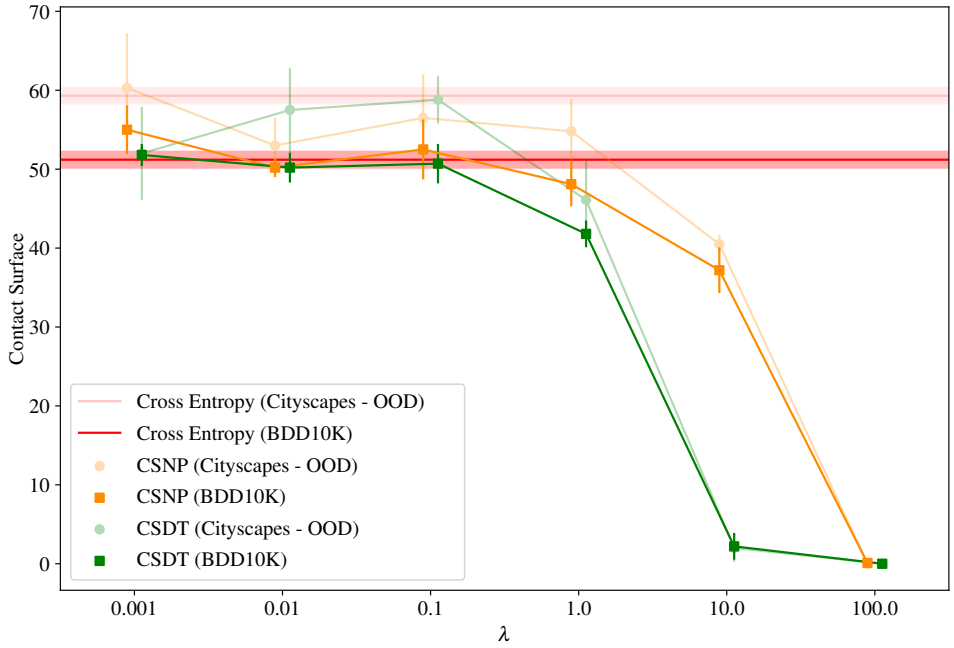


Figure 5.14: Contact surface results for the autonomous driving datasets semi-supervised learning experiments (lower is better).

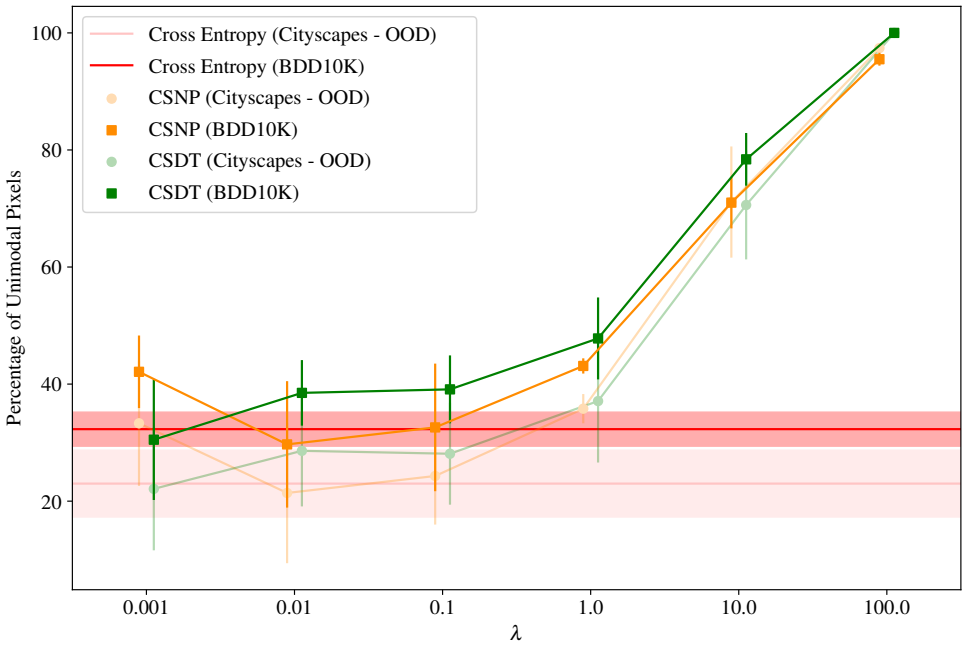


Figure 5.15: Percentage of unimodal pixels results for the autonomous driving datasets semi-supervised learning experiments (higher is better).

	Cross-Entropy
BDD10K (*wroadagents_nodrivable_noabstract*)	**38.1 ± 1.1**
Cityscapes (**OOD /** *wroadagents_nodrivable_noabstract*)	**36.0 ± 1.1**

	CSNP					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	35.0 ± 0.5	32.9 ± 1.6	33.8 ± 1.9	35.2 ± 1.5	32.6 ± 1.0	30.0 ± 1.3
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	34.0 ± 0.8	31.2 ± 1.5	32.3 ± 2.1	34.4 ± 1.5	32.1 ± 1.1	29.8 ± 1.7

	CSDT2					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	34.7 ± 3.0	35.8 ± 0.6	35.2 ± 1.1	33.3 ± 0.9	24.9 ± 1.3	25.1 ± 0.9
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	31.6 ± 3.2	34.3 ± 1.0	33.9 ± 0.4	33.5 ± 0.6	24.0 ± 0.7	24.7 ± 1.0

Table 5.9: Dice coefficient (macro average) results for the autonomous driving datasets semi-supervised learning experiments (higher is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy
BDD10K (*wroadagents_nodrivable_noabstract*)	51.22 ± 2.13
Cityscapes (**OOD /** *wroadagents_nodrivable_noabstract*)	59.34 ± 1.96

	CSNP					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	55.00 ± 3.06	50.16 ± 1.21	52.55 ± 3.80	48.09 ± 2.78	37.18 ± 2.88	0.09 ± 0.02
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	60.34 ± 6.93	53.00 ± 3.47	56.54 ± 5.53	54.83 ± 4.10	40.53 ± 1.24	0.01 ± 0.00

	CSDT2					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	51.78 ± 1.40	50.18 ± 1.92	50.73 ± 2.47	41.81 ± 1.68	2.17 ± 1.70	**0.00 ± 0.00**
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	52.00 ± 5.92	57.53 ± 5.33	58.75 ± 3.01	46.14 ± 5.15	1.99 ± 1.76	**0.00 ± 0.00**

Table 5.10: Contact surface results for the autonomous driving datasets semi-supervised learning experiments (lower is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy
BDD10K (*wroadagents_nodrivable_noabstract*)	32.3 ± 5.8
Cityscapes (**OOD /** *wroadagents_nodrivable_noabstract*)	23.0 ± 11.4

	CSNP					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	42.1 ± 6.2	29.7 ± 10.8	32.6 ± 10.9	43.1 ± 1.3	71.0 ± 4.4	95.5 ± 1.1
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	33.3 ± 10.7	21.4 ± 12.0	24.3 ± 8.3	35.8 ± 2.5	71.1 ± 9.5	97.4 ± 0.6

	CSDT2					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	30.5 ± 10.3	38.5 ± 5.6	39.1 ± 5.8	47.8 ± 7.0	78.4 ± 4.5	$\mathbf{100.0 \pm 0.0}$
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	22.1 ± 10.5	28.6 ± 9.5	28.1 ± 8.7	37.1 ± 10.5	70.6 ± 9.3	$\mathbf{100.0 \pm 0.0}$

Table 5.11: Percentage of unimodal pixels results for the autonomous driving datasets semi-supervised learning experiments (higher is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

5.4 Discussion

A comparison between the biomedical and autonomous driving datasets results concludes that the biomedical results are significantly better in terms of the Dice coefficient performance – the performance gains with the autonomous driving datasets are still positive, but not as large, since these datasets are considerably more complex, with a greater variety of scenarios and segmentation classes.

The pixel-wise CO2 method performed well in both the biomedical and autonomous driving datasets, especially when considered in an out-of-distribution domain – it could potentially be applied in a real-world scenario in order to improve the generalization capabilities of perception algorithms.

The spatial methods CSNP and CSDT2 in their current form may not be applicable, at least at high regularization weights, to the autonomous driving scenario. The scene perspective from the car makes it so that there is a large amount of valid contact surface between non-ordinally adjacent classes. Adaptations of these methods that consider this type of contact could be devised in the future. However, in the biomedical datasets, these methods had a good performance, with their greatest difficulty being the existence of occlusions.

As was seen, the choice of λ value, i.e., regularization weight, is critical for the performance of the proposed methods. Because of this, in order to be applied to different domains, there should be an empirical study of the influence of the lambda value in the segmentation performance in that specific domain and the choice of a value that is a balance between the ordinal metrics and the Dice coefficient, i.e., a value that promotes some unimodality and spatial consistency but also does not hurt Dice performance to the point where it is unusable.

Finally, it can be seen that the Dice values for the autonomous driving datasets are underwhelming (less than 50%), and the proposed algorithms, when solely used, although they result in some Dice performance gain, were not enough to substantially increase it. Therefore, in the real world, it would be desirable to use algorithms that are not solely based on RGB image segmentation but that use a combination of sensors, like LiDARs.

Chapter 6

Conclusions

The present chapter concludes the dissertation document. Section 6.1 revisits and summarizes the key conclusions from the work carried out. Section 6.2 delineates future work that could be explored in the context of ordinal segmentation and autonomous driving.

6.1 Final Remarks

This dissertation explores the proposition that introducing domain knowledge to deep neural networks used for scene parsing in autonomous driving can improve their generalization capabilities, making them more robust, reliable, and safer to employ in real-world scenarios. With this in mind, the focus of the work was: (**1**) the development of novel ordinal segmentation loss functions, which seek to imbue the networks with ordinal constraints during training; and (**2**) the transposition of autonomous driving to an ordinal domain, including the adaptation of the proposed methods to domains with arbitrary ordinal hierarchies.

Two categories of loss functions for ordinal segmentation were studied: (**1**) pixel-wise, where each pixel is treated individually by promoting unimodality in its probability distribution; and (**2**) spatial, where each pixel is considered in the context of its neighborhood and the contact surface between non-ordinally adjacent classes is minimized. The following losses were proposed: (**1**) the pixel-wise adaptation of the CO2 loss for segmentation; (**2**) the spatial CSNP loss, which considers only the immediate neighbor pixels; and (**3**) the spatial CSDT loss, which considers the global neighborhood. In addition, two metrics were proposed to evaluate the network output's ordinal consistency: (**1**) the percentage of unimodal pixels and (**2**) the contact surface between the segmentation masks of non-ordinally adjacent classes.

The proposed methods were initially validated with five biomedical datasets, where it was clear, by analyzing the ordinal metrics, that their usage resulted in more ordinally consistent models and a good amount of ordinal consistency was achieved without a major negative impact on the Dice coefficient results. To evaluate the impact of the methods in autonomous driving domains, they were trained with the BDD100K dataset and tested in two different scenarios: (**1**) on

the BDD100K dataset; and (**2**) in an out-of-distribution domain, through the Cityscapes dataset, to evaluate their generalization ability. Furthermore, two additional experiences were conducted – the models were trained with: (**1**) scaled-down versions of the BDD100K dataset, to evaluate how the network learns with scarce data; and (**2**) semi-supervised learning, to evaluate if using the methods with a mixture of labeled and unlabeled samples would improve the network's learning.

It was observed that promoting spatial constraints, beyond improving the spatial metric, also results in improvements in the pixel-wise metric and vice-versa. It was also clear that at high lambda values, all methods suffer from over-regularization – the applied ordinal constraints are excessive and result in predictions that deviate too much from the ground truth.

The CO2 pixel-wise loss achieved the best overall results compared with cross-entropy, achieving a maximum Dice coefficient absolute improvement of 1.5% (4% in relative terms) when testing with BDD100K. In an out-of-distribution domain, the models trained with this loss achieved absolute gains of 4.2% (11.5% in relative terms), which proves that it helps the model generalize better to unseen situations. When trained with reduced-scale versions of the BDD10K dataset, the loss obtained maximum absolute improvements of 1.2% (5.7% in relative terms) when tested with the BDD10K dataset and of 5.3% (15.7% in relative terms) when tested in an out-of-distribution domain.

The CSNP and CSDT spatial losses served their purpose in reducing the contact surface between non-ordinally adjacent classes but may not apply to various real-world scenarios with occlusions and unexpected perspectives. These particularities make it so that contact between non-ordinally adjacent masks is sometimes valid. Still, the proposed methods only break these scenarios at high lambda values, making them useful for promoting ordinal consistency at low regularization weights.

In conclusion, incorporating ordinal consistency into autonomous driving scene parsing models showed promising results, including developing generalizable models that exhibit improved learning capabilities with limited data availability.

6.2 Future Work

In terms of future directions for research in the topic of this dissertation, the division into two primary categories is suggested: (**1**) the development of ordinal segmentation methods; and (**2**) the exploration of ordinality applied to autonomous driving domains.

Regarding ordinal segmentation, the following areas have the potential for further investigation:

- Development of more flexible spatial ordinal segmentation methods, allowing for limited contact between non-ordinally adjacent classes, such as in the case of occlusions and different perspectives.

- Exploration of an unsupervised adaptation of the CO2 loss, enabling its application in semi-supervised learning scenarios, and the evaluation of its performance.
- Development of novel methods that leverage ordinal constraints not necessarily consisting of augmented loss functions.

In the realm of autonomous driving and ordinality, the following topics can be explored:

- Usage of datasets containing more ordinal relations, such as those incorporating segmented road markings.
- Experimentation with novel ordinal mask setups.

These preliminary ideas lay the foundation for future work. Furthermore, it is essential to consider the exploration of novel approaches for the introduction of domain knowledge, which could enhance the reliability of these algorithms and facilitate the safer application of deep learning perception pipelines in real-world scenarios.

References

[1] World Health Organization. Global status report on road safety 2018. Technical report, World Health Organization, 2018.

[2] World Health Organization. Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, June 2022. Accessed: 2023-02-05.

[3] Santokh Singh. Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. *Traffic Safety Facts - Crash Stats*, February 2015. Number: DOT HS 812 115.

[4] Badr Ben Elallid, Nabil Benamar, Abdelhakim Senhaji Hafid, Tajjeeddine Rachidi, and Nabil Mrani. A Comprehensive Survey on the Application of Deep and Reinforcement Learning Approaches in Autonomous Driving. *Journal of King Saud University - Computer and Information Sciences*, 34(9):7366–7390, October 2022.

[5] On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, April 2021.

[6] Darsh Parekh, Nishi Poddar, Aakash Rajpurkar, Manisha Chahal, Neeraj Kumar, Gyanendra Prasad Joshi, and Woong Cho. A Review on Autonomous Vehicles: Progress, Methods and Challenges. *Electronics*, 11(14):2162, January 2022. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation, March 2015. arXiv:1411.4038 [cs].

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. arXiv:1505.04597 [cs] version: 1.

[9] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, October 2016. arXiv:1511.00561 [cs].

[10] Jigang Tang, Songbin Li, and Peng Liu. A review of lane detection methods based on deep learning. *Pattern Recognition*, 111:107623, March 2021.

[11] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial Examples for Semantic Segmentation and Object Detection, July 2017. arXiv:1703.08603 [cs].

[12] 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, April 2016. arXiv:1604.01685 [cs].

[13] Ozan Öztürk, Batuhan Sariturk, and Dursun Seker. Comparison of Fully Convolutional Networks (FCN) and U-Net for Road Segmentation from High Resolution Imageries. *International Journal of Environment and Geoinformatics*, 7:272–279, September 2020.

[14] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for Deep Learning: A Taxonomy, October 2017. arXiv:1710.10686 [cs, stat].

[15] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 12(1):1040, January 2022. Publisher: Nature Publishing Group.

[16] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. Incorporating Domain Knowledge into Deep Neural Networks, March 2021. arXiv:2103.00180 [cs].

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.

[18] Mattia Silvestri, Michele Lombardi, and Michela Milano. Injecting Domain Knowledge in Neural Networks: a Controlled Experiment on a Constrained Problem, February 2020. arXiv:2002.10742 [cs].

[19] Pedro M. Ferreira, Filipe Marques, Jaime S. Cardoso, and Ana Rebelo. Physiological Inspired Deep Neural Networks for Emotion Recognition. *IEEE Access*, 6:53930–53943, 2018. Conference Name: IEEE Access.

[20] Kelwin Fernandes and Jaime S. Cardoso. Ordinal Image Segmentation using Deep Neural Networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2018. ISSN: 2161-4407.

[21] Jie Zhang, Yi Xu, Bingbing Ni, and Zhenyu Duan. Geometric Constrained Joint Lane Segmentation and Lane Boundary Detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11205, pages 502–518, Cham, 2018. Springer International Publishing.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[23] Michael Crawshaw. Multi-Task Learning with Deep Neural Networks: A Survey, September 2020. arXiv:2009.09796 [cs, stat].

[24] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks, June 2017. arXiv:1706.05098 [cs, stat].

[25] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 160–167, New York, NY, USA, July 2008. Association for Computing Machinery.

[26] Joaquim F. Pinto da Costa, Hugo Alonso, and Jaime S. Cardoso. The unimodal model for the classification of ordinal data. *Neural Networks*, 21(1):78–91, January 2008.

[27] Jianlin Cheng. A neural network approach to ordinal regression, April 2007. arXiv:0704.1028 [cs].

[28] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks, June 2016. arXiv:1606.00298 [cs].

[29] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D. Plumbley. Detection and Classification of Acoustic Scenes and Events. *IEEE Transactions on Multimedia*, 17(10):1733–1746, October 2015.

[30] Eibe Frank and Mark Hall. A Simple Approach to Ordinal Classification. In Luc De Raedt and Peter Flach, editors, *Machine Learning: ECML 2001*, Lecture Notes in Computer Science, pages 145–156, Berlin, Heidelberg, 2001. Springer.

[31] Christopher Beckham and Christopher Pal. Unimodal probability distributions for deep ordinal classification, June 2017. arXiv:1705.05278 [stat].

[32] Tomé Albuquerque, Ricardo Cruz, and Jaime S. Cardoso. Ordinal losses for classification of cervical cancer risk. *PeerJ Computer Science*, 7:e457, April 2021. Publisher: PeerJ Inc.

[33] Jaime S. Cardoso and Maria J. Cardoso. Towards an intelligent medical system for the aesthetic evaluation of breast cancer conservative treatment. *Artificial Intelligence in Medicine*, 40(2):115–126, June 2007.

[34] Intel. Intel & MobileODT Cervical Cancer Screening. https://kaggle.com/competitions/intel-mobileodt-cervical-cancer-screening. Accessed: 2023-06-22.

[35] Ana F. Sequeira, João C. Monteiro, Ana Rebelo, and Hélder P. Oliveira. MobBIO: A multimodal database captured with a portable handheld device. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 133–139, January 2014.

[36] Ching-Wei Wang, Cheng-Ta Huang, Jia-Hong Lee, Chung-Hsing Li, Sheng-Wei Chang, Ming-Jhih Siao, Tat-Ming Lai, Bulat Ibragimov, Tomaž Vrtovec, Olaf Ronneberger, Philipp Fischer, Tim F. Cootes, and Claudia Lindner. A benchmark for comparison of dental radiography analysis algorithms. *Medical Image Analysis*, 31:63–76, July 2016.

[37] Kelwin Fernandez and Carolina Chang. Teeth/Palate and Interdental Segmentation Using Artificial Neural Networks. In Nadia Mana, Friedhelm Schwenker, and Edmondo Trentin, editors, *Artificial Neural Networks in Pattern Recognition*, Lecture Notes in Computer Science, pages 175–185, Berlin, Heidelberg, 2012. Springer.

[38] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[39] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018.

[40] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[41] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[42] Duc Duy Pham, Gurbandurdy Dovletov, and Josef Pauli. A Differentiable Convolutional Distance Transform Layer for Improved Image Segmentation. In Zeynep Akata, Andreas Geiger, and Torsten Sattler, editors, *Pattern Recognition*, Lecture Notes in Computer Science, pages 432–444, Cham, 2021. Springer International Publishing.

[43] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs].

Appendix A

PyTorch Code Samples for the Proposed Loss Regularization Terms

A.1 CSNP

```
def CSNP(P, K):
    loss = 0
    count = 0

    # for each pair of non-ordinally adjacent classes
    for k1 in range(K):
        for k2 in range(K):
            if abs(k2 - k1) <= 1:
                continue

            # more weight to more ordinally distant classes
            ordinal_multiplier = abs(k2 - k1) - 1

            dx = P[:,  k1, :, :-1] * P[:, k2, :, 1:]
            dy = P[:,  k1, :-1, :] * P[:, k2, 1:, :]

            loss += ordinal_multiplier * (torch.mean(dx) + torch.mean(dy)) / 2
            count += 1

    if count != 0:
        loss /= count
    return loss
```

A.2 CSDT2

```
def CSDT2(P, K, threshold=.5, max_dist=10.):
    loss = 0
    count = 0

    activations = 1. * (P > threshold)
    DT = distance_transform(activations)

    # cap the maximum distance at 10
    max_dist_DT = (DT >= max_dist) * max_dist
    # select the values with a distance < 10
    DT *= DT < max_dist
    # add the capped values
    DT += max_dist_DT

    # for each pair of non-ordinally adjacent classes
    for k1 in range(K):
        for k2 in range(k1 + 2, K):
            # more weight to more ordinally distant classes
            ordinal_multiplier = abs(k2 - k1) - 1

            d_k1, d_k2 = DT[:, k1], DT[:, k2]
            p_k1, p_k2 = P[:, k1], P[:, k2]

            calc = p_k1 * d_k2 + p_k2 * d_k1
            calc = calc[calc != 0]

            loss += ordinal_multiplier * torch.mean(calc)
            count += 1

    if count != 0:
        loss /= count

    loss /= max_dist
    loss *= -1 # maximize

    return loss
```

Appendix B

Additional Metrics for the Experimental Results

B.1 Biomedical Datasets

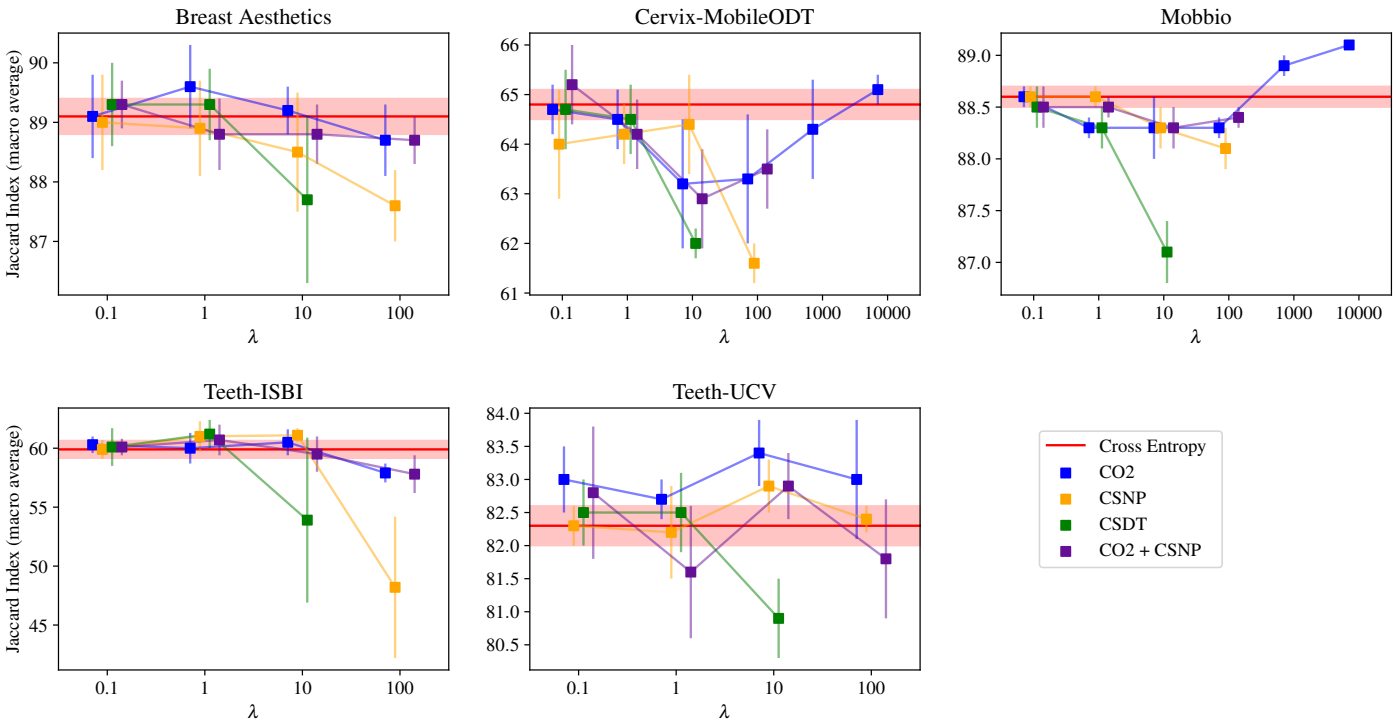


Figure B.1: Jaccard index (macro average) results for the biomedical datasets (higher is better).

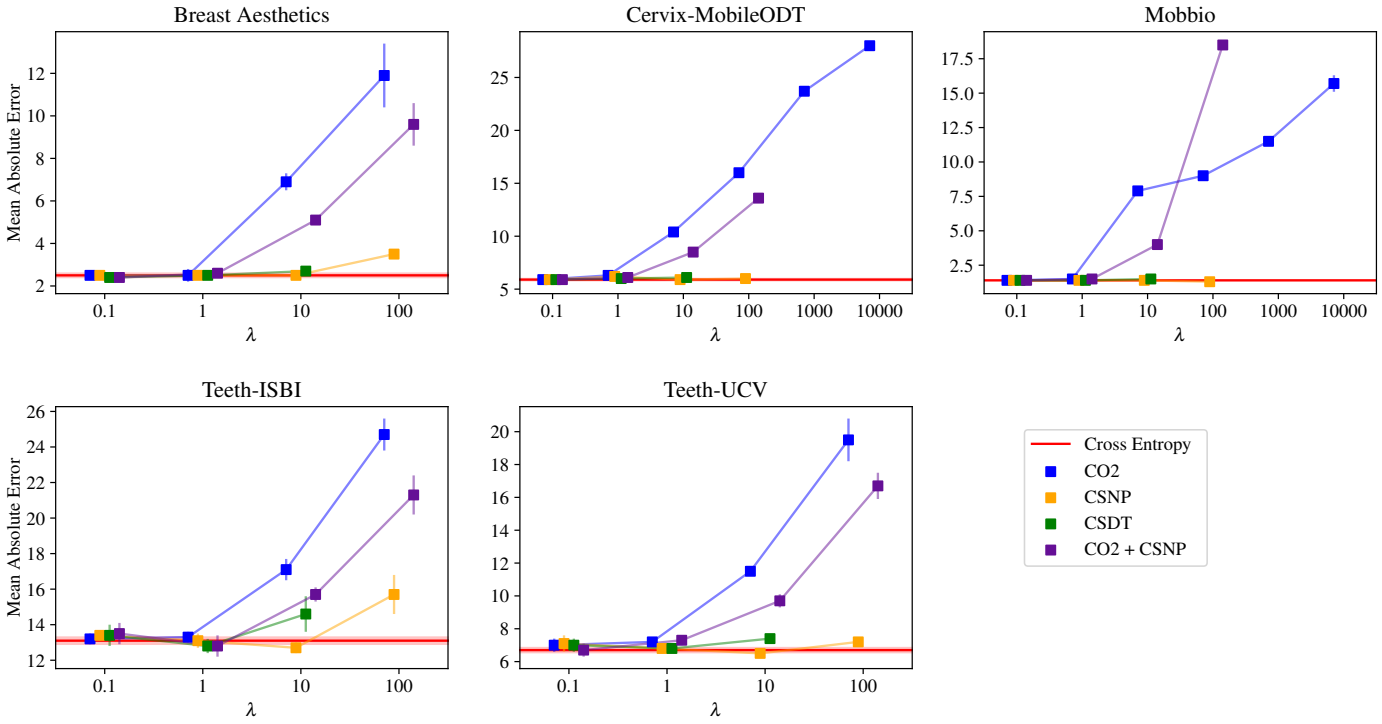


Figure B.2: Mean absolute error results for the biomedical datasets (lower is better). Displayed values are multiplied by 10^2, in order to facilitate analysis.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
Breast Aesthetics	**89.1 ± 0.6**	76.2 ± 3.5	**89.1 ± 0.7**	10.6 ± 0.6
Cervix-MobileODT	**64.8 ± 0.6**	63.1 ± 0.8	64.2 ± 0.7	52.4 ± 1.3
Mobbio	88.6 ± 0.2	88.1 ± 0.3	88.4 ± 0.2	87.8 ± 0.2
Teeth-ISBI	**59.9 ± 1.5**	8.2 ± 5.1	59.5 ± 1.0	10.4 ± 0.9
Teeth-UCV	82.3 ± 0.6	52.6 ± 5.8	81.6 ± 0.8	27.2 ± 0.2

	CO2					
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$	$\lambda = 1000.0$	$\lambda = 10000.0$
Breast Aesthetics	**89.1 ± 0.7**	**89.6 ± 0.7**	**89.2 ± 0.4**	88.7 ± 0.6		
Cervix-MobileODT	**64.7 ± 0.5**	64.5 ± 0.6	63.2 ± 1.3	63.3 ± 1.3	**64.3 ± 1.0**	**65.1 ± 0.3**
Mobbio	88.6 ± 0.1	88.3 ± 0.1	88.3 ± 0.3	88.3 ± 0.1	88.9 ± 0.1	**89.1 ± 0.0**
Teeth-ISBI	**60.3 ± 0.7**	**60.0 ± 1.3**	**60.5 ± 1.1**	57.9 ± 0.8		
Teeth-UCV	**83.0 ± 0.5**	82.7 ± 0.3	**83.4 ± 0.5**	**83.0 ± 0.9**		

	CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	**89.0 ± 0.8**	**88.9 ± 0.8**	88.5 ± 1.0	87.6 ± 0.6
Cervix-MobileODT	64.0 ± 1.1	64.2 ± 0.6	**64.4 ± 1.0**	61.6 ± 0.4
Mobbio	88.6 ± 0.1	88.6 ± 0.1	88.3 ± 0.2	88.1 ± 0.2
Teeth-ISBI	**59.9 ± 0.8**	**61.0 ± 1.3**	**61.1 ± 0.6**	48.2 ± 6.0
Teeth-UCV	82.3 ± 0.3	82.2 ± 0.7	**82.9 ± 0.4**	82.4 ± 0.2

	CSDT2		
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$
Breast Aesthetics	**89.3 ± 0.7**	**89.3 ± 0.6**	87.7 ± 1.4
Cervix-MobileODT	**64.7 ± 0.8**	**64.5 ± 0.7**	62.0 ± 0.3
Mobbio	88.5 ± 0.2	88.3 ± 0.2	87.1 ± 0.3
Teeth-ISBI	**60.1 ± 1.6**	**61.2 ± 1.2**	53.9 ± 7.0
Teeth-UCV	82.5 ± 0.5	82.5 ± 0.6	80.9 ± 0.6

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	**89.3 ± 0.4**	**88.8 ± 0.6**	88.8 ± 0.5	88.7 ± 0.4
Cervix-MobileODT	**65.2 ± 0.8**	64.2 ± 0.7	62.9 ± 1.0	63.5 ± 0.8
Mobbio	88.5 ± 0.2	88.5 ± 0.1	88.3 ± 0.2	88.4 ± 0.1
Teeth-ISBI	**60.1 ± 0.7**	**60.7 ± 1.3**	59.5 ± 1.5	57.8 ± 1.6
Teeth-UCV	**82.8 ± 1.0**	81.6 ± 1.0	**82.9 ± 0.5**	81.8 ± 0.9

Table B.1: Jaccard index (macro average) results for the biomedical datasets (higher is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
Breast Aesthetics	**2.5 ± 0.2**	14.2 ± 2.1	4.4 ± 0.6	26.9 ± 1.0
Cervix-MobileODT	**5.9 ± 0.2**	6.1 ± 0.1	**5.9 ± 0.1**	10.7 ± 1.3
Mobbio	1.4 ± 0.0	1.5 ± 0.0	1.4 ± 0.0	3.5 ± 0.7
Teeth-ISBI	**13.1 ± 0.4**	30.8 ± 0.5	17.4 ± 0.3	30.1 ± 0.2
Teeth-UCV	**6.7 ± 0.3**	25.4 ± 1.4	11.1 ± 0.5	29.3 ± 0.3

	CO2					
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$	$\lambda = 1000.0$	$\lambda = 10000.0$
Breast Aesthetics	**2.5 ± 0.2**	**2.5 ± 0.3**	6.9 ± 0.4	11.9 ± 1.5		
Cervix-MobileODT	**5.9 ± 0.2**	6.3 ± 0.1	10.4 ± 0.3	16.0 ± 0.3	23.7 ± 0.4	28.0 ± 0.2
Mobbio	1.4 ± 0.0	1.5 ± 0.0	7.9 ± 0.0	9.0 ± 0.1	11.5 ± 0.1	15.7 ± 0.6
Teeth-ISBI	13.2 ± 0.3	13.3 ± 0.3	17.1 ± 0.6	24.7 ± 0.9		
Teeth-UCV	7.0 ± 0.4	7.2 ± 0.2	11.5 ± 0.2	19.5 ± 1.3		

	CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	**2.5 ± 0.2**	**2.5 ± 0.2**	**2.5 ± 0.2**	3.5 ± 0.2
Cervix-MobileODT	**5.9 ± 0.2**	6.2 ± 0.3	**5.9 ± 0.2**	**6.0 ± 0.2**
Mobbio	1.4 ± 0.0	1.4 ± 0.0	1.4 ± 0.0	**1.3 ± 0.0**
Teeth-ISBI	13.4 ± 0.3	**13.1 ± 0.4**	**12.7 ± 0.3**	15.7 ± 1.1
Teeth-UCV	7.1 ± 0.5	**6.8 ± 0.3**	**6.5 ± 0.2**	7.2 ± 0.2

	CSDT2		
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$
Breast Aesthetics	**2.4 ± 0.2**	**2.5 ± 0.2**	2.7 ± 0.2
Cervix-MobileODT	**5.9 ± 0.2**	**6.0 ± 0.2**	6.1 ± 0.1
Mobbio	1.4 ± 0.0	1.4 ± 0.0	1.5 ± 0.0
Teeth-ISBI	**13.4 ± 0.6**	**12.8 ± 0.4**	14.6 ± 1.0
Teeth-UCV	7.0 ± 0.4	**6.8 ± 0.2**	7.4 ± 0.2

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
Breast Aesthetics	**2.4 ± 0.2**	2.6 ± 0.1	5.1 ± 0.2	9.6 ± 1.0
Cervix-MobileODT	**5.9 ± 0.2**	6.1 ± 0.2	8.5 ± 0.1	13.6 ± 0.4
Mobbio	1.4 ± 0.0	1.5 ± 0.0	4.0 ± 0.0	18.5 ± 0.1
Teeth-ISBI	13.5 ± 0.6	**12.8 ± 0.6**	15.7 ± 0.4	21.3 ± 1.1
Teeth-UCV	**6.7 ± 0.4**	7.3 ± 0.3	9.7 ± 0.4	16.7 ± 0.8

Table B.2: Mean absolute error results for the biomedical datasets (lower is better). Displayed values are multiplied by 10^2, in order to facilitate analysis. The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

B.2 Autonomous Driving Datasets

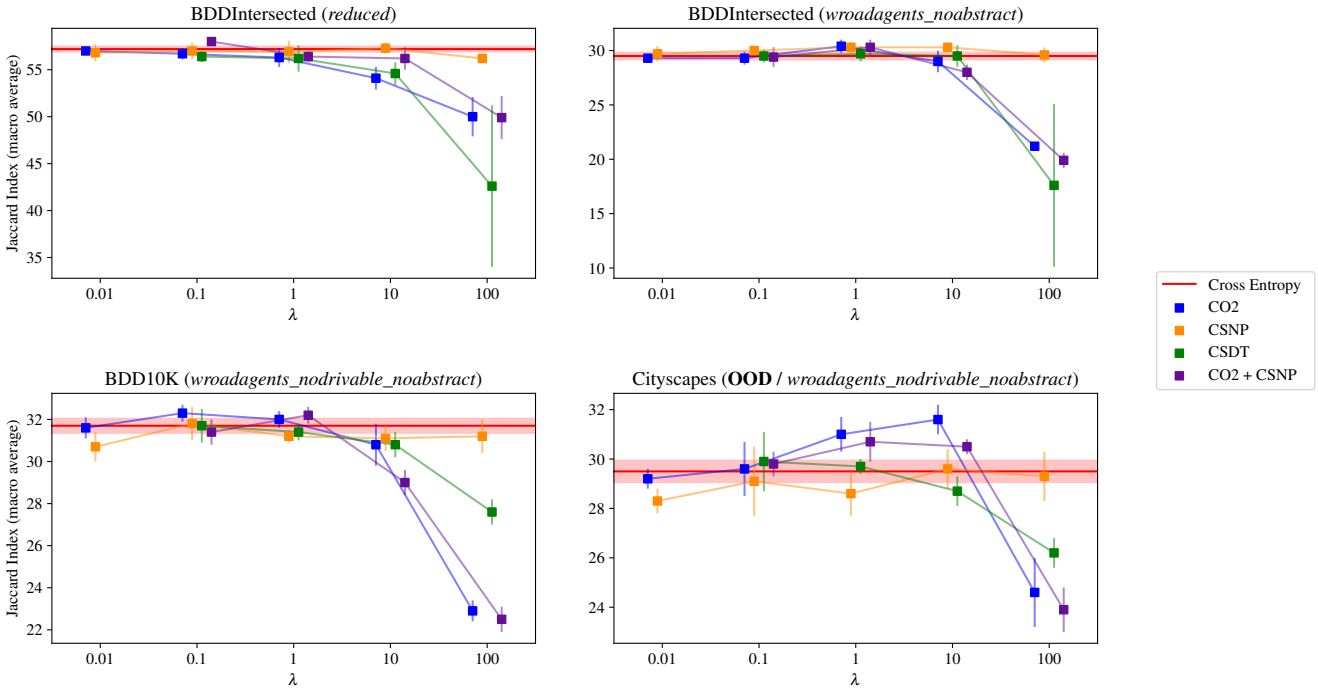


Figure B.3: Jaccard index (macro average) results for the autonomous driving datasets (higher is better).

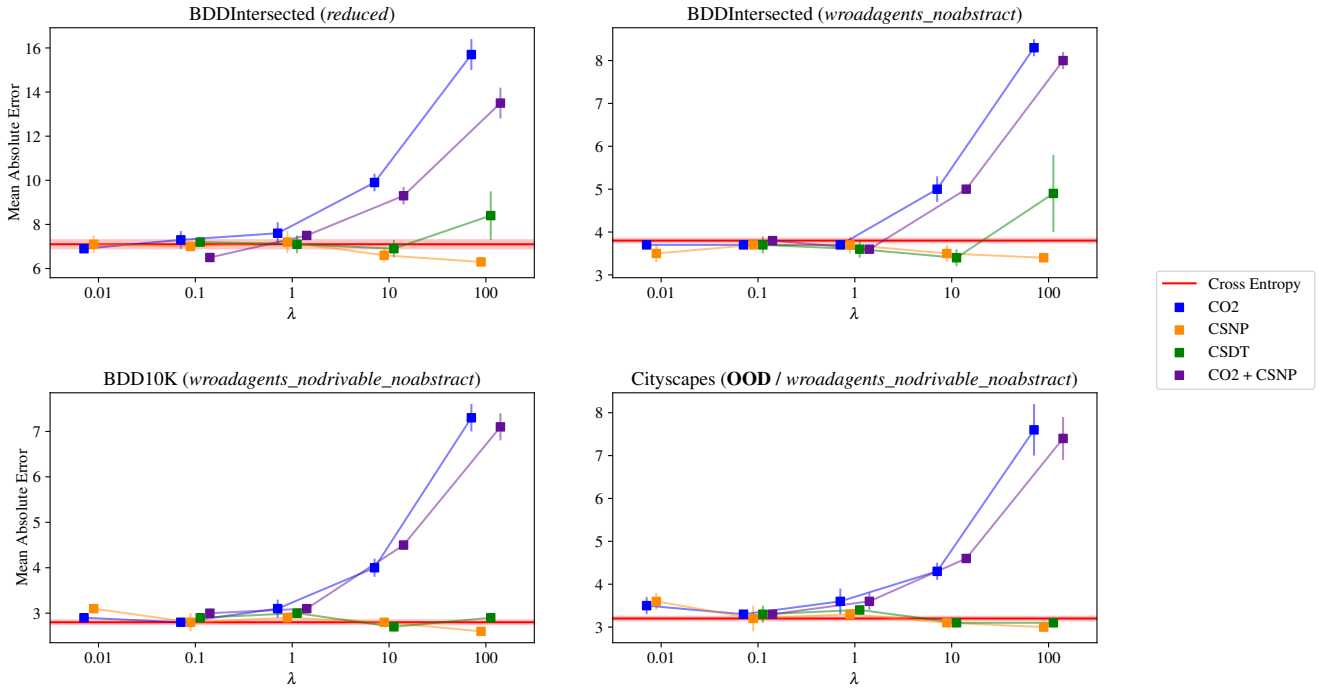


Figure B.4: Mean absolute error results for the autonomous driving datasets (lower is better). Displayed values are multiplied by 10^2, in order to facilitate analysis.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
BDDIntersected (*reduced*)	57.2 ± 0.6	$\mathbf{57.3 \pm 1.1}$	$\mathbf{57.6 \pm 0.6}$	57.5 ± 0.5
BDDIntersected (*wroadagents_noabstract*)	29.5 ± 0.7	29.1 ± 0.4	$\mathbf{29.8 \pm 0.6}$	29.3 ± 0.8
BDD10K (*wroadagents_nodrivable_noabstract*)	31.7 ± 0.7	30.7 ± 0.4	29.9 ± 0.6	31.2 ± 0.2
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	29.5 ± 0.9	28.9 ± 0.4	28.7 ± 0.9	29.4 ± 0.4
BDDIntersected (*wroadagents*)		$\mathbf{22.8 \pm 0.3}$	22.3 ± 0.2	22.2 ± 0.3
BDD10K (*wroadagents_nodrivable*)		22.8 ± 0.3	$\mathbf{23.1 \pm 0.6}$	$\mathbf{23.3 \pm 0.3}$
Cityscapes (**OOD** / *wroadagents_nodrivable*)		$\mathbf{21.7 \pm 0.3}$	$\mathbf{21.4 \pm 0.4}$	$\mathbf{21.5 \pm 0.4}$

	CO2				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	57.0 ± 0.3	56.7 ± 0.6	56.3 ± 1.0	54.1 ± 1.2	50.0 ± 2.1
BDDIntersected (*wroadagents_noabstract*)	29.3 ± 0.4	29.3 ± 0.6	$\mathbf{30.4 \pm 0.6}$	29.0 ± 1.0	21.2 ± 0.2
BDD10K (*wroadagents_nodrivable_noabstract*)	31.6 ± 0.5	$\mathbf{32.3 \pm 0.4}$	$\mathbf{32.0 \pm 0.4}$	30.8 ± 1.0	22.9 ± 0.5
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	29.2 ± 0.4	29.6 ± 1.1	$\mathbf{31.0 \pm 0.7}$	$\mathbf{31.6 \pm 0.6}$	24.6 ± 1.4

	CSNP				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	56.8 ± 0.9	57.0 ± 0.9	56.9 ± 1.2	57.3 ± 0.6	56.2 ± 0.4
BDDIntersected (*wroadagents_noabstract*)	$\mathbf{29.7 \pm 0.7}$	$\mathbf{30.0 \pm 0.3}$	$\mathbf{30.3 \pm 0.5}$	$\mathbf{30.3 \pm 0.3}$	29.6 ± 0.7
BDD10K (*wroadagents_nodrivable_noabstract*)	30.7 ± 0.7	$\mathbf{31.8 \pm 0.8}$	31.2 ± 0.3	31.1 ± 0.6	31.2 ± 0.8
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	28.3 ± 0.5	29.1 ± 1.4	28.6 ± 0.9	29.6 ± 0.8	29.3 ± 1.0

	CSDT2			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	56.4 ± 0.6	56.2 ± 1.4	54.6 ± 1.2	42.6 ± 8.6
BDDIntersected (*wroadagents_noabstract*)	29.5 ± 0.6	$\mathbf{29.7 \pm 0.7}$	29.5 ± 1.0	17.6 ± 7.5
BDD10K (*wroadagents_nodrivable_noabstract*)	$\mathbf{31.7 \pm 0.8}$	31.4 ± 0.4	30.8 ± 0.6	27.6 ± 0.6
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	29.9 ± 1.2	29.7 ± 0.3	28.7 ± 0.6	26.2 ± 0.6

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	$\mathbf{58.0 \pm 0.2}$	56.4 ± 0.6	56.2 ± 1.2	49.9 ± 2.3
BDDIntersected (*wroadagents_noabstract*)	29.4 ± 0.9	$\mathbf{30.3 \pm 0.7}$	28.0 ± 0.7	19.9 ± 0.7
BDD10K (*wroadagents_nodrivable_noabstract*)	31.4 ± 0.6	$\mathbf{32.2 \pm 0.4}$	29.0 ± 0.6	22.5 ± 0.6
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	29.8 ± 0.5	30.7 ± 0.8	30.5 ± 0.3	23.9 ± 0.9

Table B.3: Jaccard index (macro average) results for the autonomous driving datasets (higher is better). The smaller-sized results in the first table fragment are for the mask setups with abstract classes, which are only able to be used with the ordinal segmentation methods by Fernandes et al. [20]. The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy	**Ordinal Segmentation** (Fernandes et al. [20])		
	-	Ordinal Encoding	Pixel-Wise Consistency	Decision Boundary Parallelism
BDDIntersected (*reduced*)	7.1 ± 0.4	7.8 ± 0.6	7.0 ± 0.3	7.3 ± 0.3
BDDIntersected (*wroadagents_noabstract*)	3.8 ± 0.1	3.9 ± 0.1	3.6 ± 0.1	3.7 ± 0.1
BDD10K (*wroadagents_nodrivable_noabstract*)	2.8 ± 0.1	2.9 ± 0.1	3.0 ± 0.1	2.9 ± 0.1
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	3.2 ± 0.1	3.2 ± 0.1	3.2 ± 0.1	3.2 ± 0.2
BDDIntersected (*wroadagents*)		**3.0 ± 0.2**	**2.9 ± 0.1**	**3.0 ± 0.2**
BDD10K (*wroadagents_nodrivable*)		2.3 ± 0.0	**2.2 ± 0.1**	**2.2 ± 0.0**
Cityscapes (**OOD** / *wroadagents_nodrivable*)		**2.6 ± 0.1**	**2.5 ± 0.1**	**2.5 ± 0.1**

	CO2				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	6.9 ± 0.2	7.3 ± 0.4	7.6 ± 0.5	9.9 ± 0.4	15.7 ± 0.7
BDDIntersected (*wroadagents_noabstract*)	3.7 ± 0.1	3.7 ± 0.1	3.7 ± 0.1	5.0 ± 0.3	8.3 ± 0.2
BDD10K (*wroadagents_nodrivable_noabstract*)	2.9 ± 0.1	2.8 ± 0.1	3.1 ± 0.2	4.0 ± 0.2	7.3 ± 0.3
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	3.5 ± 0.2	3.3 ± 0.1	3.6 ± 0.3	4.3 ± 0.2	7.6 ± 0.6

	CSNP				
	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	7.1 ± 0.4	7.0 ± 0.1	7.2 ± 0.5	**6.6 ± 0.3**	**6.3 ± 0.2**
BDDIntersected (*wroadagents_noabstract*)	3.5 ± 0.2	3.7 ± 0.1	3.7 ± 0.2	**3.5 ± 0.2**	**3.4 ± 0.1**
BDD10K (*wroadagents_nodrivable_noabstract*)	3.1 ± 0.1	2.8 ± 0.2	2.9 ± 0.1	2.8 ± 0.1	**2.6 ± 0.0**
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	3.6 ± 0.2	**3.2 ± 0.3**	3.3 ± 0.1	3.1 ± 0.1	**3.0 ± 0.1**

	CSDT2			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	7.2 ± 0.2	7.1 ± 0.4	6.9 ± 0.4	8.4 ± 1.1
BDDIntersected (*wroadagents_noabstract*)	3.7 ± 0.2	3.6 ± 0.2	**3.4 ± 0.2**	4.9 ± 0.9
BDD10K (*wroadagents_nodrivable_noabstract*)	2.9 ± 0.1	3.0 ± 0.1	2.7 ± 0.1	2.9 ± 0.1
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	3.3 ± 0.2	3.4 ± 0.1	**3.1 ± 0.1**	3.1 ± 0.1

	CO2 + CSNP			
	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDDIntersected (*reduced*)	6.5 ± 0.2	7.5 ± 0.2	9.3 ± 0.4	13.5 ± 0.7
BDDIntersected (*wroadagents_noabstract*)	3.8 ± 0.1	3.6 ± 0.1	5.0 ± 0.1	8.0 ± 0.2
BDD10K (*wroadagents_nodrivable_noabstract*)	3.0 ± 0.1	3.1 ± 0.1	4.5 ± 0.1	7.1 ± 0.3
Cityscapes (**OOD** / *wroadagents_nodrivable_noabstract*)	3.3 ± 0.1	3.6 ± 0.2	4.6 ± 0.1	7.4 ± 0.5

Table B.4: Mean absolute error results for the autonomous driving datasets (lower is better). Displayed values are multiplied by 10^2, in order to facilitate analysis. The smaller-sized results in the first table fragment are for the mask setups with abstract classes, which are only able to be used with the ordinal segmentation methods by Fernandes et al. [20]. The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

B.2.1 Dataset Scale Variance

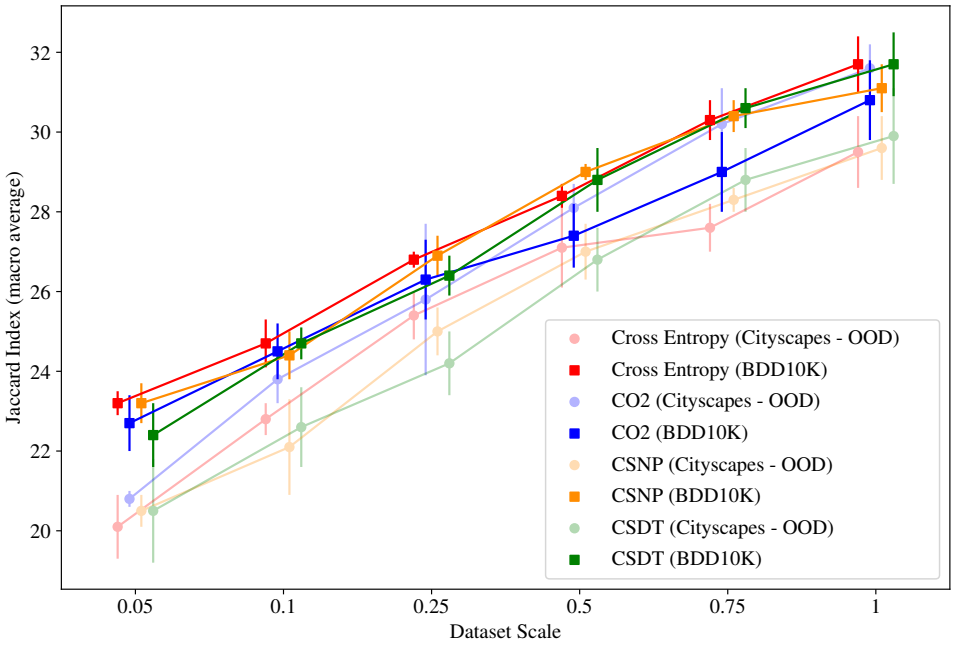


Figure B.5: Jaccard index (macro average) results for the autonomous driving datasets scale variation experiments (higher is better).

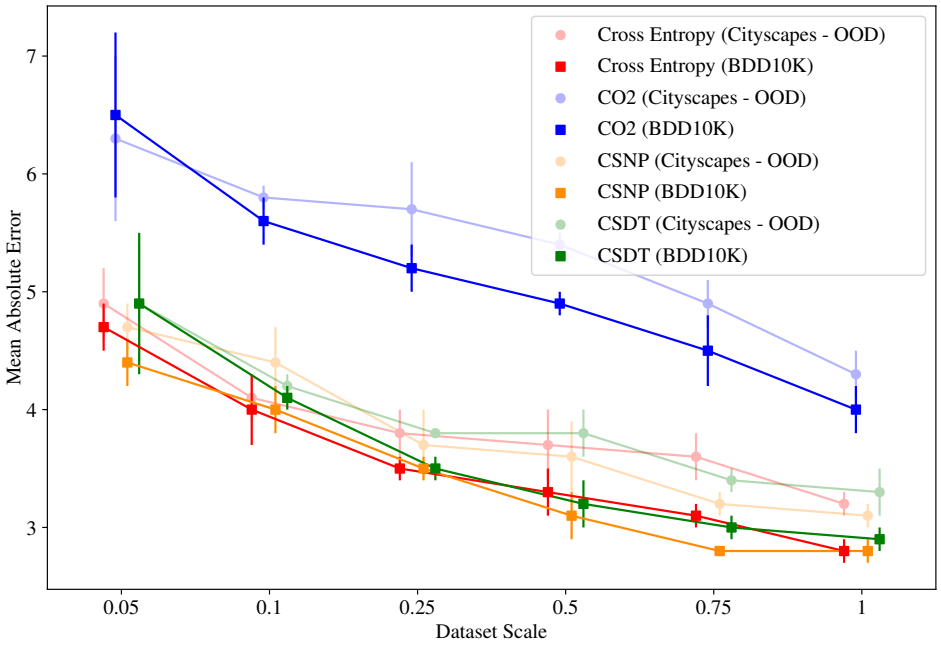


Figure B.6: Mean absolute error results for the autonomous driving datasets scale variation experiments (lower is better).

B.2.2 Semi-Supervised Learning

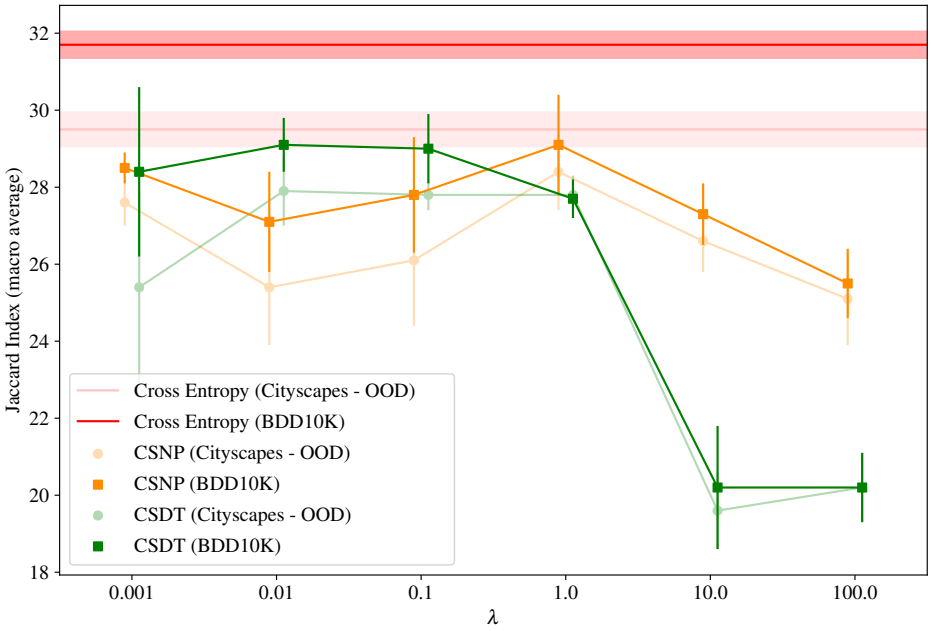


Figure B.7: Jaccard index (macro average) results for the autonomous driving datasets semi-supervised learning experiments (higher is better).

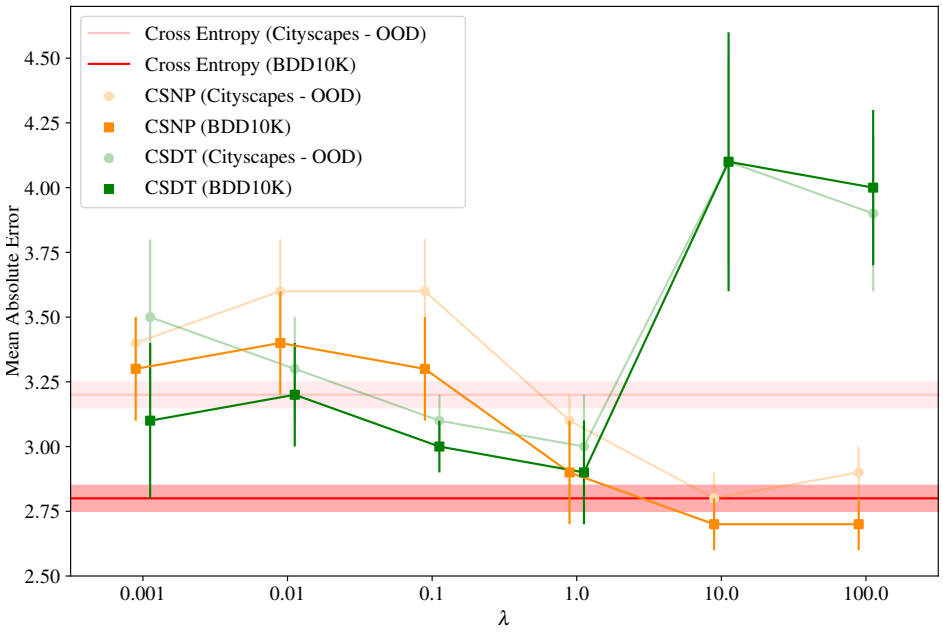


Figure B.8: Mean absolute error results for the autonomous driving datasets semi-supervised learning experiments (lower is better).

	Cross-Entropy
BDD10K (*wroadagents_nodrivable_noabstract*)	$\mathbf{31.7 \pm 0.7}$
Cityscapes (**OOD /** *wroadagents_nodrivable_noabstract*)	$\mathbf{29.5 \pm 0.9}$

	CSNP					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	28.5 ± 0.4	27.1 ± 1.3	27.8 ± 1.5	29.1 ± 1.3	27.3 ± 0.8	25.5 ± 0.9
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	27.6 ± 0.6	25.4 ± 1.5	26.1 ± 1.7	$\mathbf{28.4 \pm 1.0}$	26.6 ± 0.8	25.1 ± 1.2

	CSDT2					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	28.4 ± 2.2	29.1 ± 0.7	29.0 ± 0.9	27.7 ± 0.5	20.2 ± 1.6	20.2 ± 0.9
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	25.4 ± 2.5	27.9 ± 0.9	27.8 ± 0.4	27.8 ± 0.5	19.6 ± 1.0	20.2 ± 0.9

Table B.5: Jaccard index (macro average) results for the autonomous driving datasets semi-supervised learning experiments (higher is better). The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.

	Cross-Entropy
BDD10K (*wroadagents_nodrivable_noabstract*)	2.8 ± 0.1
Cityscapes (**OOD /** *wroadagents_nodrivable_noabstract*)	3.2 ± 0.1

	CSNP					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	3.3 ± 0.2	3.4 ± 0.2	3.3 ± 0.2	2.9 ± 0.2	$\mathbf{2.7 \pm 0.1}$	2.7 ± 0.1
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	3.4 ± 0.1	3.6 ± 0.2	3.6 ± 0.2	3.1 ± 0.1	$\mathbf{2.8 \pm 0.1}$	2.9 ± 0.1

	CSDT2					
	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
BDD10K (**SSL /** *wroadagents_nodrivable_noabstract*)	3.1 ± 0.3	3.2 ± 0.2	3.0 ± 0.1	2.9 ± 0.2	4.1 ± 0.5	4.0 ± 0.3
Cityscapes (**OOD / SSL /** *wroadagents_nodrivable_noabstract*)	3.5 ± 0.3	3.3 ± 0.2	3.1 ± 0.1	3.0 ± 0.2	4.1 ± 0.4	3.9 ± 0.3

Table B.6: Mean absolute error results for the autonomous driving datasets semi-supervised learning experiments (lower is better). Displayed values are multiplied by 10^2, in order to facilitate analysis. The values highlighted in **bold** are the best-achieved results for the dataset in the corresponding line, as determined through Welch's t-test.