Efficient Estimation and Exploitation of Predictive Uncertainties in Deep Learning-based Machine Vision

Steven Landgraf

Doctoral Thesis
Karlsruhe, 2025

Efficient Estimation and Exploitation of Predictive Uncertainties in Deep Learning-based Machine Vision

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für
Bauingenieur-, Geo- und Umweltwissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Steven Landgraf
aus Karlsruhe

Tag der mündlichen Prüfung:	03.07.2025
Referent:	Prof. Dr.-Ing. Markus Ulrich Institut für Photogrammetrie und Fernerkundung Karlsruher Institut für Technologie
Korreferent:	Prof. Dr. rer. nat. Martin Breunig Geodätisches Institut Karlsruher Institut für Technologie

Karlsruhe (2025)

Steven Landgraf
Efficient Estimation and Exploitation of Predictive Uncertainties
in Deep Learning-based Machine Vision
Doctoral Thesis
Date of examination: 03.07.2025
Referees:
Prof. Dr.-Ing. Markus Ulrich
Prof. Dr. rer. nat. Martin Breunig

Karlsruhe Institute of Technology
Department of Civil Engineering, Geo and Environmental Sciences
Institute of Photogrammetry and Remote Sensing
Kaiserstr. 12
76131 Karlsruhe

Abstract

Deep neural networks have emerged as the cornerstone of machine vision tasks, powering remarkable advancements in domains like autonomous driving, medical imaging, and industrial inspection by leveraging their ability to extract hierarchical representations from vast datasets. These unparalleled capabilities have widened the gap between traditional methods and deep learning, cementing their widespread adoption. However, their success is tempered by significant limitations like overconfidence, poor interpretability, and susceptibility to domain shifts and adversarial attacks. These shortcomings, reminiscent of the Dunning-Kruger effect in human decision-making, pose severe risks in safety-critical applications where erroneous predictions can have dire consequences. Uncertainty quantification has been recognized as a promising strategy to address these issues, enhancing trustworthiness by providing insights into prediction reliability and enabling risk-aware decision-making, such as anticipating failure cases or triggering verification steps. Yet, existing approaches often falter, burdened by high computational costs, intricate design requirements, and potential accuracy trade-offs, rendering them impractical for real-time or resource-constrained settings.

This thesis bridges this gap by developing practical, efficient solutions that balance uncertainty quality and computational feasibility across semantic segmentation, monocular depth estimation, and their joint application in multi-task learning. It systematically evaluates the quality of existing uncertainty quantification methods and develops strategies for efficient uncertainty estimation as well as the exploitation of uncertainties during training to enhance model performance.

More specifically, the following novel contributions are presented: In semantic segmentation, DUDES (**D**eep **U**ncertainty **D**istillation using **E**nsembles for **S**egmentation) employs knowledge distillation, training a lightweight student model to mimic a Deep Ensemble's uncertainty estimates, significantly lowering inference time while preserving quality and robustness – even under domain shifts. U-CE (**U**ncertainty-aware **C**ross-**E**ntropy) further advances this task by weighting the common Cross-Entropy loss with dynamic uncertainty estimates, guiding optimization toward challenging regions and boosting accuracy. For monocular depth estimation, combining several existing uncertainty quantification methods with the DepthAnythingV2 foundation model reveals that fine-tuning based on Gaussian Negative Log-Likelihood achieves reliable uncertainty estimates without sacrificing efficiency or predictive performance. In the joint task setting, EMUFormer (**E**fficient **M**ulti-task **U**ncertainty Vision Trans**former**) builds on the previous findings and combines uncertainty distillation and fine-tuning based on Gaussian Negative Log-Likelihood to achieve state-of-the-art results on the widely recognized Cityscapes and NYUv2 datasets, two critical benchmarks for evaluating semantic segmentation and depth estimation in urban driving and indoor scene understanding, respectively. Additionally, this approach delivers reliable uncertainties for both tasks that are comparable or superior to a Deep Ensemble, despite being an order of magnitude more efficient.

Ultimately, these findings demonstrate that uncertainty quantification can be both computationally viable and a driver of enhanced model performance, advancing uncertainty-aware machine vision. This work paves the way for future research that views uncertainty not as a mere auxiliary output but as a core component of machine vision systems.

Kurzfassung

Neuronale Netze haben sich durch bemerkenswerte Fortschritte im autonomen Fahren, der medizinischen Bildgebung und der industriellen Inspektion als Grundpfeiler der maschinellen Bildverarbeitung etabliert. Dies gelingt durch die Extraktion hierarchischer Repräsentationen aus zunehmend größeren Datensätzen. Ihr Erfolg wird jedoch durch erhebliche Einschränkungen wie Selbstüberschätzung, mangelnde Interpretierbarkeit und Anfälligkeit für Domänenverschiebungen und adversariale Angriffe getrübt. Diese Mängel, die an den Dunning-Kruger-Effekt in menschlichen Entscheidungen erinnern, stellen in sicherheitskritischen Anwendungen, in denen fehlerhafte Vorhersagen schwerwiegende Folgen haben können, ernsthafte Risiken dar. Die Quantifizierung von Unsicherheiten gilt als vielversprechende Strategie, um diese Probleme anzugehen, indem sie die Vertrauenswürdigkeit erhöht, Einblicke in die Zuverlässigkeit von Vorhersagen bietet und risikobewusstes Entscheiden ermöglicht, etwa durch das Aufdecken von Fehleinschätzungen oder das Auslösen von zusätzlichen Verifikationsschritten. Bestehende Ansätze scheitern jedoch häufig an ihren hohen Rechenanforderungen, komplexen Designanforderungen und möglichen Genauigkeitseinbußen, was sie für Echtzeit- oder ressourcenbeschränkte Anwendungen unpraktisch macht.

Diese Arbeit schließt diese Lücke, indem sie praktische, effiziente Lösungen entwickelt, die Unsicherheitsqualität und rechentechnische Umsetzbarkeit bei semantischer Segmentierung, monokularer Tiefenschätzung und deren gemeinsamer Anwendung ausbalancieren. Sie bewertet systematisch die Qualität bestehender Unsicherheitsquantifizierungsmethoden und entwickelt Strategien zur effizienten Unsicherheitsschätzung sowie zur Ausnutzung von Unsicherheiten während des Trainings, um die Modellleistung zu steigern.

Genauer gesagt werden die folgenden neuen Beiträge präsentiert: Bei der semantischen Segmentierung verwendet DUDES (**D**eep **U**ncertainty **D**istillation using **E**nsembles for **S**egmentation) Wissensdestillation, um ein effizientes Modell zu trainieren, das die Unsicherheitsschätzungen eines rechenintensiven Deep Ensembles nachahmt, wodurch die Inferenzzeit erheblich reduziert wird, während Qualität und Robustheit – selbst bei Domänenverschiebungen – erhalten bleiben. U-CE (**U**ncertainty-aware **C**ross-**E**ntropy) verbessert diese Aufgabe weiter, indem es die gängige Cross-Entropy Verlustfunktion dynamisch mit Unsicherheitsschätzungen gewichtet, um die Optimierung auf schwierige Bereiche zu lenken und so die Genauigkeit zu steigern. Bei der monokularen Tiefenschätzung zeigt die Synthese einer Reihe bestehender Methoden zur Quantifizierung von Unsicherheiten mit dem DepthAnythingV2 Foundation Modell, dass ein Fine-Tuning basierend auf Gaussian Negative Log-Likelihood zuverlässige Unsicherheitsschätzungen ohne Einbußen bei der Effizienz oder Vorhersageleistung ermöglicht. Im gemeinsamen Aufgabenkontext baut EMUFormer (**E**fficient **M**ulti-task **U**ncertainty Vision Trans**former**) auf den vorherigen Erkenntnissen auf und kombiniert Unsicherheitsdestillation und Fine-Tuning mithilfe von Gaussian Negative Log-Likelihood, um auf den weithin anerkannten Datensätzen Cityscapes und NYUv2 einen neuen Stand der Technik zu erzielen. Zudem liefert dieser Ansatz zuverlässige Unsicherheiten für beide Aufgaben, die einem Deep Ensemble mindestens gleichkommen und deren Quantifizierung eine Größenordnung effizienter ist.

Diese Ergebnisse zeigen, dass Unsicherheitsquantifizierung sowohl effizient implementierbar als auch ein Treiber für verbesserte Modellleistung sein kann und die unsicherheitsbewusste Bildverarbeitung voranbringen. Diese Arbeit ebnet den Weg für zukünftige Forschung, die Unsicherheit nicht nur als Nebenprodukt, sondern als zentralen Bestandteil von Bildverarbeitungssystemen betrachtet.

Dedication

To our beloved dog, who not only brought immense joy into our everyday life but also helped shape the man that I am today. Your presence is dearly missed, and this work is lovingly dedicated to your memory.

Acknowledgements

I am deeply grateful to everyone who supported me on this journey.

First and foremost, I am sincerely grateful to my supervisor, Markus Ulrich, for his guidance, trust, and ongoing encouragement. Beyond his academic mentorship, which played a crucial role in shaping this research, I greatly appreciate him as a person. His humility, integrity, and sincerity make him not only an outstanding scholar but also a truly admirable individual.

I would also like to thank Rongjun Qin, who generously hosted me during my research stay at the Ohio State University. His expertise, openness, and hospitality created an intellectually stimulating and welcoming environment that greatly enriched my academic work. More than that, this stay abroad had a profound personal impact and became one of the most formative experiences of my life.

Special thanks also go to Markus Hillemann, whose support, insights, and eagerness to engage in technical and conceptual discussions have been invaluable throughout all stages of this journey.

I am also grateful to the many collaborators – fellow researchers, colleagues, and the dedicated students I had the privilege to supervise. Your contributions, feedback, and novel perspectives helped refine my thinking and added great depth to this work.

To my family, especially my parents, Michael and Katrin Landgraf, thank you for your unwavering support, encouragement, and love. Your belief in me has been a source of strength throughout my entire life. I consider myself truly blessed to have such amazing parents.

Finally, to my partner Karo – you are truly my better half. You don't just complement me; you complete me in ways I never thought possible. Since the day we met, my life has been brighter and more meaningful in every way. Your unwavering support, love, and belief in me make even the toughest challenges feel achievable, and I am certain they will continue to guide and uplift me on our road ahead.

Contents

1 Introduction

> *The best way to predict the future is to create it.*
>
> — **Abraham Lincoln**
> (16th U.S. President)

This thesis, titled "Efficient Estimation and Exploitation of Predictive Uncertainties in Deep Learning-based Machine Vision", revolves around two key themes: Enabling efficient uncertainty quantification and leveraging predictive uncertainties to enhance the performance of Deep Learning (DL)-based machine vision models. With this thesis, we[1] advocate for a shift toward uncertainty-aware machine vision that integrates uncertainty as a core component of Deep Neural Networks (DNNs).

In the following sections, we will motivate our work and define the scope of this thesis. Finally, this Chapter outlines the structure of the thesis and provides reading guidelines for potential readers who may not be interested in reading the entirety of it.

1.1 Motivation

In recent years, DNNs have emerged as the predominant solution for fundamental machine vision tasks, such as Semantic Segmentation (SS) or Monocular Depth Estimation (MDE). These methods showcase unparalleled performance, enabling impressive advancements in applications such as autonomous driving [60, 35, 7, 168, 178], medical imaging [193, 230, 239, 137, 220, 155], and industrial inspection [253, 99]. Their ability to learn hierarchical representations from vast amounts of data has led to an ever-growing gap in performance between traditional approaches and DL [139], resulting in widespread adoption across various domains [144]. However, this success is shadowed by several critical limitations. Similar to how humans often make poor decisions and reach erroneous conclusions while overestimating their abilities – a phenomenon known as the Dunning-Kruger effect [135] – DL models frequently exhibit overconfidence in scenarios where their predictions are unreliable [84, 277]. Additionally, they suffer from a lack of interpretability [75], inability to handle out-of-domain (OOD) samples [154, 187] or domain shifts [204], and a sensitivity to adversarial attacks [219, 241, 249]. Needless to say, these issues are particularly detrimental in safety-critical applications, where the consequences of erroneous predictions can be severe.

Quantifying the uncertainty of a model's predictions has been identified as a promising approach to not only mitigate these risks but ultimately increase trustworthiness in high-stakes scenarios [147, 155, 154, 190, 191, 143, 168]. By providing insights into where a model is uncertain, Uncertainty Quantification (UQ) can enable risk-aware decision-making, such as

[1] Although I am the sole author of this thesis, which represents a selected part of my research over the past 3.5 years, I will use the plural form throughout for two reasons: It is common in scientific writing, and it acknowledges the indispensable contributions of my co-authors.

preemptive identification of failure cases, providing feedback to human operators, or triggering additional verification steps [193, 60, 67]. Furthermore, uncertainty estimates can be crucial in adapting to heterogeneous and limited data, mitigating the risk of overfitting and facilitating more robust generalization [233, 74]. Besides, they also enable advanced learning paradigms such as active learning [68, 33, 298, 198], where the uncertainty can guide sample selection for labeling, and reinforcement learning [67, 111, 122, 171], where it aids in safer exploration and policy refinement. Despite numerous promising attempts [172, 67, 138, 263, 265, 164, 191, 6], existing methods often suffer from high computational costs, the need for careful design choices, changes in the training process, or a potential deterioration in prediction accuracy [97, 189]. These drawbacks make it undeniably difficult to develop and deploy uncertainty-aware machine vision systems in real-time and resource-constrained applications.

This thesis aims to bridge this gap by exploiting UQ methods in the context of SS and MDE, both as standalone tasks and jointly within multi-task learning. Specifically, it investigates the quality of existing UQ methods, explores strategies for efficient UQ, and integrates uncertainty estimates into the training process to enhance model performance. Moreover, it examines the trade-offs between UQ quality and computational efficiency, to develop practical solutions for real-world applications. By addressing these challenges, this work contributes to the development of safer and more robust machine vision systems, ultimately advancing their reliable deployment in high-stakes environments.

1.2 Scope

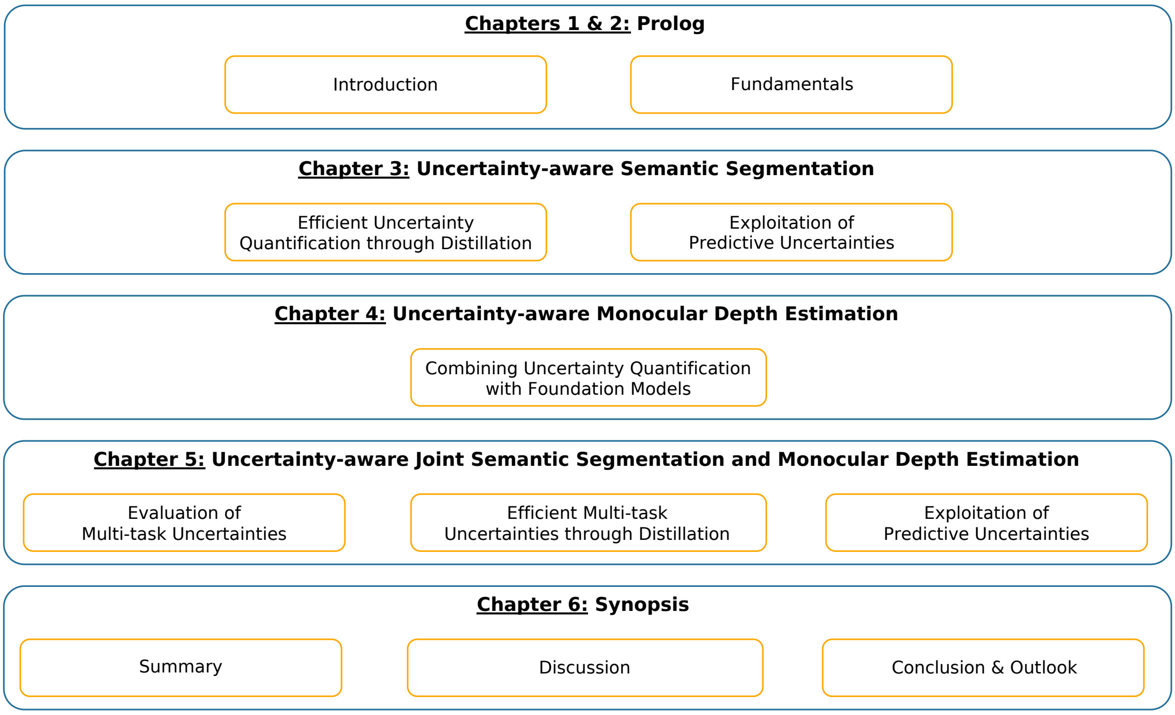

Chapters 1 & 2: Prolog
Introduction
Fundamentals
Chapter 3: Uncertainty-aware Semantic Segmentation
Efficient Uncertainty Quantification through Distillation
Exploitation of Predictive Uncertainties
Chapter 4: Uncertainty-aware Monocular Depth Estimation
Combining Uncertainty Quantification with Foundation Models
Chapter 5: Uncertainty-aware Joint Semantic Segmentation and Monocular Depth Estimation
Evaluation of Multi-task Uncertainties
Efficient Multi-task Uncertainties through Distillation
Exploitation of Predictive Uncertainties
Chapter 6: Synopsis
Summary
Discussion
Conclusion & Outlook

Figure 1.1: Schematic overview of the structure of this thesis.

The scope of this thesis encompasses the exploration and development of methodologies to integrate UQ into DL-based machine vision systems, with the overarching goal of enhancing their robustness and trustworthiness while keeping efficiency in mind for real-world applications. As illustrated by Figure 1.1, the focus of this thesis is structured around UQ in three key tasks: SS, MDE, and their joint application within a multi-task learning framework. Each Chapter tackles specific challenges and opportunities for the exploitation of UQ, contributing to the broader objective of creating safer and more efficient machine vision systems.

Uncertainty-aware Semantic Segmentation. Chapter 3 deals with the foundational task in SS, where pixel-wise classification enables a granular understanding of images through the distinct separation of different object classes and background regions. However, real-world applications do not only demand models that perform accurately but also provide meaningful confidences for their predictions. In this context, this thesis presents two distinct contributions:

1. **Efficient Uncertainty Quantification through Distillation:** Given the high computational cost of traditional UQ methods, this work explores a simple yet highly effective distillation strategy to reduce the inference overhead while maintaining uncertainty quality and predictive performance.
2. **Exploitation of Predictive Uncertainties:** By emphasizing regions of high uncertainty in the training process, valuable insights are leveraged to guide the model's learning more effectively.

Uncertainty-aware Monocular Depth Estimation. Chapter 4 investigates MDE, which describes the task of inferring depth information from a single image and is critical for applications such as autonomous navigation and 3D scene understanding. Foundation models, characterized by their ability to generalize across a wide range of tasks due to extensive pre-training on large datasets, have shown remarkable capabilities in this domain. However, their impressive performance can often lead to naive deployment, overlooking critical aspects such as the predictive uncertainty. To address this gap in the current literature, this thesis contributes:

3. **Combining Uncertainty Quantification with Foundation Models:** By integrating multiple UQ methods with a state-of-the-art foundation model, this work seeks to balance the benefits of large-scale pre-training with the need for reliable uncertainty estimates.

Uncertainty-aware Joint Semantic Segmentation and Monocular Depth Estimation. Since many real-world applications are multi-modal in nature, Chapter 5 combines SS and MDE within a multi-task learning framework, which introduces unique challenges and opportunities for the exploitation of uncertainties. Consequently, in this context, this thesis puts forward three contributions:

4. **Evaluation of Multi-task Uncertainties:** By evaluating multiple UQ approaches within a multi-task learning framework, this work provides a novel perspective on the interplay between tasks with regard to the uncertainty quality and predictive performance.

5. **Efficient Multi-task Uncertainties through Distillation:** Building on the principles of knowledge distillation, this work develops a highly effective strategy to enable high-quality multi-task uncertainties with a single forward pass.
6. **Exploitation of Predictive Uncertainties:** By incorporating predictive uncertainties of a Deep Ensemble (DE) teacher into the training process of a single student model, this work achieves new state-of-the-art results on two common benchmark datasets.

1.3 Structure and Reading Guidelines

As shown by Figure 1.1, this thesis consists of three parts: The prolog, the main contributions, and the synopsis. The remainder of the prolog contains the introduction of essential mathematical foundations of DL in Chapter 2, offering a concise yet thorough primer on DL-based machine vision for the subsequent Chapters. It also provides an overview of UQ in the context of DL, establishing common challenges, the importance of UQ, types of uncertainty, and a comprehensive taxonomy of UQ methods. Finally, it presents a compact review of available knowledge distillation techniques. This Chapter serves as a stepping stone for readers unfamiliar with the mathematical underpinnings of DL-based machine vision or UQ, ensuring both accessibility and laying the groundwork for the technical depth to follow.

Chapters 3, 4, and 5 represent the main contributions of this thesis, each focusing on a specific task within the broader scope of machine vision: SS, MDE, and their joint application within a multi-task learning framework. These Chapters share a coherent structure, each comprising an introduction, a review of related work, a detailed explanation of the proposed methodologies, a comprehensive set of experiments, and a conclusion that reflects on the findings and their implications. Theoretically, each Chapter is self-reliant and can be read independently of the remaining ones for readers who are only interested in a certain task. Readers who are more generally interested are encouraged to read all of the Chapters, as they all provide unique insights and therefore a broader perspective on efficient estimation and exploitation of uncertainties in DL-based machine vision.

Finally, Chapter 6 synthesizes the findings from all preceding Chapters, offering a holistic discussion of the implications of this uncertainty-aware machine vision research. This Chapter further provides concluding remarks encompassing key insights of Chapters 3, 4, and 5 and outlines future research opportunities.

Throughout this thesis, previously published contributions are marked by vertical lines, as exemplified here, to distinguish them from newly written content. Aside from minor editorial changes to correct errors, improve readability, or ensure consistency with the terminology and formatting of this thesis, they are adopted from their original publication without changes. In some cases, parts may be omitted or the order slightly adjusted to enhance the reading flow and coherence within the Chapter. Readers seeking the full details and contexts of these prior works are encouraged to refer to the original publications directly. It is also worth noting that while Chapters 3, 4, and 5 are presented in an order reflecting the logical development of the underlying research, the contents of Chapters 4 and 5 were largely studied concurrently rather than sequentially. Consequently, insights gained from Chapter 5, particularly regarding the choice of the uncertainty threshold, are already applied in Chapter 4. To contextualize the presented research, the following provides a concise overview of all adopted publications.

[147] S. Landgraf, K. Wursthorn, M. Hillemann, and M. Ulrich. "DUDES: Deep Uncertainty Distillation Using Ensembles for Semantic Segmentation". In: *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 92.2 (2024), pp. 101–114.

[143] S. Landgraf, M. Hillemann, K. Wursthorn, and M. Ulrich. "Uncertainty-aware Cross-Entropy for Semantic Segmentation". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 10 (2024), pp. 129–136.

[146][a] S. Landgraf, R. Qin, and M. Ulrich. "A Critical Synthesis of Uncertainty Quantification and Foundation Models in Monocular Depth Estimation". In: *arXiv preprint arXiv:2501.08188* (2025).

[141][b] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich. "A Comparative Study on Multi-task Uncertainty Quantification in Semantic Segmentation and Monocular Depth Estimation". In: *tm-Technisches Messen* (2025).

[142][c] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich. "Efficient Multi-task Uncertainties for Joint Semantic Segmentation and Monocular Depth Estimation". In: *DAGM German Conference on Pattern Recognition*. Springer. 2024, pp. 348–364.

[a] This work is still under review for publication in Pattern Recognition, with the preprint available on arXiv as cited.

[b] This paper represents an extended version of a previously published conference paper [140].

[c] This thesis adopts an extended version of this work, which received a Best Paper Honorable Mention at the German Conference on Pattern Recognition 2024 and was invited for submission to the International Journal of Computer Vision, where it is currently under review.

2 Fundamentals

This Chapter includes elements from

[147] S. Landgraf, K. Wursthorn, M. Hillemann, and M. Ulrich. “DUDES: Deep Uncertainty Distillation Using Ensembles for Semantic Segmentation”. In: *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 92.2 (2024), pp. 101–114,

which are marked with a blue line.

This Chapter also includes elements from

[146] S. Landgraf, R. Qin, and M. Ulrich. “A Critical Synthesis of Uncertainty Quantification and Foundation Models in Monocular Depth Estimation”. In: *arXiv preprint arXiv:2501.08188* (2025),

which are marked with an orange line.

This Chapter also includes elements from

[142] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich. “Efficient Multi-task Uncertainties for Joint Semantic Segmentation and Monocular Depth Estimation”. In: *DAGM German Conference on Pattern Recognition*. Springer. 2024, pp. 348–364,

which are marked with a green line.

The following Chapter introduces the fundamental concepts that underpin this thesis. Section 2.1 covers the mathematical foundations of Deep Learning (DL), from simple feed-forward neural networks to more complex architectures like Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs). Section 2.2 provides an overview of DL-based Uncertainty Quantification (UQ), highlighting key challenges, the importance of uncertainties, their categorization, and methods for quantifying them. Finally, Section 2.3 presents a compact review of knowledge distillation techniques.

2.1 Deep Learning

DL is a subset of Machine Learning and can be viewed as a hierarchy of parameterized basis functions, forming what we call neural networks. Each layer serves as a building block, combining simple functions to handle complex data transformations. While modularity makes DL widely applicable, scaling – through large datasets and computational resources – unlocks its full potential. As a result, DL has achieved unparalleled performance not only in image recognition [134, 59, 260, 258] but also in speech recognition [184, 101, 43], natural language processing [38, 118, 257], brain circuit reconstruction [100], particle accelerator data analysis [37], and genomic research [156, 205, 285].

Mathematical Foundations. Linear regression [73, 72] plays a crucial role in understanding the mathematical underpinnings of DL. It models the relationship between input-output pairs $\{(x_1, y_1), \dots, (x_N, y_N)\}$, where each input $x_i \in \mathbb{R}^M$ and corresponding output $y_i \in \mathbb{R}^D$, using a linear function

$$\hat{y}(x) = Wx + b \ , \tag{2.1}$$

where $W \in \mathbb{R}^{D \times M}$ and $b \in \mathbb{R}^D$. The parameters W and b define different kinds of linear transformations on the input x, and their optimal values are typically determined by minimizing a cost function that accounts for all the inputs X and corresponding outputs Y, which represent the ground truth targets.

However, real-world data often requires non-linear mappings. Linear basis function regression [18] solves this problem. Here, we consider linear combinations of F non-linear transformations $\psi_f(x)$ of the input x. Then, linear regression is performed on the feature vector $\Psi(x) = [\psi_1(x), \dots, \psi_F(x)]$ itself instead of x. The basis functions $\psi_f(x)$ can take various forms, such as polynomials, wavelets, or sinusoidal functions of different frequencies [18].

To allow even more flexibility, we can use parameterized basis functions [18, 66]. Instead of constraining the basis functions to be fixed and mutually orthogonal, we may define them to be ψ^{w_f, b_f}, where ψ_f is applied to the affine transformation $\langle w_f, x \rangle + b_f$, with $\langle w_f, x \rangle$ denoting the inner product between w_f and x. For instance, choosing $\psi_f(\cdot) = \sin(\cdot)$ results in a parameterized basis function

$$\psi_f^{w_f, b_f}(x) = \sin(\langle w_f, x \rangle + b_f) \ , \tag{2.2}$$

where w_f and b_f are learnable parameters representing the weights and biases for the f-th basis function. The feature vector, which consists of the outputs of these basis functions, serves as the input to a linear transformation.

As a result, the model output can be expressed as

$$\hat{y}(x) = W_2 \Psi^{W_1, b_1}(x) + b_2 \ , \tag{2.3}$$

where $\Psi^{W_1, b_1}(x) = \psi(W_1 x + b_1)$, with $W_1 \in \mathbb{R}^{F \times M}$ as a weight matrix, b_1 a vector of F elements, $W_2 \in \mathbb{R}^{D \times F}$ another weight matrix, and b_2 a vector of D elements. These parameters define a structured mapping from the input space to the output space, where $\psi(\cdot)$ introduces non-linearity. Analogous to the simple linear regression, the learnable parameters W_1, b_1, W_2, and b_2 can then be optimized by minimizing a cost function that accounts for all inputs X and corresponding ground truth data Y. For instance, minimizing the average squared error over Y is a common choice.

Combining these simple basis functions as layers inside a hierarchical neural network gives us the foundation to build specialized DL models. For regression tasks, we simply stack these layers, while for classification, we typically apply a logistic function at the end to convert the outputs into a pseudo-probability vector. The logistic function is a common choice for binary classification as it maps the model's outputs to $[0, 1]$. In contrast, the softmax function generalizes this to multi-class settings by ensuring the output forms a valid pseudo-probability distribution. Because of its importance, the softmax function and its properties are discussed in more detail in the following paragraphs.

Next, a simple neural network is introduced to connect common notations in DL with the mathematical formalism of the aforementioned linear basis function models. This is followed by a review of several specialized models for image processing.

Feed-forward Neural Networks. For the sake of simplicity, and because the extension to multiple layers is straightforward, we will only go over a simple feed-forward neural network [179, 227, 226, 232] with a single hidden layer.

Let the input to the model be denoted as $x \in \mathbb{R}^M$, often referred to as the network's input layer. The input undergoes an affine transformation, producing a hidden layer of size F, represented as

$$h = \phi(W_1 x + b_1) \ , \tag{2.4}$$

where $W_1 \in \mathbb{R}^{F \times M}$ is a weight matrix, $b_1 \in \mathbb{R}^F$ is a bias, and $\phi(\cdot)$ is an element-wise non-linearity such as Rectified Linear Unit (ReLU) [65, 64, 194]. The hidden layer is then followed by a second affine transformation to produce the final model output

$$\hat{y}(x) = hW_2 + b_2 = \phi(W_1 x + b_1)W_2 + b_2 \ , \tag{2.5}$$

where $W_2 \in \mathbb{R}^{D \times F}$ and $b_2 \in \mathbb{R}^D$ define the weights and bias of the output layer that map the hidden layer to a D-dimensional output.

For regression tasks, the model can be trained by minimizing the Mean Squared Error (MSE) loss

$$\mathcal{L}_{\mathrm{MSE}}(X, Y) = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2 \ , \tag{2.6}$$

where $\{y_1, ..., y_N\}$ are ground truth outputs, and $\{\hat{y}_1, ..., \hat{y}_N\}$ are the outputs of the model with corresponding inputs $\{x_1, ..., x_N\}$. N denotes the total number of samples in the available dataset.

For multi-class classification, the output logits $\hat{y} \in \mathbb{R}^D$ are transformed via the softmax function to produce normalized pseudo-probabilities

$$p(\hat{y})_i = \frac{\exp(\hat{y}_i)}{\sum_{c=1}^{C} \exp(\hat{y}_c)} \ , \tag{2.7}$$

where $p(\hat{y})_i$ denotes the predicted score for class i, and C is the number of classes. In this setting, the output dimensionality D corresponds to the number of classes, i.e., $D = C$. Although the outputs lie in the interval $[0, 1]$ and sum to one, they are not calibrated, and thus, are more accurately viewed as pseudo-probabilities.

Training for classification often involves minimizing the categorical Cross-Entropy (CE) loss

$$\mathcal{L}_{\mathrm{CE}}(X, Y) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \log(p(\hat{y})_{n,c}) \ . \tag{2.8}$$

The general idea of training a model, i.e., minimizing its loss function concerning W_1, W_2, b_1, and b_2 through backpropagation [98], is that it generalizes to unseen test data (X_{test}, Y_{test}). A major challenge with such an approach is usually overfitting, where the loss decreases on the training data (X, Y) but increases on unseen test data (X_{test}, Y_{test}).

A common solution to address overfitting is the incorporation of a regularization term. For instance, L2 regularization (or weight decay) penalizes large weights by adding the squared norms of the parameters to the loss, weighted by regularization coefficients $\lambda_1, \lambda_2, \lambda_3$, resulting in

$$\mathcal{L}_{\text{regularized}}(W_1, W_2, b) = \mathcal{L}(X, Y) + \lambda_1 \|W_1\|_F^2 + \lambda_2 \|W_2\|_F^2 + \lambda_3 \|b\|_2^2 \ , \tag{2.9}$$

where $\mathcal{L}(X, Y)$ represents the primary loss (e.g. MSE or CE), while the additional terms serve to constrain the magnitude of the weights and biases. $\|\cdot\|_F$ denotes the Frobenius norm [77]. A high regularization rate tends to produce a histogram of model weights that is akin to a normal distribution and a mean weight of 0, thereby reducing the chances of overfitting and improving the model's generalization to unseen data [78, 136].

As discussed earlier, these simple model architectures can be extended to multiple layers, which leads to models that are capable of capturing hierarchical representations of highly complex input data, such as images, which we will review next.

Convolutional Neural Networks (CNNs). CNNs [149, 150] are inspired by the natural visual perception mechanisms of living organisms, particularly the hierarchical structure of receptive fields in the visual cortex [113]. They are designed to efficiently process grid-like data, such as images, by recursively applying convolutions and pooling layers in combination with simple feed-forward neural networks.

The convolutional layer aims to extract meaningful feature representations by applying a set of learnable kernels over local regions in the input, as illustrated by Figure 2.1. As shown in this simplified example, the first kernel preserves the information from the blue channel of the input image while it discards the green and red channels, resulting in blue features in the layer output. Conversely, the second kernel disregards the blue and green channels and only retains information from the red channel of the input image, leading to red features in the output layer. Obviously, this is a simplified portrayal of how convolutional layers work. In reality, kernels are usually not composed of the same values in each spatial location and do not only retain information of a single channel, but rather act as feature detectors of all kinds.

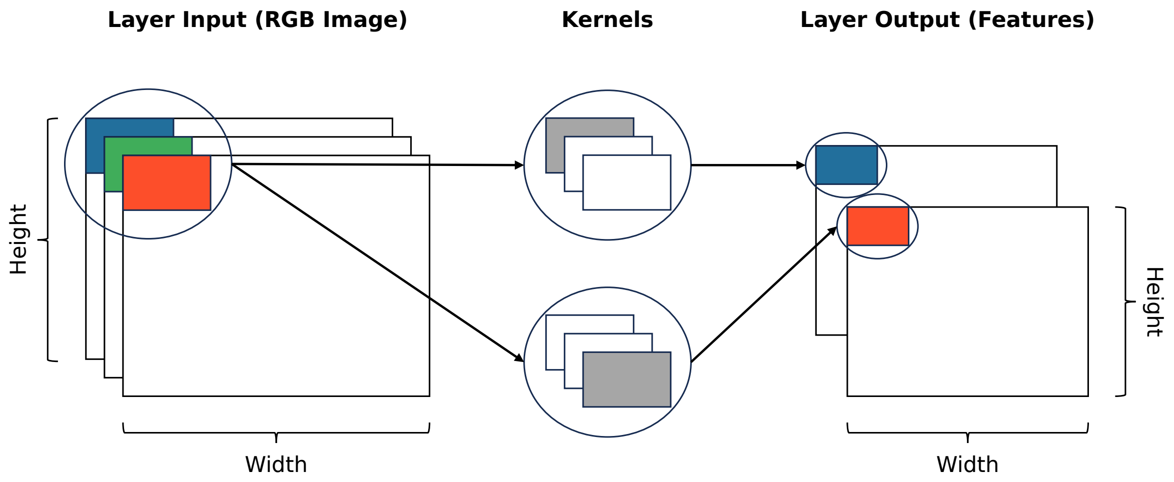


Figure 2.1: Simplified illustration of a convolutional layer. Inspired by [66].

Mathematically, the feature value at location (i, j) in the f-th feature map of the l-th layer is calculated as

$$z^l_{i,j,f} = \phi(W^l_f x^l_{i,j} + b^l_f) \ , \tag{2.10}$$

where W^l_f is the respective weight, b^l_f is the corresponding bias term, $x^l_{i,j}$ is the input patch centered at location (i, j), and $\phi(\cdot)$ is an activation function such as ReLU to introduce non-linearity. It is worth noting that the learnable kernel W^l_f is shared for the entire input, which allows CNNs to drastically reduce the number of parameters compared to a fully connected network, while maintaining the ability to detect patterns such as edges, textures, and even more abstract features at higher layers.

The output of a convolutional layer typically passes through a pooling operation (e.g., average pooling [149, 150] or max pooling [218, 117]), which downsamples the spatial resolution to reduce computational cost and improve robustness by preserving important information while discarding irrelevant details. Stacking multiple convolutional and pooling layers enables CNNs to efficiently learn hierarchical feature representations, where earlier layers capture low-level features (e.g., edges and corners) and deeper layers extract high-level semantic features (e.g., object parts and categories). Finally, these features are usually fed into one or multiple fully-connected layers, which aim to perform high-level reasoning by combining all the features from the previous layer to generate global semantic information [246, 297].

CNNs have been crucial in achieving state-of-the-art performance in image classification, object detection, and segmentation tasks. For example, the seminal AlexNet [134] architecture demonstrated that deep CNNs, combined with large-scale datasets such as ImageNet [47], could outperform traditional machine vision techniques by a groundbreaking margin. Subsequent architectures like VGGNet [246], Inception [258], and ResNet [95] introduced further innovations in terms of layer design and network scaling.

Despite their success, CNNs inherently rely on inductive biases such as local receptive fields and spatial hierarchies, which can limit their ability to model long-range dependencies and global relationships. Recently, inspired by the revolutionary success of relying entirely on self-attention in natural language processing [268], ViTs [53] have emerged as a promising alternative to CNNs.

Vision Transformers (ViTs). ViTs adopt the transformer architecture, originally designed for sequence-to-sequence tasks in natural language processing [268], to machine vision. Unlike CNNs, which operate on local regions of an image with their convolutional kernels, ViTs leverage self-attention mechanisms to capture global dependencies of image patches that are treated as sequences – just like tokens (i.e., words) in natural language processing applications [53].

In ViTs, an image is first divided into fixed-size patches, which are then linearly embedded and sequentially concatenated with positional encodings to preserve spatial information. These embedded image patch sequences are then fed into a series of transformer blocks, each consisting of multi-head self-attention and feed-forward neural networks. The self-attention mechanism at each layer is mathematically defined as:

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \ , \tag{2.11}$$

where $Q, K, V \in \mathbb{R}^{n \times d_k}$ are the query, key, and value matrices derived from n input embeddings. d_k is the dimension of the key vectors. The scaling factor $\frac{1}{\sqrt{d_k}}$ is used to prevent the dot products from growing too large in magnitude, which could lead to small gradients and unstable training dynamics.

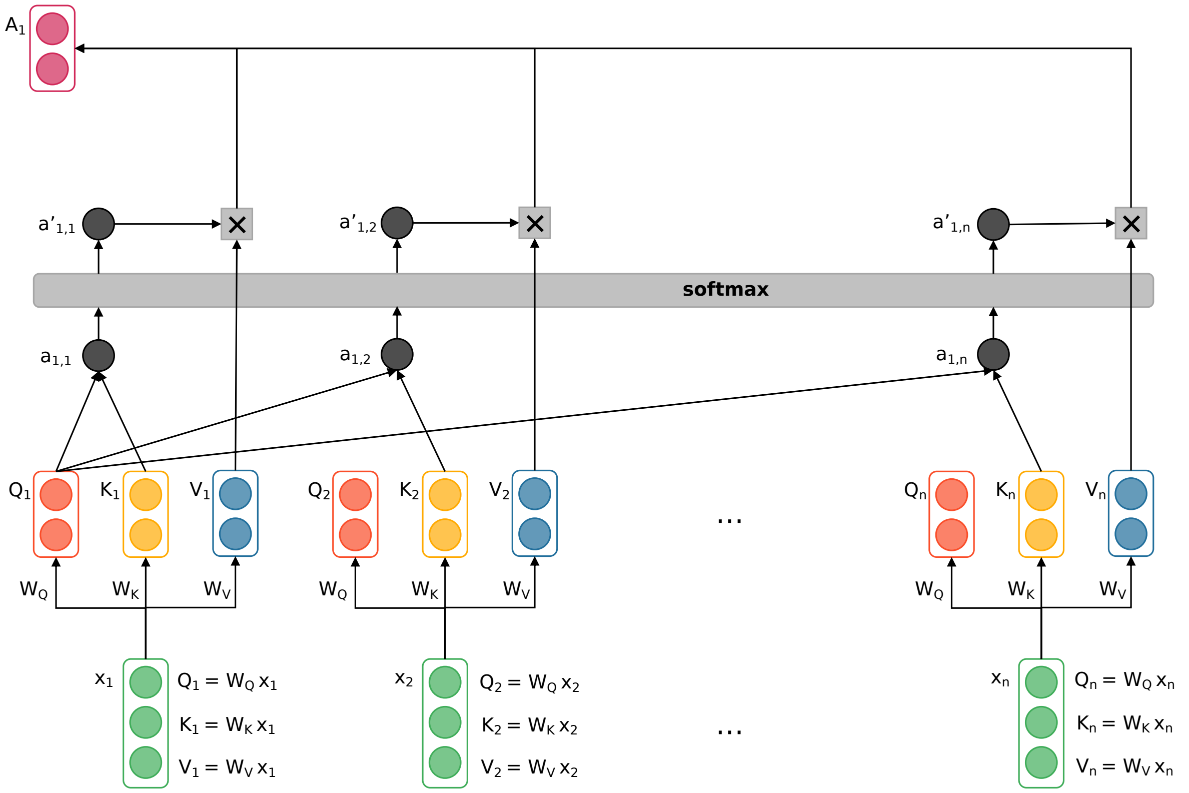


Figure 2.2: Simplified illustration of the single-head self-attention mechanism. Inspired by [152].

Figure 2.2 shows how the single-head self-attention mechanism works for a given input patch x_i with a total number of n input patches. First, the input is multiplied by the query, key, and value weight matrices W_Q, W_K, and W_V, respectively. During training, these weight matrices are learned. Afterwards, a similarity between each query Q_i and every key K_j is computed through a simple dot product, effectively determining the level of attention $a_{i,j}$ given to other parts of the input x_j when encoding an input x_i at a specific position. For example, as shown by Figure 2.2, for the first input x_1, a total number of n attention scores $a_{1,j}$ for $(j = 1, \ldots, n)$ are computed by estimating the dot product of the first query Q_1 and all the keys K_i. Subsequently, a softmax function (cf. Equation 2.7) is utilized to obtain normalized attention scores $a'_{i,j}$. Finally, the values V_i are multiplied by the normalized attention scores $a'_{i,j}$ and summed across all available positions to obtain a final self-attention representation A_i for each input x_i. For multi-head self-attention, the input is projected into multiple subspaces, using separate weight matrices W_Q, W_K, and W_V for each head, and the self-attention operation is performed independently in each subspace. The outputs of all attention heads are then concatenated and linearly transformed to produce the final output. This formulation allows ViTs to dynamically focus on the most relevant parts of the image while capturing long-range dependencies and relationships between patches, regardless of their spatial proximity. Self-attention is a key factor in the success of ViTs, as it enables to adaptively determine the importance of different image regions for any given task.

Although ViTs lack key inductive biases inherent to CNNs, such as translation equivariance and locality, their global attention mechanism enables them to capture complex, long-range patterns that span the entire image. This holistic understanding allows ViTs to outperform CNNs on various vision tasks. However, without the strong inductive biases that aid CNNs in generalizing from limited data, ViTs require extensive pre-training and more computational resources, making them less efficient in data-scarce settings [53, 90].

2.2 Uncertainty Quantification

Deep Neural Networks (DNNs) have revolutionized numerous research fields, including computer vision, natural language processing, and robotics, due to their remarkable ability to learn complex patterns from large datasets. This success has catalyzed the adoption of DNNs in high-risk applications, such as medical image analysis [193, 230, 239, 137, 220, 155] and autonomous driving [60, 35, 7, 168, 178]. However, despite their impressive capabilities, the real-world deployment of DNNs in mission- and safety-critical domains remains constrained.

Challenges in Safety-Critical Deployment. The limitations of DNNs in critical applications primarily stems from the following factors:

1. **Lack of Interpretability:** DNNs are often perceived as "black boxes", providing predictions without intuitive explanations [75]. This opaqueness hinders trust, especially in scenarios where understanding the rationale behind a decision is vital, such as a medical diagnosis. Moreover, non-existent decision-making processes undermine the collaboration between DNN-based systems and human experts, particularly in interdisciplinary domains like healthcare [155, 9] or industrial monitoring [145, 139, 144].

2. **Inability to Handle out-of-domain (OOD) Samples and Domain Shifts:** DNNs struggle to differentiate between in-domain (ID) and OOD samples [154, 187]. Additionally, they exhibit sensitivity to domain shifts [204], which can lead to significant performance degradation. This issue is exacerbated in dynamic environments, such as autonomous driving, where real-world conditions frequently diverge from training data assumptions [235, 109].

3. **Overconfidence in Predictions:** DNNs often produce overconfident outputs, even for incorrect predictions [84, 277]. This lack of calibration can be detrimental in high-risk settings where reliable confidence estimates are essential [9, 67, 84, 125, 126, 138]. In mission- or safety-critical contexts, such as industrial inspection [253] or medical diagnostics [180], deploying uncalibrated models may lead to unanticipated failures, despite seemingly high-confidence predictions.

4. **Sensitivity to Adversarial Attacks:** Adversarial perturbations – small, imperceptible changes to input data – can dramatically affect the output of a model [219, 241, 249]. For instance, in autonomous driving [24, 48], a carefully crafted perturbation to a traffic sign could cause a vehicle's DNN to misclassify a "STOP" sign as a regular speed limit sign, leading to potentially catastrophic consequences.

These challenges highlight the critical need to not only focus on improved performance of these systems but also advancements in terms of reliability and interpretability. Without addressing these concerns, deploying DNNs in safety-critical environments could lead to unintended risks and consequences, undermining their potential benefits and trust.

Importance of Uncertainty Quantification. Incorporating UQ into DNNs addresses many of the previously mentioned challenges: Beyond tackling issues such as overconfidence and sensitivity to domain shifts, UQ significantly improves the deployability of DNNs by dealing with their lack of interpretability. Providing uncertainties can enhance decision-making processes and enable broader, more reliable adoption in diverse applications:

- **Ensuring Safety in High-Risk Applications:** As mentioned before, reliable uncertainty estimates are indispensable in many real-world applications. Identifying high-uncertainty predictions can trigger human intervention or additional verification steps [193, 60, 67]. Uncertainty-aware systems can also enable adaptive responses to changing conditions or unexpected scenarios like adversarial attacks or OOD samples.
- **Adapting to Heterogeneous and Limited Data:** In domains like remote sensing, where data sources are highly diverse and labeled datasets are scarce, UQ helps mitigate the risks of overfitting and enables more robust generalization [233, 74]. Similarly, in industrial settings, where precision and reliability are critical, UQ allows for more informed decision-making [253, 262]. UQ also facilitates model transferability across domains, reducing the reliance on extensive domain-specific fine-tuning [209].
- **Enabling Uncertainty-Driven Learning Techniques:** Uncertainty estimates are integral to advanced learning paradigms such as active learning, where they guide the selection of the most informative samples for labeling [68, 33, 298, 198], and reinforcement learning, where they contribute to safer exploration and policy improvement [67, 111, 122, 171]. These techniques leverage uncertainty to maximize efficiency, which is particularly valuable in scenarios with resource constraints.

Systematically integrating UQ into DNN workflows offers researchers and practitioners the chance to overcome key barriers to the adoption of artificial intelligence in mission- and safety-critical domains. Uncertainty-aware systems have the potential to not only enhance reliability and explainability but also bridge the gap between experimental research and real-world deployment. The role of UQ is pivotal in shaping the future of trustworthy artificial intelligence.

Uncertainty Quantification Methods. DNNs, with their large number of parameters and inherent non-linearities, make exact posterior probability estimation of outputs intractable [19, 168]. To address this critical limitation, a variety of approximative UQ methods have been proposed [172, 67, 138, 263, 265, 164, 191, 6], which can be categorized by architectural design [75], the type of uncertainty they aim to quantify [97], or their underlying Bayesian principles [1, 181]. Since the scope of this thesis encompasses the exploration and development of practical solutions for efficient estimation and exploitation of uncertainties with real-world machine vision applications in mind, we disregard the type of uncertainty for now and adopt a mixture of the remaining taxonomies, leading to the following:

1. **Sampling-based Methods:** Sampling-based approaches are the most prominent UQ methods due to their ease of use and effectiveness. These methods either derive the uncertainty from the model's parameters (e.g., Bayesian Neural Networks) [67, 195, 279, 236, 221, 222, 133, 214, 170] or their architectures (e.g., Ensembling) [175, 138, 263, 276]. While they usually produce the most accurate uncertainty estimates, the computational cost associated with the necessity of multiple forward passes often makes them impractical for real-world applications that require fast inference times or that are running on resource-constrained devices. One of the simplest and most prevalent sampling-based UQ methods is Monte Carlo Dropout (MCD) [67]. MCD approximates a Gaussian process by keeping dropout layers active during both training and testing. Originally, dropout layers were solely introduced as a regularization technique during training to prevent overfitting [251]. With MCD, however, they are also used during test time to turn a deterministic model into a stochastic one to sample from the posterior distribution. The uncertainty of a given model can then easily be estimated by calculating the standard deviation (or variance) over the samples. Another widespread sampling-based UQ method are Deep Ensembles (DEs) [138], which are generally considered the state of the art for UQ across various tasks [204, 282, 87, 140]. They consist of a collection of independently trained models, ideally each initialized with random weights and optimized with random data augmentations to maximize the diversity among the ensemble members [61]. The high effectiveness of DEs comes at the cost of a very high computational overhead due to the need to train and evaluate multiple models. A more efficient variant of DEs are Deep Sub-Ensembles (DSEs) [263]. They exploit subnetworks within a single model to produce diverse predictions without the need for a full ensemble of models. They offer an effective trade-off between uncertainty quality and computational cost, which can easily be tuned based on the given constraints.

2. **Deterministic Approaches:** Next to these approximate UQ methods, there has also been an increasing interest in using deterministic single forward-pass methods, which need less memory and have a lower inference time. For instance, Van Amersfoort et al. [265] and Liu et al. [164] build on the idea of a well-regularized feature space in which they quantify the uncertainty through distance-aware output layers. Although these methods perform well, they are not quite competitive with DEs and require a substantial adaptation of the training process. Mukhoti et al. [191] propose to simplify the aforementioned approaches by using Gaussian Discriminant Analysis post-training for feature-space density estimation. With their approach, they manage to perform on par with a DE in some settings but still require a more sophisticated training approach. In general, these deterministic single-forward pass methods are a worthwhile alternative to the traditional UQ methods [172, 67, 138], yet they all introduce conceptual complexity that require changes in the architecture, the training process, and introduce additional hyperparameters. Another common approach for more efficient UQ is evidential DL [10, 26, 173, 240, 6], which only requires a single forward pass during inference. Detrimentally, training is more complicated, requires OOD samples, and there is no guarantee that the prediction accuracy of a regular network can be achieved [97, 189].

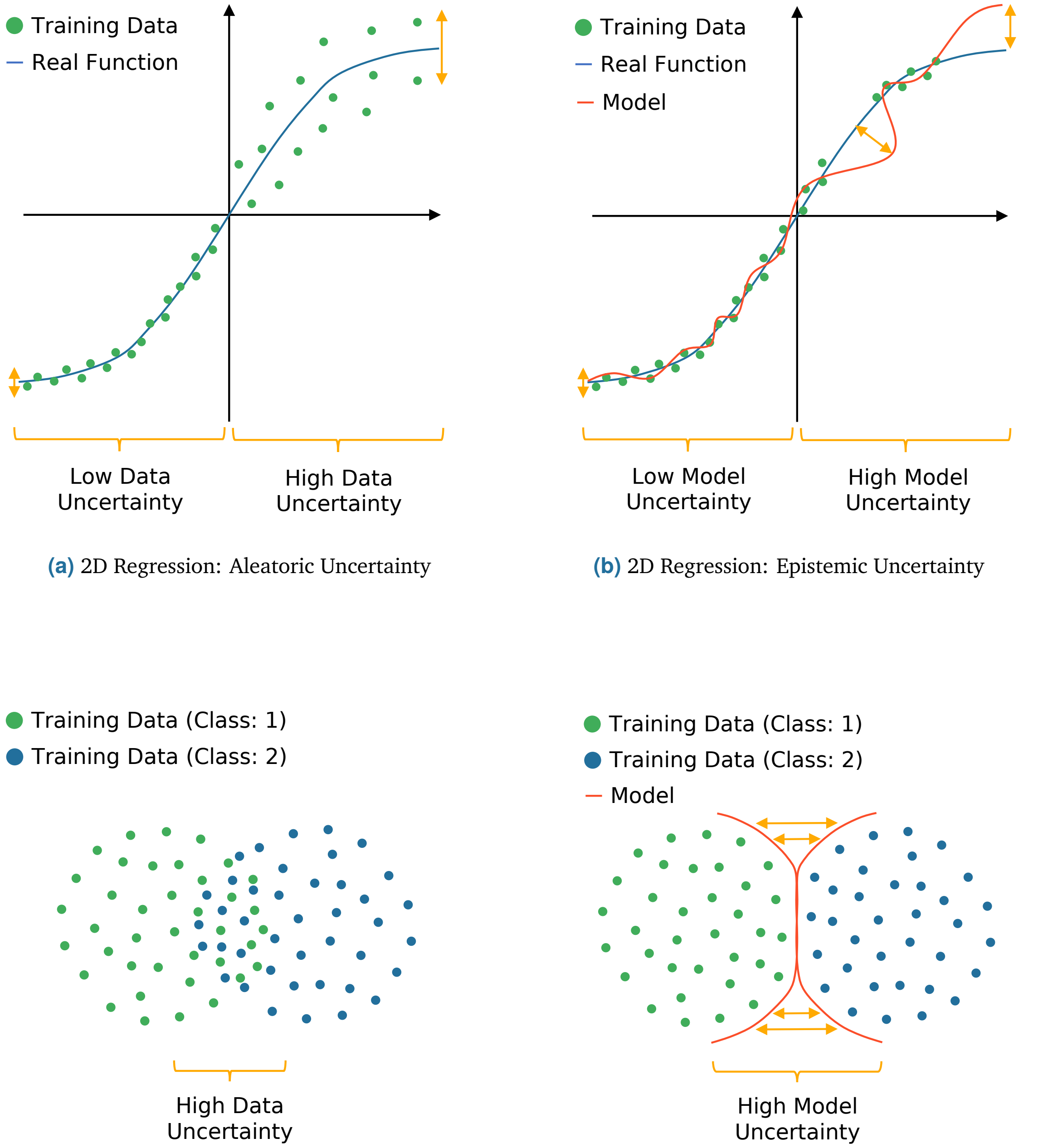
Types of Uncertainty. While not the primary focus of this thesis, uncertainty in DL is typically categorized into aleatoric and epistemic components [50], following the taxonomy originally introduced by Hacking [88]:

1. **Aleatoric uncertainty** (often referred to as data uncertainty) arises from inherent randomness or variability in the data, such as noise in measurements or ambiguities in labels, as shown by the left side of Figure 2.3. This type of uncertainty is intrinsic to the data-generating process and cannot be reduced by collecting more data[1]. For instance, the ambiguity around object boundaries due to image resolution or overlapping structures can lead to high aleatoric uncertainty. Aleatoric uncertainty can be further divided into homoscedastic, which remains constant across inputs, and heteroscedastic, which varies depending on the properties of the input [126]. UQ methods that aim to explain the uncertainty from the inherent randomness or noise in the data either construct a distribution over the prediction via discriminative models [82, 126, 208, 280, 228, 28] or generative models [202, 71, 215, 131, 55, 25]. While these methods allow for the quantification of the aleatoric uncertainty, in principle, they depend on new training strategies, need to modify the network's architecture, or convergence of the loss function is not guaranteed [97].

2. **Epistemic uncertainty** (often referred to as model uncertainty) reflects a lack of knowledge about the model or its parameters, as shown by the right side of Figure 2.3. This uncertainty often stems from insufficient data, limited diversity in training samples, or structural simplifications in the model. Unlike aleatoric uncertainty, epistemic uncertainty is reducible by improving the model or incorporating more representative training data. Epistemic uncertainty is particularly evident in OOD scenarios, where the test data distribution differs from the training distribution. Epistemic uncertainty can be quantified based on methods that target the parameters of a DNN (e.g., Bayesian Neural Networks) [67, 195, 279, 236, 221, 222, 133, 214, 170], architecture (e.g, Ensembling) [175, 138, 263, 276], or sample density [265, 264, 164, 44, 278]. Although these methods can capture valuable model uncertainty information, they generally suffer from high computational cost, lack rigorous theoretical analysis, or require careful design choices and modifications to the training process [97]

Predictive Uncertainty. While decomposing uncertainty into aleatoric and epistemic components can benefit specific applications, such as active learning [66] or out-of-distribution detection [68], this thesis focuses on predictive uncertainty as a combined measure of both types. Predictive uncertainty – i.e., any uncertainty estimate that corresponds to a model's prediction – provides a holistic view of the confidence of the model and is particularly relevant in real-world machine vision applications, where the overall reliability of predictions outweighs the need to distinguish between aleatoric and epistemic uncertainty. While this distinction is addressed shortly in the later parts of the thesis, the general approach of neglecting uncertainty disentanglement aligns with recent findings of Mucsányi et al. [189]. They reveal that, despite substantial theoretical efforts, disentangling these uncertainty types remains an unresolved challenge in practice. Their analysis attributes this difficulty to strong internal correlations and the inherent challenge of clearly separating aleatoric and epistemic components, limiting their practical utility in real-world applications. Consequently, this thesis aims to develop practical solutions that deliver reliable uncertainty estimates – regardless of type.

[1] Although aleatoric uncertainty is often considered *irreducible*, it can technically be *reducible* by increasing the measurement precision, i.e., by upgrading the underlying system, which is responsible for the data-generating process.



(a) 2D Regression: Aleatoric Uncertainty

(b) 2D Regression: Epistemic Uncertainty

(c) Multi-class Classification: Aleatoric Uncertainty

(d) Multi-class Classification: Epistemic Uncertainty

Figure 2.3: Illustration of aleatoric (data) and epistemic (model) uncertainty. The first row portrays the distinction between low and high (a) aleatoric uncertainty and (b) epistemic uncertainty for a two-dimensional function, along with the corresponding training data. The second row extends these concepts to a multi-class scenario. Inspired by [75].

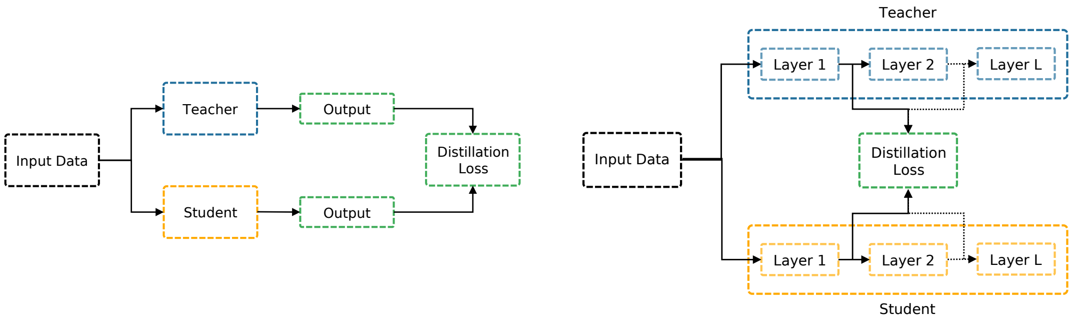
2.3 Knowledge Distillation

As a result of increasingly fast GPUs, ever-growing compute clusters, and techniques like batch normalization [115], it has become possible to train ViTs with up to 22 billion parameters [45], enabling groundbreaking performance. However, the huge computational cost and storage requirements make deployment of such models not only impractical but often impossible in real-time and resource-constrained applications.

To alleviate this challenge, several efficiency techniques have been proposed:

1. **Parameter Pruning and Sharing:** These methods allow for the removal of nonessential parameters from DNNs without significant degradation of the performance. This category can be further divided into model quantization [281], model binarization [112, 40], structural matrices [248], and parameter sharing [91, 210].
2. **Low-rank Factorization:** These methods exploit the redundancy present in DNNs by employing matrix decomposition [49].
3. **Knowledge Distillation:** Knowledge distillation transfers knowledge from a complex teacher model into a smaller student model. The student learns to imitate the teacher's predictions, minimizing output differences. Incorporating the teacher's knowledge typically produces a compact student model with comparable performance [102, 224, 174, 80]

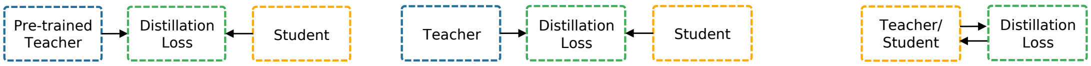
Distillation Techniques. Figure 2.4 illustrates the most relevant knowledge distillation techniques for this thesis, following the taxonomy of Gou et al. [80]. Response-based distillation targets the last output layer of the student-teacher framework, training the student to mimic the teacher model's predictions. Although conceptually simple, this approach is highly effective for model compression and widely used in different tasks and applications due to its computational efficiency. Feature-based distillation extends beyond the final layer by incorporating intermediate layers, using the teacher model's feature maps to supervise the student's training. Typically, this method builds on response-based distillation by directly matching the feature activations of a more capable teacher. It was first introduced by Romero et al. [224] with FitNets, which leveraged intermediate hints to train thinner, deeper student networks.



(a) Response-based distillation

(b) Feature-based distillation

Figure 2.4: Illustration of response-based and feature-based distillation. Inspired by [80].



(a) Offline distillation (b) Online distillation (c) Self-distillation

Figure 2.5: Illustration of offline, online, and self-distillation. Inspired by [80].

Distillation Schemes. Figure 2.5 depicts various distillation schemes, categorized into three types based on whether the teacher model is optimized alongside the student [80]. The most prevalent and straightforward scheme, offline distillation, is inspired by the original method by Hinton et al. [101]. Within offline distillation, the knowledge of a pre-trained teacher is transferred into a student model within a two-stage framework: First, the teacher model is trained on a set of training samples. Afterwards, the pre-trained teacher is utilized to guide the training of the student model. To overcome the sequential nature and inflexibility of this two-stage framework, online distillation optimizes both teacher and student models simultaneously, enabling end-to-end training. Self-distillation, a special case of online distillation, employs the same model as both teacher and student, enhancing regularization. Naturally, these distillation schemes can be combined to leverage their complementary strengths [255].

Uncertainty Distillation. The concept of knowledge distillation has recently gained traction in the context of efficient UQ, aiming to enable real-time estimation of uncertainties [243, 12, 106, 247]. While some previous works employ MCD to estimate uncertainties for the student to learn [243, 86, 12], the majority proposes to use a DE [106, 46, 147, 247, 174]. Among these, Deng et al. [46] are the only ones to consider a multi-task problem in the field of emotion recognition.

3 Uncertainty-aware Semantic Segmentation

This Chapter includes elements from

[147] S. Landgraf, K. Wursthorn, M. Hillemann, and M. Ulrich. "DUDES: Deep Uncertainty Distillation Using Ensembles for Semantic Segmentation". In: *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 92.2 (2024), pp. 101–114,

which are marked with a blue line.

This Chapter also includes elements from

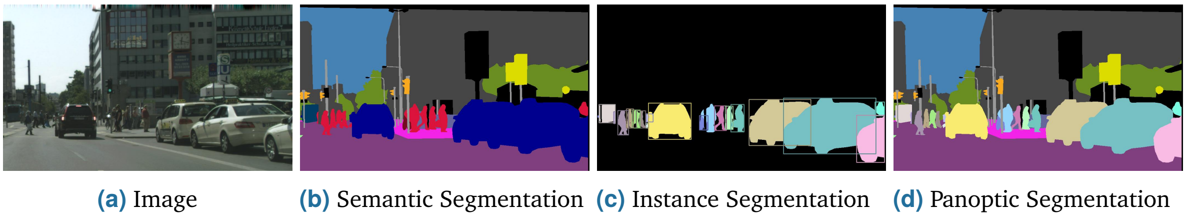
[143] S. Landgraf, M. Hillemann, K. Wursthorn, and M. Ulrich. "Uncertainty-aware Cross-Entropy for Semantic Segmentation". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 10 (2024), pp. 129–136,

which are marked with a orange line.

The following Chapter deals with Uncertainty Quantification (UQ) in Semantic Segmentation (SS), which is one of the most foundational machine vision tasks to enable thorough image understanding by assigning a distinct class label to each pixel.

Segmentation. SS represents the task of assigning a class label to each pixel in an image, effectively grouping all objects of the same category without distinguishing individual instances [167, 185, 92, 148, 157, 85]. In contrast, instance segmentation extends this approach by not only classifying each pixel but also identifying separate instances of the same class [93, 185, 89, 81, 242]. However, this distinction alone does not fully capture complex real-world scenes where both regions (e.g., sky, road) and countable objects (e.g., cars, pedestrians) coexist. To address this, panoptic segmentation unifies both approaches by segmenting each pixel into either a *thing* category (i.e., distinguishable object instances) or a *stuff* category (i.e., homogeneous regions) [129, 36, 158]. Figure 3.1 shows a qualitative comparison of these segmentation tasks, highlighting the differences in how each approach interprets the given image.

Although panoptic segmentation enables a more comprehensive scene understanding, SS is often sufficient for most applications. Similarly, while instance segmentation provides additional object-level information, it fails to capture important regions like roads, which are essential for real-world applications such as autonomous driving. Moreover, SS remains the least complex approach and maintains the lowest computational cost, making it a practical choice for the following uncertainty-aware research.



(a) Image (b) Semantic Segmentation (c) Instance Segmentation (d) Panoptic Segmentation

Figure 3.1: Qualitative comparison of semantic, instance, and panoptic segmentation for a given input image. Taken from Kirillov et al. [129].

Challenges. Despite the remarkable success of Deep Learning (DL) in SS, several critical challenges, which often hinder employment in safety-critical applications, remain as already discussed in Chapter 2.2. In short, neural networks often exhibit overconfidence in their predictions, even for incorrect classifications [84, 277]. Additionally, these models struggle with domain shifts and out-of-domain (OOD) samples, leading to performance degradation in real-world scenarios [154, 187, 204]. Furthermore, the lack of interpretability limits trust, particularly in applications where understanding the decision-making process is indispensable, such as a medical diagnosis [75, 155, 9]. Addressing these issues requires methods that not only improve predictive performance but also provide reliable uncertainty estimates – all without introducing significant computational overhead like most traditional UQ methods do [97, 189].

Research Questions. In an effort to contribute to the advancement of uncertainty-aware SS, the following Chapter investigates two key research questions:

1. How can we enable efficient and reliable UQ in SS while maintaining technical simplicity?
2. How can we exploit uncertainty estimates to guide the optimization process?

In response to the first question, we present a novel approach for efficient and reliable UQ, which we call **D**eep **U**ncertainty **D**istillation using **E**nsembles for **S**egmentation (DUDES), as illustrated by Figure 3.2. DUDES applies student-teacher distillation with a Deep Ensemble (DE) to accurately approximate predictive uncertainties while maintaining simplicity and adaptability. In comparison to the DE teacher, the student only needs a single forward pass to obtain predictive uncertainties, which massively reduces the inference time and eliminates the computational overhead that is associated with having to deal with multiple models and forward passes. DUDES simultaneously simplifies and outperforms previous work on DE-based uncertainty distillation.

Regarding the second question, we present a novel **U**ncertainty-aware **C**ross-**E**ntropy (U-CE) loss that addresses this gap by incorporating dynamic uncertainty estimates into the training process as shown in Figure 3.3. Through pixel-wise uncertainty weighting of the well-known Cross-Entropy (CE) loss, we harness the valuable insights provided by the uncertainties for more effective training. With U-CE, we manage to train models that are naturally capable of predicting meaningful uncertainties after training while simultaneously improving their segmentation performance.

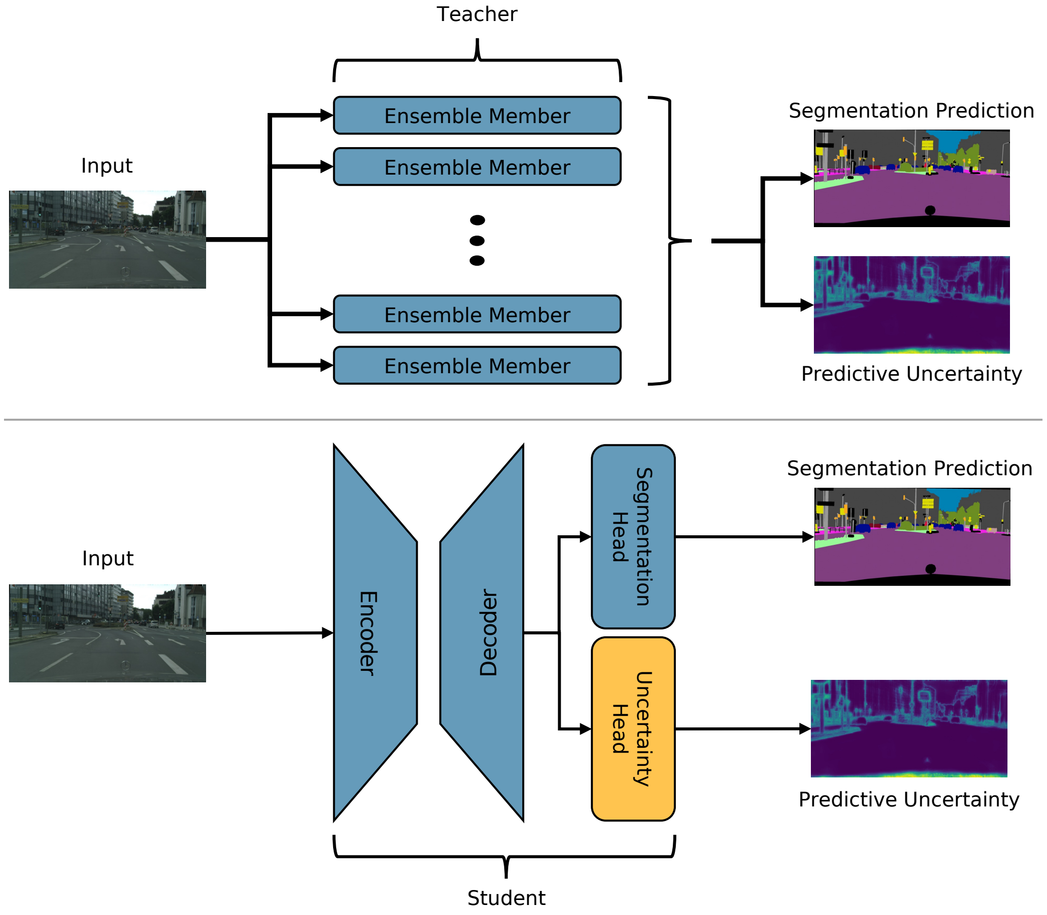


Figure 3.2: DUDES applies student-teacher distillation with a DE to accurately approximate predictive uncertainties with a single forward pass while maintaining simplicity and adaptability.

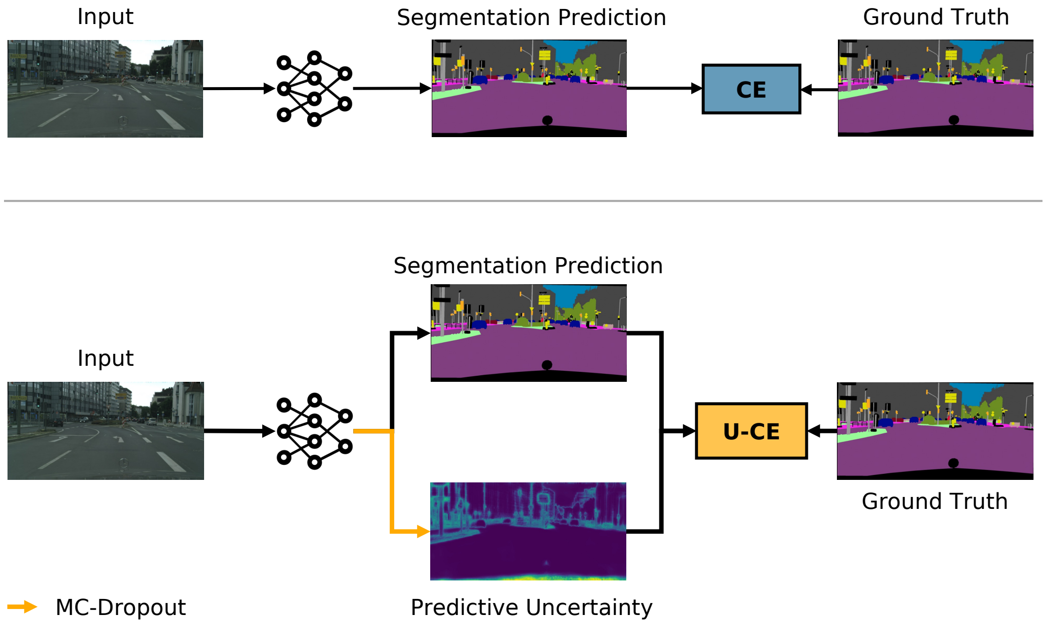


Figure 3.3: U-CE introduces an uncertainty-aware CE loss that dynamically incorporates predictive uncertainties into the training process.

Outline and Structure. This Chapter is structured as follows:

1. Section 3.1 offers an overview of related work on UQ and uncertainty-aware SS.
2. Section 3.2 showcases DUDES, our novel approach for efficient and reliable UQ.
3. Section 3.3 presents U-CE, an uncertainty-aware CE loss function that incorporates predictive uncertainties into training to enhance segmentation performance.
4. Section 3.4 concludes with a summary of key findings and their implications for uncertainty-aware SS.

3.1 Related Work

In this Section, we summarize the related work on UQ and uncertainty-aware SS. Related work on knowledge distillation can be found in Section 2.3.

Uncertainty Quantification

The simplest way to quantify uncertainty in SS is by using the softmax probabilities, which are naturally provided by the design of the model. However, while these softmax predictions are easy to implement, they tend to be overconfident and require calibration to ensure that the predicted pseudo-probabilities align with the actual likelihood, thus providing reliable confidence estimates [84, 277].

For a broader review of related work on UQ, refer to the corresponding Section 2.2 in the fundamentals Chapter.

Uncertainty-aware Segmentation

In the domain of uncertainty-aware segmentation, researchers have explored various techniques to incorporate uncertainty measures into the training process. While traditional UQ methods have successfully been employed in tasks such as visual bias mitigation in classification [254], these techniques have been largely overlooked or underutilized in the field of SS. We provide an overview of notable works that leverage uncertainty-aware techniques for segmentation tasks in various domains. Additionally, we discuss how U-CE addresses the gap towards full utilization of traditional UQ methods during training.

Hard Example Focus. Some of the earlier work on more effective training was originally designed for Object Detection. For example, Lin et al. [162] introduced a loss that down-weights the contribution of easy examples to shift the focus more towards hard examples. Another closely related technique is online hard example mining by Shrivastava et al. [244]. They propose to automatically select hard examples to only learn from them and completely ignore the easy examples. By now, both methods have been successfully adapted for SS [116, 275].

Dataset Balancing. More closely related to our work, Bischke et al. [17] and Bressan et al. [21] propose to leverage uncertainties to improve training on imbalanced aerial image datasets. The former use the per-class uncertainty of the model together with the median frequency to balance training [17]. We argue that dynamically weighting each pixel individually during training, which is what U-CE does, is even more valuable. The latter utilize pixel-wise weights, but only consider the class and labeling uncertainty [21] instead of the predictive uncertainties like U-CE.

Uncertainty-aware Weighting. In addition to these methods, Chen et al. [27] propose to transform the embeddings of the last layer from Euclidean space into Hyperbolic space to dynamically weight pixels based on the hyperbolic distance, which they interpret as uncertainty. Similarly, Bian et al. [15] propose an uncertainty estimation and segmentation module to estimate uncertainties that they use to improve the segmentation performance. Unlike U-CE, however, these two works do not incorporate traditional UQ methods into training.

3.2 Efficient Uncertainty Quantification through Distillation

The following introduces DUDES, a novel student-teacher distillation approach to efficiently approximate predictive uncertainties. By distilling the UQ capabilities of a computationally expensive teacher into a single student model, DUDES significantly reduces inference time while preserving the reliability of the uncertainty estimates of the teacher.

Research Gap. We believe that the process of ensemble-based uncertainty distillation can be improved upon by simplification. Instead of distilling the entire uncertainty map, which is what Holder and Shafique [106] propose, we only consider the uncertainty of the respective predicted class, which we refer to as the predictive uncertainty. This basic, yet highly effective, simplification ensures that the student's segmentation performance is not degraded and the corresponding uncertainties can be learned more easily. As a result, we manage to train a student model that achieves similar or better segmentation performance than the DE teacher and does not suffer from any systematic shortcomings concerning the UQ. Additionally, DUDES does not rely on custom segmentation or uncertainty head architectures and introduces only a single uncertainty loss without hyperparameters. Thereby, we provide a distinct improvement over all of the shortcomings and complexities of previous work.

3.2.1 Methodology

In the following, we provide an overview of DUDES, explain the methodology behind our uncertainty distillation approach, and lay out the implementation details.

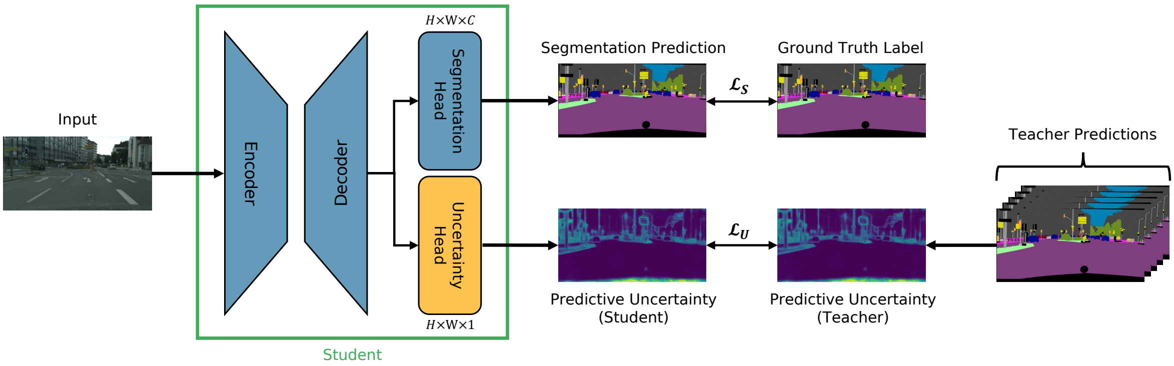


Figure 3.4: A schematic overview of the training process of the student model of DUDES. DUDES is an easy-to-adapt framework for efficiently estimating predictive uncertainty through student-teacher distillation. The student model simultaneously outputs a segmentation prediction alongside a corresponding uncertainty prediction. Training the student involves a regular segmentation loss with the ground truth labels and an additional uncertainty loss. As ground truth uncertainties, we compute the predictive uncertainty of a DE, thereby acting as the teacher.

Overview

DUDES is an easy-to-adapt framework for efficient and reliable UQ through student-teacher distillation. The overall goal is to train a student model that can simultaneously output a segmentation prediction and a corresponding predictive uncertainty in the form of standard deviations that correlate with wrongly classified or OOD pixels with a single forward pass, as shown in Figure 3.2. Although the student and the teacher could be trained jointly, in principle, we propose a two-step framework for the sake of simplicity and computational constraints:

1. Training the teacher with the ground truth labels
2. Training the student with the ground truth labels and the teacher's uncertainty predictions

As shown in Figure 3.4, the training of the student model consists of two loss components. The first component $\mathcal{L}_S$ assesses the dissimilarity between the student's segmentation prediction and the ground truth labels, while the second component $\mathcal{L}_U$ evaluates the disparity between the uncertainty prediction of the student and the output of one of the UQ methods described in Section 2.2. As mentioned before, we propose to use a DE as the teacher for the concrete implementation of DUDES. DEs are simple to implement, easily parallelizable, require little tuning, and represent the current state of the art UQ method [204, 87, 282]. Nevertheless, since DUDES is flexible with regards to the chosen UQ method, the DE can simply be replaced by any other UQ method as long as the resulting uncertainty measure is limited between 0 and 1, which we will show in Section 3.2.3.

Teacher. For the reasons stated above, we use a DE as the teacher. The DE consists of ten regular SS models that are not pre-trained, thus following prior work on DE-based UQ [138, 61]. By randomly initializing all the parameters before training, we aim to capture different aspects of the input data distribution for each ensemble member, boosting the overall performance of the teacher, robustness, and UQ capabilities. During inference, each ensemble member produces slightly different predictions, which can be averaged to obtain a mean softmax pseudo-probability

$$\bar{p}(\hat{y}) = \frac{1}{M} \sum_{m=1}^{M} p(\hat{y}_m) \ , \tag{3.1}$$

where $p(\hat{y}_m) \in \mathbb{R}^C$ is the softmax pseudo-probability vector from the m-th ensemble member, containing C elements corresponding to the number of classes in the SS task, and M is the number of members. Following Holder and Shafique [106], we compute the standard deviation over all the softmax pseudo-probabilities to obtain the teacher's predictive uncertainty

$$\sigma = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(p(\hat{y}_m) - \bar{p}(\hat{y})\right)^2} \ . \tag{3.2}$$

Student. As our student has to output a corresponding predictive uncertainty in addition to the segmentation prediction, we add a second head to the decoder of the segmentation model. We propose to use an additional uncertainty head that is identical to the regular segmentation head of the segmentation model, except for the output layer. For the segmentation head, we use a softmax activation to obtain class-wise pseudo-probabilities. Whereas for the uncertainty head, we use a sigmoid activation that limits the outputs between 0 and 1. Our uncertainty head only needs one output channel instead of the number of classes, as needed by the segmentation head. Since this is a key modification to improve upon previous work, we will discuss this simplification in detail in Section 3.2.4. In contrast to the randomly initialized ensemble members, the parameters of the student are initialized with ImageNet pre-training [47] to drastically reduce the required training time, as shown in Section 3.2.2.

Uncertainty Distillation

To efficiently estimate the predictive uncertainty of the DE with a single student model, we utilize student-teacher distillation as Figure 3.4 shows.

Segmentation Loss. The main objective function that is being minimized for the segmentation task is the well-known categorical CE loss

$$\mathcal{L}_S = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \log\left(p_{n,c}(\hat{y})\right) \ , \tag{3.3}$$

where $\mathcal{L}_S$ is the segmentation loss for a single image, N is the number of pixels in the image, C is the number of classes, $y_{n,c}$ is the one-hot encoded ground truth, and $p_{n,c}(\hat{y})$ is the predicted pseudo-probability from the model output $\hat{y}$. The categorical CE loss measures the dissimilarity between the ground truth probability distribution and the predicted pseudo-probability distribution. By minimizing this loss during training, the model is encouraged to produce pixel-wise class predictions that are as close as possible to the ground truth classes.

Uncertainty Loss. To distill the predictive uncertainties of our teacher into the student, we introduce an additional uncertainty loss, which is formulated as the Root Mean Squared Logarithmic Error (RMSLE)

$$\mathcal{L}_U = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\log(\sigma_n + 1) - \log(\hat{\sigma}_n + 1)\right)^2} \ , \tag{3.4}$$

where $\mathcal{L}_U$ is the uncertainty loss for a single image, N is the number of pixels, $\sigma_n(x)$ is the teacher's uncertainty for the n-th pixel (as ground truth), and $\hat{\sigma}_n(x)$ is the student's predicted uncertainty. The uncertainty of the teacher represents the standard deviation of the softmax pseudo-probabilities of the predicted class in the segmentation map. By minimizing the RMSLE during training, the student is encouraged to produce uncertainty estimates that are as close as possible to those predicted by the teacher. The natural logarithm $log(\cdot)$ provides special attention to the pixels where uncertainties are higher by penalizing underestimations more than overestimations.

Total Loss. The total loss combines the two previous losses as

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_U \ . \tag{3.5}$$

For the sake of simplicity and because of the empirical results, which we will demonstrate in Section 3.2.2 and 3.2.3, we refrain from introducing additional hyperparameters to weight the individual losses. However, it is worth mentioning that, depending on the application, the introduction of weights for the individual loss terms could be valuable.

3.2.2 Experiments

In this Section, we describe a variety of experiments that demonstrate the advantages of DUDES. Firstly, we go over our experimental setup. Secondly, we compare the student and the teacher quantitatively. More specifically, we examine the class-wise segmentation performance as well as the class-wise uncertainties. In addition, we investigate the uncertainty quality and highlight the substantial difference in terms of inference time and trainable parameters between the teacher and the student model. Thirdly, we evaluate the student's predictions qualitatively. Fourthly, we assess how well our student model performs on OOD datasets in comparison to the teacher. Lastly, we provide two ablation studies, which explore the influence of the number of ensemble members and analyze the impact of pre-training.

Experimental Setup

Architecture. For our baseline SS model, we use a DeepLabv3+ [29] as the decoder and a ResNet-18 (RN18) [95] as the backbone because they both are very commonly used architectures for SS. All ensemble members are trained with just the segmentation loss (cf. Equation 3.3) and follow the training procedure of the student with regards to data augmentations and hyperparameters.

Training. To prevent overfitting, we apply the following data augmentation strategy to all training procedures: Random scaling with a scaling factor between 0.5 and 2.0, random cropping with the crop size of 768 × 768, and random horizontal flipping with a flip chance of 50 %. Besides, we employ a Stochastic Gradient Descent optimizer [223] with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 0.0005 as optimizer-specific hyperparameters. In all experiments, the learning rate of the decoder is ten times higher than that of the backbone [284]. Additionally, we use polynomial learning rate scheduling to decay the initial learning rate during the training process:

$$lr = lr_{\text{base}} \cdot \left(1 - \frac{\text{iteration}}{\text{total iterations}}\right)^{0.9} \tag{3.6}$$

where lr is the current learning rate, and lr_{base} is the initial base learning rate. In all training processes, we train for 200 epochs with a batch size of 16 on a NVIDIA A100 GPU. We empirically found this to be sufficient for the models to converge and did not employ any early stopping techniques.

Dataset. Our experiments are based on the Cityscapes dataset [39], a freely available urban street scene dataset. It consists of 2975 training images, 500 validation images, and 1525 test images. Since the test images are not publicly available, we use the validation images for testing in all of our experiments. Each RGB image is 2048 × 1024 pixels in size, with each pixel assigned to one of 19 class labels or a void label. The void ground truth pixels are excluded during training and evaluation in the segmentation task, but they are used to qualitatively evaluate the uncertainty outputs as they indicate the ability of the model to distinguish between in-domain (ID) and OOD samples. Additionally, we test our student model and teacher ensemble on Foggy Cityscapes [235] and Rain Cityscapes [109] to investigate the potential of DUDES for OOD detection.

Metrics. For quantitative evaluations, we primarily report the mean Intersection over Union (mIoU) [148] to measure the quality of the segmentation prediction. In addition, we use the Expected Calibration Error (ECE) [192] to evaluate the calibration of the softmax pseudo-probabilities. Lastly, we report the mean class-wise Predictive Uncertainty (mUnc) [106] to compare the uncertainty of the student with that of the teacher.

Quantitative Evaluation

Tables 3.1 and 3.2 give a detailed quantitative comparison between the student's and the teacher's Intersection over Union (IoU) as well as their predictive uncertainties. The results of Holder and Shafique [106] have been included as they are the most relevant previous work on DE-based student-teacher distillation for efficient UQ. Their teacher is based on 25 DeepLabv3+ models with a MobileNet backbone [108], whereas our teacher consists of 10 DeepLabv3+ models with a RN18 backbone. The MobileNet backbone and our RN18 backbone have been shown to have very similar performance [16]. Both students are initialized with ImageNet pre-training [47] and evaluated on the Cityscapes validation dataset [39].

	Road	Sidewalk	Building	Wall	Fence	Pole	Tr. Light	Tr. Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorbike	Bicycle	Mean
Teacher IoU (Theirs)	0.961	0.794	0.914	0.432	0.563	0.584	0.620	0.730	0.917	0.596	0.937	0.782	0.553	0.935	0.668	0.793	0.677	0.534	0.743	0.723
Student IoU (Theirs)	0.964	0.772	0.900	0.426	0.547	0.476	0.511	0.652	0.904	0.564	0.919	0.739	0.498	0.921	0.617	0.723	0.625	0.493	0.689	0.681
Teacher IoU (Ours)	0.978	0.822	0.907	0.504	0.545	0.549	0.578	0.693	0.915	0.627	0.943	0.754	0.535	0.932	0.696	0.759	0.640	0.476	0.695	0.713
Student IoU (Ours)	0.980	0.835	0.914	0.467	0.557	0.591	0.633	0.733	0.918	0.631	0.942	0.790	0.579	0.939	0.747	0.838	0.694	0.501	0.736	0.738
Difference (Theirs) ↑	**0.003**	-0.022	-0.014	**-0.006**	-0.016	-0.108	-0.109	-0.078	-0.013	-0.032	-0.018	-0.043	-0.055	-0.014	-0.051	-0.070	-0.052	-0.041	-0.054	-0.042
Difference (Ours) ↑	0.002	**0.013**	**0.007**	-0.037	**0.012**	**0.042**	**0.055**	**0.040**	**0.003**	**0.004**	**-0.001**	**0.036**	**0.044**	**0.007**	**0.051**	**0.079**	**0.054**	**0.025**	**0.041**	**0.025**

Table 3.1: Quantitative comparison between the student's and the teacher's class-wise IoU ↑. Results of Holder and Shafique [106] are indicated by 'Theirs'.

	Road	Sidewalk	Building	Wall	Fence	Pole	Tr. Light	Tr. Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorbike	Bicycle	Mean
Teacher Unc. (Theirs)	0.029	0.097	0.055	0.210	0.147	0.100	0.128	0.108	0.028	0.129	0.030	0.068	0.125	0.030	0.176	0.155	0.257	0.165	0.082	0.111
Student Unc. (Theirs)	0.032	0.086	0.077	0.155	0.141	0.133	0.135	0.111	0.055	0.127	0.046	0.097	0.127	0.046	0.108	0.100	0.127	0.135	0.130	0.104
Teacher Unc. (Ours)	0.024	0.064	0.038	0.150	0.165	0.100	0.142	0.102	0.025	0.101	0.031	0.109	0.120	0.043	0.175	0.163	0.195	0.158	0.108	0.106
Student Unc. (Ours)	0.018	0.065	0.035	0.144	0.160	0.128	0.112	0.097	0.027	0.126	0.025	0.117	0.105	0.038	0.200	0.144	0.171	0.190	0.150	0.108
Difference (Theirs) ↓	0.003	-0.011	0.022	-0.055	-0.006	0.033	0.007	0.003	0.027	0.002	0.016	0.029	0.002	0.016	-0.068	-0.055	-0.130	-0.030	0.048	-0.007
Difference (Ours) ↓	-0.006	0.001	-0.003	-0.006	-0.005	0.028	-0.03	-0.005	0.002	0.025	-0.006	0.008	-0.015	-0.005	0.025	-0.019	-0.024	0.032	0.042	0.002

Table 3.2: Quantitative comparison between the student's and the teacher's class-wise predictive uncertainties. For easier interpretation, the differences are highlighted based on the absolute differences being: ≤ 0.01, ≤ 0.02, ≤ 0.03, ≤ 0.04, ≤ 0.05, ≤ 0.06, ≥ 0.06. Results of Holder and Shafique [106] are indicated by 'Theirs'.

Segmentation Prediction. As shown in Table 3.1, our student network outperforms the teacher on the segmentation task for all classes except for *wall* and *sky*, with an average improvement of 0.025 in mIoU. We attribute this improvement to the ImageNet pre-training of the student as compared to the randomly initialized ensemble members of the teacher. In comparison, the student by Holder and Shafique [106] showed a mIoU deterioration of 0.042.

Uncertainty Prediction. Table 3.2 shows that our student approximates the uncertainties of the teacher very accurately: In 10 out of the 19 classes, our student's class-wise uncertainties deviate by less than 0.01 compared to that of the teacher. Our student manages to deviate by less than 0.03 in 17 out of the 19 classes, with a maximum deviation of 0.042 for the *bicycle* class. On the other hand, the student by Holder and Shafique [106] deviates by less than 0.01 in 5 out of the 19 classes and by less than 0.03 in only 13 out of the 19 classes. The maximum difference of their student is 0.130 for the *train* class. On average across all classes, the uncertainties of both students deviate only slightly from those of the teachers, with our student model deviating by 0.002 and the student by Holder and Shafique [106] deviating by -0.007. Both students struggle with accurately approximating the teacher's uncertainties for the classes *truck*, *bus*, *train*, *motorbike*, and *bicycle*, with an average absolute deviation of 0.028 for our student and 0.066 for theirs [106]

Figure 3.5 displays another comparison between the student's and the teacher's ability to approximate reliable uncertainties: For this analysis, we progressively ignored an increasing percentage of uncertain pixels in the segmentation prediction and simultaneously re-evaluated the mIoU. For this, the pixels were sorted based on their predictive uncertainty in descending order. This initially removes the pixels with the most uncertain segmentation predictions from the evaluation until only the pixels with the most certain predictions are left. Consequently, meaningful uncertainties should result in a monotonically increasing function.

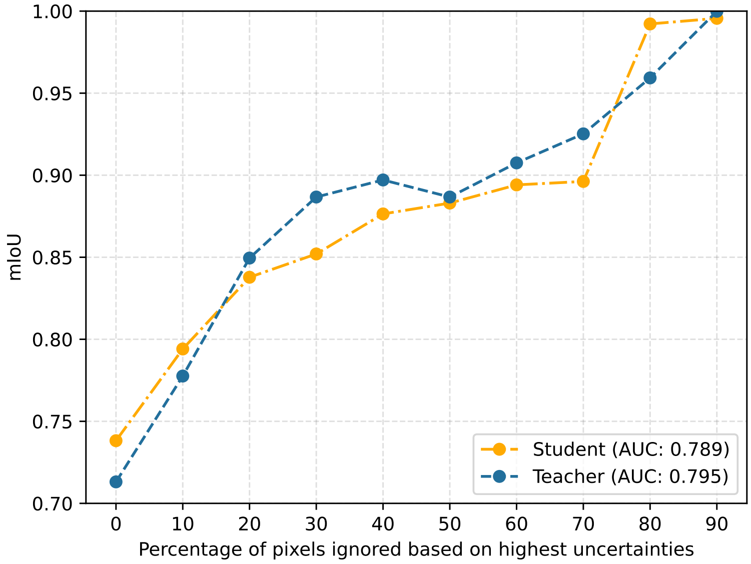


Figure 3.5: Comparison between the student's and the teacher's mIoU. We progressively ignore an increasing percentage of pixels in the segmentation prediction and simultaneously re-evaluate the mIoU. The pixels are sorted based on their predictive uncertainty in descending order, thus removing the most uncertain segmentation predictions first. Additionally, we report the Area Under the Curve (AUC).

As Figure 3.5 shows, the student as well as the teacher experience an almost linear rise in mIoU from 0.738 and 0.713, respectively, to almost 1.0 after removing 90 % of the most uncertain pixels. Both models attain a similar relative increase in mIoU by disregarding the first 10 % of the most uncertain pixels. Up until ignoring 70 % of the pixels, the teacher reaches a mIoU of 0.925, while the student only attains 0.896. Beyond this point, the student's mIoU surpasses that of the teacher, with the student achieving 0.992 after ignoring 80 % of the pixels with the highest uncertainties, while the teacher only reaches 0.959. This analysis yields two key findings: Firstly, predictive uncertainties prove to be related to the correctness of the prediction and hence provide an effective approach for identifying misclassified pixels. Secondly, our student's predictive uncertainties deviate only slightly from the uncertainties of the teacher, revealing that they are equally meaningful, as proven by the AUC of 0.789 for our student and 0.795 for the teacher.

Inference Time. Table 3.3 compares the inference time for a single image and the number of trainable parameters between the baseline, the teacher, and the student model. The experiment was conducted on a common NVIDIA GeForce RTX 3090 GPU with 24GB of memory. There is only an insignificant difference of 0.2 milliseconds in inference time between the baseline and the student, despite the ability of the student to output an additional predictive uncertainty. Furthermore, inference of the student is roughly 11.7 times faster than that of the teacher. The number of trainable parameters shows the efficiency of the student network. The additional uncertainty head of the student network only adds 257 parameters to the baseline model.

	Inference time [ms]	Trainable Parameters
Baseline	18.3 ± 0.4	12,333,923
Teacher	217.1 ± 0.8	123,339,230
Student	18.5 ± 0.4	12,334,180

Table 3.3: Comparison of the inference time for a single image in milliseconds and the number of trainable parameters between the baseline, the teacher, and the student model. The inference time and corresponding standard deviation are based on 25 forward passes.

Qualitative Evaluation

Figure 3.6 displays four example images from the Cityscapes validation set and their corresponding ground truth labels, our student's segmentation prediction, a binary accuracy map, and the student's uncertainty prediction. The binary accuracy map visualizes incorrectly predicted pixels and void classes in white and correctly predicted pixels in black.

Visually, for large areas and well-represented classes like road, sidewalk, building, sky, and car, the student's segmentation is almost free of errors. This supports the quantitative evaluation described in Table 3.2. Like most segmentation models, our student struggles with class transitions, areas with lots of inherent noise, or areas that belong to the void class, which is visualized by the binary accuracy map.

A comparison of the binary accuracy map and our student's uncertainty prediction adds to the observations laid out in Table 3.2 and Figure 3.5: The uncertainty prediction reliably returns high uncertainties for wrongly classified pixels and OOD samples, which both are visualized as white pixels in the binary accuracy map. For example, in the first image of Figure 3.6, our student correctly predicts high uncertainties in the noisy parts of the background and for fine geometric structures like traffic lights. Conversely, the student predicts very low uncertainties for the road, buildings, sky, and vegetation. The second example image confirms this observation and adds two valuable insights about the quality of the student's uncertainty predictions. Firstly, although the train in the left part of the image is predicted correctly for the most part, the student still predicts high uncertainties. This is intuitively comprehensible and even desirable, given that the train class is highly underrepresented in the dataset – constituting only 0.15 % of the labeled pixels [39] – and is therefore inherently more challenging to detect reliably. Secondly, the student predicts high uncertainties in the bottom part of the image, where reflections on the hood of the car cause incoherent segmentation predictions. The third image exemplifies another quality of the predictive uncertainty of our student. In this case, the student struggles to correctly segment the truck in the right part of the image. Simultaneously, the student predicts high uncertainties for the entire truck, thus indicating the wrong segmentation prediction. The fourth image demonstrates the potential capability of the student model to identify OOD samples: For areas that belong to the void class, high uncertainties are predicted.

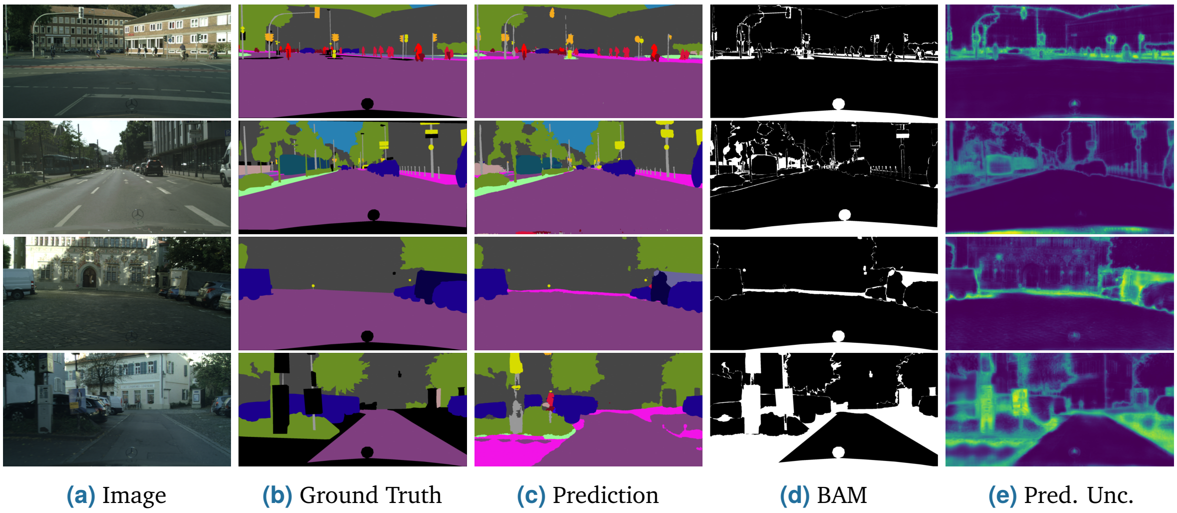


Figure 3.6: Qualitative examples from the Cityscapes validation set. White pixels in the binary accuracy map (BAM) are either incorrect predictions or void classes. The latter appear black in the ground truth labels. For the uncertainty prediction, brighter pixels represent higher predictive uncertainties.

Potential for Out-of-Domain Detection

To investigate the potential of DUDES for OOD detection, we evaluate our student model and the teacher ensemble on Foggy Cityscapes [235] and Rain Cityscapes [109] without re-training them. Despite considerable effort, we were unable to reproduce the results from Holder and Shafique [106], and therefore cannot conduct a direct comparison of their work against ours for this OOD examination.

Quantitative Evaluation. As Tables 3.4 and 3.5 show, our student model compares quite well with the teacher. Across all six validation datasets with varying amounts of simulated fog and rain, our student performs better on the segmentation task. It also manages to output similar predictive uncertainties, although it underestimates them with increasing intensity of fog and rain in comparison to the teacher. Potentially, this gap can be closed by incorporating hold-out samples or additional data augmentations during the distillation process to improve the ability of the student to generalize to OOD tasks. This certainly remains an interesting research question for future work.

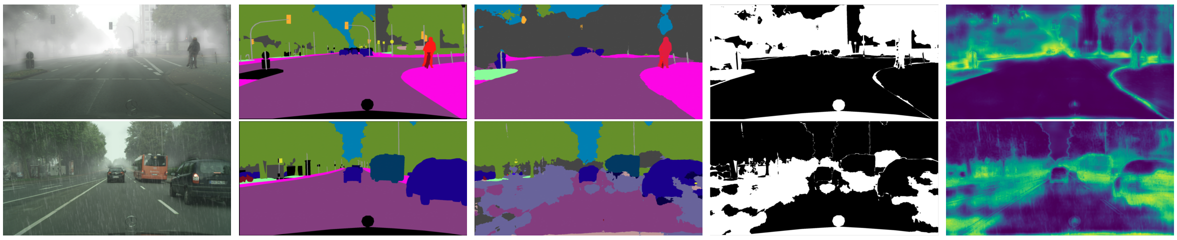
	$\text{Fog}_{\beta=0.005}$		$\text{Fog}_{\beta=0.01}$		$\text{Fog}_{\beta=0.02}$	
	mIoU ↑	mUnc	mIoU ↑	mUnc	mIoU ↑	mUnc
Teacher	0.642	0.123	0.576	0.141	0.467	0.162
Student	0.674	0.120	0.605	0.128	0.493	0.143

Table 3.4: Comparison between the student's and the teacher's mIoU and mUnc on the validation set of the Foggy Cityscapes dataset [235]. β denotes the attenuation coefficient and controls the thickness of the fog. Higher β values result in thicker fog.

	Rain_1		Rain_2		Rain_3	
	mIoU ↑	mUnc	mIoU ↑	mUnc	mIoU ↑	mUnc
Teacher	0.477	0.132	0.406	0.149	0.349	0.162
Student	0.483	0.123	0.422	0.134	0.361	0.140

Table 3.5: Comparison between the student's and the teacher's mIoU and mUnc on the validation set of the Rain Cityscapes dataset [109]. We evaluate on three sets of parameters, where Rain_1 uses [0.01, 0.005, 0.01], Rain_2 uses [0.02, 0.01, 0.005], and Rain_3 uses [0.03, 0.015, 0.002] for attenuation coefficients α and β and the raindrop radius a. α and β determine the degree of simulated rain and fog in the images.

Qualitative Evaluation. Figure 3.7 supports the quantitative findings with qualitative examples. As expected, the simulated fog and rain degrade the quality of the segmentation prediction considerably. Nevertheless, the student model exhibits valuable predictive uncertainty estimations, particularly in regions with numerous incorrect classifications. Generally, this adds to the observations of Figure 3.6: High uncertainties of the student correlate with wrongly classified pixels and OOD samples.



(a) Image (b) Ground Truth (c) Prediction (d) BAM (e) Pred. Unc.

Figure 3.7: Qualitative examples from the Foggy Cityscapes (top) and Rain Cityscapes (bottom) validation set. White pixels in the binary accuracy map (BAM) are either incorrect predictions or void classes. The latter appear black in the ground truth labels. For the uncertainty prediction, brighter pixels represent higher predictive uncertainties.

Ablation Studies

Number of Ensemble Members. An essential part of DUDES is the quality of the teacher's uncertainty prediction because it represents an upper bound for the uncertainty quality that can be expected from the student. Figure 3.8 shows the impact of the number of ensemble members on the mIoU and mUnc. Naturally, adding more ensemble members improves the segmentation results. The mIoU increases from 0.706 when using just two ensemble members to a maximum of 0.714 for twelve members. More importantly for DUDES, the mUnc increases from 0.092 for just two ensemble members to a maximum of 0.107 for six members. Adding more ensemble members to the teacher does not change the uncertainty prediction substantially, as the mUnc stays within 0.106 and 0.107 until all twenty members are included. Overall, using ten members appears to strike a balance between segmentation performance and computational efficiency. While the mIoU is only 0.001 lower compared to using twelve members, opting for ten members reduces the computational cost in terms of training time and memory footprint considerably. These findings go along with prior work on DE-based UQ [61, 138]. Consequently, we propose to use ten ensemble members for DUDES, which should be sufficient for most applications.

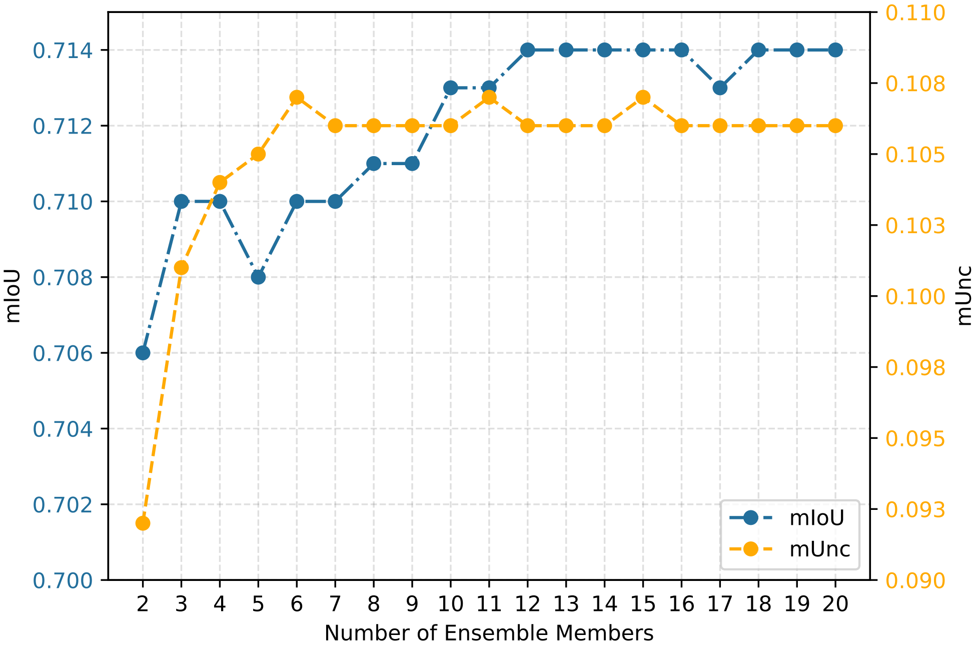


Figure 3.8: Ablation study on the impact of the number of ensemble members on the mIoU and mUnc.

Impact of Pre-training. Table 3.6 shows the results of another ablation study on the impact of ImageNet [47] pre-training on the mIoU, mUnc, and ECE. We comprehensively compare the baseline segmentation model with our student model and our teacher model, which consists of ten randomly initialized baseline models. The study does not examine the impact of ImageNet pre-training on the ensemble members, as this would lead to less reliable uncertainties compared to random initialization [61, 138].

While training for 200 epochs and using random initialization, our student underperforms the baseline model by 0.039 and the teacher by 0.067 with an mIoU of 0.646 on the segmentation task. Our randomly initialized student also underestimates the teacher's uncertainties by

0.009 with a mUnc of 0.097. When using ImageNet pre-training for the baseline model and our student, both significantly improve their mIoU with 0.737 and 0.738, respectively. The student also manages to approximate the predictive uncertainties better with a mUnc of 0.108, which is close to the 0.106 of the teacher. It is worth noting that similar performance can also be achieved by randomly initializing our student when the number of training epochs is quadrupled to 800. This concurs with the findings of He et al. [94]. As a consequence, we suggest using ImageNet pre-training for the student to improve convergence speed. On top of that, using ImageNet pre-training leads to a lower ECE.

	Training Epochs	mIoU ↑	ECE ↓	mUnc
Teacher$_{Random,n=10}$	200	0.713	0.021	0.106
Baseline$_{Random}$	200	0.685	0.031	-
Student$_{Random}$	200	0.646	0.045	0.097
Baseline$_{ImageNet}$	200	0.737	0.019	-
Student$_{ImageNet}$	200	0.738	0.025	0.108
Student$_{Random}$	800	0.739	0.037	0.105

Table 3.6: Ablation study on the impact of ImageNet [47] pre-training on the mIoU, mUnc, and ECE [192]. We evaluate the ECE based on the softmax pseudo-probabilities with the l_1 norm and a bin size of 10.

3.2.3 Adaptability Experiments

In this Section, we provide more evidence for the simplicity and generalizability of the methodology of DUDES by incorporating additional experiments with a modern Vision Transformer (ViT)-based architecture, Monte Carlo Dropout (MCD) as the UQ method, and a different dataset. Firstly, we lay out our adapted experimental setup. Secondly, we provide a quantitative as well as qualitative evaluation to finalize this Section.

Experimental Setup

Architecture. As shown by Figure 3.9, we use a state-of-the-art ViT-based architecture, SegFormer-B5 [284], pre-trained on ImageNet [47] as the backbone of a U-Net [225] decoder as the baseline model for the student and the teacher. Following Section 3.2.1, we add a second uncertainty head to the U-Net decoder. It mirrors the segmentation head but differs in the output layer, which has a single output channel instead of C channels, where C denotes the number of classes.

Uncertainty Quantification Method. For UQ, we apply MCD to our teacher, replacing the use of a DE. Since the SegFormer [284] already applies dropout layers throughout the entire network, we follow their work and consider two common dropout rates, 20 % and 50 %, for the teacher model. To train the student model, we leave the dropout layers of the teacher activated and sample ten times to obtain the predictive uncertainty for the uncertainty distillation [67, 243, 87].

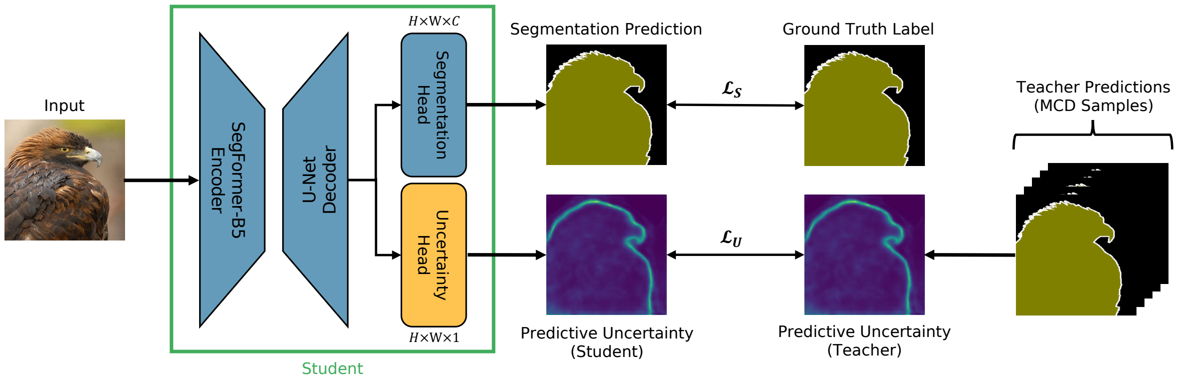


Figure 3.9: A schematic overview of the adapted training process of the student model of DUDES. Instead of using a DE, we apply MCD to our student. Additionally, we use the state-of-the-art SegFormer [284] as the backbone for a U-Net decoder [225]. As ground truth uncertainties, we compute the predictive uncertainty of the MCD samples of the teacher.

Training. During the training processes, we make three changes compared to Section 3.2.1. Firstly, we decrease the initial learning rate to 0.001, which we empirically found to work better with the new architecture. Additionally, we apply color jittering during the distillation process to improve the quality of the student's uncertainty estimates. Specifically, we use the TorchVision library [176] to randomly adjust the brightness, contrast, saturation, and hue of the input image. Following Shen et al. [243], who showed that such augmentation helps prevent the student from underestimating the teacher's test-time uncertainty distribution when training and distilling with the same dataset, we apply random variations in the range of $[-0.2, 0.2]$ for each of the four components. Lastly, we change the crop size to 256×256.

Dataset. In addition to a different architecture and UQ method, we also use the Pascal VOC 2012 [58] dataset for evaluation. Unlike Cityscapes, Pascal VOC 2012 consists of only 1464 training images and 1449 validation images with varying resolutions, 20 semantic object classes, and 1 background class. Additionally, the dataset is less homogeneous than Cityscapes. These properties make it inherently difficult to achieve accurate segmentation results with corresponding uncertainty estimates.

Quantitative Evaluation

Table 3.7 shows a comparison between the baseline segmentation model, two teacher models with dropout rates of 20 % and 50 %, respectively, and two student models with an additional uncertainty head for UQ.

Overall, the results align with the experimental findings on the Cityscapes dataset in Section 3.2.2. Our student models outperform their respective teacher models on the segmentation task while also capturing their predictive uncertainties. They only slightly underestimate their respective teachers' uncertainty, which we attribute to suboptimal hyperparameters and the inherently challenging properties of the Pascal VOC 2012 dataset. As a consequence, our student models match the performance of the baseline model on the segmentation task, while being slightly better calibrated in terms of ECE, and they are able to output a meaningful predictive uncertainty, without significantly increasing the inference time.

	Dropout	Inference Time [ms]	mIoU ↑	ECE ↓	mUnc
Baseline	-	33.1 ± 0.7	0.788	0.027	-
Teacher$_A$	20 %	355.5 ± 1.3	0.761	0.013	0.096
Teacher$_B$	50 %		0.656	0.008	0.157
Student$_A$	-	34.5 ± 1.3	0.787	0.022	0.081
Student$_B$	-		0.784	0.024	0.135

Table 3.7: Quantitative comparison between the baseline's, the student's, and the teacher's inference time, mIoU, ECE [192], and mUnc on the Pascal VOC 2012 dataset. Student$_A$ uses the uncertainties provided by Teacher$_A$ during training, whereas Student$_B$ uses the uncertainties provided by Teacher$_B$.

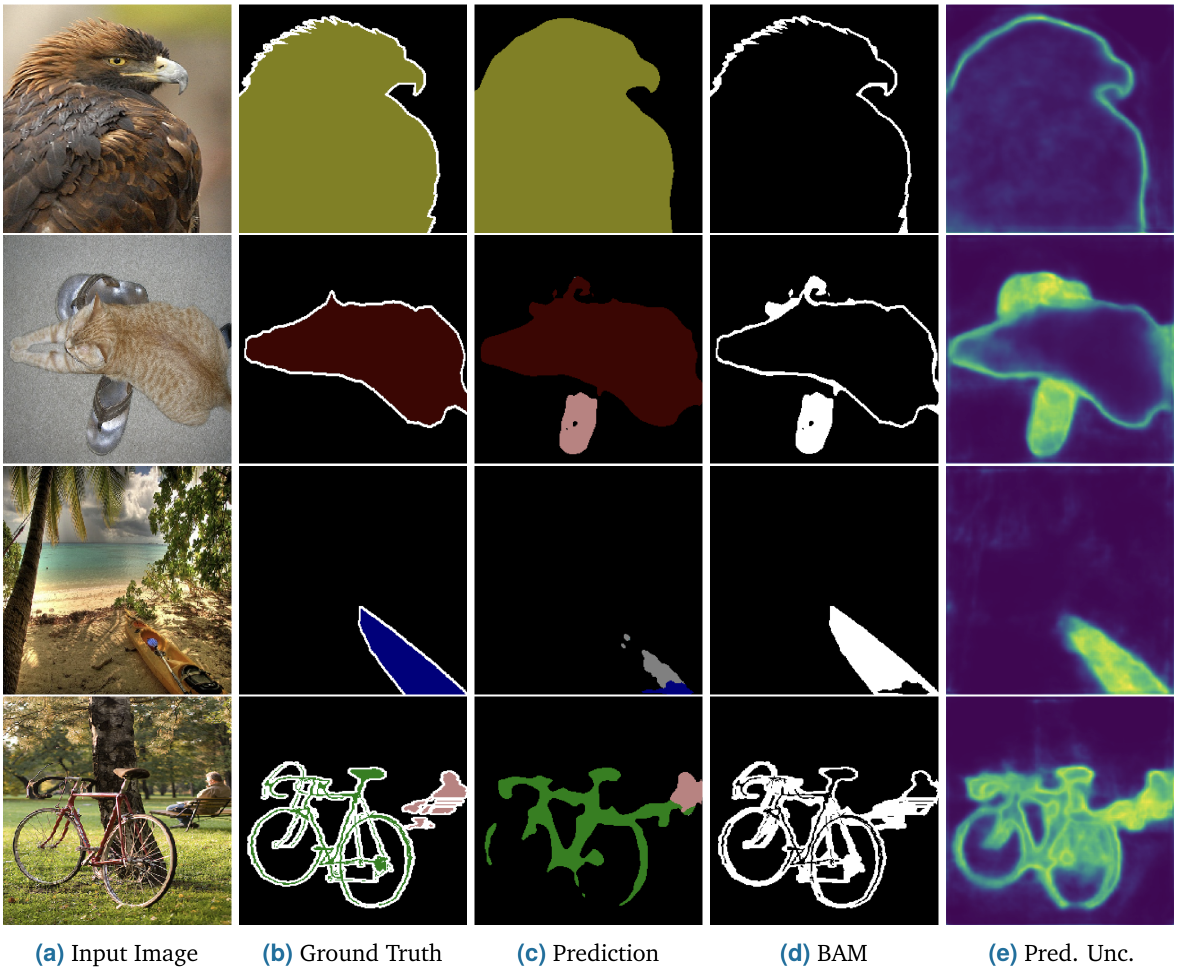

(a) Input Image **(b)** Ground Truth **(c)** Prediction **(d)** BAM **(e)** Pred. Unc.

Figure 3.10: Qualitative examples of the student$_B$ model that is based on a dropout rate of 50 % on the Pascal VOC 2012 validation set. White pixels in the binary accuracy map (BAM) are either incorrect predictions or belong to the void class. Latter appears white in the ground truth label. For the uncertainty prediction, brighter pixels represent higher predictive uncertainties.

Qualitative Evaluation

Figure 3.10 corroborates the quantitative findings. The student$_B$ model, based on a dropout rate of 50 %, predicts high uncertainties for object boundaries, entirely wrong or missing classifications, and areas with fine-grained details that are challenging to classify. In contrast, easy-to-classify areas and background pixels exhibit low uncertainties.

For example, in the first image, our student segments the depicted eagle almost perfectly and accordingly only predicts high uncertainties for the object boundaries. Conversely, in the second and third images, our student either wrongly classifies pixels that should belong to the background or fails to classify parts of the object. Nonetheless, in both cases, high uncertainties are predicted for these areas, providing valuable information. Similarly, the student predicts high uncertainties for both the bicycle and the human sitting on a bench in the fourth image, as they are challenging to classify through all of the fine-grained details and noise.

3.2.4 Discussion

DUDES applies student-teacher distillation with a DE to accurately approximate predictive uncertainties with a single forward pass while maintaining simplicity and adaptability. Against the teacher, the needed inference time per image is reduced by an order of magnitude, and the computational overhead in comparison to the baseline is negligible. Additionally, the student exhibits impressive potential for identifying wrongly classified pixels and OOD samples within an image by leveraging its uncertainties. Based on these observations, one could easily introduce an uncertainty-based threshold for OOD detection. However, it is essential to acknowledge that there remains a challenge in distinguishing between misclassified pixels and OOD samples, as both may trigger the threshold. Disentangling the predictive uncertainty into its two components – aleatoric and epistemic – could be a possible solution for this problem. Aleatoric uncertainty reflects inherent noise or ambiguity in the data, which could be higher for misclassified pixels, where prediction quality suffers due to confusing in-distribution patterns. In contrast, epistemic uncertainty indicates the model's lack of knowledge. Therefore, OOD samples should have higher epistemic uncertainty because they fall outside the training distribution.

DUDES represents a simple yet highly effective new approach for UQ. In contrast to the work by Holder and Shafique [106], DUDES requires no major changes to the student's architecture compared to the baseline and introduces only a single uncertainty loss without additional hyperparameters, yet delivers substantial improvements over their work. Firstly, our student model slightly outperforms its teacher in the segmentation task by 0.025 mIoU while their student suffers from a segmentation performance degradation in comparison to its teacher by 0.042. Secondly, our student approximates its teacher's predictive uncertainties more closely than the student model by Holder and Shafique [106]. More precisely, their student tends to underestimate uncertainties for classes with high uncertainties and vice versa, whereas our student does not suffer from any systematic shortcomings.

A major factor in the effectiveness of DUDES lies in the simplification of what is distilled. Instead of distilling the entire uncertainty map of the teacher, which is what Holder and Shafique [106] proposed, we only use the predictive uncertainty of the teacher. The teacher's uncertainty map is calculated by computing the standard deviation of the softmax pseudo-probability maps of the individual models along the class dimension. In the case of multi-class

SS, the resulting uncertainty map has dimensions of $H \times W \times C$, where C is the number of classes, H is the image height, and W is the image width. For DUDES, the class dimension is reduced to 1 by only considering the uncertainty of the predicted class in the segmentation map. Due to this simplification, the segmentation performance of the student is not hindered, and the predictive uncertainties can be learned more accurately.

We acknowledge the simplification in the uncertainty distillation to be a potential limitation of DUDES as the student is only capable of estimating the uncertainty of the predicted class. However, there are practically no negative implications of this limitation since the remaining uncertainties are usually discarded anyway. Hence, DUDES remains useful for efficiently estimating predictive uncertainties for a wide range of applications while being easy to adapt.

We believe that DUDES has the potential to provide a new promising paradigm in reliable UQ by focusing on simplicity and efficiency. Except for the computational overhead during training, we found no apparent reason not to employ our proposed method in SS applications where safety and reliability are critical.

3.3 Exploitation of Predictive Uncertainties

The upcoming Section introduces U-CE, a novel loss function that integrates dynamic uncertainty estimates into the training process. By applying pixel-wise uncertainty weighting to the standard CE loss, U-CE enhances segmentation performance and trains models that are naturally capable of quantifying meaningful uncertainties.

Research Gap. In contrast to existing literature on uncertainty-aware segmentation, U-CE fully utilizes predictive uncertainties dynamically during training. By pixel-wise uncertainty weighting of the CE loss, U-CE harnesses valuable insights from the uncertainties to guide the optimization process. This approach enables more effective training, resulting in improved segmentation performance.

3.3.1 Methodology

In the following, we provide an overview of U-CE, explain our novel uncertainty-aware CE loss, and outline the implementation details.

Overview

The central idea of U-CE is to incorporate predictive uncertainties into the training process to enhance segmentation performance. As depicted in Figure 3.11, we propose two simple yet highly effective adaptations to the regular training process:

1. During training, we sample from the posterior distribution with MCD to obtain predictive uncertainties alongside the regular segmentation prediction.
2. We apply pixel-wise weighting to the regular CE loss based on the collected uncertainties.

For our specific case, MCD emerges as the preferred option to compute the predictive uncertainties due to its ease of use, minimal impact on the training process, and computational efficiency compared to DEs. However, it is worth noting that other UQ methods could also be utilized for U-CE.

With predictive uncertainties, we refer to the standard deviation of the softmax pseudo-probabilities of the predicted class provided by MCD sampling.

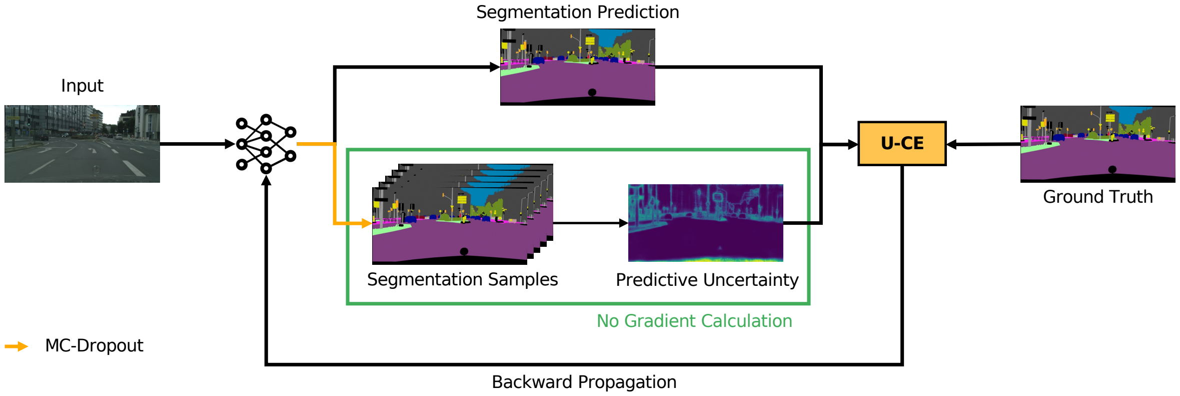


Figure 3.11: A schematic overview of the training process of U-CE. U-CE integrates the predictive uncertainties of a MCD model into the training process to enhance segmentation performance. In comparison to most applications of MCD, U-CE utilizes the uncertainties not only at test time but also dynamically during training by applying pixel-wise weighting to the regular CE loss.

Uncertainty-aware Cross-Entropy

Segmentation Sampling. In contrast to typical usage of MCD, U-CE incorporates the sampling process from the posterior distribution not only at test time but also during training. To compute the necessary uncertainties for our uncertainty-aware CE loss, we perform β sampling iterations at each training step. This generates β segmentation samples in addition to the regular segmentation prediction. Notably, gradient computation is disabled during the sampling process as it is unnecessary for backward propagation, which relies solely on the regular segmentation prediction. By disabling gradient computation during sampling, we reduce the additional computational overhead of U-CE in terms of training time and GPU memory usage.

Uncertainty-aware Cross-Entropy Loss. The final objective function of U-CE builds upon the well-known categorical CE loss and can be defined as:

$$\mathcal{L}_{\text{U-CE}} = -\frac{1}{N}\sum_{n=1}^{N} \omega_n \sum_{c=1}^{C} y_{n,c} \cdot \log(p_{n,c}(\hat{y})) \ , \tag{3.7}$$

where $\mathcal{L}_{\text{U-CE}}$ is the loss for a single image, N is the number of pixels, C is the number of classes, $y_{n,c}$ is the ground truth label, $p_{n,c}(\hat{y})$ is the predicted pseudo-probability, and ω_n is the pixel-wise uncertainty weight. It is worth noting that Equation 3.7 simplifies to the regular CE loss by setting w_n to one for all pixels.

Pixel-wise Uncertainty Weight. The pixel-wise uncertainty weight w_n can be formulated as

$$\omega_n = (1 + \hat{\sigma}_n)^{\alpha} \ , \tag{3.8}$$

where $\hat{\sigma}_n$ denotes the predictive uncertainty, and α controls the influence of the uncertainties in an exponential manner. The predictive uncertainty $\hat{\sigma}$ corresponds to the standard deviation of the softmax pseudo-probabilities of the predicted class of the segmentation samples.

3.3.2 Experiments

In this Section, we conduct an extensive range of experiments to demonstrate the value of incorporating predictive uncertainties into the training process. Firstly, we provide quantitative results comparing regular CE to U-CE under diverse settings. Secondly, we analyze qualitative examples. Lastly, we provide multiple ablation studies.

Experimental Setup

Architecture. For all of our experiments, we employ DeepLabv3+ [29] as the decoder and either a RN18 or ResNet-101 (RN101) [95] as the encoder. Both backbones are commonly used for SS [185, 299], making our work highly comparable and serving as an excellent baseline for future research.

Monte Carlo Dropout. To convert our architectures into MCD models, we add a dropout layer after each of the four residual block layers of the ResNets, inspired by Kendall et al. [125] and Gustafsson et al. [87].

Training. For all training processes, we use a Stochastic Gradient Descent optimizer [223] with a base learning rate of 0.01, momentum of 0.9, and weight decay of 0.0001. Additionally, we multiply the learning rate of the decoder and segmentation head by ten. Finally, we employ polynomial learning rate scheduling to decay the initial learning rate during the training process (cf. Equation 3.6). In all training processes, we use a batch size of 16 and train on four NVIDIA A100 GPUs with 40 GB of memory using mixed precision [183].

Datasets. All of our experiments are based on either the Cityscapes dataset [39] or the ACDC dataset [234]. Both datasets are publicly available street scene datasets aimed at advancing the current state of the art in autonomous driving. The former consists of 2975 training images, 500 validation images, and 1525 test images. The latter contains 1600 training images, 406 validation images, and 2000 test images. Although both datasets share the same 19 evaluation classes and a void class, the ACDC dataset exclusively focuses on four adverse conditions: fog, nighttime, rain, and snow.

Data Augmentations. To prevent overfitting, we apply a common data augmentation strategy for all training procedures, regardless of the dataset or architecture used. The strategy includes the following steps:

1. Random scaling with a factor between 0.5 and 2.0.
2. Random cropping with a crop size of 768 × 768 pixels.
3. Random horizontal flipping with a flip chance of 50 %.

Evaluation. Since both test splits are withheld for benchmarking purposes, we utilize the validation images for testing in all our experiments. Unless otherwise specified, we only report single forward pass results based on the original validation images without resizing or sampling for a fair comparison between all of the models. Also, we set the number of segmentation samples β to ten by default.

Metrics. For quantitative evaluations, we primarily report the mIoU to measure the segmentation performance. In addition to the mIoU, we also utilize the ECE [192] to evaluate the calibration as well as the mUnc to quantitatively compare the resulting uncertainties.

Quantitative Evaluation

Tables 3.8 and 3.9 outline a quantitative comparison between Focal Loss (FL) [162], regular CE training, and our proposed U-CE loss using two different α values for various dropout ratios and training lengths on the Cityscapes [39] and ACDC [234] datasets. For FL, we followed the original publication and set the focusing parameter γ to 2.0 as this worked best in their experiments [162].

FL [162] performed the worst in all our experiments, possibly due to insufficient hyperparameter tuning. Remarkably, U-CE$_{\alpha=10}$ achieves the highest mIoU across all dropout ratios, even outperforming dropout-free baseline models in most cases. Notably, U-CE$_{\alpha=10}$ achieves a maximum improvement of up to 0.093 mIoU over regular CE when training on ACDC [234] for 200 epochs using a RN18 with a dropout ratio of 40 %. On average, U-CE$_{\alpha=10}$ outperforms CE by 0.020 on Cityscapes [39] and by 0.046 on ACDC [234]. Interestingly, U-CE$_{\alpha=1}$ also matches or improves upon regular CE training in most cases. On average, U-CE$_{\alpha=1}$ outperforms CE by 0.003 on Cityscapes and by 0.013 on ACDC.

Table 3.10 provides additional information on the ECE and mUnc for CE and U-CE using a dropout ratio of 20 %. In comparison to regular CE and U-CE$_{\alpha=1}$, which exhibit similar results, U-CE$_{\alpha=10}$ not only improves segmentation performance but also yields slightly better calibrated networks, as measured by the ECE. Moreover, the mUnc is also slightly lower for U-CE$_{\alpha=10}$.

Overall, Tables 3.8, 3.9, and 3.10 provide strong evidence for the effectiveness of leveraging predictive uncertainties in the training process. The impact of quantifying these uncertainties on the training time will be explored in the ablation studies discussed in Section 3.3.2.

	Encoder	200 Epochs				500 Epochs			
		FL [162]	CE	U-CE$_{\alpha=1}$	U-CE$_{\alpha=10}$	FL [162]	CE	U-CE$_{\alpha=1}$	U-CE$_{\alpha=10}$
Dropout (0 %)	RN18	0.660	**0.700**	-	-	0.700	**0.720**	-	-
Dropout (10 %)	RN18	0.661	0.694	0.696	**0.716**	0.699	0.723	0.723	**0.742**
Dropout (20 %)	RN18	0.654	0.690	0.695	**0.718**	0.694	0.719	0.726	**0.735**
Dropout (30 %)	RN18	0.643	0.682	0.690	**0.710**	0.691	0.719	0.724	**0.741**
Dropout (40 %)	RN18	0.622	0.666	0.677	**0.705**	0.681	0.711	0.711	**0.737**
Dropout (50 %)	RN18	0.582	0.643	0.653	**0.696**	0.655	0.690	0.694	**0.726**
Dropout (0 %)	RN101	0.731	**0.746**	-	-	0.756	**0.761**	-	-
Dropout (10 %)	RN101	0.728	0.748	0.751	**0.761**	0.753	0.763	0.766	**0.775**
Dropout (20 %)	RN101	0.726	0.746	0.748	**0.766**	0.753	0.763	0.770	**0.777**
Dropout (30 %)	RN101	0.718	0.745	0.747	**0.761**	0.755	0.764	0.766	**0.775**
Dropout (40 %)	RN101	0.712	0.747	0.740	**0.758**	0.750	0.761	0.765	**0.782**
Dropout (50 %)	RN101	0.703	0.741	0.737	**0.759**	0.744	0.766	0.766	**0.773**

Table 3.8: Quantitative comparison on the Cityscapes dataset [39] for different dropout ratios. The provided numbers represent the mIoU. Best respective results are marked in **bold**.

	Encoder	200 Epochs				500 Epochs			
		FL [162]	CE	U-CE$_{\alpha=1}$	U-CE$_{\alpha=10}$	FL [162]	CE	U-CE$_{\alpha=1}$	U-CE$_{\alpha=10}$
Dropout (0 %)	RN18	0.501	**0.563**	-	-	0.576	**0.622**	-	-
Dropout (10 %)	RN18	0.502	0.555	0.564	**0.600**	0.574	0.621	0.628	**0.650**
Dropout (20 %)	RN18	0.490	0.546	0.561	**0.605**	0.569	0.615	0.620	**0.650**
Dropout (30 %)	RN18	0.466	0.522	0.543	**0.592**	0.549	0.596	0.616	**0.643**
Dropout (40 %)	RN18	0.426	0.489	0.508	**0.582**	0.511	0.568	0.588	**0.639**
Dropout (50 %)	RN18	0.399	0.477	0.493	**0.563**	0.482	0.533	0.560	**0.624**
Dropout (0 %)	RN101	0.604	**0.650**	-	-	0.663	**0.688**	-	-
Dropout (10 %)	RN101	0.589	0.645	0.653	**0.670**	0.658	0.684	0.693	**0.699**
Dropout (20 %)	RN101	0.588	0.641	0.650	**0.658**	0.652	0.685	0.687	**0.702**
Dropout (30 %)	RN101	0.573	0.627	0.643	**0.653**	0.651	0.684	0.685	**0.699**
Dropout (40 %)	RN101	0.547	0.611	0.631	**0.654**	0.631	0.678	0.678	**0.700**
Dropout (50 %)	RN101	0.523	0.580	0.602	**0.637**	0.611	0.660	0.674	**0.702**

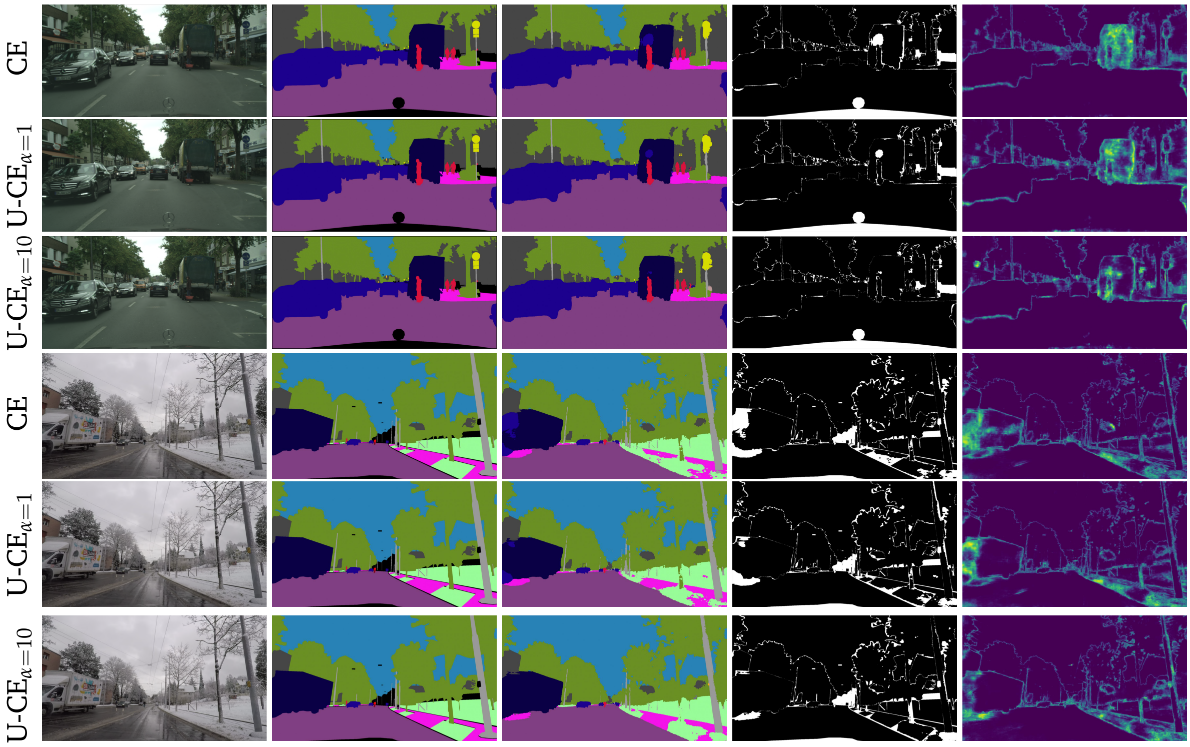
Table 3.9: Quantitative comparison on the ACDC dataset [234] for different dropout ratios. The provided numbers represent the mIoU. Best respective results are marked in **bold**.

	Encoder	200 Epochs			500 Epochs		
		mIoU ↑	ECE ↓	mUnc	mIoU ↑	ECE ↓	mUnc
CE	RN18	0.690	0.035	0.088	0.719	0.025	0.088
U-CE$_{\alpha=1}$	RN18	0.695	0.036	0.089	0.726	0.027	0.088
U-CE$_{\alpha=10}$	RN18	0.718	0.029	0.085	0.735	0.018	0.084
CE	RN101	0.746	0.026	0.080	0.763	0.041	0.076
U-CE$_{\alpha=1}$	RN101	0.748	0.024	0.079	0.770	0.041	0.076
U-CE$_{\alpha=10}$	RN101	0.766	0.022	0.073	0.777	0.040	0.073

Table 3.10: A more detailed quantitative comparison between regular CE and U-CE on the Cityscapes dataset [39] using a dropout ratio of 20 %.

Qualitative Evaluation

In addition to the quantitative evaluation, we also provide qualitative examples in Figure 3.12. The first three rows depict results from models with a RN18 backbone and a dropout ratio of 20 %, trained for 200 epochs with CE, U-CE$_{\alpha=1}$, U-CE$_{\alpha=10}$ on Cityscapes [39]. The last three rows show examples from models using a RN101 backbone and a dropout ratio of 20 %, trained for 500 epochs on the ACDC dataset [234]. The binary accuracy map visualizes incorrectly predicted pixels and void classes in white, and correctly predicted pixels in black.



(a) Input Image **(b)** Ground Truth **(c)** Prediction **(d)** BAM **(e)** Pred. Unc.

Figure 3.12: Qualitative results on the Cityscapes and ACDC validation sets. White pixels in the binary accuracy map (BAM) are either incorrect predictions or void classes, which appear black in the ground truth label. For the uncertainty, brighter pixels represent higher predictive uncertainties. The first three rows depict results from models with a RN18 backbone and dropout ratio of 20 %, trained for 200 epochs on Cityscapes [39]. The last three rows show examples from models using a RN101 backbone and a dropout ratio of 20 %, trained for 500 epochs on the ACDC dataset [234].

Generally, for large areas and well-represented classes like road, building, sky, and car, all models perform exceptionally well with minimal errors. Furthermore, there is a strong correlation between the binary accuracy map and the predictive uncertainty, indicating that all models provide meaningful uncertainties.

Nonetheless, there are nuanced differences between the models. For example, in the first two rows of Figure 3.12, which represent models trained with CE and U-CE$_{\alpha=1}$, there are noticeable misclassifications on top of the human standing in front of the truck. Naturally, this area is also accompanied by high uncertainties. In contrast, the model trained with U-CE$_{\alpha=10}$ exhibits significantly fewer difficulties, resulting in a better segmentation prediction and lower uncertainties.

A similar situation is observable in the last three rows, showing examples from the more challenging ACDC dataset [234]. Here, the model trained with regular CE struggles to correctly segment the truck on the left as well as differentiate between the sidewalk and the terrain on the right side of the image. The model trained with U-CE$_{\alpha=1}$ does slightly better in these areas, but is equally uncertain. Only the model trained with U-CE$_{\alpha=10}$ successfully classifies the truck and differentiates between the sidewalk and the terrain decently. Consequently, the predictive uncertainty is also lower in these areas.

In summary, the qualitative findings presented in Figure 3.12 concur with our quantitative evaluation, manifesting the efficacy of U-CE across different datasets and architectures.

Ablation Studies

In addition to the quantitative and qualitative evaluation, we also present multiple ablation studies. Unless otherwise noted, we confined all of the ablation studies to models that use a RN18 as the backbone, have a dropout ratio of 20 %, and were trained for 200 epochs.

Impact of α. The most influential hyperparameter of U-CE is α as it exponentially controls the weighting of the CE loss. Table 3.11 demonstrates the impact of different α values on the mIoU for both backbones, RN18 and RN101, on both Cityscapes and ACDC. Evidently, the segmentation performance consistently improves as α increases until it reaches ten, which stands as the best value in three out of four cases across the two datasets and architectures. Thus, using ten as the default value for α seems to be a fair estimation to achieve the best results, not only for the mentioned cases but potentially for other applications as well. Further increasing α leads to a degradation in mIoU. Additionally, training becomes more unstable as models overly focus on uncertain pixels, resulting in some models failing to converge properly. Nonetheless, U-CE exhibits robustness against changes in α, offering a wide range of valid hyperparameters that lead to improved segmentation results compared to regular CE training.

α	1	2	4	6	8	10	12	14	16
RN18 (Cityscapes)	0.695	0.700	0.707	0.712	0.715	**0.718**	0.710	0.470	0.709
RN101 (Cityscapes)	0.748	0.752	0.756	0.761	0.764	**0.766**	0.763	0.758	0.726
RN18 (ACDC)	0.561	0.569	0.576	0.588	0.588	**0.605**	0.603	0.601	0.375
RN101 (ACDC)	0.650	0.650	0.657	0.655	0.660	0.658	**0.667**	0.645	0.199

Table 3.11: Ablation study on the impact of α. The provided numbers represent the mIoU ↑. Best respective results are marked in **bold**.

Impact of β. Table 3.12 exhibits another ablation study on the number of segmentation samples β. Interestingly, there is no clear benefit of sampling more often than six times, especially concerning the training time. U-CE$_{\beta=6}$ increases the approximate training time by just 10 % whilst improving the mIoU by 2.6 % over regular CE training, whereas U-CE$_{\beta=10}$ extends it by roughly 35 % without any further improvements. For comparison, Gal and Ghahramani [67] recommend sampling ten times to get a reasonable estimation of the predictive mean and uncertainty.

β	0	2	6	10	14	18
CE	0.690 (1:49)	-	-	-	-	-
U-CE$_{\alpha=10}$	-	0.711 (1:52)	0.716 (2:01)	0.716 (2:27)	0.716 (2:53)	0.717 (3:17)

Table 3.12: Ablation study on the number of segmentation samples β. In addition to the mIoU ↑, we provide the training time in hours:minutes ↓ in paranthesis.

Impact of Data Augmentations. The impact of various data augmentation strategies on CE and U-CE is demonstrated in Table 3.13. The results show that incorporating additional data augmentations improves the mIoU across the board. More importantly, this ablation study confirms that U-CE consistently outperforms CE across different data augmentation strategies, except for U-CE$_{\alpha=1}$ without random flipping and random scaling, indicating its effectiveness in improving segmentation performance.

	Random Flipping	Random Scaling	mIoU ↑
CE	×	×	0.661
	✓	×	0.670
	×	✓	0.686
	✓	✓	0.690
U-CE$_{\alpha=1}$	×	×	0.658
	✓	×	0.678
	×	✓	0.691
	✓	✓	0.695
U-CE$_{\alpha=10}$	×	×	0.696
	✓	×	0.701
	×	✓	0.718
	✓	✓	0.718

Table 3.13: Ablation study on the impact of various data augmentation strategies.

Impact of lr_{base}. Table 3.14 shows the ablation study on the base learning rate lr_{base}. The most notable comparison is between regular CE and U-CE$_{\alpha=1}$, which demonstrates that U-CE is not limited to specific learning rates. U-CE$_{\alpha=1}$ consistently outperforms regular CE for all examined base learning rates. Moreover, U-CE$_{\alpha=10}$ exceeds the results of CE and U-CE$_{\alpha=1}$ for all base learning rates except 10^{-1}, which caused divergence. Overall, this ablation study confirms the value of leveraging predictive uncertainties during training, irrespective of the learning rate, which is arguably the single most important hyperparameter in DL [11].

lr_{base}	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
CE	0.505	0.690	0.559	0.356	0.189
U-CE$_{\alpha=1}$	**0.560**	0.695	0.576	0.369	0.193
U-CE$_{\alpha=10}$	0.020	**0.718**	**0.650**	**0.476**	**0.253**

Table 3.14: Ablation study on the base learning rate lr_{base}. The provided numbers represent the mIoU ↑. Best results are marked in **bold**.

3.3.3 Discussion

In contrast to previous approaches, U-CE fully leverages predictive uncertainties obtained by MCD during training. As a result, we manage to train models that not only improve their segmentation performance but are also naturally capable of predicting meaningful uncertainties after training as well.

While U-CE appears to have no apparent shortcomings, except for a minor increase in training time, we acknowledge the need for a transparent discussion about its potential limitations. We aim to effectively guide future work in pushing the boundaries of state-of-the-art techniques, especially in safety-critical applications like autonomous driving.

Limitations. One limitation of U-CE arises in the absence of densely annotated ground truth labels. If most pixels are either labeled as background or designated to be ignored while training, U-CE will likely offer next to no benefit, except for a higher loss around object boundaries. Additionally, U-CE may not contribute to improved segmentation performance if the network is already overfitting the training data. Having said that, the impact of U-CE on the generalization ability of a trained model needs further examination.

Overall, we believe that U-CE presents a promising paradigm in SS by dynamically leveraging uncertainties to create more robust and reliable models. Despite a minor increase in training time and room for further improvement, we see no reason not to employ U-CE in comparison to regular CE.

3.4 Conclusion

Summary. In this Chapter, two fundamental research questions related to UQ in SS were addressed: (1) How to enable efficient and reliable UQ while maintaining technical simplicity, and (2) how to exploit uncertainty estimates to guide the optimization process. To this end, we introduced two novel approaches – DUDES and U-CE – that contribute to the advancement of uncertainty-aware segmentation models.

First, we presented DUDES, which appropriates the concept of knowledge distillation for efficient UQ. By leveraging a lightweight student model trained to mimic the uncertainty estimates of a DE-based teacher. DUDES provides reliable predictive uncertainties with a single forward pass while maintaining architectural simplicity and ease of implementation. Based on our extensive evaluations, DUDES not only significantly reduces the computational overhead compared to a DE but also achieves comparable uncertainty estimates and segmentation performance. The ability to detect wrongly classified pixels and OOD samples by high uncertainties further underscores its potential for real-world applications.

Second, we introduced U-CE, an uncertainty-aware CE loss that dynamically incorporates predictive uncertainties into the training process. By applying pixel-wise uncertainty weighting, U-CE allows the model to adapt its learning to regions of higher uncertainty, leading to improved segmentation performance. Through a comprehensive set of experiments, we not only confirm the effectiveness of U-CE over regular CE training but also find that the trained models are naturally capable of predicting meaningful uncertainties after training.

Together, these contributions demonstrate that UQ can be both computationally efficient – and therefore usable in real-world applications – as well as practically useful for improving segmentation performance itself. While DUDES enables real-time UQ without sacrificing accuracy, U-CE exploits predictive uncertainties to guide learning, resulting in more robust and reliable models. These findings not only highlight the importance and potential of uncertainty-aware SS but also open up opportunities for further exploration in other critical machine vision tasks such as Object Detection, Pose Estimation, or Monocular Depth Estimation (MDE).

Future Work. In terms of future work, multiple promising research opportunities could build upon the findings of DUDES and U-CE to accelerate uncertainty-aware SS for safety-critical applications.

For DUDES, an interesting direction could be the exploration of different knowledge distillation techniques, which were already outlined in Section 2.3. For example, in the context of feature-based distillation, it would be highly interesting to investigate the impact of feature map diversity inside a DE and how this information could be used to refine the uncertainty distillation process for the student. Moreover, online distillation and self-distillation present promising avenues to streamline the current two-step framework. This would also allow for the integration of dynamic uncertainty-aware training techniques like U-CE, effectively combining both approaches for efficient estimation and exploitation of predictive uncertainties for enhanced SS.

For U-CE, several promising directions could further enhance its effectiveness. One potential avenue is integrating state-of-the-art UQ methods like DEs to improve the uncertainty quality, which we were unable to explore due to computational constraints. However, improved uncertainty estimates will likely lead to more effective uncertainty-weighting. In the same sense, it would be worth investigating the introduction of warmup epochs, which we deliberately omitted to avoid introducing additional hyperparameters. This could help stabilize early training dynamics and improve uncertainty estimates, possibly leading to better overall performance. Another potential improvement would be the incorporation of statistical hypothesis testing to replace the hyperparameter α. This would be beneficial in two ways: Firstly, it would remove the most influential hyperparameter of U-CE. Secondly, and maybe more importantly, it would leverage the entire uncertainty distribution over all classes and not just the uncertainty of the predicted class. Finally, we encourage other researchers to incorporate U-CE into state-of-the-art SS approaches and to explore its usefulness in other computer vision tasks that rely on pixel-wise predictions, such as MDE.

We hope that these findings and suggestions will inspire further research into exploiting uncertainties as a core component of modern SS pipelines, ultimately leading to more reliable machine vision systems for the real world.

4 Uncertainty-aware Monocular Depth Estimation

This Chapter includes elements from

[146] S. Landgraf, R. Qin, and M. Ulrich. “A Critical Synthesis of Uncertainty Quantification and Foundation Models in Monocular Depth Estimation”. In: *arXiv preprint arXiv:2501.08188* (2025),

which are marked with a blue line.

In the following Chapter, we will deal with Uncertainty Quantification (UQ) in Monocular Depth Estimation (MDE), which is another detrimental machine vision task that aims to estimate a depth value for each pixel from a single image. In particular, we will target metric depth estimation, as it is crucial for real-world applications such as robotics [52, 229], augmented reality [123], and autonomous driving [288, 283].

Monocular Depth Estimation. At its core, MDE aims to transform a single image into a depth map by regressing range values for each pixel, all without exploiting direct range or stereo measurements. Theoretically, MDE is a geometrically ill-posed problem that is fundamentally ambiguous and can only be solved with the help of prior knowledge about object shapes, sizes, scene layouts, and occlusion patterns [52, 177, 128, 302]. This inherent requirement for scene understanding perfectly aligns MDE with Deep Learning (DL) approaches, which have proven proficient in encoding potent priors [14, 212, 290, 291]. These models benefit from extreme scaling, i.e., training on massive datasets and increasing model size, which facilitates the emergence of high-level visual scene understanding.

Based on these findings, a plethora of models have been proposed to address the challenges of MDE, with recent state-of-the-art solutions often leveraging large Vision Transformers (ViTs) trained on internet-scale data [31, 32, 159, 217, 14, 212, 290, 291], yielding foundation models capable of generalizing to a wide range of applications and scenes. A particularly challenging yet crucial application in fields such as robotics [52, 229], augmented reality [123], and autonomous driving [288, 283] is the estimation of absolute distances in real-world units (e.g., meters), commonly referred to as metric depth estimation. This task is especially difficult due to inherent metric ambiguities caused by different camera models and scene variations. Fortunately, these foundation models can successfully be fine-tuned in the respective domain [14, 212, 290, 291] to determine exceptionally accurate metric depths.

Challenges. However, the strong performance of foundation models on common benchmarks [245, 76, 39, 250] can lead to naive deployment, potentially overlooking their limitations. As illustrated in Figure 4.1, even state-of-the-art foundation models are not immune to inaccuracies. This can be particularly detrimental in safety-critical applications where errors can have serious consequences. There are multiple challenges associated with real-world deployment of DL models, including the lack of transparency due to the "black box" character of end-to-end systems [230, 75], the inability to distinguish between in-domain (ID) and out-of-domain (OOD) samples [153, 154], the tendency to be overconfident [84], and the sensitivity to adversarial attacks [219, 241, 249].

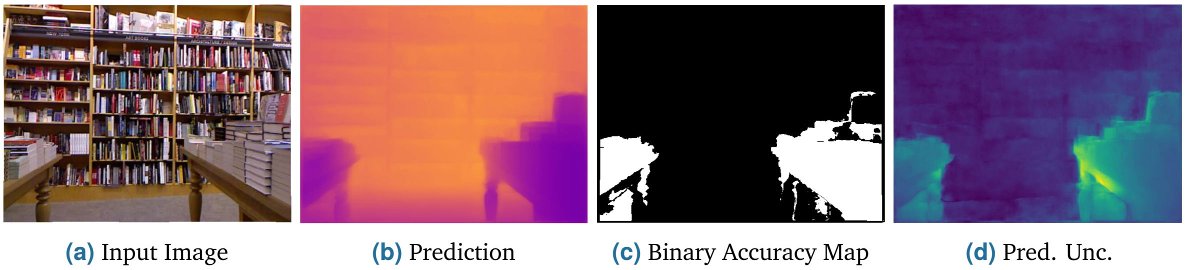


Figure 4.1: Qualitative example of a fine-tuned DepthAnythingV2 [291] for metric MDE on the NYUv2 dataset [245], using a ViT-S encoder and Monte Carlo Dropout (MCD) [67] for an additional uncertainty estimate, which represents the variance of the metric depth estimate. The binary accuracy map is based on the δ_1 error. The strong correlation between erroneous predictions and high uncertainties highlights the potential of integrating UQ methods with foundation models for MDE.

Research Questions. The following Chapter delves into a crucial research question to establish powerful foundation models in MDE that are aware of their uncertainties, defined as the actual variances of metric depth estimates:

- How can we synthesize UQ and foundation models for metric MDE to balance the benefits of large-scale pre-training with the need for reliable uncertainty estimates?

To answer this question and to bridge the gap between ground-breaking results in research and safe, reliable deployment in real-world applications, we investigate multiple UQ methods in combination with MDE foundation models. We specifically focus on combining the state-of-the-art DepthAnythingV2 foundation model [291] with five different UQ methods to enable pixel-wise variance measures for metric depth estimation:

1. **Learned Confidence (LC)** [269]: Confidences, interpreted as uncertainties, are learned by extending the primary objective function with an additional loss term.
2. **Gaussian Negative Log-Likelihood (GNLL)** [201]: Predictions are treated as samples from a Gaussian distribution, with the network outputting both a predictive mean and its corresponding variance, which is learned implicitly through minimizing the GNLL.
3. **Monte Carlo Dropout (MCD)** [67]: Dropout layers remain active during inference, sampling from the posterior distribution to estimate a predictive mean and variance.
4. **Deep Sub-Ensembles (DSEs)** [263]: A Deep Ensemble (DE) [138] is approximated by multiplying a subset of the model's layers instead of using the full model.
5. **Test-Time Augmentation (TTA)** [9]: Perturbations applied to inputs during inference produce unique samples, enabling computation of a predictive mean and corresponding variance.

If we can leverage UQ to correlate high uncertainties with erroneous predictions, it opens up the possibility of safer deployment of these models in real-world applications, as shown by Figure 4.1.

Outline and Structure. The structure of the following Chapter is outlined as follows:

1. Section 4.1 presents an overview of related work on MDE and previous attempts to incorporate UQ in this task.
2. Section 4.2 provides a detailed description of the utilized foundation model, DepthAnythingV2 [290] – and its predecessor, DepthAnythingV1 [290].
3. Section 4.3 lays out the methodology for this critical synthesis of UQ and foundation models, including extensive experiments on four diverse datasets.
4. Section 4.4 concludes this Chapter with a concise summary of key findings and a call for future research to include UQ in MDE.

4.1 Related Work

The following Section introduces related work in MDE and previous attempts to synthesize UQ with this foundational machine vision task.

Monocular Depth Estimation

Foundations. MDE is a dense regression task that aims to predict a depth value for each pixel in a given input image. The pioneering work of Eigen et al. [57] laid the foundation for MDE by directly predicting depth using a multi-scale neural network. This seminal approach demonstrated that Convolutional Neural Networks (CNNs) could effectively learn spatial hierarchies and capture depth cues from monocular images, thus inspiring a plethora of subsequent methods [186, 177, 128, 8]. While most introduce novel architectures or loss functions, Fu et al. [63] reformulate depth estimation as an ordinal regression (sometimes also referred to as ordinal classification in some contexts) problem through discretization of the depth ranges. Another innovative approach by Yuan et al. [295] incorporates neural conditional random fields to model contextual dependencies, further refining depth predictions. Besides, Patil et al. [206] impose geometric constraints based on piecewise planarity priors.

Vision Transformers (ViTs). Naturally, the rise of ViTs [53] has also significantly impacted the field of MDE. These models employ the self-attention mechanism of the transformer to aggregate depth information across a more extensive field of view to capture long-range dependencies and global context, leading to more accurate and consistent depth maps. Inherently, multiple approaches have successfully adapted ViTs to MDE [2, 160, 289, 3, 303, 200, 216, 56, 293, 211, 124].

Hybrid Approaches. Beyond more traditional regression-based methods, some works creatively treat MDE as a combined regression-classification task. By discretizing the depth range into bins, these methods simplify the learning task and improve performance in some cases. Notable examples include AdaBins [13], BinsFormer [161], and LocalBins [14].

Generative Models. A more recent trend in MDE includes repurposing generative models such as diffusion models [54, 120, 237, 238, 124, 207], effectively building on its predecessor, generative adversarial networks [42, 5].

Depth in the Wild. Estimating depth "in the wild" has become an increasingly important area of research in MDE. It refers to the challenge of predicting accurate depth estimates in unconstrained environments, where lighting, scene structure, and camera parameters vary significantly. With the increasing availability of computing resources, researchers have found scaling to be a valuable tool to tackle this challenge. By constructing large and diverse depth datasets and leveraging powerful foundation models, MDE has become more accessible and robust to real-world use. Foundation models are large neural networks pre-trained on internet-scale data, which allows them to develop a high-level visual understanding that can either be used directly or fine-tuned for a variety of downstream tasks [20].

Ordinal Depth. One of the earlier works aimed at addressing depth in the wild by leveraging the scale of data [31]. Building on this idea, Chen et al. [32] introduced the OASIS dataset, a large-scale dataset specifically designed for depth and normal estimation. It is worth noting, however, that both of these works primarily focus on relative (ordinal) depth, which only estimates the depth order instead of providing absolute measurements. While ordinal depth can provide valuable information about the structure of a given scene, its practical use is limited.

Affine-invariant Depth. To overcome the limitations of ordinal depth, several studies have explored affine-invariant depth estimation, which provides depth estimates up to an unknown affine transformation. In other words, the absolute scale and offset of the depth map can vary while the relative depth differences are preserved. For instance, models trained on the MegaDepth dataset [159], which uses multi-view internet photo collections along with structure-from-motion and multi-view stereo methods to create depth maps, can generalize well to unseen images. Another significant contribution is MiDaS [217], which achieves the ability to generalize across a variety of scenes and conditions through training on a mixture of multiple datasets.

Metric Depth. For applications that often require absolute distances, such as robotics [52, 229], augmented reality [123], and autonomous driving [288, 283], metric depth estimation is crucial. Metric depth estimation aims to provide absolute depth measurements in real-world units (e.g., meters or centimeters). Inconveniently, zero-shot generalization is particularly challenging due to the metric ambiguities introduced by different camera models. Aside from some works that explicitly incorporate camera intrinsics as an additional input [83, 292] to directly solve this issue, current state-of-the-art metric depth estimation approaches still rely on fine-tuning powerful foundation models in the respective domain [14, 212, 290, 291].

Uncertainty Quantification

In the following, we will go over previous attempts to fuse UQ with MDE. For a broader review of related work on UQ, refer to the corresponding Section 2.2 in the fundamentals Chapter.

Uncertainty Quantification in Self-supervised Monocular Depth Estimation. While substantial progress has been made in developing effective UQ methods, integrating these techniques with MDE comes with some unique challenges like the limited ground truth data, which is expensive and difficult to acquire, especially at scale. For that reason, a significant body of research has focused on UQ in self-supervised MDE [213, 103, 199, 34, 51]. While they all explore different strategies to estimate the uncertainty, some even manage to leverage the uncertainty to enhance model performance [213, 199, 34].

Uncertainty Quantification in Supervised Monocular Depth Estimation. For supervised learning scenarios, one common approach to UQ involves modeling the regression output as a parametric distribution and training the model to estimate its parameters [126, 201]. This approach allows the model to not only output the depth estimate but also to measure the corresponding uncertainty, typically represented as the variance of the distribution. Similarly, Yu et al. [294] propose an auxiliary network that exploits the output and the intermediate representations of the main model to estimate the uncertainty. Hornauer et al. [107] introduce a post hoc UQ method that relies on gradients extracted with an auxiliary loss function. This technique utilizes TTA [9] to investigate the correspondence of the depth prediction for an image and its horizontally flipped counterpart. Another innovative approach is proposed by Franchi et al. [62], who address computational efficiency by optimizing a set of latent prototypes. The uncertainty is quantified by examining the position of an input sample in the prototype space. Lastly, Mi et al. [182] developed a training-free UQ approach based on tolerable perturbations during inference and using the variance of multiple outputs as a surrogate for the uncertainty.

4.2 Baseline Model

DepthAnythingV2 [291] is one of the most recent state-of-the-art foundation models for MDE, which can easily be fine-tuned for metric depth estimation, making it the perfect candidate for exploring the combination of various UQ methods with MDE foundation models. Consequently, we want to provide a more detailed overview of this model, which was built upon the framework established by its predecessor, DepthAnythingV1 [290].

DepthAnythingV1. Yang et al. [291] laid the groundwork for creating a versatile foundation model for MDE using the DINOv2 encoder [203] for feature extraction with the DPT decoder [216] for depth regression. The training process of DepthAnythingV1 involves a semi-supervised approach using a student-teacher framework. The teacher model generates pseudo-labels for an extremely large corpus of unlabeled images (approx. 62 million from 8 public datasets), while the student is trained on both the pseudo-labels and a set of 1.5 million labeled images from 6 public datasets. To ensure robustness of the learned representations, they additionally apply strong image perturbations for the student. These include strong color distortion like color jittering and Gaussian blurring, and strong spatial distortion through CutMix [296].

To further refine the model's capabilities, they also introduce an auxiliary feature alignment loss. It measures the cosine similarity between the features of the student model and those of a frozen DINOv2 encoder, which is a powerful model for semantically related tasks like image retrieval and Semantic Segmentation (SS). This potent addition to the training process helps imbue the DepthAnythingV1 model with high-level semantic understanding to further improve the depth estimation.

DepthAnythingV2. Building on the success of DepthAnythingV1, Yang et al. [291] quickly introduced several key advancements that enable finer and more robust depth predictions. These improvements are centered around the following three strategies:

1. **Synthetic Data for Label Accuracy:** One of the most significant changes for DepthAnythingV2 is the replacement of all labeled real images with synthetic images. This alteration is motivated by the desire to eliminate label noise and address the lack of detail often ignored in real datasets. In contrast to real images, synthetic data allows for precise depth training and avoids the inconsistencies found in real-world labels.
2. **Scaling up the Teacher Model:** To mitigate the drawbacks of synthetic images, such as distribution shifts and restricted scene coverage, the capacity of the teacher model significantly increased. DepthAnythingV2 employs DINOv2-G, the most powerful variant of the DINOv2 encoder [216].
3. **Leveraging Pseudo-Labeled Real Images:** To bridge the gap between synthetic images and the complexity of real-world scenes, DepthAnythingV2 incorporates large-scale pseudo-labeled real images into its training pipeline. This not only expands the lacking scene coverage of the synthetic images but also ensures that the model is exposed to a wide variety of real-world scenarios, improving its generalization capabilities.

These advancements establish DepthAnythingV2 as one of the most powerful currently available MDE foundation models, combining high-quality depth predictions with robust fine-tuning capabilities for metric depth estimation. Given these attributes, DepthAnythingV2 serves as an ideal candidate for exploring foundation model uncertainty in MDE. We aim to assess various UQ methods in combination with this foundation model, enabling accurate metric depth estimation and pixel-wise uncertainty measures for real-world applications.

4.3 Combining Uncertainty Quantification with Foundation Models

The subsequent Section explores the critical synthesis of UQ and MDE foundation models to balance the benefits of large-scale pre-training with the need for reliable uncertainty estimates. By evaluating the DepthAnythingV2 foundation model with five distinct UQ methods, we investigate their feasibility to provide pixel-wise uncertainty estimates for real-world deployment.

Research Gap. Despite significant advancements in UQ for MDE, integrating these techniques with large-scale foundation models remains unexplored. We aim to address this gap by combining multiple UQ methods with the state-of-the-art DepthAnythingV2 foundation model [291], enabling pixel-wise uncertainty estimates in addition to metric depth measurements. Our findings contribute to a more nuanced understanding of the capabilities and limitations of these models, emphasizing the importance of not only striving for higher performance but also making MDE more reliable and trustworthy for real-world use.

4.3.1 Methodology

Our primary research question is straightforward: How can we bridge the gap between groundbreaking results in research and safe deployment in real-world applications that need robust metric depth estimates with corresponding uncertainties?

As Figure 4.2 shows, we study five different approaches to not only estimate metric depths but also their corresponding uncertainties: LC [269], GNLL [201], MCD [67], DSEs [263], and TTA [9]. DEs are not employed here, as they require random weight initialization for at least parts of the network [138, 61], which would prohibit leveraging the pre-trained weights of the DepthAnythingV2 model, which are essential for the competitive metric depth estimation performance that we want to preserve at all cost.

The upper two approaches of Figure 4.2, LC and GNLL, are fairly simple since they only require adding a second output channel to the already existing depth head. The first output channel generates metric depth maps, while the second yields uncertainty estimates.

The third option, MCD, is equally straightforward since it only requires activating all of the already existing dropout layers in the model while fine-tuning. During inference, these dropout layers are kept active, and multiple depth outputs are sampled. By computing the mean and variance, the final depth map and the corresponding uncertainty are obtained.

The fourth option, DSE, is possibly the most complicated and requires significant changes to the architecture. Instead of using just one depth head, a DSE of randomly initialized depth heads is created. During inference, every depth head predicts slightly different depth samples. Similar to the third option, MCD, the final depth map and uncertainty can be obtained by computing the mean and variance.

Finally, as Figure 4.2 shows, we also examine TTA, which does not require any changes to the fine-tuning process of DepthAnythingV2. Instead, we apply horizontal and vertical flipping during test-time to create two additional inputs, to create a total of three unique depth samples. Based on these, we compute the mean and variance to obtain the final depth map and the uncertainty.

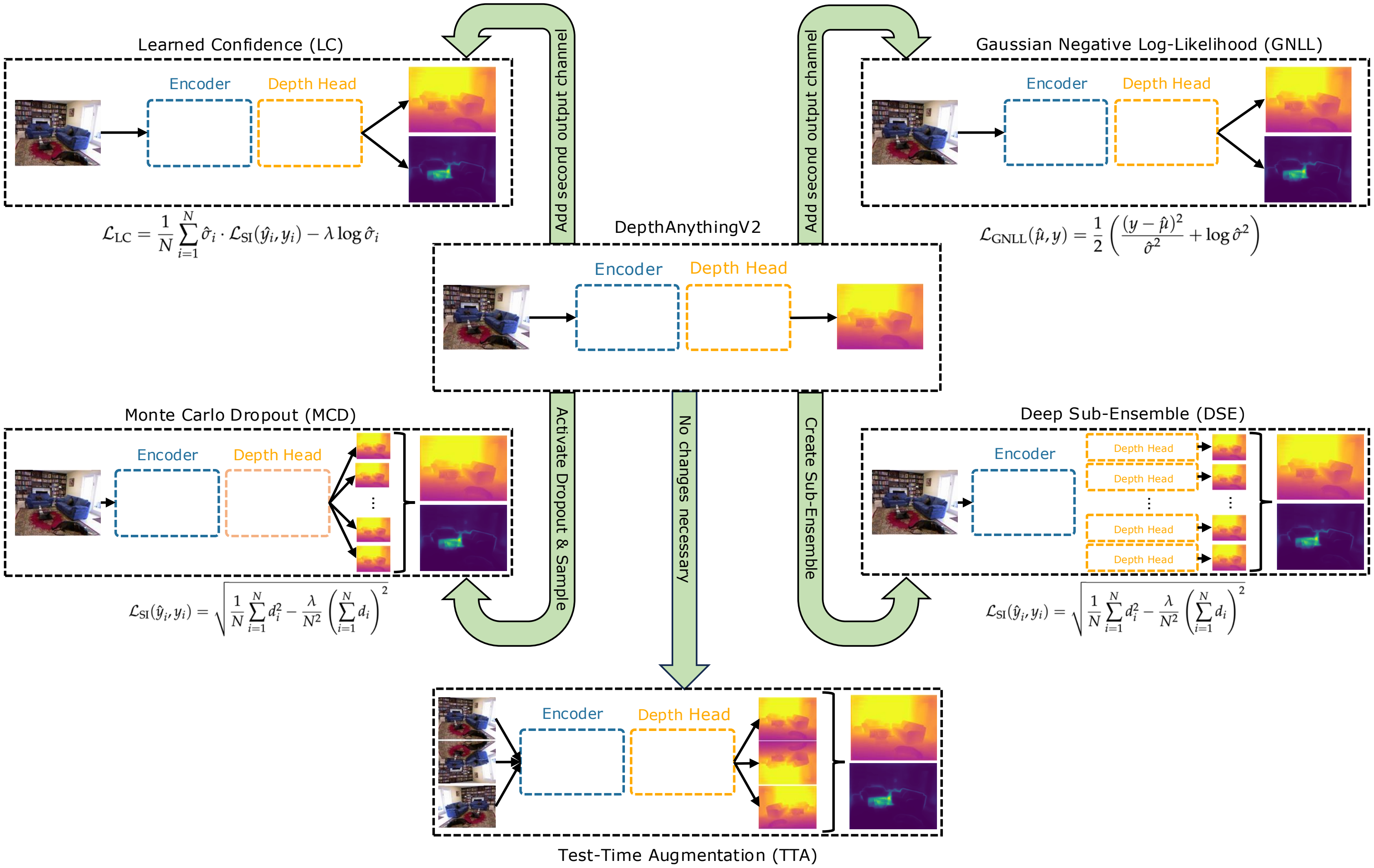

Learned Confidence (LC)
Encoder
Depth Head
$\mathcal{L}_{\text{LC}} = \frac{1}{N}\sum_{i=1}^{N} \hat{\sigma}_i \cdot \mathcal{L}_{\text{SI}}(\hat{y}_i, y_i) - \lambda \log \hat{\sigma}_i$
Add second output channel
Add second output channel
Gaussian Negative Log-Likelihood (GNLL)
Encoder
Depth Head
$\mathcal{L}_{\text{GNLL}}(\hat{\mu}, y) = \frac{1}{2}\left(\frac{(y-\hat{\mu})^2}{\hat{\sigma}^2} + \log \hat{\sigma}^2\right)$
DepthAnythingV2
Encoder
Depth Head
Monte Carlo Dropout (MCD)
Encoder
Depth Head
$\mathcal{L}_{\text{SI}}(\hat{y}_i, y_i) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2 - \frac{\lambda}{N^2}\left(\sum_{i=1}^{N} d_i\right)^2}$
Activate Dropout & Sample
No changes necessary
Create Sub-Ensemble
Deep Sub-Ensemble (DSE)
Encoder
Depth Head
Depth Head
Depth Head
Depth Head
$\mathcal{L}_{\text{SI}}(\hat{y}_i, y_i) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2 - \frac{\lambda}{N^2}\left(\sum_{i=1}^{N} d_i\right)^2}$
Encoder
Depth Head
Test-Time Augmentation (TTA)

Figure 4.2: A schematic overview of how to fuse the five different UQ approaches with the DepthAnythingV2 foundation model [291].

Baseline Model

To enable a fair and consistent comparison across all five uncertainty-aware methods, we use the standard DepthAnythingV2 model [291] as a baseline. For training the baseline models, we simply follow the fine-tuning recommendations of DepthAnythingV2 [291]. In the case of metric depth estimation, they suggest fine-tuning solely using the scale-invariant loss

$$\mathcal{L}_{\mathrm{SI}}(\hat{y}_i, y_i) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2 - \frac{\lambda}{N^2}\left(\sum_{i=1}^{N} d_i\right)^2} \ , \tag{4.1}$$

where N is the number of pixels with valid ground truth, $\hat{y}_i$ is the predicted depth of the i-th pixel, y_i is the true depth, $d_i = \log y_i - \log \hat{y}_i$, and $\lambda = 0.2$.

Learned Confidence

The general approach of LC was originally proposed by Wan et al. [269] for classification tasks, but has already been adapted before for regression tasks by Wang et al. [274]. The confidences, which we interpret as uncertainties, can simply be learned in addition to the primary objective function by adding a second output channel to the depth head for the uncertainty and optimizing

$$\mathcal{L}_{\mathrm{LC}} = \frac{1}{N}\sum_{i=1}^{N} \hat{\sigma}_i \cdot \mathcal{L}_{\mathrm{SI}}(\hat{y}_i, y_i) - \lambda \log \hat{\sigma}_i \ , \tag{4.2}$$

where $\hat{\sigma}_i$ is the confidence (uncertainty) and $\lambda = 0.2$, following Wang et al. [274].

Gaussian Negative Log-Likelihood Loss

For depth regression, neural networks are usually only trained to output a predictive mean $\hat{\mu}$. To also approximate the corresponding variance $\hat{\sigma}^2$, i.e., the uncertainty, we follow the approach of Nix and Weigend [201]: By treating the neural network prediction as a sample from a Gaussian distribution and adding a second output channel to the depth head for the uncertainty, we can minimize the GNLL loss

$$\mathcal{L}_{\mathrm{GNLL}}(\hat{\mu}, y) = \frac{1}{2}\left(\frac{(y-\hat{\mu})^2}{\hat{\sigma}^2} + \log \hat{\sigma}^2\right) \ . \tag{4.3}$$

Analogous to Equation 4.2, there is no ground truth for the uncertainty, which means that $\hat{\sigma}^2$ is solely learned implicitly through the optimization of the predictive means $\hat{\mu}$ based on the ground truth labels y.

Monte Carlo Dropout

Using MCD [67] to estimate the predictive mean $\hat{\mu}$ and the corresponding uncertainty, i.e., the variance $\hat{\sigma}^2$, is fairly straightforward. Since the DepthAnythingV2 model already applies dropout layers throughout its architecture, we simply have to activate them not only during training but also during inference to sample from the posterior of the network.

To compute the predictive mean $\hat{\mu}$, we take the average of all the samples:

$$\hat{\mu} = \frac{1}{T} \sum_{t=1}^{T} \hat{y}_t \ , \tag{4.4}$$

where T is the number of samples and $\hat{y}_t$ is the t-th depth prediction of the model.

The uncertainty can be quantified by the variance

$$\hat{\sigma}^2 = \frac{1}{T-1} \sum_{t=1}^{T} (\hat{y}_t - \hat{\mu})^2 \ . \tag{4.5}$$

We adhere to the fine-tuning recommendations of DepthAnythingV2 [291], using the scale-invariant loss $\mathcal{L}_{SI}(y, \hat{y})$ from Equation 4.1 as the objective function.

Deep Sub-Ensemble

DSEs [263] enable the approximation of DEs. While a DE requires multiple models to be trained and used during inference, the DSE only requires a subset of the layers to be multiplied. As shown by Figure 4.2, we use a shared encoder for multiple randomly initialized depth heads. To maximize the diversity across the depth heads and decrease the training time, we cycle through the heads during training. Per training batch, only one head is optimized; the others are ignored. During inference, however, each depth head predicts a unique sample $\hat{y}_t$ based on the extracted feature from the encoder. Similar to MCD, we can compute the mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of all the samples (cf. Equations 4.4 and 4.5) to get the desired output, i.e., a final depth prediction and a corresponding uncertainty.

As for MCD, we follow the fine-tuning recommendations of DepthAnythingV2 [291], using the scale-invariant loss $\mathcal{L}_{SI}(y, \hat{y})$ from Equation 4.1 as the objective function.

Test-Time Augmentation

In contrast to the other four UQ approaches, we just fine-tune the DepthAnythingV2 model with the scale-invariant loss $\mathcal{L}_{SI}$ from Equation 4.1 for metric depth estimation and apply TTA after training [9]. As shown by Figure 4.2, we flip the input image vertically as well as horizontally and perform inference with each. As a result, we obtain three unique depth samples $\hat{y}_t$ that we can use to compute the mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ (cf. Equations 4.4 and 4.5).

4.3.2 Experiments

In this Section, we conduct an extensive set of experiments to answer the question on how to fuse UQ with metric MDE. Firstly, we describe the experimental setup. Secondly, we provide quantitative results comparing all five UQ methods applied to the DepthAnythingV2 foundation model on four diverse datasets with varying encoder sizes. Lastly, we offer qualitative examples of the UQ methods for all four datasets, showcasing the potential for foundation model uncertainty in metric MDE.

Experimental Setup

Training. For all training processes, we follow the default settings of DepthAnythingV2 for metric depth fine-tuning [291], using an AdamW optimizer [169] with a base learning rate of $6 \cdot 10^{-5}$, a weight decay of 0.01, and a polynomial learning rate scheduler (cf. Equation 3.6). Every model is trained for 25 epochs with an effective batch size of 16, using four NVIDIA A100 GPUs. We do not employ any early stopping techniques and hence only evaluate the final model checkpoints.

Datasets. We conduct our experiments on four highly different datasets, simulating a broad range of real-world applications, as shown by Table 4.1. Cityscapes [39] provides an urban street scene benchmark dataset with high-resolution images. In contrast, NYUv2 [245] presents indoor scenes with a very low image resolution. UseGeo [197] covers high-resolution aerial images, which are often neglected in the computer vision community despite their significance in many real-world applications. Finally, the HOPE [261] dataset offers a variety of household objects, originally designed for Pose Estimation. The main reason why the HOPE dataset is so interesting is that the training dataset is based on almost 50,000 synthetic images, whereas the test dataset consists of just 457 real images. A unique challenge that is often overlooked but fairly common, especially in robotics [104, 256, 105].

	Scene / Application	Resolution (W × H)	Training Images	Test Images
Cityscapes [39]	Outdoor	2048 × 1024	2975	500
NYUv2 [245]	Indoor	640 × 480	795	654
UseGeo [197]	Aerial	1989 × 1320	551	277
HOPE [261]	Robotics	1920 × 1080	49450 (synth.)	457 (real)

Table 4.1: Overview of metric depth datasets that we used for evaluation.

Data Augmentations. Regardless of the trained model, we apply random cropping with a crop size of 756 × 756 pixels on all datasets except NYUv2, which uses 630 × 476 pixels, and random horizontal flipping with a flip chance of 50 %. For testing purposes, we use the original image resolutions as shown by Table 4.1.

Metrics. For quantitative evaluations of the metric depth estimation, we report all standard metrics widely used in MDE research [177, 8]:

1. Root Mean Squared Error (RMSE), which measures overall depth prediction accuracy,
2. Absolute Relative Error (AbsRel), which captures relative depth deviations,
3. Logarithmic Mean Absolute Error (log10), which emphasizes errors on a logarithmic scale,
4. and three threshold-based accuracies (δ_1, δ_2, δ_3), which assess the proportion of predictions within specific error thresholds.

In contrast to classification tasks like SS, prediction errors are continuous in MDE. As a result, there is no natural binary notion of correctness, prohibiting straightforward utilization of calibration metrics like Expected Calibration Error (ECE), which rely on categorical outcomes.

To evaluate the uncertainty quality in this setting, we instead adopt the following metrics proposed by Mukhoti et al. [190]:

1. **p(accurate|certain)**: The probability that the model is accurate on its output given that the uncertainty is below a specified threshold.
2. **p(uncertain|inaccurate)**: The probability that the uncertainty of the model exceeds a specified threshold given that the prediction is inaccurate.
3. **PAvPU**: The combined measure that captures both accurate pixels among certain predictions and uncertain pixels among inaccurate predictions.

The conditional probabilities p(accurate|certain) and p(uncertain|inaccurate) can be calculated as

$$\begin{aligned} p(\text{accurate}|\text{certain}) &= \frac{n_{ac}}{(n_{ac} + n_{ic})} \ , \\ p(\text{uncertain}|\text{inaccurate}) &= \frac{n_{iu}}{(n_{iu} + n_{ic})} \ , \end{aligned} \tag{4.6}$$

where n_{ac} represents the number of pixels that are accurate and certain, n_{ic} the number of pixels that are inaccurate and certain, and n_{iu} the number of pixels that are inaccurate and uncertain. Finally, we can combine both desired cases of accurate and certain pixels as well as inaccurate and uncertain pixels to compute PAvPU as

$$\text{PAvPU} = \frac{(n_{ac} + n_{iu})}{n_{ac} + n_{au} + n_{ic} + n_{iu}} \ , \tag{4.7}$$

where n_{au} is the number of pixels that are accurate and uncertain. Clearly, these metrics depend on the choice of the accuracy threshold as well as the uncertainty threshold, which is what we will go over next.

To determine whether a depth prediction is accurate or inaccurate, we use the strictest threshold-based accuracy δ_1, which can be defined as

$$\delta_1 = max\left(\frac{y}{\hat{y}}, \frac{\hat{y}}{y}\right) \ . \tag{4.8}$$

In this case, the accuracy is the percentage of pixels where $\delta_1 < 1.25$, indicating that the predicted depth is within 25 % of the ground truth depth.

To simulate real-world deployment scenarios, we define a pixel as certain or uncertain based on whether its uncertainty falls below or above the median uncertainty of a given image. This approach is based on findings from Section 5.3.2 in the following Chapter 5, where a comparison between multiple uncertainty thresholds demonstrated that the median serves as the most robust default choice.

Uncertainty Quantification. Unless stated otherwise, for MCD and DSE, the final depth map and corresponding uncertainty are computed using ten samples or ten depth heads, respectively, in accordance with the findings in [138, 61, 147, 143].

Quantitative Evaluation

Efficiency. As shown by Table 4.2, training times are comparable to the DepthAnythingV2 baseline model for all methods, with LC and GNLL matching it exactly in both training and inference time. DSE and MCD increase training times by around 5 % - 10 %. While DSE nearly triples inference time and roughly doubles the trainable parameters, MCD requires roughly 10 times the inference time due to the costly sampling process that roughly scales linearly with the number of samples. TTA also triples inference time as it requires three forward passes.

NYUv2 (indoor)		Trainable Parameters [M] ↓	FLOPs [G] ↓	Training Time [mm:ss] ↓	Inference Time [ms] ↓	FPS ↑
ViT-S	Baseline	24.8	66.82	00:27	12.9 ± 0.1	77.5
	TTA	24.8	66.82	00:27	40.0 ± 0.9	25.0
	LC	24.8	66.83	00:27	12.9 ± 0.2	77.5
	GNLL	24.8	66.83	00:27	12.9 ± 0.2	77.5
	DSE	49.3	177.29	00:29	36.5 ± 0.5	27.4
	MCD	24.8	66.82	00:30	128.4 ± 0.5	7.8
ViT-B	Baseline	97.5	217.49	00:48	24.9 ± 0.5	40.2
	TTA	97.5	217.49	00:48	75.5 ± 1.2	13.2
	LC	97.5	217.50	00:49	25.1 ± 0.5	39.8
	GNLL	97.5	217.50	00:49	25.0 ± 0.7	40.0
	DSE	195.5	608.09	00:52	61.9 ± 0.8	16.2
	MCD	97.5	217.49	00:54	248.4 ± 2.7	4.0
ViT-L	Baseline	335.3	741.40	02:11	57.1 ± 3.5	17.5
	TTA	335.3	741.40	02:11	172.6 ± 11.6	5.8
	LC	335.3	741.41	02:14	57.0 ± 3.9	17.5
	GNLL	335.3	741.41	02:13	57.1 ± 3.3	17.5
	DSE	613.8	2204.07	02:17	131.6 ± 5.2	7.6
	MCD	335.3	741.40	02:26	569.8 ± 24.3	1.8

Table 4.2: Efficiency comparison between the five chosen UQ methods: TTA, LC, GNLL, DSE, and MCD for three different encoder sizes: ViT-S, ViT-B, ViT-L on the NYUv2 [245] dataset. We compare the number of trainable parameters, FLOPs, training time per epoch on a single A100 GPU, inference time per image, and the respective FPS. The mean inference time and corresponding standard deviation are based on 1000 forward passes.

NYUv2. Table 4.3 shows a quantitative comparison for the NYUv2 dataset [245]. LC, GNLL, and DSE maintain depth quality similar to the baseline, with DSE and GNLL even surpassing it for the ViT-S and ViT-B encoders, respectively. TTA and MCD exhibit a slight degradation but remain competitive.

Regarding the uncertainty quality, GNLL emerges as the top-performing method, outperforming all others in terms of p(acc|cer) and p(unc|inacc) across all three encoder sizes, achieving impressive values of up to 98.0 % and 91.2 %, respectively. For PAvPU, both GNLL and MCD deliver the best results. While all other methods are somewhat competitive with each other, LC clearly falls behind concerning p(unc|inacc), achieving only 26.8 % for ViT-L.

Cityscapes. For the Cityscapes dataset [39], as shown by Table 4.4, TTA stands out by significantly outperforming the baseline across all three encoder sizes in terms of depth quality. In contrast, LC, GNLL, and DSE exhibit noticeable degradation, while MCD performs the worst.

For uncertainty quality, GNLL is the standout performer, decisively surpassing all other methods for all three metrics and encoder sizes. It achieves remarkable values of 69.2 % for p(acc|cer), 70.6 % for p(unc|inacc), and 69.5 % for PAvPU, setting a benchmark for reliability in this dataset. The remaining methods deliver less consistent results, with TTA generally performing the worst, showing values as low as 28.8 % for p(acc|cer), 42.0 % for p(unc|inacc), and 39.6 % for PAvPU.

UseGeo. For the UseGeo dataset [197], as presented by Table 4.5, the depth quality results are inconsistent, with methods sometimes surpassing and at other times falling short of the baseline. TTA is the only approach that consistently outperforms the baseline across all three encoder sizes.

In terms of uncertainty quality, all methods deliver near-perfect results for p(acc|cer) of at least 96.3 %. For p(unc|inacc) and PAvPU, MCD generally performs best with values of up to 67.2 % and 50.4 %, respectively. In contrast to the other datasets, GNLL significantly lags behind the other methods on UseGeo. This is likely due to the large depth values in UseGeo, which cause higher GNLL loss values. The GNLL loss includes a logarithmic term that penalizes high uncertainty estimates, which are amplified by large depth values. Notably, no hyperparameter adjustments were made to address this, ensuring comparability but potentially hindering GNLL's optimization in this particular case.

HOPE. Table 4.6 shows the final quantitative comparison for the HOPE dataset [261]. In terms of depth quality, there are only marginal differences between the baseline and all uncertainty approaches, with RMSE values ranging between 0.215 to 0.277.

Regarding the uncertainty quality, GNLL once again asserts itself as the best option for all three metrics across all three encoder sizes, with only one minor exception. GNLL achieves results of up to 49.7 % for p(acc|cer), 62.2 % for p(unc|inacc), and 59.6 % for PAvPU. Mirroring its strong performance on NYUv2 and Cityscapes (cf. Tables 4.3 and 4.4), GNLL's dominance is evident, while the other methods remain fairly competitive with each other. LC, however, lags significantly behind across all three uncertainty quality metrics, underscoring GNLL's reliability and consistency.

NYUv2 (indoor)		RMSE ↓	AbsRel ↓	log10 ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑	p(acc|cer) ↑	p(unc|ina) ↑	PAvPU ↑
ViT-S	Baseline	**0.340**	0.093	0.039	0.928	0.988	0.997	-	-	-
	TTA	0.399	0.111	0.049	0.881	0.984	0.997	0.903	0.602	0.522
	LC	0.343	0.090	0.039	0.930	0.988	0.997	0.920	0.369	0.490
	GNLL	0.342	0.094	0.040	0.924	0.987	0.997	**0.953**	**0.846**	0.529
	DSE	**0.340**	0.092	0.039	0.926	0.988	0.997	0.937	0.704	0.511
	MCD (10%)	0.422	0.121	0.050	0.867	0.973	0.992	0.900	0.692	**0.533**
ViT-B	Baseline	0.307	0.080	0.034	0.948	0.991	0.998	-	-	-
	TTA	0.359	0.099	0.435	0.910	0.988	0.998	0.924	0.599	0.514
	LC	0.314	0.085	0.036	0.943	0.991	0.998	0.926	0.292	0.483
	GNLL	**0.305**	0.079	0.034	0.949	0.991	0.998	**0.966**	**0.826**	0.517
	DSE	0.309	0.080	0.034	0.947	0.991	0.998	0.955	0.698	0.507
	MCD (10%)	0.339	0.091	0.039	0.925	0.986	0.997	0.952	0.774	**0.527**
ViT-L	Baseline	**0.270**	0.068	0.030	0.964	0.993	0.998	-	-	-
	TTA	0.324	0.087	0.039	0.938	0.992	0.998	0.944	0.598	0.507
	LC	0.275	0.069	0.030	0.963	0.993	0.998	0.946	0.268	0.483
	GNLL	0.285	0.072	0.031	0.959	0.992	0.998	**0.980**	**0.912**	0.521
	DSE	0.280	0.070	0.030	0.961	0.993	0.998	0.969	0.734	0.508
	MCD (10%)	0.339	0.091	0.039	0.925	0.986	0.997	0.967	0.798	**0.522**

Table 4.3: Quantitative comparison on the NYUv2 [245] dataset between the five chosen UQ methods: TTA, LC, GNLL, DSE, and MCD for three different encoder sizes: ViT-S, ViT-B, ViT-L. Best results for RMSE and the three uncertainty metrics are marked in **bold** for each encoder.

Cityscapes (outdoor)		RMSE ↓	AbsRel ↓	log10 ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑	p(acc|cer) ↑	p(unc|ina) ↑	PAvPU ↑
ViT-S	Baseline	7.138	0.219	0.084	0.704	0.964	0.991	-	-	-
	TTA	**6.474**	0.223	0.089	0.576	0.965	0.991	0.335	0.421	0.410
	LC	7.492	0.206	0.078	0.733	0.958	0.991	0.658	0.619	0.611
	GNLL	7.739	0.236	0.089	0.669	0.949	0.987	**0.692**	**0.706**	**0.695**
	DSE	7.714	0.234	0.088	0.681	0.954	0.990	0.658	0.654	0.649
	MCD (10%)	8.527	0.307	0.113	0.417	0.920	0.984	0.328	0.525	0.516
ViT-B	Baseline	6.884	0.252	0.095	0.599	0.965	0.992	-	-	-
	TTA	**5.757**	0.247	0.095	0.526	0.967	0.993	0.288	0.420	0.396
	LC	7.711	0.285	0.107	0.434	0.958	0.991	0.329	0.506	0.502
	GNLL	7.092	0.244	0.094	0.613	0.966	0.991	**0.594**	**0.635**	**0.629**
	DSE	7.824	0.271	0.102	0.534	0.957	0.991	0.490	0.587	0.588
	MCD (10%)	8.268	0.288	0.107	0.488	0.939	0.989	0.449	0.579	0.583
ViT-L	Baseline	6.655	0.256	0.097	0.558	0.972	0.993	-	-	-
	TTA	**5.298**	0.227	0.088	0.608	0.979	0.994	0.371	0.431	0.416
	LC	7.392	0.280	0.105	0.416	0.969	0.993	0.292	0.488	0.488
	GNLL	6.562	0.234	0.092	0.628	0.970	0.990	**0.581**	**0.620**	**0.607**
	DSE	7.522	0.272	0.103	0.500	0.966	0.993	0.446	0.568	0.570
	MCD (10%)	8.268	0.288	0.107	0488	0.939	0.989	0.480	0.584	0.584

Table 4.4: Quantitative comparison on the Cityscapes [39] dataset between the five chosen UQ methods: TTA, LC, GNLL, DSE, and MCD for three different encoder sizes: ViT-S, ViT-B, ViT-L. Best results for RMSE and the three uncertainty metrics are marked in **bold** for each encoder.

UseGeo (aerial)		RMSE ↓	AbsRel ↓	log10 ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑	p(acc|cer) ↑	p(unc|ina) ↑	PAvPU ↑
ViT-S	Baseline	7.366	0.077	0.032	0.973	0.994	0.999	-	-	-
	TTA	7.078	0.074	0.031	0.977	0.995	0.999	0.976	0.647	0.500
	LC	8.213	0.086	0.036	0.958	0.995	1.000	0.963	0.481	**0.506**
	GNLL	7.467	0.076	0.032	0.979	0.993	0.998	0.976	0.237	0.500
	DSE	7.259	0.076	0.032	0.976	0.994	0.999	0.971	0.467	0.495
	MCD (10%)	**6.682**	0.068	0.029	0.982	0.994	0.998	**0.982**	**0.672**	0.501
ViT-B	Baseline	6.386	0.067	0.028	0.981	0.995	0.999	-	-	-
	TTA	**6.060**	0.631	0.027	0.985	0.995	0.999	**0.984**	0.609	0.499
	LC	6.612	0.068	0.029	0.975	0.995	1.000	0.967	0.454	0.492
	GNLL	7.810	0.080	0.035	0.972	0.990	0.997	0.971	0.293	0.499
	DSE	6.491	0.068	0.028	0.980	0.994	0.999	0.981	0.559	0.502
	MCD (10%)	6.641	0.070	0.029	0.978	0.994	0.999	0.982	**0.657**	**0.504**
ViT-L	Baseline	6.173	0.065	0.027	0.981	0.995	0.999	-	-	-
	TTA	5.898	0.063	0.026	0.982	0.995	0.999	0.980	0.596	0.499
	LC	**5.406**	0.056	0.024	0.986	0.995	1.000	**0.985**	0.477	0.500
	GNLL	7.082	0.073	0.031	0.980	0.991	0.998	0.980	0.294	0.498
	DSE	6.260	0.067	0.028	0.980	0.995	1.000	0.977	0.577	0.497
	MCD (10%)	6.697	0.071	0.029	0.981	0.995	0.999	0.982	**0.669**	**0.501**

Table 4.5: Quantitative comparison on the UseGeo [197] dataset between the five chosen UQ methods: TTA, LC, GNLL, DSE, and MCD for three different encoder sizes: ViT-S, ViT-B, ViT-L. Best results for RMSE and the three uncertainty metrics are marked in **bold** for each encoder.

HOPE (robotics)		RMSE ↓	AbsRel ↓	log10 ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑	p(acc|cer) ↑	p(unc|ina) ↑	PAvPU ↑
ViT-S	Baseline	0.263	0.265	0.115	0.537	0.821	0.942	-	-	-
	TTA	0.264	0.262	0.114	0.539	0.818	0.943	**0.421**	0.564	0.552
	LC	**0.259**	0.262	0.114	0.537	0.822	0945	0.339	0.476	0.474
	GNLL	0.277	0.287	0.124	0.492	0.795	0.929	0.404	**0.575**	**0.567**
	DSE	0.262	0.263	0.117	0.544	0.809	0.933	0.393	0.528	0.522
	MCD (10%)	0.274	0.286	0.118	0.514	0.816	0.937	0.398	0.562	0.553
ViT-B	Baseline	0.222	0.232	0.094	0.616	0.892	0.969	-	-	-
	TTA	0.221	0.230	0.093	0.621	0.891	0.968	0.462	0.573	0.558
	LC	0.224	0.227	0.095	0.616	0.883	0.966	0.330	0.419	0.427
	GNLL	0.223	0.225	0.094	0.619	0.892	0.972	**0.497**	**0.622**	**0.596**
	DSE	**0.218**	0.230	0.094	0.625	0.889	0.967	0.448	0.546	0.543
	MCD (10%)	0.245	0.269	0.106	0.560	0.851	0.955	0.405	0.555	0.543
ViT-L	Baseline	0.223	0.238	0.096	0.588	0.906	0.980	-	-	-
	TTA	0.217	0.232	0.094	0.599	0.904	0.980	0.436	0.576	0.557
	LC	**0.215**	0.226	0.092	0.604	0.911	0.980	0.325	0.426	0.441
	GNLL	0.229	0.244	0.099	0.588	0.888	0.973	**0.460**	**0.608**	**0.586**
	DSE	0.235	0.252	0.098	0.594	0.893	0.974	0.442	0.574	0.569
	MCD (10%)	0.249	0.272	0.107	0.557	0.859	0.966	0.416	0.580	0.563

Table 4.6: Quantitative comparison on the HOPE [261] dataset between the five chosen UQ methods: TTA, LC, GNLL, DSE, and MCD for three different encoder sizes: ViT-S, ViT-B, ViT-L. Best results for RMSE and the three uncertainty metrics are marked in **bold** for each encoder.

Qualitative Evaluation

We provide handpicked qualitative examples of all five different UQ methods with varying encoder sizes for all four datasets in Figure 4.3, highlighting the potential for foundation model uncertainty in metric MDE.

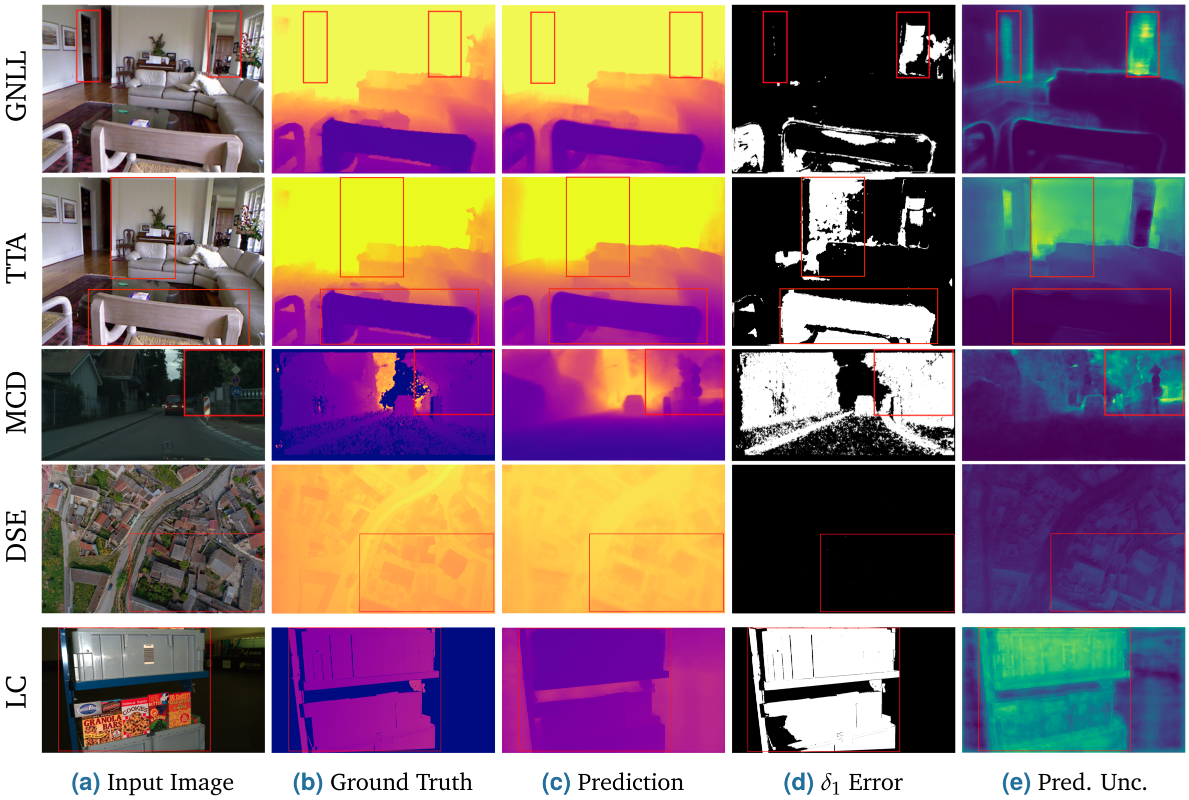


Figure 4.3: Qualitative examples for indoor [245], outdoor [39], aerial [197], and robotics [261] scenarios with varying UQ approaches and encoder sizes. Red rectangles are added to highlight interesting areas.

NYUv2. The first row shows the GNLL (ViT-S) results, demonstrating high prediction quality overall. Uncertainty is notably elevated around object boundaries and the two open doors in the background, suggesting these regions are edge cases. This likely stems from the model's limited exposure to such depth ranges during training, as the maximum depth is limited to 5m, which is common practice on NYUv2.

The second row presents results from TTA (ViT-S), revealing mixed results in terms of prediction and uncertainty quality. While the model assigns high uncertainties to the background, including some erroneous predictions, it fails to recognize the large prediction error on the chair's backrest. This comparison suggests that GNLL provides more meaningful uncertainties, supporting the quantitative findings in Section 4.3.2.

Cityscapes. In the third row, the MCD (ViT-B) predictions exhibit multiple errors. However, in parts, there is a correlation between uncertainty and challenging regions, indicating the model's awareness of its limitations.

UseGeo. The fourth row shows DSE (ViT-B) results, where relative depth predictions are plausible despite reduced accuracy in absolute terms. Across the entire image, especially in the bottom right, uncertainties are heightened for the buildings, emphasizing that the model is aware of key areas where depth errors are most likely.

HOPE. The fifth row presents LC (ViT-L) results, where the model struggles with the absolute depth of the large foreground object. At the same time, the entire object is highlighted by high uncertainty, reflecting the model's strong awareness of its prediction error in this case.

4.4 Conclusion

Summary. In this Chapter, we investigated the research question on how to fuse UQ with metric MDE, addressing the challenge of achieving reliable uncertainty estimates without compromising prediction accuracy or computational efficiency. To this end, we conducted a comprehensive evaluation of five UQ approaches – LC, GNLL, MCD, DSEs, and TTA – applied to the state-of-the-art DepthAnythingV2 foundation model. Our evaluation spanned four diverse datasets (NYUv2, Cityscapes, UseGeo, and HOPE), ensuring a broad assessment across different domains relevant to real-world applications.

Our findings demonstrate that fine-tuning with GNLL emerges as the most promising approach, providing high-quality uncertainty estimates while maintaining depth prediction accuracy and computational efficiency comparable to the baseline model. These results highlight that UQ can be both effective and efficient, enabling safer and more explainable metric depth estimates without introducing significant overhead.

Future Work. By demonstrating the feasibility of synthesizing UQ with MDE, we pave the way for future research that prioritizes not only performance but also explainability. Naturally, several open research directions remain.

For instance, future work could investigate whether the findings of this uncertainty-aware MDE research persist in low-data regimes. Exploring the interplay between uncertainty quality and unsupervised, active, self-supervised, or semi-supervised learning paradigms could help reduce dependency on large-scale annotated fine-tuning datasets while maintaining predictive reliability. Additionally, future research could inspect uncertainty-aware domain adaptation. While our evaluation covered diverse datasets, we did not examine whether uncertainty-aware MDE models remain reliable on OOD samples or how easily they adapt to new domains, which would be critical for many real-world applications like autonomous driving.

We hope that these findings and suggestions will inspire continued research into uncertainty-aware foundation models, ultimately contributing to the development of more reliable machine vision systems for deployment in real-world scenarios.

5 Uncertainty-aware Joint Semantic Segmentation and Monocular Depth Estimation

This Chapter includes elements from

[141] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich. “A Comparative Study on Multi-task Uncertainty Quantification in Semantic Segmentation and Monocular Depth Estimation”. In: *tm-Technisches Messen* (2025),

which are marked with a blue line.

This Chapter also includes elements from

[142] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich. “Efficient Multi-task Uncertainties for Joint Semantic Segmentation and Monocular Depth Estimation”. In: *DAGM German Conference on Pattern Recognition*. Springer. 2024, pp. 348–364,

which are marked with an orange line.

The following Chapter explores Uncertainty Quantification (UQ) in joint Semantic Segmentation (SS) and Monocular Depth Estimation (MDE) within a multi-task learning framework. As discussed in Chapters 3 and 4, these tasks are fundamental to machine vision: SS assigns distinct class labels to each pixel, while MDE predicts pixel-wise distances from the camera. Combining these two tasks enables more comprehensive scene understanding, which is beneficial for many real-world applications such as autonomous driving [30] or robotics [196] that are multi-modal in nature.

Multi-task Learning. Multi-task learning is a learning paradigm that aims to improve generalization and efficiency by training a single model to perform multiple related tasks simultaneously. By sharing representations across multiple tasks, multi-task learning can mitigate overfitting, reduce computational costs, and exploit cross-task dependencies to enhance the individual performance on each task [231, 259, 301, 41, 266, 300].

Although SS and MDE are often treated as separate tasks, they are inherently interconnected: The former provides contextual information about objects and regions in an image, while the latter captures their spatial arrangement in 3D space. Consequently, one can leverage their complementary nature by jointly modeling these tasks to enhance efficiency or even performance [273, 188, 121, 286, 165, 163, 196, 96, 69, 119, 127, 166, 23, 287, 22]. For example, semantic information can guide the depth estimation by providing priors on object shapes and sizes, whereas depth cues can help resolve segmentation ambiguities in noisy areas of the image, such as object borders.

Challenges. While multi-task learning has the potential to improve the predictive performance over single-task learning, several key challenges remain. As discussed in Chapter 2.2, Deep Learning (DL) models often exhibit overconfidence [84, 277], which makes deployment in safety-critical applications questionable. Moreover, they are known to suffer severe performance degradation under domain shifts [154, 187, 204], which, in real-world scenarios, are the rule, not the exception. Additionally, the lack of interpretability limits trust, especially in fields like medical diagnosis [75, 155, 9]. Most notably, even powerful foundation models that are trained on internet-scale data are far from perfect, as examined in the previous Chapter 4.

It is obvious that these issues require methods that not only focus on obtaining a slightly higher accuracy on a given benchmark dataset but also incorporate reliable uncertainty estimates. Surprisingly, however, quantifying predictive uncertainties in the context of joint SS and MDE has not been thoroughly explored yet.

Research Question. To address this notable gap in the literature, we investigate the following key research questions:

1. How do existing UQ methods perform for joint SS and MDE?
2. How can we enable efficient and reliable UQ in this multi-task learning framework?
3. How can we exploit predictive uncertainties during training to optimize performance?

Regarding the first question, we combine three common UQ methods – Monte Carlo Dropout (MCD), Deep Sub-Ensembles (DSEs), and Deep Ensembles (DEs) – with joint SS and MDE and evaluate their performance. In this context, we reveal the potential benefit of multi-task learning concerning the uncertainty quality compared to solving SS and MDE separately. Additionally, we examine the influence of employing different uncertainty thresholds to determine whether a pixel is certain or uncertain and include an out-of-domain (OOD) evaluation.

In response to the second and third questions, we propose a novel **E**fficient **M**ulti-task **U**ncertainty Vision Trans**former** (EMUFormer). Thereby, we show that by leveraging the predictive uncertainties during training through the use of the Gaussian Negative Log-Likelihood (GNLL) loss, EMUFormer achieves new state-of-the-art results on Cityscapes and NYUv2, while providing predictive uncertainties for both tasks that are comparable or superior to a DE despite being an order of magnitude more efficient.

Outline and Structure. The Chapter is organized as follows:

1. Section 5.1 offers an overview of related work on joint SS and MDE.
2. Section 5.2 provides a detailed description of the utilized baseline models: SegFormer [284], DepthFormer, and SegDepthFormer.
3. Section 5.3 showcases our comprehensive multi-task uncertainty evaluation.
4. Section 5.4 presents EMUFormer, our student-teacher distillation approach for efficient and reliable multi-task uncertainties.
5. Section 5.5 concludes with a summary of key findings and suggestions for future work.

5.1 Related Work

In this Section, we summarize related work on joint SS and MDE. For a broader review of related work on UQ and knowledge distillation, refer to the corresponding Sections 2.2 and 2.3 in the fundamentals Chapter, respectively.

Joint Semantic Segmentation and Monocular Depth Estimation

SS and MDE are both fundamental problems in image understanding that involve pixel-wise predictions based on a single input image. Motivated by the strong correlation and complementary properties of the two tasks, multiple previous works have focused on solving both tasks jointly [273, 188, 121, 286, 165, 163, 196, 96, 69, 119, 127, 166, 23, 287, 22]. Other multi-task approaches with joint representation sharing [300] or methods that leverage the depth map to improve the SS prediction [110, 270] are not relevant for our work and, therefore, are not covered by this review.

Wang et al. [273] and Liu et al. [165] propose frameworks for combining SS and MDE using conditional random fields. In contrast, Mousavian et al. [188] train parts of the model for each task separately and then fine-tune the full model on both tasks with a single loss function. Multiple previous works introduce attention mechanisms to improve results [121, 23, 166, 69]. Gao et al. [70] and Kendall et al. [127] introduce confidences to weight the individual losses accordingly. Xu et al. [286] propose a multi-task prediction-and-distillation network, where the predictions of intermediate auxiliary tasks are the multi-modal input for the final task – a concept that multiple works [267, 196] followed. Finally, there are multiple works [163, 96, 119] that propose specialized architectures, focusing on task-relevant feature separation, geometric constraints, and dynamic loss balancing, respectively.

5.2 Baseline Models

Hereinafter, we go over the three baseline models that we use for our comprehensive multi-task uncertainty evaluation and as a foundation for our EMUFormer approach:

1. SegFormer [284] for solving the SS task.
2. DepthFormer for solving the MDE task.
3. SegDepthFormer for solving the joint SS and MDE task.

We selected the SegFormer model for its architectural simplicity, computational efficiency, and competitive performance, and derived DepthFormer and SegDepthFormer from it. For each model, we briefly describe its architecture, training criterion, and how we obtain a measurement for the uncertainty.

5.2.1 SegFormer

Architecture. For solely solving the SS task, we use SegFormer [284], a modern Transformer-based architecture that stands out because of its high efficiency and performance. Thus, it is particularly suitable for real-time UQ. As depicted in Figure 5.1, SegFormer consists of two main modules: A hierarchical Transformer-based encoder that generates high-resolution coarse features and low-resolution fine features, and a lightweight all-MLP segmentation decoder. The latter fuses the multi-level features of the encoder to produce pseudo-probabilities with the softmax activation function, which can be formulated as

$$p(\hat{y})_i = \frac{\exp(\hat{y}_i)}{\sum_{c=1}^{C} \exp(\hat{y}_c)} \quad , \tag{5.1}$$

where $p(\hat{y})_i$ is the softmax score for class i, C is the class count, and $\hat{y}$ represents the logits. Since SegFormer [284] only outputs logits at a $\frac{H}{4} \times \frac{W}{4}$ resolution given an input image of size $H \times W$, we use bilinear interpolation [284] before applying the softmax function on $\hat{y}$ to obtain the original resolution for the final segmentation prediction.

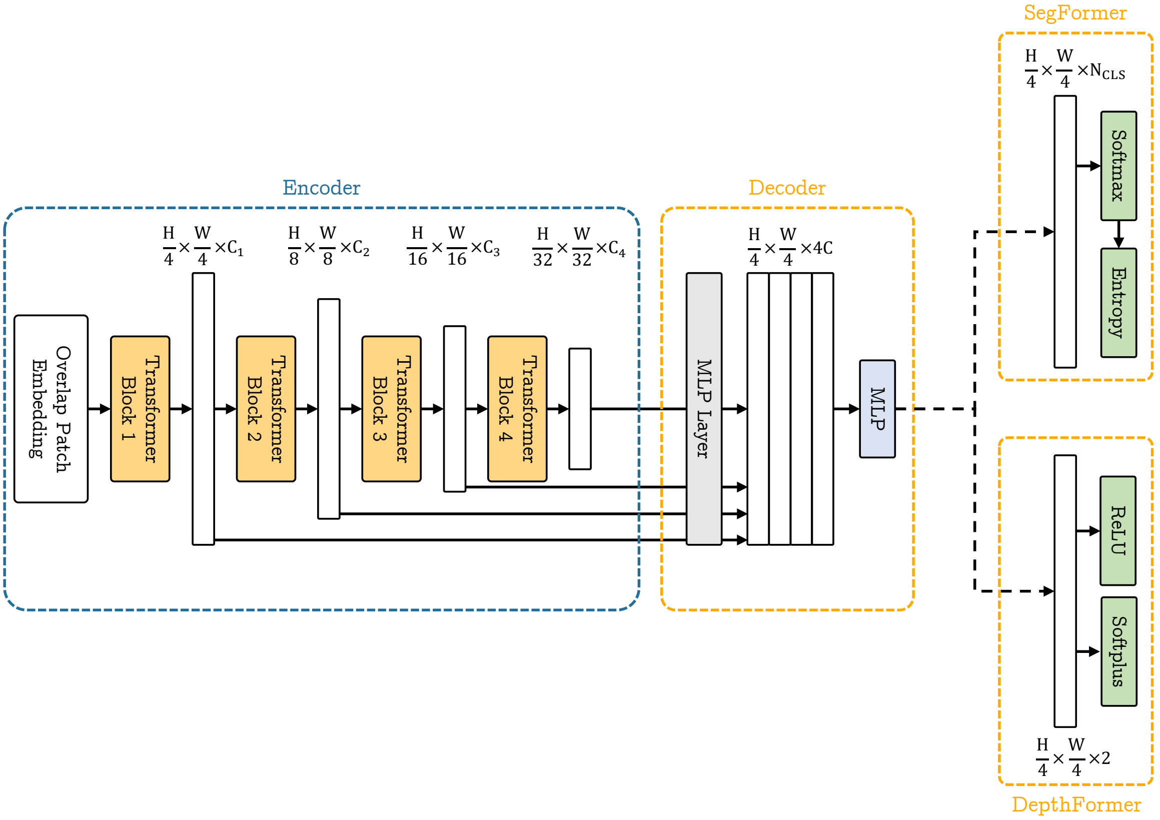

SegFormer
$\frac{H}{4}\times\frac{W}{4}\times N_{CLS}$
Softmax
Entropy
Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
Overlap Patch Embedding
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
ReLU
Softplus
$\frac{H}{4}\times\frac{W}{4}\times 2$
DepthFormer

Figure 5.1: A schematic overview of the SegFormer [284] and DepthFormer architectures. Both models share the same hierarchical Transformer-based encoder that generates high-resolution coarse features and low-resolution fine features, and a lightweight all-MLP segmentation decoder. They only differ in the number of output channels and in terms of output activations.

Training Criterion. For the objective function during training, we use the well-known categorical Cross-Entropy (CE) loss

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \cdot \log(p_{n,c}(\hat{y})) \ , \tag{5.2}$$

where $\mathcal{L}_{\text{CE}}$ is the loss for an image, N is the pixel count, C is the number of classes, $y_{n,c}$ is the ground truth, and $p_{n,c}(\hat{y})$ is the predicted pseudo-probability.

Uncertainty Measure. We compute the predictive entropy

$$H(p(\hat{y})) = -\sum_{c=1}^{C} p(\hat{y})_c \cdot \log(p(\hat{y})_c) \ . \tag{5.3}$$

which serves as a measure for the uncertainty [126]. Unlike standard deviation or variance (cf. Equation 4.5), which require multiple samples generated by UQ methods, predictive entropy is derived directly from the models' softmax pseudo-probability, enabling comparison of the baseline models' uncertainty with that of applied UQ methods, where the mean softmax pseudo-probability of all the samples is used to compute the predictive entropy.

5.2.2 DepthFormer

Architecture. Inspired by the efficiency and performance of SegFormer [284], we propose DepthFormer for MDE. As Figure 5.1 shows, we use the same hierarchical Transformer-based encoder as SegFormer to generate high-level and low-level features. Similarly, those multi-level features are fused in an all-MLP decoder. In contrast to SegFormer, the output layer differs by having two output channels: One for the predictive mean $\hat{\mu}(x)$ and one for the predictive variance $\hat{\sigma}^2(x)$ [168].

Predictive Mean. The first output channel uses a Rectified Linear Unit (ReLU) output activation function to produce the predictive mean $\hat{\mu}(x)$ based on an input image x:

$$\hat{\mu}(x) = \max(0, \hat{y}(x)) \ . \tag{5.4}$$

Predictive Variance. The second output channel applies a Softplus activation to produce the predictive variance

$$\hat{\sigma}^2(x) = \log(1 + \exp(\hat{y}(x))) \ . \tag{5.5}$$

The Softplus activation is a smooth and fully differentiable approximation of the ReLU function, including at $\hat{y} = 0$. In our experiments, we found that Softplus yields more stable predictive variances than ReLU, consistent with the observations by Lakshminarayanan et al. [138]. Intuitively, its strictly positive and smooth output avoids zero gradients and enables more reliable learning of uncertainties, particularly in regions with low predicted variance.

Training Criterion. For regression tasks, neural networks typically output only a predictive mean $\hat{\mu}$ and the parameters are, in the most straightforward approach, optimized by minimizing the Mean Squared Error (MSE). However, the MSE does not cover uncertainty. Therefore, we follow the approach of Nix and Weigend [201] instead: By treating the neural networks prediction as a sample from a Gaussian distribution with the predictive mean $\hat{\mu}$ and corresponding predictive variance $\hat{\sigma}^2$, we can minimize the GNLL loss using the ground truth depth y:

$$\mathcal{L}_{\text{GNLL}} = \frac{1}{2}\left(\frac{(y-\hat{\mu})^2}{\hat{\sigma}^2} + \log \hat{\sigma}^2\right) \ . \tag{5.6}$$

Uncertainty Measure. Through GNLL minimization, DepthFormer not only optimizes the predictive means, but also inherently learns the corresponding variances, which serve as a measure of the uncertainty [126, 168].

5.2.3 SegDepthFormer

Architecture. In order to jointly solve SS and MDE, we propose SegDepthFormer. The architecture, which is shown in Figure 5.2, comprises three modules: A hierarchical Transformer-based encoder, an all-MLP segmentation decoder, and an all-MLP depth decoder. The encoder and segmentation decoder are adapted from SegFormer [284] (cf. Section 5.2.1), while the depth decoder is from DepthFormer (cf. Section 5.2.2). Both decoders fuse the multi-level features obtained through the shared encoder to predict a final segmentation mask and a pixel-wise depth estimation, respectively.

Training Criterion. SegDepthFormer is trained to minimize the weighted sum of the two previously described objective functions:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_{\text{GNLL}} \ , \tag{5.7}$$

where λ_1 is a simple weighting factor. Because both loss values are of similar magnitude, we set $\lambda_1 = 1$. Tuning λ_1 might slightly improve the performance, however.

Uncertainty Measure. The respective uncertainty is obtained by computing the predictive entropy $H(p(\hat{y}))$ (cf. Equation 5.3) for the segmentation task or by the predictive variance $\hat{\sigma}^2(x)$ (cf. Equation 5.5), which is learned implicitly through the optimization of $\mathcal{L}_{\text{GNLL}}$ (cf. Equation 5.6).

5.3 Evaluation of Multi-task Uncertainties

This Section investigates the integration of MCD, DSEs, and DEs with joint SS and MDE. In this way, we explore the potential benefits of multi-task learning for the uncertainty quality and analyze the influence of different uncertainty thresholds for distinguishing whether a pixel is certain or uncertain.

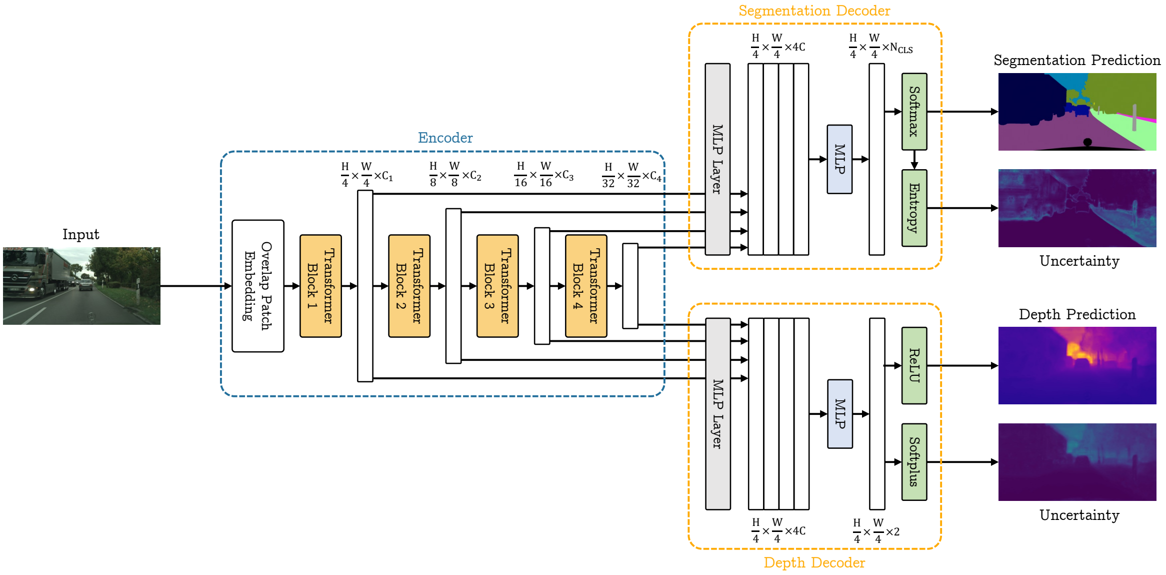

Segmentation Decoder
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{CLS}$
Segmentation Prediction
MLP Layer
MLP
Softmax
Entropy
Uncertainty
Encoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
Input
Overlap Patch Embedding
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
Depth Prediction
MLP Layer
MLP
ReLU
Softplus
Uncertainty
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times 2$
Depth Decoder

Figure 5.2: A schematic overview of the SegDepthFormer architecture. The model combines the SegFormer [284] architecture with a lightweight all-MLP depth decoder.

Research Gap. Remarkably, to the best of our knowledge, quantifying uncertainties in joint SS and MDE has been overlooked entirely. Therefore, we compare multiple UQ methods for this joint task and show how multi-task learning influences the quality of the uncertainty quality in comparison to solving both tasks separately.

5.3.1 Methodology

We evaluate MCD [67], DEs [138], and DSEs [263], motivated by their simplicity, ease of implementation, parallelizability, minimal tuning requirements, and state-of-the-art performance. Learned Confidence (LC) [269] and Test-Time Augmentation (TTA) [9] are excluded from this evaluation due to their poor performance in Chapter 4, and GNLL [201] is left out because it assumes continuous, Euclidean targets, whereas SS requires discrete categorical predictions. Applying GNLL to SS would lead to unstable optimization without yielding coherent probabilistic outputs over the classes.

Monte Carlo Dropout. MCD depends on the number and placement of dropout layers, and particularly the dropout rate. We adopt the original SegFormer [284] layer placement and consider two dropout rates, 20 % and 50 %. We sample ten times to obtain the prediction and predictive uncertainty [67, 87].

Deep Ensemble. DEs achieve the best results if they are trained to explore diverse modes in function space, which we accomplish by randomly initializing all decoder heads, using random augmentations (cf. Section 5.3.2), and by applying random shuffling of the training data points [138, 61]. We report results of a DE with ten members, following the suggestions of previous work [138, 61, 147].

Deep Sub-Ensemble. Consistent with DEs and MCD, we train the DSE with ten decoder heads for each task on top of a shared encoder [263]. During training, we only optimize a single decoder head per training batch and alternate between them. Thereby, we aim to introduce as much randomness as possible, analogous to the training of DEs. For inference, we utilize all decoder heads.

5.3.2 Experiments

The following Section describes a variety of experiments, including quantitative results, the impact of the number of ensemble members, the impact of the uncertainty threshold, and an OOD evaluation.

Experimental Setup

Predictions. For the SS task, we compute the mean softmax pseudo-probability of all samples. For the MDE task, we first apply ReLU and then compute the mean depth of the corresponding samples.

Uncertainty. For the SS task, we measure the predictive uncertainty using the predictive entropy (cf. Equation 5.3) [190], which is computed either using the softmax pseudo-probability of a single baseline model or based on the mean softmax pseudo-probabilities across multiple samples from a UQ method. In contrast, for the depth estimation task, we explicitly distinguish between baseline models and UQ methods. For baseline models, we estimate the uncertainty directly using the predictive variance $\hat{\sigma}^2$ (cf. Equation 5.5). For models employing UQ, we compute the predictive uncertainty by combining the mean predictive variance and the variance of the depth predictions across available samples, as proposed by Loquercio et al. [168]:

$$\hat{\sigma}^2_{\text{UQ}} = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}_t^2}_{\text{aleatoric}} + \underbrace{\frac{1}{T-1}\sum_{t=1}^{T}(\hat{y}_t - \hat{\mu})^2}_{\text{epistemic}} \quad . \tag{5.8}$$

Here, T is the number of samples, $\hat{\sigma}_t^2$ is the predictive variance of the t-th sample, $\hat{y}_t$ is the t-th depth prediction, and $\hat{\mu}$ is the predictive mean of all depth predictions. While the primary focus of this thesis is not on decomposing uncertainty into aleatoric and epistemic components, this formulation captures both data uncertainty (aleatoric) and model uncertainty (epistemic) [168]. We therefore differentiate between baseline models, which are only capable of capturing aleatoric uncertainty, and UQ methods, which compute a more holistic uncertainty encompassing both aleatoric and epistemic components.

Datasets. We conduct all experiments on Cityscapes [39] and NYUv2 [245]. Cityscapes, with 2975 training and 500 validation images, is a popular urban street scene benchmark dataset. Notably, the depth values are based on the disparity of stereo camera images. NYUv2 contains 795 training and 654 testing images of indoor scenes.

Data Augmentations. Regardless of the trained model, we apply random scaling with a factor between 0.5 and 2.0, random cropping with a crop size of 768 × 768 pixels on Cityscapes and 480 × 640 pixels on NYUv2, and random horizontal flipping with a flip chance of 50 %.

Implementation Details. For all training processes, we use AdamW [169] optimizer with a base learning rate of $6 \cdot 10^{-5}$ and employ a polynomial rate scheduler (cf. Equation 3.6). Besides, we use a batch size of 8 and train for 250 epochs on Cityscapes and for 100 epochs on NYUv2, respectively. The encoders of the baseline models are initialized with weights pretrained on ImageNet [47] and then trained for 250 epochs on Cityscapes and for 100 epochs on NYUv2, respectively. We use the SegFormer-B2 [284] backbone for all experiments.

Metrics. For SS, we report mean Intersection over Union (mIoU) [148] and Expected Calibration Error (ECE) [192]. For MDE, we use Root Mean Squared Error (RMSE) [177, 8] and the adopted uncertainty quality metrics from Mukhoti and Gal [190], which we already discussed in Section 4.3.2.

For the sake of simplicity and to simulate real-world employment, we set the uncertainty threshold to the mean uncertainty of a given image for all evaluations, unless noted otherwise. This choice reflects standard practice [190] at the time this research was conducted and was deemed appropriate given that the threshold is consistently applied across all UQ methods. We also conduct a comparative analysis of different thresholds in Section 5.3.2, including the median and a statistically robust approach, which does not use the normal standard deviation, but one that is resilient to outliers. To achieve this, as described by Steger et al. [253], we take the median uncertainty

$$\hat{\sigma}_M = \text{Med}(\hat{\sigma}) \ . \tag{5.9}$$

Subsequently, a robust measure of the standard deviation can be derived with

$$\hat{\sigma}_R = \frac{\text{Med}(|\hat{\sigma} - \hat{\sigma}_M|)}{0.6745} \ , \tag{5.10}$$

where the correction factor in the denominator is chosen in such a way that, for normally distributed uncertainties, the standard deviation aligns with one of a normal distribution.

Finally, by using the median as a central value and extending it by a range proportional to the robust standard deviation, the robust threshold can be defined as

$$\Theta = \hat{\sigma}_M + \tau \cdot \hat{\sigma}_R \ , \tag{5.11}$$

where τ is a tunable factor. It is worth mentioning, however, that using a negative scaling factor $\tau < 0$ resulted in unstable evaluation results, as it led to the threshold being set to $\Theta = 0$ in some cases, which is why we discarded that option altogether.

Quantitative Results

In this Section, we describe the results of our joint uncertainty evaluation quantitatively. We compare combinations of the baseline models SegFormer, DepthFormer, and SegDepthFormer (cf. Section 5.2) with the UQ methods MCD, DSE, and DEs (cf. Section 5.3.1). Tables 5.1 and 5.2 contain a detailed comparison, primarily focusing on the uncertainty quality.

		Semantic Segmentation					Monocular Depth Estimation				
		mIoU ↑	ECE ↓	p(acc$\mid$cer) ↑	p(unc$\mid$inacc) ↑	PAvPU ↑	RMSE ↓	p(acc$\mid$cer) ↑	p(unc$\mid$inacc) ↑	PAvPU ↑	Inference Time [ms]
Baseline	SegFormer	0.772	0.033	0.882	0.395	0.797	-	-	-	-	17.90 ± 0.47
	DepthFormer	-	-	-	-	-	7.452	0.749	0.476	0.766	17.59 ± 0.82
	SegDepthFormer	0.738	0.028	0.913	0.592	0.826	7.536	0.745	0.472	0.762	22.04 ± 0.27
MCD (20 %)	SegFormer	0.759	**0.007**	0.883	0.424	0.780	-	-	-	-	177.13 ± 0.64
	DepthFormer	-	-	-	-	-	7.956	0.749	0.555	0.739	139.32 ± 0.78
	SegDepthFormer	0.738	0.020	0.911	0.592	0.803	7.370	0.761	0.523	0.757	202.23 ± 0.39
MCD (50 %)	SegFormer	0.662	0.028	0.883	0.485	0.760	-	-	-	-	176.98 ± 0.53
	DepthFormer	-	-	-	-	-	21.602	0.181	0.366	0.431	139.81 ± 1.20
	SegDepthFormer	0.640	0.021	0.906	0.616	0.782	8.316	0.733	**0.558**	0.723	203.82 ± 0.81
DSE	SegFormer	0.772	0.037	0.890	0.456	0.797	-	-	-	-	132.30 ± 3.16
	DepthFormer	-	-	-	-	-	**7.036**	0.762	0.467	0.772	91.82 ± 2.01
	SegDepthFormer	0.749	0.009	**0.931**	**0.696**	**0.844**	7.441	0.751	0.463	0.766	212.11 ± 8.44
DE	SegFormer	**0.784**	0.033	0.887	0.416	0.798	-	-	-	-	667.51 ± 2.89
	DepthFormer	-	-	-	-	-	7.222	0.759	0.486	0.771	626.79 ± 2.05
	SegDepthFormer	0.755	0.015	0.917	0.609	0.828	7.156	**0.763**	0.493	**0.773**	743.23 ± 32.95

Table 5.1: Quantitative comparison on Cityscapes [39] between SegFormer, DepthFormer, and SegDepthFormer, each paired with MCD, DSEs, and DEs, respectively. See Section 5.3.2 for a concise description of how the uncertainties are calculated. Best results are marked in **bold**.

		Semantic Segmentation					Monocular Depth Estimation				
		mIoU ↑	ECE ↓	p(acc$\mid$cer) ↑	p(unc$\mid$inacc) ↑	PAvPU ↑	RMSE ↓	p(acc$\mid$cer) ↑	p(unc$\mid$inacc) ↑	PAvPU ↑	Inference Time [ms]
Baseline	SegFormer	0.470	0.159	0.768	0.651	**0.734**	-	-	-	-	18.09 ± 0.41
	DepthFormer	-	-	-	-	-	0.554	0.786	0.449	0.610	17.51 ± 0.87
	SegDepthFormer	0.466	0.151	0.769	0.659	0.733	0.558	0.776	0.446	0.594	22.31 ± 0.23
MCD (20 %)	SegFormer	0.422	0.102	0.767	0.706	0.724	-	-	-	-	222.67 ± 0.61
	DepthFormer	-	-	-	-	-	0.605	0.741	0.478	0.568	139.58 ± 052
	SegDepthFormer	0.433	0.093	0.771	0.710	0.725	0.610	0.731	0.450	0.560	251.25 ± 0.81
MCD (50 %)	SegFormer	0.273	0.083	0.705	**0.722**	0.713	-	-	-	-	223.25 ± 0.82
	DepthFormer	-	-	-	-	-	0.978	0.516	**0.492**	0.526	139.27 ± 0.69
	SegDepthFormer	0.272	0.084	0.702	0.721	0.711	0.837	0.576	0.473	0.525	251.98 ± 0.60
DSE	SegFormer	0.469	0.092	0.776	0.681	0.726	-	-	-	-	180.42 ± 3.93
	DepthFormer	-	-	-	-	-	0.547	0.782	0.423	0.596	91.66 ± 0.26
	SegDepthFormer	0.461	**0.077**	0.776	0.692	0.723	0.584	0.738	0.403	0.573	261.69 ± 5.10
DE	SegFormer	**0.486**	0.125	0.782	0.675	**0.734**	-	-	-	-	715.97 ± 7.55
	DepthFormer	-	-	-	-	-	**0.524**	**0.808**	0.475	**0.613**	624.30 ± 2.07
	SegDepthFormer	0.481	0.122	**0.783**	0.682	0.733	0.552	0.785	0.453	0.590	788.76 ± 2.00

Table 5.2: Quantitative comparison on NYUv2 [245] between SegFormer, DepthFormer, and SegDepthFormer, each paired with MCD, DSEs, and DEs, respectively. See Section 5.3.2 for a concise description of how the uncertainties are calculated. Best results are marked in **bold**.

Single-task vs. Multi-task. Looking at the differences between the single-task models, SegFormer and DepthFormer, and the multi-task model, SegDepthFormer, the single-task models generally deliver slightly better prediction performance. However, SegDepthFormer exhibits greater uncertainty quality for the SS task in comparison to SegFormer. This is particularly evident for p(unc|inacc) on Cityscapes. For the depth estimation task, there is no significant difference in terms of uncertainty quality.

Baseline Models. As expected, the baseline models have the lowest inference times, being 5 to 30 times faster without using any UQ method. While their prediction performance turns out to be quite competitive, only beaten by DEs, they show poor calibration and uncertainty quality for SS. Surprisingly, the uncertainty quality for the depth estimation task is very decent, often only surpassed by the DE.

Uncertainty Quantification Methods. MCD causes a significantly higher inference time compared to the respective baseline model. Additionally, leaving dropout activated during inference to sample from the posterior has a detrimental effect on the prediction performance, particularly with a 50 % dropout ratio. Nevertheless, MCD outputs well-calibrated softmax pseudo-probabilities and uncertainties, although the results should be interpreted with caution because of the deteriorated prediction quality. Across both datasets, DSEs show comparable prediction performance compared with the baseline models. Notably, DSEs consistently demonstrate a high uncertainty quality across all metrics, particularly in the segmentation task on Cityscapes. In accordance with previous work [87], DEs emerge as state-of-the-art, delivering the best prediction performance and mostly superior uncertainty quality. At the same time, DEs suffer from the highest computational cost, which scales approximately linearly with the number of members. Hence, we will explore the impact of ensemble members next.

Impact of Ensemble Members

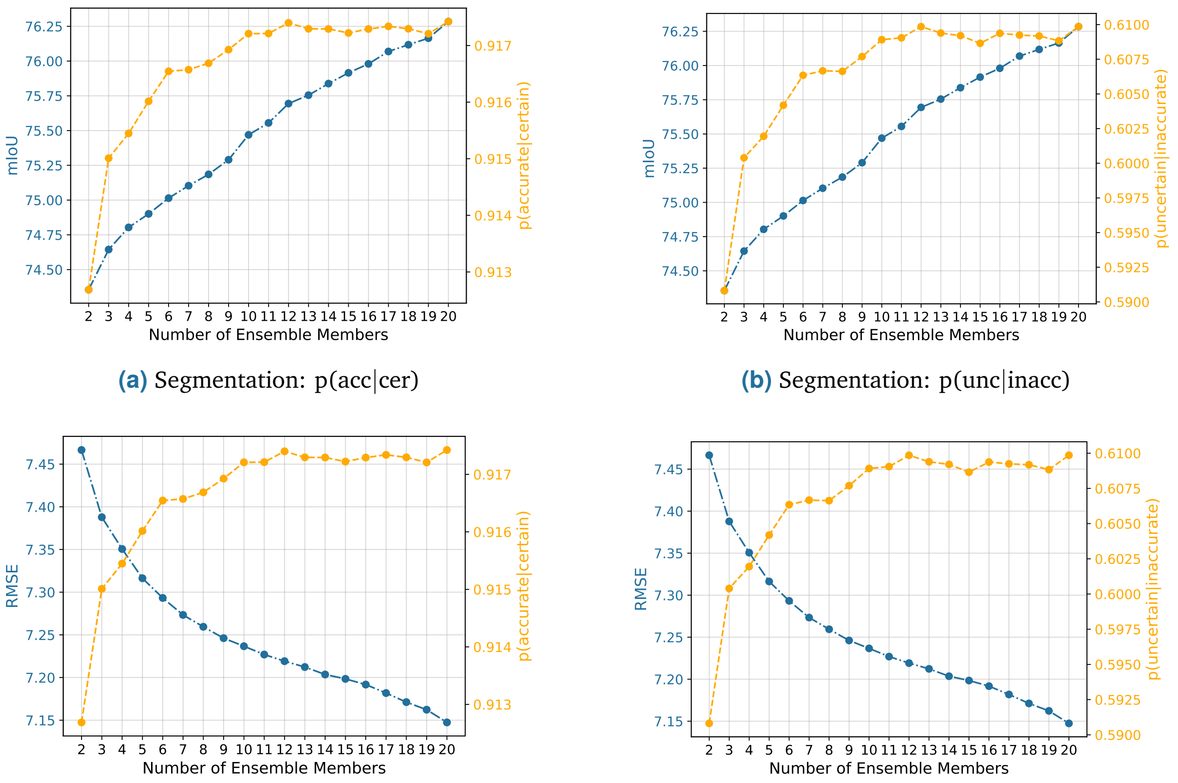
As Figure 5.3 shows, increasing the number of ensemble members improves the mIoU from 74.35 % with just two members to a maximum of over 76.25 % with twenty members. Similarly, the RMSE decreases from 7.46 to around 7.15. More notably, however, the uncertainty quality, measured by p(acc|cer) and p(unc|inacc), does not show significant improvement after 12 members for both tasks. Given the high computational cost of adding more members to a DE and the diminishing returns in both predictive performance, but mainly uncertainty quality, a configuration of about ten members appears to be a reasonable trade-off. These findings align with prior work on DE-based UQ [61, 147, 143], where ten members are often recommended as the default choice.

Impact of Uncertainty Threshold

In this Section, we examine the impact of employing different thresholds to classify pixels as certain or uncertain, incorporating a robust alternative based on a standard deviation designed to mitigate outliers (cf. Equation 5.11) [253].

As shown by Tables 5.3 and 5.4, the median threshold consistently performs best for p(acc|cer) and p(unc|inacc), demonstrating its ability in correlating correct predictions with low uncertainty and incorrect predictions with high uncertainty across both tasks and datasets. However, for the combined uncertainty quality metric, PAvPU, the mean threshold often achieves the highest scores, indicating that it provides fewer accurate labels with high uncertainty than the median threshold. While the robust threshold is theoretically promising and provides higher scores for p(acc|cer) and p(unc|inacc) than the mean threshold and higher scores for PAvPU than the median threshold on the Cityscapes dataset, its performance is less convincing on the NYUv2 dataset, particularly for p(unc|inacc). This suggests that the robust threshold may require further tuning of the clipping factor τ or more sophisticated adaptations to perform consistently across diverse datasets to match the reliability of the other two approaches.

Overall, the choice of the uncertainty threshold significantly impacts the uncertainty quality metrics, influencing the correlation between accurate predictions with low uncertainty and inaccurate predictions with high uncertainty. However, the results indicate that this impact is consistent across different methods, suggesting that the threshold selection influences the metrics independently of the underlying model or approach used.



(a) Segmentation: p(acc|cer)

(b) Segmentation: p(unc|inacc)

(c) Depth: p(acc|cer)

(d) Depth: p(unc|inacc)

Figure 5.3: Impact of the number of ensemble members on the predictive performance and uncertainty quality for a SegDepthFormer DE on the Cityscapes dataset [39].

		Semantic Segmentation			Monocular Depth Estimation		
		p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Baseline	Mean	0.913	0.592	0.826	0.745	0.472	0.762
	Median	0.947	0.852	0.611	0.870	0.832	0.750
	Robust $_{(\tau=1)}$	0.939	0.806	0.667	0.810	0.685	0.769
	Robust $_{(\tau=2)}$	0.935	0.780	0.693	0.779	0.596	0.766
DE	Mean	0.917	0.609	**0.828**	0.763	0.492	**0.773**
	Median	**0.950**	**0.861**	0.612	**0.878**	**0.834**	0.743
	Robust $_{(\tau=1)}$	0.943	0.816	0.670	0.822	0.691	0.770
	Robust $_{(\tau=2)}$	0.939	0.790	0.698	0.791	0.596	0.770

Table 5.3: Uncertainty threshold comparison on the Cityscapes dataset [39] between the SegDepthFormer baseline model and a SegDepthFormer DE. Best results are marked in **bold**.

		Semantic Segmentation			Monocular Depth Estimation		
		p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Baseline	Mean	0.769	0.659	**0.733**	0.776	0.446	0.594
	Median	0.810	0.783	0.698	0.787	**0.591**	0.532
	Robust $(\tau = 1)$	0.743	0.572	0.691	0.772	0.303	0.670
	Robust $(\tau = 2)$	0.706	0.423	0.667	0.766	0.172	0.717
DE	Mean	0.783	0.682	**0.733**	0.785	0.453	0.590
	Median	**0.816**	**0.785**	0.696	**0.796**	0.589	0.528
	Robust $(\tau = 1)$	0.741	0.544	0.688	0.784	0.309	0.676
	Robust $(\tau = 2)$	0.701	0.381	0.662	0.779	0.178	**0.728**

Table 5.4: Uncertainty threshold comparison on the NYUv2 dataset [245] between the SegDepthFormer baseline model and a SegDepthFormer DE. Best results are marked in **bold**.

Out-of-Domain Evaluation

In the following, we analyze the prediction and uncertainty quality of the SegDepthFormer baseline model and a corresponding DE with 10 members on two OOD datasets: Foggy Cityscapes [235] and Rain Cityscapes [109]. Both models were originally trained on the standard Cityscapes dataset and are evaluated without further fine-tuning. Quantitative comparisons are presented in Tables 5.5 and 5.6.

		Semantic Segmentation					Monocular Depth Estimation			
		mIoU ↑	ECE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	RMSE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Cityscapes	Baseline	0.738	0.028	0.913	0.592	0.826	7.536	0.745	0.472	0.762
	DE	0.755	0.015	0.917	0.609	0.828	7.156	0.763	0.493	0.773
$\text{Foggy}_{\beta=0.005}$	Baseline	0.707	0.035	0.906	0.602	0.818	8.061	0.731	0.481	0.751
	DE	0.727	0.028	0.914	0.627	0.822	7.487	0.758	0.509	0.765
$\text{Foggy}_{\beta=0.01}$	Baseline	0.674	0.054	0.899	0.606	0.814	8.628	0.715	0.475	0.741
	DE	0.699	0.056	0.910	0.637	0.817	7.971	0.750	0.511	0.761
$\text{Foggy}_{\beta=0.02}$	Baseline	0.609	0.078	0.875	0.593	0.798	9.844	0.697	0.467	0.730
	DE	0.639	0.045	0.895	0.644	0.803	9.213	0.738	0.517	0.760

Table 5.5: Quantitative comparison of the SegDepthFormer baseline model and a SegDepthFormer DE with 10 members on the Foggy Cityscapes validation dataset [235] without fine-tuning. β denotes the attenuation coefficient and controls the thickness of the fog. Higher β values result in thicker fog. The original Cityscapes and the Foggy Cityscapes datasets share the same validation images, enabling a fair comparison between in-domain (ID) and OOD results.

Foggy Cityscapes. Compared to the original Cityscapes dataset, the Foggy Cityscapes dataset reveals significant performance degradation, as shown in Table 5.5. For the strongest fog, the baseline model's performance drops from 0.738 to 0.609 mIoU and from 7.536 to 9.844 RMSE, while the DE model declines from 0.755 to 0.639 mIoU and from 7.156 to 9.213 RMSE. Calibration quality also worsens, with ECE increasing from 0.028 to 0.078 for the baseline and from 0.015 to 0.045 for the DE. For uncertainty quality, the baseline exhibits consistent deterioration in p(acc|cer) and PAvPU with denser fog but maintains relatively stable p(unc|inacc) values, likely due to an increasing number of inaccurate pixels offsetting the apparent uncertainty degradation. The DE shows less performance degradation in mIoU and RMSE, while improving p(unc|inacc) under severe fog, highlighting its robustness in OOD UQ.

		Semantic Segmentation					Monocular Depth Estimation			
		mIoU ↑	ECE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	RMSE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Rain$_1$	Baseline	0.608	0.020	0.936	0.658	0.810	7.187	0.792	0.558	0.767
	DE	0.673	0.004	0.954	0.741	0.813	6.740	0.804	0.559	0.767
Rain$_2$	Baseline	0.611	0.031	0.928	0.670	0.802	8.043	0.771	0.543	0.756
	DE	0.645	0.012	0.948	0.750	0.806	7.516	0.785	0.544	0.759
Rain$_3$	Baseline	0.582	0.045	0.917	0.671	0.795	8.848	0.751	0.534	0.749
	DE	0.612	0.023	0.943	0.756	0.799	8.294	0.767	0.535	0.755

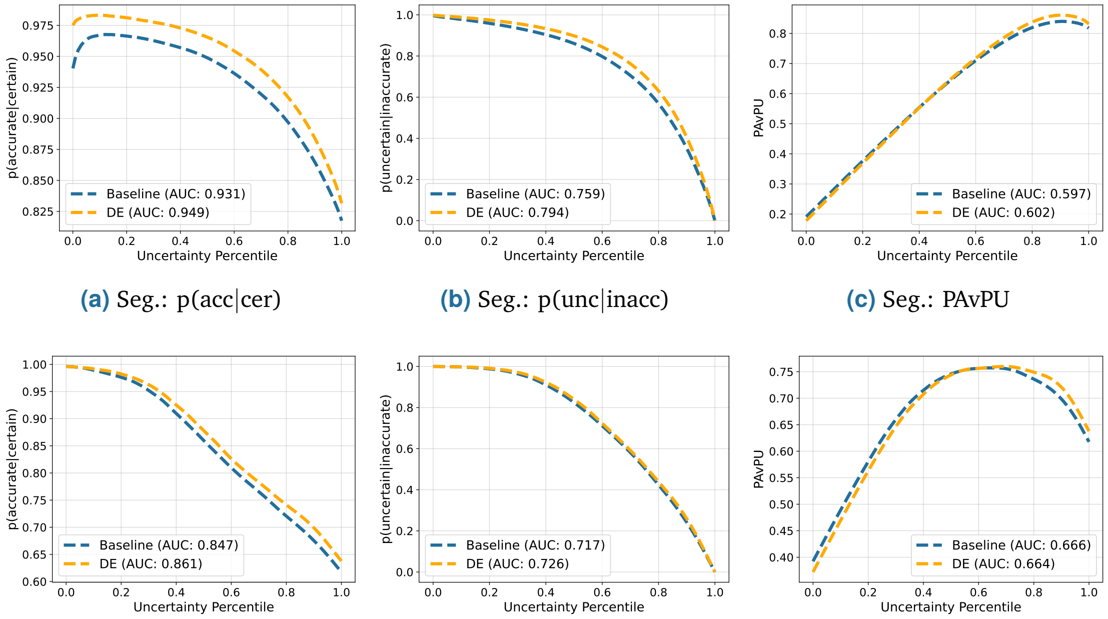
Table 5.6: Quantitative comparison of the SegDepthFormer baseline model and a SegDepthFormer DE with 10 members on the Rain Cityscapes validation dataset [109] without fine-tuning. β denotes the attenuation coefficient and controls the thickness of the fog. Higher β values result in thicker fog. We evaluate on three sets of parameters, where Rain$_1$ uses [0.01, 0.005, 0.01], Rain$_2$ uses [0.02, 0.01, 0.005], and Rain$_3$ uses [0.03, 0.015, 0.002] for attenuation coefficients α and β as well as the raindrop radius a. α and β determine the degree of simulated rain and fog in the images.

Rain Cityscapes. Table 5.6 highlights performance trends across varying levels of simulated rain. Under Rain$_1$, the baseline model achieves 0.608 mIoU and 7.187 RMSE, while the DE model improves these metrics to 0.673 mIoU and 6.740 RMSE. Calibration quality also favors the DE, with ECE values of 0.020 for the baseline and 0.004 for the DE. For uncertainty quality, p(acc|cer), p(unc|inacc), and PAvPU consistently show better results for the DE, with a single exception where results are equal. As rain intensity increases to Rain$_2$ and Rain$_3$, both models experience performance degradation, but the DE retains superior metrics, achieving 0.612 mIoU and 8.294 RMSE under Rain$_3$ compared to 0.582 mIoU and 8.848 RMSE for the baseline. Additionally, the DE sustains better calibration and uncertainty metrics, demonstrating enhanced robustness to OOD conditions.

Uncertainty Quality. To account for the impact of the uncertainty threshold on uncertainty quality metrics, as discussed in Section 5.3.2, we assess the OOD uncertainty quality across the entire percentile spectrum and utilize the Area Under the Curve (AUC) as a comprehensive evaluation metric. The AUC is chosen because it summarizes the uncertainty quality across all threshold levels, providing a robust measure that mitigates the sensitivity to specific threshold values. Figure 5.4 provides a detailed comparison between the SegDepthFormer baseline and the DE on the Rain$_3$ Cityscapes validation dataset. The DE demonstrates consistent superiority over the baseline in p(acc|cer) (AUC: 0.931 vs. 0.949 for segmentation and 0.847 vs. 0.861 for depth) and p(unc|inacc) (AUC: 0.759 vs. 0.794 for segmentation and 0.717 vs. 0.726 for depth). For PAvPU, the DE performs comparably to the baseline (AUC: 0.597 vs. 0.602 for segmentation and 0.666 vs. 0.664 for depth), aligning with the findings in Table 5.6.

5.4 Efficient Estimation and Exploitation of Multi-task Uncertainties

In the upcoming Section, we introduce EMUFormer, a novel student-teacher approach that combines previous findings on efficient UQ from Chapter 3 and the GNLL-based fine-tuning approach of the preceding Chapter 4. By doing so, EMUFormer delivers new state-of-the-art



(a) Seg.: p(acc|cer)

(b) Seg.: p(unc|inacc)

(c) Seg.: PAvPU

(d) Depth: p(acc|cer)

(e) Depth: p(unc|inacc)

(f) Depth: PAvPU

Figure 5.4: OOD uncertainty quality evaluation between the baseline SegDepthFormer and a SegDepthFormer DE with 10 members on the Rain$_3$ Cityscapes validation dataset [109]. Rain$_3$ uses [0.03, 0.015, 0.002] for attenuation coefficients α and β and the raindrop radius a. We compare the three uncertainty metrics p(acc|cer), p(unc|inacc), and PAvPU for different uncertainty thresholds. Additionally, we report the AUC.

performance on Cityscapes [39] and NYUv2 [245] while providing uncertainty estimates comparable to or better than a DE, albeit with significantly higher efficiency.

Research Gap. As noted before in Section 5.3, quantifying uncertainties in joint SS and MDE has not been explored yet. For this reason, we aim to set a new baseline for efficient estimation and exploitation of predictive uncertainties in this highly relevant multi-task setting. Additionally, most of the previous work on joint SS and MDE use out-of-date architectures and require complex adaptations to either the model, the training process, or both. Contrastingly, in order to push the state of the art forward, we adapt a modern Vision Transformer (ViT)-based architecture similar to Xu et al. [287]. To maintain methodological simplicity and transparency of the results, we do not use strategies like cross-task attention mechanisms, contrastive self-supervised learning algorithms, or a loss weighting strategy like that of Kendall et al. [127], and nevertheless achieve superior results. However, these strategies could be applied to our method as well, potentially further improving the results.

5.4.1 Methodology

In the following, we explain our student-teacher distillation framework for efficient multi-task uncertainties, which we call EMUFormer. Our objective with EMUFormer is threefold:

1. Achieve state-of-the-art joint SS and MDE results
2. Estimate well-calibrated predictive uncertainties for both tasks

3. Avoid introducing additional computational overhead during inference

To achieve these goals, EMUFormer employs a two-step student-teacher distillation framework:

1. Training a teacher with ground truth labels
2. Training the student with ground truth labels while distilling the teacher's predictive uncertainties

In principle, any architecture capable of outputting a SS mask along with a predictive mean and variance for MDE is suitable for EMUFormer.

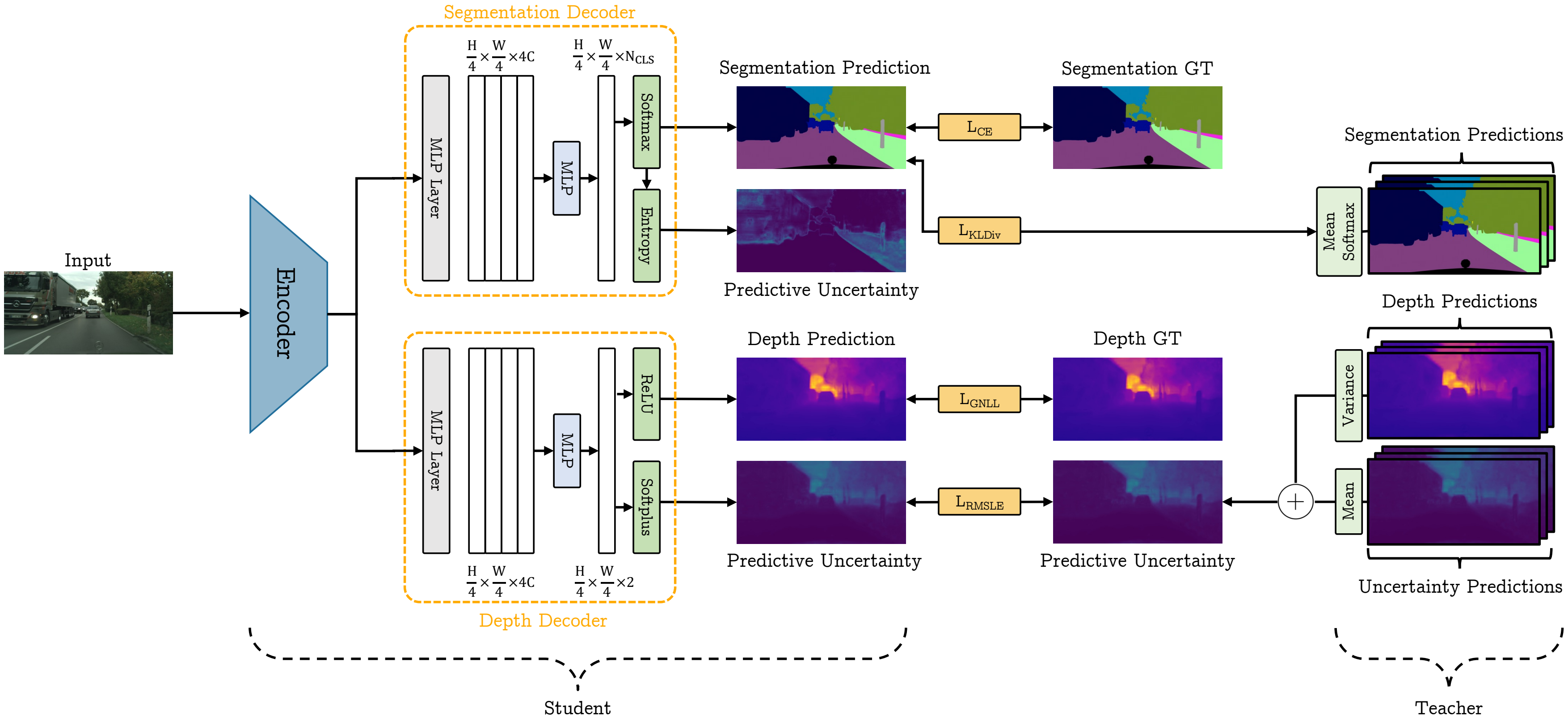


Figure 5.5: A schematic overview of EMUFormer. In addition to the regular CE loss for the SS task and the GNLL loss, EMUFormer utilizes two additional losses that distill the predictive uncertainties of the teacher into the student model.

Student. To solve both predictive tasks simultaneously, we propose to use SegDepthFormer (cf. Section 5.2.3), which is a modified version of SegFormer [284]. In addition to the efficient yet effective hierarchical Transformer-based encoder and all-MLP segmentation decoder of SegFormer, we add an all-MLP depth decoder, as shown in the left part of Figure 5.5.

Teacher. In principle, our framework is flexible with regard to the type of teacher. We select a DE that is known for producing high-quality estimates [204, 282, 87], which is also the case for multi-task UQ [141].

Improving Uncertainty Distillation. To determine both predictive uncertainties for the uncertainty distillation, we compute multiple prediction samples from the teacher, as shown by Figure 5.5. Since we use the same dataset for training and distillation, the student may underestimate the epistemic uncertainty component of the teacher because of overfitting. Hence, we add color jittering as an additional data augmentation to the teacher's input $\tilde{x}$, which was shown to be helpful by previous work on uncertainty distillation [243, 147]. The color jitter causes the teacher's uncertainty distribution on the training dataset to be more closely related to the test-time distribution.

Training Criterion. As Figure 5.5 shows, EMUFormer is trained to minimize the weighted sum of four objective functions, resulting in

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_{\text{GNLL}} + \lambda_2 \mathcal{L}_{\text{KL}} + \lambda_3 \mathcal{L}_{\text{RMSLE}} \ , \tag{5.12}$$

where

- $\mathcal{L}_{\text{CE}}$ is the categorical CE loss for the SS task,
- $\mathcal{L}_{\text{GNLL}}$ is the GNLL loss for the MDE task,
- $\mathcal{L}_{\text{KL}}$ is the Kullback-Leibler divergence loss for the segmentation uncertainty distillation,
- $\mathcal{L}_{\text{RMSLE}}$ is the root mean squared logarithmic error for the depth uncertainty distillation,
- and the weighting factors are empirically set to $\lambda_1 = \lambda_3 = 1$, $\lambda_2 = 10$, in order to balance the multi-task loss components by scaling their contributions to similar magnitudes.

Segmentation Criterion. For the SS task, we use the well-known categorical CE loss

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \cdot \log(p_{n,c}(\hat{y})) \ . \tag{5.13}$$

where $\mathcal{L}_{\text{CE}}$ is the CE loss for a single image, N is the number of pixels in the image, C is the number of classes, $y_{n,c}$ is the corresponding ground truth label, and $p_{n,c}(\hat{y})$ is the predicted softmax pseudo-probability.

Depth Criterion. For regression tasks, neural networks typically output only a predictive mean $\hat{\mu}$ and the parameters are, in the most straightforward approach, optimized by minimizing the MSE. However, the MSE does not cover uncertainty. Therefore, we follow the approach of Nix and Weigend [201] instead: By treating the neural network's prediction as a sample from a Gaussian distribution with the predictive mean $\hat{\mu}$ and corresponding predictive variance $\hat{\sigma}^2$, we can minimize the GNLL loss

$$\mathcal{L}_{\mathrm{GNLL}} = \frac{1}{2}\left(\frac{(y-\hat{\mu})^2}{\hat{\sigma}^2} + \log\hat{\sigma}^2\right) \quad . \tag{5.14}$$

where y is the ground truth depth.

Since this is a major insight of our work, we want to highlight that, usually, $\hat{\sigma}^2$ is solely learned implicitly through the optimization of the predictive means based on the ground truth labels. In the case of EMUFormer, however, the network is also being trained to mimic the predictive uncertainty of the teacher in parallel. Consequently, the depth uncertainty does not need to be learned implicitly, rather, it can be used explicitly to improve the depth estimation itself.

Segmentation Uncertainty Distillation. The segmentation uncertainty knowledge of the teacher model is transferred into the student model by using the Kullback-Leibler divergence loss

$$\mathcal{L}_{\mathrm{KL}} = \sum_{c=1}^{C} q_c(\hat{y}(\tilde{x})) \cdot \log\left(\frac{q_c(\hat{y}(\tilde{x}))}{p_c(\hat{y}(x))}\right) \quad , \tag{5.15}$$

where $p_c(\hat{y}(x))$ is the student's pseudo-probability output from input x, and $q_c(\hat{y}(\tilde{x}))$ is the teacher's pseudo-probability output from a color jittered input $\tilde{x}$, and C is the number of classes. Minimizing this loss ensures that the student learns to match the well-calibrated softmax pseudo-probabilities provided by the teacher, allowing the predictive entropy

$$H(p(\hat{y})) = -\sum_{c=1}^{C} p_c(\hat{y}) \cdot \log(p_c(\hat{y})) \tag{5.16}$$

to capture the underlying predictive uncertainty.

Depth Uncertainty Distillation. Because it is not possible to match two distributions for the unbound uncertainties in the regression task, we introduce the Root Mean Squared Logarithmic Error (RMSLE) for the depth uncertainty distillation:

$$\mathcal{L}_{\mathrm{RMSLE}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\log(\sigma_n^2(\tilde{x})+1) - \log(\hat{\sigma}_n^2(x)+1)\right)^2} \quad , \tag{5.17}$$

where $\sigma_n^2(\tilde{x})$ is the teacher's predictive uncertainty based on the color jittered input image $\tilde{x}$ and $\hat{\sigma}_n^2(x)$ is the student's predictive uncertainty estimate. The natural logarithm penalizes underestimations more than overestimations, thereby providing special attention to the pixels with higher uncertainties. By minimizing the depth uncertainty loss, the student is trained to mimic the predictive uncertainty of the teacher.

5.4.2 Experiments

We conduct several experiments to demonstrate the efficiency and efficacy of EMUFormer. Firstly, we compare EMUFormer's performance with its DE teacher and the baseline models. Subsequently, we compare our results with previous state-of-the-art approaches, followed by qualitative examples. Then, we evaluate the generalizability of EMUFormer on two OOD datasets in comparison to its DE teacher and explore its capacity for domain adaptation. Subsequently, we study the impact of utilizing the distilled uncertainties in the GNLL loss. Lastly, we provide an ablation study on different backbone sizes.

Unless otherwise specified, we use SegFormer's B2 backbone as the default for all experiments as a compromise between efficiency and performance. Also, we use a SegDepthFormer DE with 10 members as the teacher for all experiments.

Experimental Setup

Datasets. We conduct all experiments on Cityscapes [39] and NYUv2 [245]. Cityscapes is a popular urban street scene benchmark dataset with 2975 training and 500 validation images. Notably, the depth values are based on the disparity of stereo camera images. NYUV2 contains 795 training and 654 testing images of indoor scenes.

For OOD evaluations, we use the validation sets of Foggy Cityscapes [235] and Rain Cityscapes [109], which introduce progressively increasing perturbations to the original urban street scenes, simulating adverse weather conditions like fog and rain.

Data Augmentations. Regardless of the trained model, we apply random scaling with a factor between 0.5 and 2.0, random cropping with a crop size of 768 × 768 pixels on Cityscapes and 480 × 640 pixels on NYUv2, and random horizontal flipping with a flip chance of 50 %.

Implementation Details. All training runs utilize the AdamW [169] optimizer with a base learning rate of 0.00006 and employ a polynomial learning rate scheduler (cf. Equation 3.6). Besides, we use a batch size of 8 and train on four NVIDIA A100 GPUs with 40 GB of memory using mixed precision [183]. The encoders of the baseline models are initialized with weights pre-trained on ImageNet [47] and subsequently trained for 250 epochs on Cityscapes and for 100 epochs on NYUv2. In contrast, EMUFormer is initialized with the weights of a pre-trained SegDepthFormer and fine-tuned for only 100 epochs on both datasets. Unless otherwise noted, we use the SegFormer-B2 [284] backbone for all experiments. To maintain simplicity and transparency, we refrain from employing common techniques such as OHEM [244], auxiliary losses, class imbalance compensation, or sliding window testing to boost performance.

Metrics. Analogous to the multi-task uncertainty evaluation from Section 5.3, we report mIoU [148] and ECE [192] for the SS task. For MDE, we use RMSE [177, 8] and the adopted uncertainty quality metrics from Mukhoti and Gal [190], which we already discussed in Section 4.3.2.

Quantitative Evaluation

Efficiency. As Table 5.7 demonstrates, EMUFormer estimates predictive uncertainties that are comparable to the Deep Ensemble teacher, while being approximately an order of magnitude more efficient. EMUFormer-B2 has only 30.5M parameters compared to over 300M in the SegDepthFormer Deep Ensemble, and achieves 44.8 FPS on a single NVIDIA A100 GPU without any runtime-specific optimizations, whereas the Deep Ensemble runs at under 5 FPS under the same conditions. This substantial improvement highlights EMUFormer's practical suitability for real-time applications.

	Seg.	Pred. Unc.	Depth	Pred. Unc.	Parameters	FLOPs [G]	FPS
a) SegFormer-B2 [284]	✓	×	×	×	27.3M	72.6	55.3
b) DepthFormer-B2	×	×	✓	×	27.3M	72.1	57.1
c) SegDepthFormer-B2	✓	×	✓	×	30.5M	120.1	44.8
DE of a)	✓	✓	×	×	273.6M	726.4	5.6
DE of b)	×	×	✓	✓	273.5M	720.8	7.2
DE of c)	✓	✓	✓	✓	305.1M	1201.1	4.9
EMUFormer-B2 (Ours)	✓	✓	✓	✓	30.5M	120.1	44.8

Table 5.7: Overview of the segmentation (Seg.), depth estimation (Depth), and UQ (Pred. Unc.) capabilities, where predictive uncertainty refers to reliable, well-calibrated uncertainties, as well as the respective number of parameters, FLOPs, and FPS for different single-task and multi-task models and their respective DE versions with 10 members. SegFormer [284] and DepthFormer represent single-task models, whereas SegDepthFormer and EMUFormer depict multi-task models. Results are based on single-scale inference conducted on the NYUv2 [245] dataset using a single NVIDIA A100 GPU.

Baseline vs. Teacher vs. Student. We present a comprehensive analysis in Table 5.8 by comparing the baseline models, their DEs versions with 10 members, and our EMUFormer. EMUFormer emerges as the standout performer on both datasets, surpassing the baseline models across all metrics, with only a single exception. Remarkably, this performance is achieved while maintaining an equivalent inference time. EMUFormer even outperforms the SegDepthFormer DE, which served as its teacher and has approximately 33 times higher inference time, in most cases. In terms of prediction performance, EMUFormer yields marginally worse segmentation results compared to the SegFormer DE on Cityscapes. However, it notably excels in the depth estimation task, especially on Cityscapes [39], which is a phenomenon we observed across all of our experiments (cf. Tables 5.9, 5.10, 5.11, 5.12, 5.13, and 5.15). We primarily attribute this success to the utilization of the predictive uncertainties inside the GNLL loss, but investigate this more thoroughly in Section 5.4.2.

Comparison with SOTA. Table 5.9 shows that EMUFormer-B5 surpasses the previous state of the art in joint SS and MDE on both Cityscapes [39] and NYUv2 [245]. For instance, on NYUv2 [245], it achieves 1.4 % higher mIoU and 0.007 lower RMSE than MTFormer [287], which also adopts a modern ViT-based architecture. In contrast to our work, however, MTFormer relies on cross-task attention and a complex self-supervised pre-training pipeline, which introduces additional complexity. Moreover, EMUFormer yields high-quality uncertainty estimates without any additional computational overhead during inference.

	Semantic Segmentation					Monocular Depth Estimation				Inference Time [ms]
	mIoU ↑	ECE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	RMSE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	
					Cityscapes					
SegFormer [284]	0.772	0.033	0.882	0.395	0.797	-	-	-	-	17.90 ± 0.47
SegFormer (DE)	**0.784**	0.033	0.887	0.416	0.798	-	-	-	-	667.51 ± 2.89
DepthFormer	-	-	-	-	-	7.452	0.749	0.476	0.766	17.59 ± 0.82
DepthFormer (DE)	-	-	-	-	-	7.222	0.759	0.486	0.771	626.79 ± 2.05
SegDepthFormer	0.738	0.028	0.913	0.592	0.826	7.536	0.745	0.472	0.762	22.04 ± 0.27
SegDepthFormer (DE)	0.755	0.015	0.917	0.609	**0.828**	7.156	0.763	**0.493**	0.773	743.23 ± 32.95
EMUFormer	0.752	**0.012**	**0.923**	**0.658**	0.811	**6.983**	**0.772**	0.491	**0.783**	22.04 ± 0.27
					NYUv2					
SegFormer [284]	0.470	0.159	0.768	0.651	0.734	-	-	-	-	18.09 ± 0.41
SegFormer (DE)	**0.486**	**0.125**	0.782	0.675	0.734	-	-	-	-	715.97 ± 7.55
DepthFormer	-	-	-	-	-	0.554	0.786	0.449	0.610	17.51 ± 0.87
DepthFormer (DE)	-	-	-	-	-	0.524	0.808	**0.475**	0.613	624.30 ± 2.07
SegDepthFormer	0.466	0.151	0.769	0.659	0.733	0.558	0.776	0.446	0.594	22.31 ± 0.23
SegDepthFormer (DE)	0.481	0.122	0.783	0.682	0.733	0.552	0.785	0.453	0.590	788.76 ± 2.00
EMUFormer	0.475	0.129	**0.787**	**0.692**	**0.737**	**0.514**	**0.810**	0.440	**0.633**	22.31 ± 0.23

Table 5.8: Quantitative comparison on the Cityscapes [39] and NYUv2 [245] datasets between the baseline models, their DE versions with ten members, and our EMUFormer. SegDepthFormer (DE) serves as the teacher.

	NYUv2		Cityscapes	
	mIoU ↑	RMSE ↓	mIoU ↑	RMSE ↓
HybridNet A2 [163]	0.343	0.682	0.666	12.09
Mousavian et al. [188]	0.392	0.816	-	-
C-DCNN [165]	0.398	0.628	-	-
BMTAS [22]	0.411	0.543	-	-
Gao et al. [70]	0.419	0.528	-	-
Nekrasov et al. [196]	0.420	0.565	-	-
CI-Net [69]	0.426	0.504	0.701	6.880
Wang et al. [273]$_{\text{CVPR'15}}$	0.442	0.745	-	-
PAD-Net [286]$_{\text{CVPR'18}}$	0.502	0.582	0.761	-
Nekrasov et al. [196]$_{\text{ICRA'19}}$	0.420	0.565	-	-
MTI-Net [267]$_{\text{ECCV'20}}$	0.490	0.529	-	-
ATRC [23]$_{\text{ICCV'21}}$	0.463	0.536	-	-
MTFormer [287]$_{\text{ECCV'22}}$	0.506	0.483	-	-
EMUFormer-B2 (Ours)	0.475	0.514	0.752	6.983
EMUFormer-B5 (Ours)	**0.520** (+0.014)	**0.476** (-0.007)	**0.771** (+0.010)	**6.157** (-0.723)

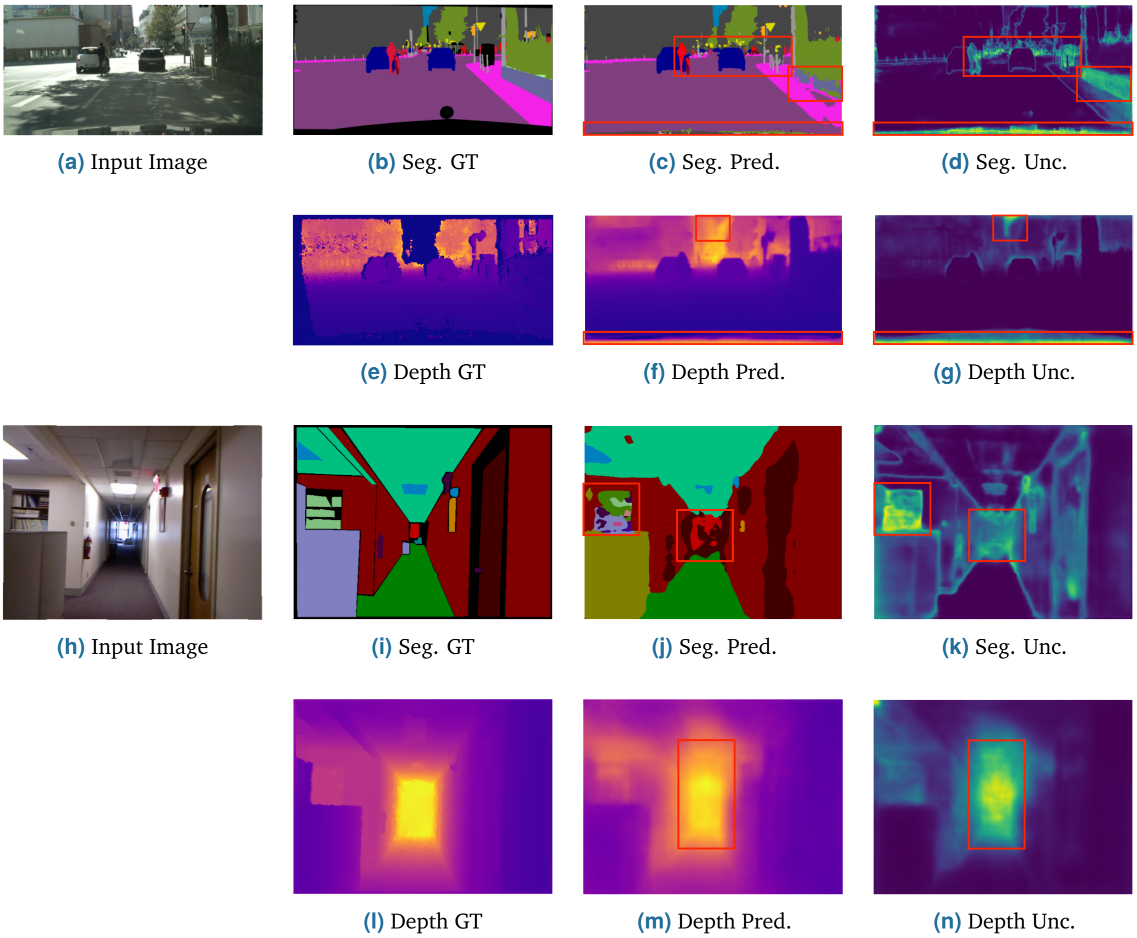
Table 5.9: Comparison against previous state-of-the-art approaches for joint SS and MDE.

Qualitative Evaluation

Cityscapes. On Cityscapes, EMUFormer demonstrates good prediction performance for both tasks, as shown by Figure 5.6. In the segmentation task, its uncertainty prediction proves particularly insightful, as highlighted by the red rectangles. For example, for the car hood, which is not part of the training labels, the model exhibits high uncertainty values, indicating its ability to capture out-of-distribution information or epistemic uncertainty. Similarly, in noisy background areas, the model effectively captures the aleatoric uncertainty and predicts high uncertainties for challenging areas like the wall on the right, demonstrating the benefit of uncertainties in identifying potential model errors.

In the depth estimation task, EMUFormer comparably predicts high uncertainty on the car hood and the sky, which are both areas without ground truth information, as well as at object boundaries, indicating sensitivity to depth discontinuities.

NYUv2. For the segmentation task, EMUFormer again outputs high uncertainties for pixels without ground truth information or that are misclassified, consistently providing useful predictive uncertainties, as shown by Figure 5.6. In the depth estimation task, the uncertainties seem to correlate with the estimated depth, providing an intuitive and helpful indication. This alignment suggests that the model effectively captures the depth prediction quality, particularly as it relates to increasing distances.



(a) Input Image (b) Seg. GT (c) Seg. Pred. (d) Seg. Unc.

(e) Depth GT (f) Depth Pred. (g) Depth Unc.

(h) Input Image (i) Seg. GT (j) Seg. Pred. (k) Seg. Unc.

(l) Depth GT (m) Depth Pred. (n) Depth Unc.

Figure 5.6: Qualitative examples of our EMUFormer-B2 on the Cityscapes [39] (top) and NYUv2 [245] (bottom) datasets. Red rectangles are added to highlight interesting areas.

Summary. In essence, the qualitative evaluation aligns with the quantitative findings of Section 5.4.2 and demonstrates the proficiency of EMUFormer in handling both the segmentation and the depth estimation tasks and its ability to generate meaningful predictive uncertainties that enable more thorough interpretations of the predictions.

Out-of-Domain Evaluation

In the following, we compare the SegDepthFormer baseline model, a SegDepthFormer DE with 10 members (teacher), and our EMUFormer on two OOD datasets: Foggy Cityscapes [235] and Rain Cityscape [109]. To evaluate the generalizability, we do not fine-tune any model.

Foggy Cityscapes. Compared to the original Cityscapes dataset, the Foggy Cityscapes dataset reveals significant performance degradation, as shown by Table 5.10. Both the baseline and DE models experience declines in predictive performance and calibration quality as the fog density increases, with the baseline showing more pronounced degradation. The DE demonstrates greater robustness, maintaining better performance and uncertainty quality even under severe fog conditions.

EMUFormer performs comparably to the DE in terms of predictive accuracy for the segmentation task while offering improved calibration, except for one case. Additionally, it delivers significantly better performance in terms of depth estimation. Regarding uncertainty quality, EMUFormer matches the DE in $p(\text{acc}|\text{cer})$ and PAvPU for SS but exhibits a notable improvement in $p(\text{unc}|\text{inacc})$. For depth estimation, EMUFormer provides equal or slightly better uncertainty quality across all evaluated metrics, further underscoring its effectiveness in handling challenging OOD scenarios without the computational overhead of a DE.

Rain Cityscapes. Table 5.11 highlights performance trends across varying levels of simulated rain. Both the baseline and DE models experience performance degradation as rain intensity increases, with the DE consistently demonstrating superior robustness in predictive performance, calibration quality, and uncertainty metrics.

Compared to the DE, EMUFormer shows mixed results under varying rain conditions. While it delivers strong calibration on par with the DE, except for one case, its performance in predictive accuracy decreases with more intense rain. Under the most challenging conditions, EMUFormer performs worse than the DE and even slightly worse than the baseline in SS accuracy, though it achieves significantly better results for depth estimation. Regarding segmentation uncertainty quality, EMUFormer is slightly worse than the DE for $p(\text{acc}|\text{cer})$ and PAvPU, but performs on par for $p(\text{unc}|\text{inacc})$. For depth uncertainty, EMUFormer matches the DE in $p(\text{acc}|\text{cer})$, performs slightly worse in $p(\text{unc}|\text{inacc})$, and slightly better in PAvPU, showcasing that it can almost match the performance of a DE in OOD scenarios while maintaining the computational efficiency of the baseline SegDepthFormer.

Summary. Overall, these results demonstrate that EMUFormer generalizes effectively to OOD scenarios without fine-tuning, achieving competitive performance compared to the DE while maintaining the computational efficiency of the baseline SegDepthFormer. EMUFormer matches or exceeds the DE in calibration and uncertainty quality for depth estimation and delivers robust segmentation performance under foggy conditions. Although performance slightly declines under heavy rain, particularly in SS, EMUFormer still offers strong depth estimation and uncertainty calibration, highlighting its ability to handle domain shifts efficiently without the need for ensemble-based methods.

		Semantic Segmentation					Monocular Depth Estimation			
		mIoU ↑	ECE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	RMSE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Cityscapes	SegDepthFormer	0.738	0.028	0.913	0.592	0.826	7.536	0.745	0.472	0.762
	SegDepthFormer (DE)	0.755	0.015	0.917	0.609	0.828	7.156	0.763	0.493	0.773
	EMUFormer	0.752	0.012	0.923	0.658	0.811	6.983	0.772	0.491	0.783
Foggy$_{\beta=0.005}$	SegDepthFormer	0.707	0.035	0.906	0.602	0.818	8.061	0.731	0.481	0.751
	SegDepthFormer (DE)	0.727	0.028	0.914	0.627	0.822	7.487	0.758	0.509	0.765
	EMUFormer	0.721	0.040	0.919	0.678	0.803	7.182	0.769	0.500	0.780
Foggy$_{\beta=0.01}$	SegDepthFormer	0.674	0.054	0.899	0.606	0.814	8.628	0.715	0.475	0.741
	SegDepthFormer (DE)	0.699	0.056	0.910	0.637	0.817	7.971	0.750	0.511	0.761
	EMUFormer	0.691	0.027	0.915	0.694	0.794	7.635	0.764	0.506	0.778
Foggy$_{\beta=0.02}$	SegDepthFormer	0.609	0.078	0.875	0.593	0.798	9.844	0.697	0.467	0.730
	SegDepthFormer (DE)	0.639	0.045	0.895	0.644	0.803	9.213	0.738	0.517	0.760
	EMUFormer	0.629	0.015	0.904	0.714	0.778	8.927	0.750	0.518	0.772

Table 5.10: Quantitative comparison between the SegDepthFormer baseline model, a SegDepthFormer DE with 10 members (teacher), and our EMUFormer (student) on the Foggy Cityscapes validation dataset [235] without fine-tuning. β denotes the attenuation coefficient and controls the thickness of the fog. Higher β values result in thicker fog. The original Cityscapes and the Foggy Cityscapes datasets share the same validation images, enabling a fair comparison between ID and OOD results.

		Semantic Segmentation					Monocular Depth Estimation			
		mIoU ↑	ECE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	RMSE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Rain$_1$	SegDepthFormer	0.608	0.020	0.936	0.658	0.810	7.187	0.792	0.558	0.767
	SegDepthFormer (DE)	0.673	0.004	0.954	0.741	0.813	6.740	0.804	0.559	0.767
	EMUFormer	0.647	0.006	0.943	0.739	0.784	6.538	0.805	0.534	0.774
Rain$_2$	SegDepthFormer	0.611	0.031	0.928	0.670	0.802	8.043	0.771	0.543	0.756
	SegDepthFormer (DE)	0.645	0.012	0.948	0.750	0.806	7.516	0.785	0.544	0.759
	EMUFormer	0.611	0.021	0.934	0.745	0.776	7.294	0.787	0.516	0.765
Rain$_3$	SegDepthFormer	0.582	0.045	0.917	0.671	0.795	8.848	0.751	0.534	0.749
	SegDepthFormer (DE)	0.612	0.023	0.943	0.756	0.799	8.294	0.767	0.535	0.755
	EMUFormer	0.576	0.026	0.928	0.751	0.767	8.033	0.772	0.510	0.761

Table 5.11: Quantitative comparison between the SegDepthFormer baseline model, a SegDepthFormer DE with 10 members (teacher), and our EMUFormer (student) on the Rain Cityscapes validation dataset [109] without fine-tuning. We evaluate on three sets of parameters, where Rain$_1$ uses [0.01, 0.005, 0.01], Rain$_2$ uses [0.02, 0.01, 0.005], and Rain$_3$ uses [0.03, 0.015, 0.002] as attenuation coefficients α and β and the raindrop radius a. α and β determine the degree of simulated rain and fog in the images.

Domain Adaptation

Domain adaptation is essential for achieving robust model performance across diverse environmental conditions by facilitating knowledge transfer to new domains. While some previous UQ distillation approaches [243, 46, 106, 147] have explored OOD performance, they have largely overlooked domain adaptation. This gap is particularly significant for applications such as autonomous driving, where re-training a teacher model – often implemented as a DE – and repeating the distillation process for every new domain is prohibitively expensive and operationally impractical. To address this, we propose a novel perspective for evaluating UQ distillation methods, emphasizing their capacity to adapt efficiently to domain shifts without requiring extensive re-training or re-distillation efforts.

More specifically, we evaluate the domain adaptation capabilities of EMUFormer by fine-tuning it on Foggy Cityscapes [235] and Rain Cityscapes [109]. This setting aligns with the simplest homogeneous domain adaptation paradigm as defined by Wang and Deng [272], where the source and target domains share identical feature spaces (semantically and dimensionally) but differ in input data distributions.

We follow the distillation process described in Section 5.4.1, with the exception of omitting the additional color jitter augmentation, as the training and distillation datasets are no longer identical, following Shen et al. [243]. EMUFormer, initially trained on Cityscapes, is fine-tuned using the ground truth labels from Foggy Cityscapes or Rain Cityscapes in conjunction with the outputs of the teacher DE trained on Cityscapes. The fine-tuning process is deliberately constrained to a single NVIDIA A100 GPU, with a maximum training duration of approximately 2.5 hours, corresponding to 10 epochs for Foggy Cityscapes and 100 epochs for Rain Cityscapes. This setup is designed to evaluate the domain adaptation capabilities of our approach while keeping computational efficiency in mind.

Quantitative Evaluation. Tables 5.12 and 5.13 show a quantitative comparison between the OOD and fine-tuning results of EMUFormer on Foggy Cityscapes and Rain Cityscapes, respectively. The results demonstrate significant benefits from domain adaptation across both SS and depth estimation tasks.

For SS, fine-tuning EMUFormer leads to improvements in mIoU of at least 2.8 % and up to 13.9 % compared to the OOD baseline and an improved softmax calibration, as measured by ECE, in 5 out of 6 cases. Segmentation uncertainty quality also improves in terms of $p(\text{acc}|\text{cer})$ and PAvPU. However, $p(\text{unc}|\text{inacc})$ shows a slight degradation, which can be attributed to the surprising strength of EMUFormer in terms of OOD performance, outperforming its DE teacher on this specific metric, as shown by Table 5.10 in the previous Section 5.4.2. In the depth estimation task, fine-tuning yields substantial performance gains, with reductions in RMSE ranging from 0.868 to 3.467. Depth uncertainty quality also improves consistently across all evaluated metrics, highlighting the robustness and strong performance of EMUFormer, particularly in terms of MDE, even while adapting to domain-specific conditions.

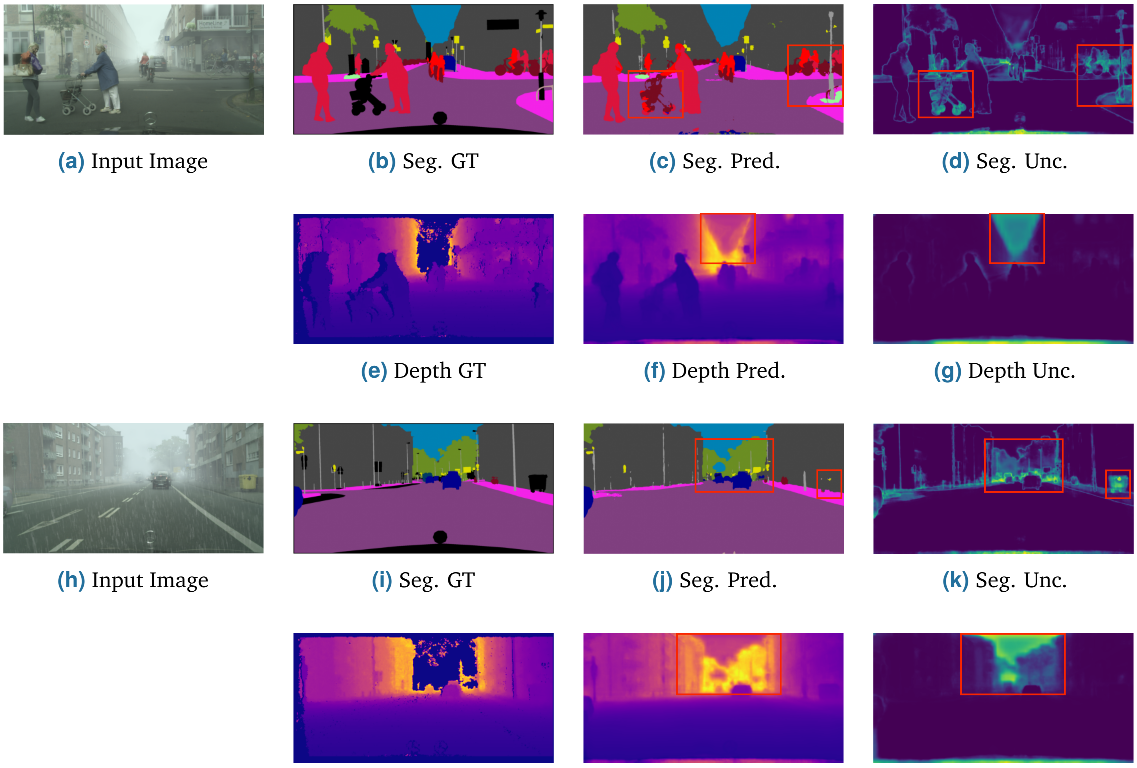
		Semantic Segmentation					Monocular Depth Estimation			
		mIoU ↑	ECE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	RMSE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Foggy$_{\beta=0.005}$	EMUFormer (OOD)	0.721	0.040	0.919	0.678	0.803	7.182	0.769	0.500	0.780
	EMUFormer (FT)	0.749 (+0.028)	0.011 (+0.029)	0.920 (+0.001)	0.631 (-0.047)	0.821 (+0.018)	6.314 (-0.868)	0.796 (+0.030)	0.522 (+0.022)	0.794 (+0.014)
Foggy$_{\beta=0.01}$	EMUFormer (OOD)	0.691	0.027	0.915	0.694	0.794	7.635	0.764	0.506	0.778
	EMUFormer (FT)	0.747 (+0.056)	0.019 (+0.008)	0.917 (+0.002)	0.635 (-0.059)	0.811 (+0.017)	5.631 (-2.004)	0.822 (+0.058)	0.547 (+0.041)	0.807 (+0.029)
Foggy$_{\beta=0.02}$	EMUFormer (OOD)	0.629	0.015	0.904	0.714	0.778	8.927	0.750	0.518	0.772
	EMUFormer (FT)	0.730 (+0.101)	0.004 (+0.011)	0.918 (+0.014)	0.662 (-0.052)	0.792 (+0.014)	5.463 (-3.464)	0.828 (+0.078)	0.563 (+0.045)	0.802 (+0.030)

Table 5.12: Quantitative comparison of out-of-domain (OOD) and fine-tuning (FT) results of our EMUFormer (student) on the Foggy Cityscapes validation dataset [235]. The parameter β, representing the attenuation coefficient, determines the fog density, with higher β values corresponding to denser fog conditions.

		Semantic Segmentation					Monocular Depth Estimation			
		mIoU ↑	ECE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑	RMSE ↓	p(acc|cer) ↑	p(unc|inacc) ↑	PAvPU ↑
Rain$_1$	EMUFormer (OOD)	0.647	0.006	0.943	0.739	0.784	6.538	0.805	0.534	0.774
	EMUFormer (FT)	0.727 (+0.080)	0.010 (-0.004)	0.958 (+0.015)	0.705 (-0.034)	0.822 (+0.038)	4.730 (-1.808)	0.866 (+0.061)	0.582 (+0.048)	0.798 (+0.024)
Rain$_2$	EMUFormer (OOD)	0.611	0.021	0.934	0.745	0.776	7.294	0.787	0.516	0.765
	EMUFormer (FT)	0.680 (+0.069)	0.018 (+0.003)	0.957 (+0.023)	0.713 (-0.032)	0.800 (+0.024)	4.941 (-2.353)	0.873 (+0.096)	0.629 (+0.113)	0.787 (+0.022)
Rain$_3$	EMUFormer (OOD)	0.576	0.026	0.928	0.751	0.767	8.033	0.772	0.510	0.761
	EMUFormer (FT)	0.715 (+0.139)	0.025 (+0.001)	0.960 (+0.052)	0.733 (-0.018)	0.793 (+0.026)	4.566 (-3.467)	0.876 (+0.104)	0.606 (+0.096)	0.799 (+0.038)

Table 5.13: Quantitative comparison of out-of-domain (OOD) and fine-tuning (FT) results of our EMUFormer (student) on the Rain Cityscapes validation dataset [109]. For fine-tuning, the student model was trained for 100 epochs using the original distillation process, except for omitting the additional color jitter augmentation. We evaluate on three sets of parameters, where Rain$_1$ uses [0.01, 0.005, 0.01], Rain$_2$ uses [0.02, 0.01, 0.005], and Rain$_3$ uses [0.03, 0.015, 0.002] for attenuation coefficients α and β and the raindrop radius a. α and β determine the degree of simulated rain and fog.

Qualitative Evaluation. Figure 5.7 presents qualitative examples of our domain-adapted, i.e., fine-tuned, EMUFormer-B2 on the most difficult versions of the Foggy Cityscapes [235] and Rain Cityscapes [109] validation datasets. EMUFormer demonstrates strong performance across both tasks and datasets, with uncertainty estimates effectively highlighting challenging regions. More specifically, on Foggy Cityscapes, segmentation uncertainty aligns with objects absent from the training data, such as an elderly person's walker in the foreground, as well as misclassified or noisy areas, exemplified by the region marked with a red rectangle in the right part of the image. Depth uncertainty is notably high for sky regions, where depth estimation is inherently ill-defined. Similarly, on Rain Cityscapes, the model assigns high uncertainty to out-of-distribution objects, like a dumpster, and to distant regions obscured by rain and fog, as indicated in the central part of the image. Depth uncertainty remains elevated for sky pixels and distant, occluded regions, reflecting the model's sensitivity to visually ambiguous or uninformative cues.



(a) Input Image

(b) Seg. GT

(c) Seg. Pred.

(d) Seg. Unc.

(e) Depth GT

(f) Depth Pred.

(g) Depth Unc.

(h) Input Image

(i) Seg. GT

(j) Seg. Pred.

(k) Seg. Unc.

(l) Depth GT

(m) Depth Pred.

(n) Depth Unc.

Figure 5.7: Qualitative examples of our domain-adapted EMUFormer-B2 on the Foggy Cityscapes [235] (top) and Rain Cityscapes [109] (bottom) datasets. Red rectangles are added to highlight interesting areas.

Summary. Overall, these findings align with the quantitative evaluations, demonstrating that our EMUFormer is capable of efficiently adapting to domain shifts without requiring extensive re-training or re-distillation efforts, while maintaining strong performance and reliable uncertainty estimates across both tasks.

Impact of Uncertainty Utilization

As described in Section 5.4.1 and shown by Equation 5.14, GNLL treats every prediction as a sample from a Gaussian distribution with a predictive mean and a corresponding predictive variance. Typically, these variances are learned implicitly through optimizing predictive means based on ground truth labels. However, with EMUFormer, the network is optimized to mimic the teacher's predictive uncertainty. This allows the depth uncertainty to be used explicitly to improve depth estimation. To explore this more thoroughly, we study the impact of the uncertainty utilization by replacing the GNLL loss with the MSE loss and the Huber loss [114], respectively, which do not account for the available predictive uncertainty.

Table 5.14 shows a quantitative comparison of the impact of the respective depth loss for EMUFormer-B2 on the Cityscapes and NYUv2 datasets. On Cityscapes, training with GNLL loss leads to the best performance across the board, especially concerning the RMSE for MDE. GNLL loss results in a RMSE of 6.983 in comparison to 7.217 and 7.340 for MSE and Huber loss [114], respectively. Similarly, on NYUv2, training with GNLL loss yields the best RMSE with 0.514 versus 0.527 and 0.533 for MSE and Huber loss [114], although at the cost of a very slight deterioration of 0.006 in mIoU. GNLL loss leads to the highest depth uncertainty quality for both datasets.

	Semantic Segmentation					Monocular Depth Estimation			
	mIoU ↑	ECE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	RMSE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑
Cityscapes									
MSE	0.749	0.014	0.922	**0.659**	0.810	7.217	0.742	0.446	0.761
Huber [114]	0.748	0.013	**0.923**	0.657	0.809	7.340	0.743	0.446	0.760
GNLL	**0.752**	**0.012**	**0.923**	0.658	**0.811**	**6.983**	**0.772**	**0.491**	**0.783**
NYUv2									
MSE	**0.481**	**0.127**	**0.788**	0.690	**0.737**	0.527	0.788	0.431	0.587
Huber [114]	**0.481**	**0.127**	**0.788**	0.689	**0.737**	0.533	0.786	0.431	0.587
GNLL	0.475	0.129	0.787	**0.692**	**0.737**	**0.514**	**0.810**	**0.440**	**0.633**

Table 5.14: Impact of the depth loss on the results of EMUFormer-B2 on Cityscapes [39] and NYUv2 [245].

Ablation Studies

Backbone Size. Table 5.15 displays a comprehensive assessment of the influence of the backbone size on Cityscapes [39] and NYUv2 [245]. In this context, we decided to evaluate the three baseline models as a DE with ten members each in comparison to EMUFormer for the smallest, B0, and the biggest, B5, backbone of SegFormer [284], respectively.

More specifically, EMUFormer emerges as the top performer on all segmentation metrics, except for the mIoU where the SegFormer DE yields slightly better results. On the Cityscapes dataset, EMUFormer stands out by delivering the best results for all depth metrics across both backbones. Notably, it achieves this superior performance while maintaining a 20 to 30 times faster inference time compared to the DEs. On NYUv2, the DepthFormer DE performs marginally better on the depth metrics, although EMUFormer remains highly competitive, especially if inference time is considered.

		Semantic Segmentation					Monocular Depth Estimation				Inference Time [ms]
		mIoU ↑	ECE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	RMSE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	
						Cityscapes					
B0	SegFormer (DE)	**0.689**	0.037	0.888	0.486	0.779	-	-	-	-	273.20 ± 1.38
	DepthFormer (DE)	-	-	-	-	-	8.452	0.692	0.414	0.719	236.13 ± 0.70
	SegDepthFormer (DE)	0.651	0.045	0.912	0.634	**0.803**	8.495	0.692	0.425	0.718	317.47 ± 15.64
	EMUFormer	0.630	**0.023**	**0.924**	**0.714**	0.791	**8.086**	**0.717**	**0.473**	**0.732**	**9.58 ± 0.07**
B5	SegFormer (DE)	**0.809**	0.032	0.896	0.435	0.819	-	-	-	-	1931.01 ± 12.77
	DepthFormer (DE)	-	-	-	-	-	6.588	0.782	0.487	0.791	1892.47 ± 9.24
	SegDepthFormer (DE)	0.789	0.037	0.928	0.657	**0.852**	6.664	0.785	0.502	0.792	2018.04 ± 32.31
	EMUFormer	0.771	**0.014**	**0.934**	**0.703**	0.845	**6.157**	**0.804**	**0.536**	**0.799**	**50.72 ± 0.45**
						NYUv2					
B0	SegFormer (DE)	**0.376**	0.105	0.743	0.701	0.718	-	-	-	-	315.42 ± 2.41
	DepthFormer (DE)	-	-	-	-	-	**0.642**	**0.720**	0.476	**0.566**	227.92 ± 2.39
	SegDepthFormer (DE)	0.375	0.097	**0.744**	0.703	0.718	0.678	0.693	0.466	0.553	346.21 ± 2.72
	EMUFormer	0.363	**0.090**	0.743	**0.713**	**0.720**	0.674	0.705	**0.498**	0.558	**10.04 ± 0.06**
B5	SegFormer (DE)	**0.534**	0.138	0.792	0.653	**0.744**	-	-	-	-	1958.46 ± 36.71
	DepthFormer (DE)	-	-	-	-	-	0.468	**0.852**	**0.505**	**0.647**	1875.53 ± 12.83
	SegDepthFormer (DE)	0.526	**0.133**	0.794	0.665	0.743	**0.451**	0.838	0.478	0.619	2038.26 ±13.06
	EMUFormer	0.520	0.134	**0.798**	**0.688**	**0.744**	0.476	0.846	0.467	**0.647**	**52.27 ± 1.40**

Table 5.15: Quantitative comparison on the Cityscapes [39] and NYUv2 [245] datasets between the three baseline models as DEs and EMUFormer with SegFormer's B0 and B5 backbone [284]. The respective SegDepthFormer DE served as the teacher for the corresponding EMUFormer.

5.5 Conclusion

Summary. Building upon the foundations established in the previous two Chapters 3 and 4 – where we explored UQ in SS and MDE – this Chapter aims to investigate three pivotal research questions regarding uncertainty-aware joint SS and MDE. Specifically, we investigated (1) how existing UQ methods perform in this joint setting, (2) how to enable efficient and reliable uncertainty estimates within this multi-task framework, and (3) how to exploit predictive uncertainties during training to optimize performance. To this end, we conducted a multi-task uncertainty evaluation and introduced EMUFormer.

Our investigation into the first research question revealed that DEs deliver the best predictive performance and uncertainty quality, albeit at significant computational cost. Additionally, we find that they exhibit greater robustness in OOD scenarios compared to the baseline. As a more practical alternative, DSEs strike a compelling balance between efficiency and effectiveness, offering comparable predictive performance and uncertainty quality with reduced overhead. Another valuable insight from this analysis is that multi-task learning can enhance the uncertainty quality of the SS task compared to the single-task approach, underscoring the synergistic benefits of joint task optimization. Furthermore, we show that while the choice of the uncertainty threshold significantly impacts metrics, its influence remains independent of the underlying model or approach, and the median uncertainty of an image proves to be a suitable default threshold.

To address the second research question, we introduced EMUFormer, a novel student-teacher distillation approach to overcome the computational limitations of many UQ methods like DEs. By distilling the UQ capabilities of a DE teacher into a lightweight student model, EMUFormer achieves state-of-the-art results in both SS and MDE on Cityscapes and NYUv2. Remarkably, it provides well-calibrated uncertainties without any additional inference time overhead, making it suitable for time-critical applications. EMUFormer even surpasses its teacher in specific scenarios – most notably in the depth estimation task – despite having an order of magnitude fewer parameters and approximately 30 times lower inference time.

For the third question, we demonstrated that EMUFormer reliably matches the DE teacher's overall performance, while consistently providing superior depth estimates. This success can be primarily attributed to the use of the GNLL loss, which is commonly employed to implicitly learn corresponding variances in addition to the predictive means. In the case of EMUFormer, however, the teacher model already provides high-quality variances through distillation, allowing for a more accurate approximation of the predictive means and their associated uncertainties. Notably, EMUFormer even exhibits robustness in OOD scenarios, matching the teacher's overall performance while consistently delivering superior depth estimates, highlighting the efficacy of exploiting uncertainties during training. Its potential for domain adaptation is equally promising, as it achieves substantial performance gains with minimal fine-tuning, filling a critical gap in current literature and opening exciting avenues for future work.

Collectively, these contributions reveal that UQ can be seamlessly integrated into complex multi-task learning frameworks without sacrificing efficiency or performance. By building on the findings of **D**eep **U**ncertainty **D**istillation using **E**nsembles for **S**egmentation (DUDES) from Chapter 3 and the GNLL-based fine-tuning approach of Chapter 4, this Chapter scales these principles to a more sophisticated joint task setting, enhancing both reliability and trustworthiness. EMUFormer's success in OOD robustness as well as domain adaptation clearly highlights the value of the consideration and exploitation of UQ for real-world, safety-critical machine vision applications.

Future Work. Looking ahead, future research could extend these ideas to other multi-task learning settings, including surface normal prediction as well as keypoint and edge detection [252]. This could also involve exploring how uncertainty estimates can be shared or combined across different tasks to improve overall robustness, perhaps discovering additional synergies between tasks. Additionally, incorporating multi-modal inputs – both visual and textual data – poses an interesting venue for future work [132].

Furthermore, advancing domain adaptation techniques within the context of UQ is essential, especially for real-world applications where constant re-training is impractical. Developing uncertainty-aware models that can efficiently and effectively adapt to new domains without extensive fine-tuning protocols should be a major focus of future research.

We hope that these findings and suggestions will inspire continued research not only into uncertainty-aware joint SS and MDE, but also into broader multi-modal and multi-task settings, in pursuit of more reliable machine vision systems for real-world deployment.

Synopsis 6

> *I do not think much of a man who is not wiser today than he was yesterday.*
>
> — **Abraham Lincoln**
> (16th U.S. President)

This final Chapter provides a synopsis encompassing key insights of Chapters 3, 4, and 5.

First, Section 6.1 concisely summarizes the preceding Chapters. Second, Section 6.2 jointly discusses all the findings of this thesis. Finally, Section 6.3 provides concluding remarks and outlines potential for future research opportunities.

6.1 Summary

This thesis, titled "Efficient Estimation and Exploitation of Predictive Uncertainties in Deep Learning-based Machine Vision", investigates the integration of Uncertainty Quantification (UQ) into foundational machine vision tasks, specifically Semantic Segmentation (SS), Monocular Depth Estimation (MDE), and their joint application within a multi-task learning framework. Although UQ has been identified as a promising approach to mitigate critical limitations of Deep Learning (DL) – such as overconfidence, lack of interpretability, and vulnerability to domain shifts – existing methods often suffer from high computational cost, the need for careful design choices, and a potential deterioration in predictive performance. It is undeniable that these drawbacks impede the development and deployment of uncertainty-aware machine vision systems in real-time and resource-constrained applications. To bridge this gap between academic research and practical real-world adoption, this thesis investigates the quality of existing UQ methods in novel settings and explores strategies to enable efficient UQ as well as exploiting these to guide the optimization process more effectively.

Uncertainty-aware Semantic Segmentation. Chapter 3 focuses on UQ in SS, which is one of the most foundational classification tasks in machine vision. Two novel approaches are introduced: **D**eep **U**ncertainty **D**istillation using **E**nsembles for **S**egmentation (DUDES) and **U**ncertainty-aware **C**ross-**E**ntropy (U-CE). DUDES employs a student-teacher distillation strategy to efficiently estimate uncertainties using a lightweight student model trained to mimic a Deep Ensemble (DE) teacher, reducing computational overhead while preserving segmentation and uncertainty quality. U-CE incorporates dynamic uncertainty estimates into the training process, weighting the regular Cross-Entropy (CE) loss to emphasize uncertain regions, thereby improving segmentation performance and enabling reliable uncertainty predictions post-training.

Uncertainty-aware Monocular Depth Estimation. Chapter 4 shifts to MDE, which represents an essential regression task in DL-based machine vision. In this context, we combined five existing UQ methods – Learned Confidence (LC), Gaussian Negative Log-Likelihood (GNLL), Monte Carlo Dropout (MCD), Deep Sub-Ensembles (DSEs), and Test-Time Augmentation (TTA) – with the DepthAnythingV2 foundation model. This analysis identifies GNLL-based fine-tuning as the most effective approach, delivering high-quality uncertainty estimates without compromising depth accuracy or computational efficiency, thus demonstrating the feasibility of synthesizing UQ with state-of-the-art MDE.

Uncertainty-aware Joint Semantic Segmentation and Monocular Depth Estimation. Chapter 5 explores UQ in the joint task of SS and MDE, effectively combining DL-based classification and regression within a multi-task learning framework. Based on our multi-task uncertainty evaluation, DEs excel in both predictive performance and uncertainty quality, though at high computational cost, whereas DSEs offer a more efficient yet competitive alternative. This analysis also reveals that multi-task learning can enhance the uncertainty quality, underscoring the synergistic benefits of joint task optimization. Additionally, this Chapter introduces the **E**fficient **M**ulti-task **U**ncertainty Vision Trans**former** (EMUFormer), a student-teacher distillation framework that achieves state-of-the-art results on the Cityscapes and NYUv2 benchmark datasets. EMUFormer builds on the findings of DUDES (Chapter 3) and the efficacy of GNLL-based fine-tuning (Chapter 4) to enable efficient estimation and exploitation of predictive uncertainties to improve explainability and predictive performance.

6.2 Discussion

A central challenge in UQ in DL-based machine vision – and the driving theme of this thesis – is the trade-off between computational efficiency and uncertainty quality, often hindering the exploitation of uncertainties. State-of-the-art methods like DEs excel at enhancing trustworthiness and reliability, for instance, by consistently correlating erroneous predictions with high uncertainties. However, their substantial computational overhead renders them impractical for real-time or resource-constrained environments. This thesis addresses this issue by introducing efficient, uncertainty-aware methods, offering practical solutions for real-world applications without compromising practicality.

The pursuit of efficient UQ is manifested by several approaches developed in this work: First, DUDES harnesses the concept of knowledge distillation to transfer the UQ capabilities of a computationally expensive DE teacher to a lightweight student model, achieving comparable uncertainty estimates with a single forward pass for SS. This significantly reduces inference time while maintaining quality and even proves to be robust against domain shifts – evidence of effective generalization rather than overfitting. Secondly, by synthesizing powerful foundation models for MDE with a variety of UQ methods, we find that GNLL-based fine-tuning strikes an appealing balance between computational efficiency and uncertainty quality without degrading predictive performance. Thirdly, EMUFormer builds on these findings and fuses both uncertainty distillation and GNLL-based fine-tuning for joint SS and MDE to deliver high-quality uncertainties with an inference time reduction of approximately 30 times. Together, these methods demonstrate that well-calibrated uncertainties can be achieved efficiently, enabling their use in time-sensitive, real-world scenarios.

Beyond efficient UQ, this thesis presents two approaches for how uncertainty can be actively exploited to enhance model performance. U-CE dynamically applies a pixel-wise weight to the loss to emphasize uncertain regions, guiding the model to focus on challenging areas and thereby improving accuracy. This transforms uncertainty from a passive, auxiliary output into an active driver of optimization. EMUFormer contributes in the same sense, leveraging high-quality uncertainties from a teacher model during training to surpass conventional GNLL-based fine-tuning, which is commonly utilized to learn uncertainties implicitly. In the case of EMUFormer, however, the teacher already provides high-quality variances through distillation, allowing for a more robust and accurate approximation of the predictive means, resulting in superior prediction and uncertainty quality. The possibility of enhancing model performance through the exploitation of predictive uncertainties underscores the value of uncertainty-aware training techniques to deliver practical benefits aside from increased explainability.

Collectively, these findings advance uncertainty-aware machine vision by addressing the efficiency-quality trade-off with practical and effective solutions. DUDES, U-CE, and EMUFormer not only make UQ computationally feasible but also demonstrate its potential to improve predictive performance, laying a robust foundation for real-world deployment across diverse machine vision tasks.

6.3 Conclusion and Outlook

This thesis advances the field of uncertainty-aware machine vision by developing practical and effective approaches for integrating UQ into SS, MDE, and their joint application. The proposed methods – DUDES, U-CE, and EMUFormer – enhance efficiency, robustness, and predictive performance, demonstrating that uncertainty is not merely an auxiliary output but can be a crucial component toward improving trustworthiness and applicability in real-world scenarios.

Building on this foundation, several exciting avenues for future research emerge, with the potential to impact the entire spectrum of machine vision tasks and learning paradigms.

Other Machine Vision Tasks. Since this thesis focuses on SS, MDE, and their joint application within a multi-task learning framework, future research could explore adopting the proposed methods to other foundational machine vision tasks like Object Detection and Pose Estimation, as well as exploring their feasibility for sophisticated decision-making processes such as autonomous driving, robot navigation or industrial inspection. This could not only enable preemptive identification of failure cases to enhance safety but also improve the performance of these systems.

Uncertainty-aware Multi-Modality. While most machine vision systems rely on visual data, integrating additional modalities – such as textual descriptions or sensor readings – could enhance the model's capabilities, particularly in scenarios where visual inputs are either limited or unreliable. Future research could explore the synergies of predictive uncertainties from these diverse sources by dynamically fusing modalities based on their reliability. For example, in autonomous driving, the predictive uncertainty could adjust the reliance on cameras versus LiDAR under varying conditions, improving safety and performance.

Uncertainty-aware Advanced Learning Paradigms. Beyond supervised learning, the exploitation of predictive uncertainties could further enhance several other learning paradigms like unsupervised learning, active learning, self-supervised learning, or semi-supervised learning. For instance, in semi-supervised learning, predictive uncertainties could refine noisy pseudo-labels, while methods like U-CE could emphasize uncertain ground truth regions to improve robustness.

Uncertainty-aware Data Augmentations. Techniques like CutMix [296] could be enhanced by using predictive uncertainties in the patch selection process. Instead of randomly choosing regions to mix, the uncertainty-aware method could select high-uncertainty patches and place them into low-uncertainty areas of other training samples. This approach would increase training difficulty, functioning as an online hard-example mining strategy [244]. By challenging the network to reinterpret complex regions in simpler contexts, this uncertainty-aware augmentation could enhance generalization and reduce overfitting. Moreover, focusing on uncertain regions during training may improve model calibration by explicitly addressing the network's blind spots.

Uncertainty-based Adversarial Attack Robustness. Given the susceptibility of DL models to adversarial perturbations, future work could introduce uncertainty-awareness to capture these attacks. Since prior studies show that adversarial attacks can target uncertainty estimates as well, it is crucial to develop training methods that ensure reliable uncertainties under such conditions [151].

Uncertainty-aware Domain Adaptation. Investigating how uncertainty-aware models handle domain adaptation is a crucial question, highly relevant for real-world deployment. Future research could assess the efficiency with which these models adapt to new domains and explore whether uncertainty can guide this process – akin to active learning – by prioritizing high-uncertainty samples for fine-tuning. Additionally, examining the extent of catastrophic forgetting [79, 130, 4] in the context of continual learning [271] could reveal critical insights into uncertainty-aware models' long-term stability and deployment potential.

Exploiting Predictive Uncertainties in Foundation Models. Due to computational constraints, this thesis did not explore large-scale foundation models, restraining investigations to efficient models. However, future work could examine whether uncertainty-aware training methods like those proposed in U-CE or EMUFormer can enhance these models' performance while contributing to their trustworthiness, potentially unlocking entirely new opportunities for deployment in high-stakes applications.

This thesis advocates for a shift toward uncertainty-aware machine vision, and I hope that it will inspire future research to go beyond incremental improvements on benchmark datasets and integrate uncertainty as a core component of Deep Neural Networks.

Bibliography

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi. “A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges”. In: *Information Fusion* 76 (2021), pp. 243–297.

[2] A. Agarwal and C. Arora. “Depthformer: Multiscale Vision Transformer for Monocular Depth Estimation with Global Local Information Fusion”. In: *IEEE International Conference on Image Processing*. IEEE. 2022, pp. 3873–3877.

[3] S. Aich, J. M. U. Vianney, M. A. Islam, and M. K. B. Liu. “Bidirectional Attention Network for Monocular Depth Estimation”. In: *IEEE International Conference on Robotics and Automation*. IEEE. 2021, pp. 11746–11752.

[4] E. L. Aleixo, J. G. Colonna, M. Cristo, and E. Fernandes. “Catastrophic Forgetting in Deep Learning: A Comprehensive Taxonomy”. In: *Journal of the Brazilian Computer Society* 30.1 (2024), pp. 175–211.

[5] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia. “Generative Adversarial Networks for Unsupervised Monocular Depth Prediction”. In: *Proceedings of the European Conference on Computer Vision Workshops*. 2018.

[6] A. Amini, W. Schwarting, A. Soleimany, and D. Rus. “Deep Evidential Regression”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14927–14937.

[7] A. Amini, A. Soleimany, S. Karaman, and D. Rus. “Spatial Uncertainty Sampling for End-to-end Control”. In: *arXiv preprint arXiv:1805.04829* (2018).

[8] V. Arampatzakis, G. Pavlidis, N. Mitianoudis, and N. Papamarkos. “Monocular Depth Estimation: A Thorough Review”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[9] M. S. Ayhan and P. Berens. “Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks”. In: *Medical Imaging with Deep Learning*. 2018.

[10] W. Bao, Q. Yu, and Y. Kong. “Evidential Deep Learning for Open Set Action Recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13349–13358.

[11] Y. Bengio. “Practical Recommendations for Gradient-based Training of Deep Architectures”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 437–478.

[12] V. Besnier, D. Picard, and A. Briot. “Learning Uncertainty for Safety-Oriented Semantic Segmentation in Autonomous Driving ”. In: *2021 IEEE International Conference on Image Processing*. IEEE, 2021, pp. 3353–3357. ISBN: 978-1-66544-115-5. DOI: 10.1109/ICIP42928.2021.9506719.

[13] S. F. Bhat, I. Alhashim, and P. Wonka. “AdaBins: Depth Estimation Using Adaptive Bins”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4009–4018.

[14] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. "ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth". In: *arXiv preprint arXiv:2302.12288* (2023).

[15] C. Bian, C. Yuan, J. Wang, M. Li, X. Yang, S. Yu, K. Ma, J. Yuan, and Y. Zheng. "Uncertainty-aware Domain Alignment for Anatomical Structure Segmentation". In: *Medical Image Analysis* 64 (2020), p. 101732.

[16] S. Bianco, R. Cadene, L. Celona, and P. Napoletano. "Benchmark Analysis of Representative Deep Neural Network Architectures ". In: *IEEE Access*. Vol. 6. 2018, pp. 64270–64277. DOI: 10.1109/ACCESS.2018.2877890.

[17] B. Bischke, P. Helber, D. Borth, and A. Dengel. "Segmentation of Imbalanced Classes in Satellite Imagery Using Adaptive Uncertainty Weighted Class Loss". In: *International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 6191–6194.

[18] C. M. Bishop and N. M. Nasrabadi. *Pattern Recognition and Machine Learning*. Vol. 4. 4. Springer, 2006.

[19] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. "Weight Uncertainty in Neural Network". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1613–1622.

[20] R. Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *arXiv preprint arXiv:2108.07258* (2021).

[21] P. O. Bressan, J. M. Junior, J. A. C. Martins, M. J. de Melo, D. N. Gonçalves, D. M. Freitas, A. P. M. Ramos, M. T. G. Furuya, L. P. Osco, J. de Andrade Silva, Z. Luo, R. C. Garcia, L. Ma, J. Li, and W. N. Gonçalves. "Semantic Segmentation with Labeling Uncertainty and Class Imbalance Applied to Vegetation Mapping". In: *International Journal of Applied Earth Observation and Geoinformation* 108 (2022), p. 102690.

[22] D. Bruggemann, M. Kanakis, S. Georgoulis, and L. Van Gool. "Automated Search for Resource-efficient Branched Multi-task Networks". In: *Procedings of the British Machine Vision Conference*. 2020, p. 359.

[23] D. Brüggemann, M. Kanakis, A. Obukhov, S. Georgoulis, and L. Van Gool. "Exploring Relational Context for Multi-task Dense Prediction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15869–15878.

[24] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao. "Adversarial Sensor Attack on Lidar-based Perception in Autonomous Driving". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 2267–2281.

[25] J. Chang, Z. Lan, C. Cheng, and Y. Wei. "Data Uncertainty Learning in Face Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5710–5719.

[26] B. Charpentier, D. Zügner, and S. Günnemann. "Posterior Network: Uncertainty Estimation Without Ood Samples via Density-based Pseudo-counts". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1356–1367.

[27] B. Chen, W. Peng, X. Cao, and J. Röning. "Hyperbolic Uncertainty Aware Semantic Segmentation". In: *IEEE Transactions on Intelligent Transportation Systems* 25.2 (2023), pp. 1275–1290.

[28] H. Chen, Z. Huang, H. Lam, H. Qian, and H. Zhang. "Learning Prediction Intervals for Regression: Generalization and Calibration". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 820–828.

[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *Proceedings of the European Conference on Computer Vision*. 2018.

[30] L. Chen, Z. Yang, J. Ma, and Z. Luo. “Driving Scene Perception Network: Real-Time Joint Detection, Depth Estimation and Semantic Segmentation”. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. 2018, pp. 1283–1291. DOI: 10.1109/WACV.2018.00145.

[31] W. Chen, Z. Fu, D. Yang, and J. Deng. “Single-image Depth Perception in the Wild”. In: *Advances in Neural Information Processing Systems* 29 (2016).

[32] W. Chen, S. Qian, D. Fan, N. Kojima, M. Hamilton, and J. Deng. “OASIS: A Large-scale Dataset for Single Image 3d in the Wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 679–688.

[33] K. Chitta, J. M. Alvarez, and A. Lesnikowski. “Large-scale Visual Active Learning with Deep Probabilistic Ensembles”. In: *arXiv preprint arXiv:1811.03575* (2018).

[34] H. Choi, H. Lee, S. Kim, S. Kim, S. Kim, K. Sohn, and D. Min. “Adaptive Confidence Thresholding for Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12808–12818.

[35] J. Choi, D. Chun, H. Kim, and H.-J. Lee. “Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 502–511.

[36] Y. Chuang, S. Zhang, and X. Zhao. “Deep Learning-based Panoptic Segmentation: Recent Advances and Perspectives”. In: *IET Image Processing* 17.10 (2023), pp. 2807–2828.

[37] T. Ciodaro, D. Deva, J. De Seixas, and D. Damazio. “Online Particle Detection with Neural Networks Based on Topological Calorimetry Information”. In: *Journal of Physics: Conference Series*. Vol. 368. 1. IOP Publishing. 2012, p. 012030.

[38] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. “Natural Language Processing (almost) From Scratch”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2493–2537.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3213–3223.

[40] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio. “Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1”. In: *arXiv preprint arXiv:1602.02830* (2016).

[41] M. Crawshaw. “Multi-task Learning with Deep Neural Networks: A Survey”. In: *arXiv preprint arXiv:2009.09796* (2020).

[42] A. CS Kumar, S. M. Bhandarkar, and M. Prasad. “Monocular Depth Prediction Using Generative Adversarial Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 300–308.

[43] G. E. Dahl, T. N. Sainath, and G. E. Hinton. “Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 8609–8613.

[44] A. Damianou. “Deep Gaussian Processes and Variational Propagation of Uncertainty”. PhD thesis. University of Sheffield, 2015.

[45] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, R. Jenatton, L. Beyer, M. Tschannen, A. Arnab, X. Wang, C. Riquelme, M. Minderer, J. Puigcerver, U. Evci, M. Kumar, S. van Steenkiste, G. F. Elsayed, A. Mahendran, F. Yu, A. Oliver, F. Huot, J. Bastings, M. P. Collier, A. Gritsenko, V. Birodkar, C. Vasconcelos, Y. Tay, T. Mensink, A. Kolesnikov, F. Pavetić, D. Tran, T. Kipf, M. Lučić, X. Zhai, D. Keysers, J. Harmsen, and N. Houlsby. "Scaling Vision Transformers to 22 Billion Parameters". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 7480–7512.

[46] D. Deng, L. Wu, and B. E. Shi. "Iterative Distillation for Better Uncertainty Estimates in Multitask Emotion Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2021, pp. 3557–3566.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei. "ImageNet : A Large-Scale Hierarchical Image Database". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPR.2009.5206848.

[48] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim. "An Analysis of Adversarial Attacks and Defenses on Autonomous Driving Models". In: *2020 IEEE International Conference on Pervasive Computing and Communications*. IEEE. 2020, pp. 1–10.

[49] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. "Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation". In: *Advances in Neural Information Processing Systems* 27 (2014).

[50] A. Der Kiureghian and O. Ditlevsen. "Aleatory or Epistemic? Does It Matter?" In: *Structural Safety* 31.2 (2009), pp. 105–112.

[51] G. Dikov and J. van Vugt. "Variational Depth Networks: Uncertainty-aware Monocular Self-supervised Depth Estimation". In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 43–60.

[52] X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass. "Towards Real-time Monocular Depth Estimation for Robotics: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 23.10 (2022), pp. 16940–16961.

[53] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2020.

[54] Y. Duan, X. Guo, and Z. Zhu. "DiffusionDepth: Diffusion Denoising Approach for Monocular Depth Estimation". In: *Proceedings of the European Conference on Computer Vision*. Springer. 2024, pp. 432–449.

[55] V. Edupuganti, M. Mardani, S. Vasanawala, and J. Pauly. "Uncertainty Quantification in Deep MRI Reconstruction". In: *IEEE Transactions on Medical Imaging* 40.1 (2020), pp. 239–250.

[56] A. Eftekhar, A. Sax, J. Malik, and A. Zamir. "Omnidata: A Scalable Pipeline for Making Multi-task Mid-level Vision Datasets From 3d Scans". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10786–10796.

[57] D. Eigen, C. Puhrsch, and R. Fergus. "Depth Map Prediction From a Single Image Using a Multi-scale Deep Network". In: *Advances in Neural Information Processing Systems* 27 (2014).

[58] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. "The Pascal Visual Object Classes (voc) Challenge". In: *International journal of computer vision* 88 (2010), pp. 303–338.

[59] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. “Learning Hierarchical Features for Scene Labeling”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2012), pp. 1915–1929.

[60] D. Feng, L. Rosenbaum, and K. Dietmayer. “Towards Safe Autonomous Driving: Capture Uncertainty in the Deep Neural Network for Lidar 3d Vehicle Detection”. In: *21st International Conference on Intelligent Transportation Systems*. IEEE. 2018, pp. 3266–3273.

[61] S. Fort, H. Hu, and B. Lakshminarayanan. “Deep Ensembles : A Loss Landscape Perspective ”. In: *arXiv preprint arXiv:1912.02757* (2019).

[62] G. Franchi, X. Yu, A. Bursuc, E. Aldea, S. Dubuisson, and D. Filliat. “Latent Discriminant Deterministic Uncertainty”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 243–260.

[63] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. “Deep Ordinal Regression Network for Monocular Depth Estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2002–2011.

[64] K. Fukushima. “Cognitron: A Self-organizing Multilayered Neural Network”. In: *Biological Cybernetics* 20.3 (1975), pp. 121–136.

[65] K. Fukushima. “Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements”. In: *IEEE Transactions on Systems Science and Cybernetics* 5.4 (1969), pp. 322–333.

[66] Y. Gal. “Uncertainty in Deep Learning”. PhD thesis. University of Cambridge, 2016.

[67] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. PMLR, 2016, pp. 1050–1059. URL: `https://proceedings.mlr.press/v48/gal16.html`.

[68] Y. Gal, R. Islam, and Z. Ghahramani. “Deep Bayesian Active Learning with Image Data”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1183–1192.

[69] T. Gao, W. Wei, Z. Cai, Z. Fan, S. Q. Xie, X. Wang, and Q. Yu. “CI-Net: A Joint Depth Estimation and Semantic Segmentation Network Using Contextual Information”. In: *Applied Intelligence* 52.15 (2022), pp. 18167–18186.

[70] T. Gao, W. Wei, X. Wang, Q. Yu, and Z. Fan. “Predictive Uncertainties for Multi-task Learning Network”. In: *International Conference on Advanced Algorithms and Neural Networks*. Vol. 12285. SPIE. 2022, pp. 294–300.

[71] Y. Gao and M. K. Ng. “Wasserstein Generative Adversarial Uncertainty Quantification in Physics-informed Neural Networks”. In: *Journal of Computational Physics* 463 (2022), p. 111270.

[72] C. F. Gauss. *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*. Vol. 7. FA Perthes, 1877.

[73] C.-F. Gauss. *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae*. Henricus Dieterich, 1823.

[74] J. Gawlikowski, S. Saha, A. Kruspe, and X. X. Zhu. “An Advanced Dirichlet Prior Network for Out-of-distribution Detection in Remote Sensing”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–19.

[75] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu. “A Survey of Uncertainty in Deep Neural Networks”. In: *Artificial Intelligence Review* 56 (2023), pp. 1513–1589.

[76] A. Geiger, P. Lenz, and R. Urtasun. “Are We Ready for Autonomous Driving? the Kitti Vision Benchmark Suite”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3354–3361.

[77] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU press, 2013.

[78] I. Goodfellow, Y. Bengio, and A. Courville. “Regularization for Deep Learning”. In: *Deep learning* (2016), pp. 216–261.

[79] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. “An Empirical Investigation of Catastrophic Forgetting in Gradient-based Neural Networks”. In: *arXiv preprint arXiv:1312.6211* (2013).

[80] J. Gou, B. Yu, S. J. Maybank, and D. Tao. “Knowledge Distillation: A Survey”. In: *International Journal of Computer Vision* 129.6 (2021), pp. 1789–1819.

[81] W. Gu, S. Bai, and L. Kong. “A Review on 2d Instance Segmentation Based on Deep Neural Networks”. In: *Image and Vision Computing* 120 (2022), p. 104401.

[82] A. B. Guillaumes. “Mixture Density Networks for Distribution and Uncertainty Estimation”. PhD thesis. Universitat Politècnica de Catalunya. Facultat d’Informàtica de Barcelona, 2017.

[83] V. Guizilini, I. Vasiljevic, D. Chen, R. Ambruș, and A. Gaidon. “Towards Zero-shot Scale-aware Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 9233–9243.

[84] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. “On Calibration of Modern Neural Networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1321–1330.

[85] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew. “A Review of Semantic Segmentation Using Deep Neural Networks”. In: *International Journal of Multimedia Information Retrieval* 7 (2018), pp. 87–93.

[86] C. Gurau, A. Bewley, and I. Posner. “Dropout Distillation for Efficiently Estimating Model Confidence”. In: *arXiv preprint arXiv:1809.10562* (2018).

[87] F. K. Gustafsson, M. Danelljan, and T. B. Schon. “Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 318–319.

[88] I. Hacking. *The Emergence of Probability: A Philosophical Study of Early Ideas About Probability, Induction and Statistical Inference*. Cambridge University Press, 1975.

[89] A. M. Hafiz and G. M. Bhat. “A Survey on Instance Segmentation: State of the Art”. In: *International Journal of Multimedia Information Retrieval* 9.3 (2020), pp. 171–189.

[90] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Y. Zhaohui, Y. Zhang, and D. Tao. “A Survey on Vision Transformer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2022), pp. 87–110.

[91] S. Han, J. Pool, J. Tran, and W. Dally. “Learning Both Weights and Connections for Efficient Neural Network”. In: *Advances in Neural Information Processing Systems* 28 (2015).

[92] S. Hao, Y. Zhou, and Y. Guo. “A Brief Survey on Semantic Segmentation with Deep Learning”. In: *Neurocomputing* 406 (2020), pp. 302–321.

[93] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. “Simultaneous Detection and Segmentation”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014, pp. 297–312.

[94] K. He, R. Girshick, and P. Dollar. “Rethinking ImageNet Pre-Training”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

[95] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[96] L. He, J. Lu, G. Wang, S. Song, and J. Zhou. “SOSD-Net: Joint Semantic Object Segmentation and Depth Estimation From Monocular Images”. In: *Neurocomputing* 440 (2021), pp. 251–263.

[97] W. He, Z. Jiang, T. Xiao, Z. Xu, and Y. Li. “A Survey on Uncertainty Quantification Methods for Deep Learning”. In: *arXiv preprint arXiv:2302.13425* (2023).

[98] R. Hecht-Nielsen. “Theory of the Backpropagation Neural Network”. In: *Neural Networks for Perception*. Elsevier, 1992, pp. 65–93.

[99] M. Heizmann, A. Braun, M. Glitzner, M. Günther, G. Hasna, C. Klüver, J. Krooß, E. Marquardt, M. Overdick, and M. Ulrich. “Implementing Machine Learning: Chances and Challenges”. In: *at-Automatisierungstechnik* 70.1 (2022), pp. 90–101.

[100] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk. “Connectomic Reconstruction of the Inner Plexiform Layer in the Mouse Retina”. In: *Nature* 500.7461 (2013), pp. 168–174.

[101] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. “Deep Neural Networks for Acoustic Modeling in Speech Recognition: the Shared Views of Four Research Groups”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[102] G. Hinton, O. Vinyals, and J. Dean. “Distilling the Knowledge in a Neural Network”. In: *NIPS Deep Learning and Representation Learning Workshop*. 2015. URL: http://arxiv.org/abs/1503.02531.

[103] N. Hirose, S. Taguchi, K. Kawano, and S. Koide. “Variational Monocular Depth Estimation for Reliability Prediction”. In: *2021 International Conference on 3d Vision (3DV)*. IEEE. 2021, pp. 637–647.

[104] T. Hodan, M. Sundermeyer, B. Drost, Y. Labbe, E. Brachmann, F. Michel, C. Rother, and J. Matas. “BOP Challenge 2020 on 6d Object Localization”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 577–594.

[105] T. Hodan, M. Sundermeyer, Y. Labbe, V. N. Nguyen, G. Wang, E. Brachmann, B. Drost, V. Lepetit, C. Rother, and J. Matas. “BOP Challenge 2023 on Detection Segmentation and Pose Estimation of Seen and Unseen Rigid Objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 5610–5619.

[106] C. J. Holder and M. Shafique. “Efficient Uncertainty Estimation in Semantic Segmentation via Distillation ”. In: *IEEE/CVF International Conference on Computer Vision Workshops*. IEEE, 2021, pp. 3080–3087. ISBN: 978-1-66540-191-3. DOI: 10.1109/ICCVW54120.2021.00343.

[107] J. Hornauer and V. Belagiannis. “Gradient-based Uncertainty for Monocular Depth Estimation”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 613–630.

[108] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. “MobileNets : Efficient Convolutional Neural Networks for Mobile Vision Applications ”. In: *arXiv preprint arXiv:1704.04861* (2017).

[109] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng. “Depth-attentional Features for Single-image Rain Removal”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8022–8031.

[110] Y. Hu, Z. Chen, and W. Lin. “RGB-D Semantic Segmentation: A Review”. In: *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2018, pp. 1–6.

[111] W. Huang, J. Zhang, and K. Huang. “Bootstrap Estimated Uncertainty of the Environment Model for Model-based Reinforcement Learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 3870–3877.

[112] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. “Binarized Neural Networks”. In: *Advances in Neural Information Processing Systems* 29 (2016).

[113] D. H. Hubel and T. N. Wiesel. “Receptive Fields and Functional Architecture of Monkey Striate Cortex”. In: *The Journal of Physiology* 195.1 (1968), pp. 215–243.

[114] P. J. Huber. “Robust Estimation of a Location Parameter”. In: *Breakthroughs in Statistics: Methodology and Distribution*. Springer, 1992, pp. 492–518.

[115] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *International Conference on Machine Learning*. pmlr. 2015, pp. 448–456.

[116] S. Jadon. “A Survey of Loss Functions for Semantic Segmentation”. In: *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*. IEEE. 2020, pp. 1–7.

[117] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. “What Is the Best Multi-stage Architecture for Object Recognition?” In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, pp. 2146–2153.

[118] S. Jean, K. Cho, R. Memisevic, and Y. Bengio. “On Using Very Large Target Vocabulary for Neural Machine Translation”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (volume 1: Long Papers)*. 2015, pp. 1–10.

[119] N. Ji, H. Dong, F. Meng, and L. Pang. “Semantic Segmentation and Depth Estimation Based on Residual Attention Mechanism”. In: *Sensors* 23.17 (2023), p. 7466.

[120] Y. Ji, Z. Chen, E. Xie, L. Hong, X. Liu, Z. Liu, T. Lu, Z. Li, and P. Luo. “DDP: Diffusion Model for Dense Visual Prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 21741–21752.

[121] J. Jiao, Y. Cao, Y. Song, and R. Lau. “Look Deeper Into Depth: Monocular Depth Estimation with Semantic Booster and Attention-driven Loss”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 53–69.

[122] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine. “Uncertainty-aware Reinforcement Learning for Collision Avoidance”. In: *arXiv preprint arXiv:1702.01182* (2017).

[123] M. Kalia, N. Navab, and T. Salcudean. “A Real-time Interactive Augmented Reality Depth Estimation Technique for Surgical Robotics”. In: *International Conference on Robotics and Automation*. IEEE. 2019, pp. 8291–8297.

[124] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler. “Repurposing Diffusion-based Image Generators for Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9492–9502.

[125] A. Kendall, V. Badrinarayanan, and R. Cipolla. “Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding”. In: *Procedings of the British Machine Vision Conference*. 2017.

[126] A. Kendall and Y. Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 5580–5590. ISBN: 9781510860964.

[127] A. Kendall, Y. Gal, and R. Cipolla. “Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491.

[128] F. Khan, S. Salahuddin, and H. Javidnia. “Deep Learning-based Monocular Depth Estimation Methods—a State-of-the-art Review”. In: *Sensors* 20.8 (2020), p. 2272.

[129] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. “Panoptic Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9404–9413.

[130] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. “Overcoming Catastrophic Forgetting in Neural Networks”. In: *Proceedings of the National Academy of Sciences* 114.13 (2017), pp. 3521–3526.

[131] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. Eslami, D. Jimenez Rezende, and O. Ronneberger. “A Probabilistic U-Net for Segmentation of Ambiguous Images”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[132] V. Kostumov, B. Nutfullin, O. Pilipenko, and E. Ilyushin. “Uncertainty-aware Evaluation for Vision-language Models”. In: *arXiv preprint arXiv:2402.14418* (2024).

[133] A. Kristiadi, M. Hein, and P. Hennig. “Learnable Uncertainty Under Laplace Approximations”. In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 344–353.

[134] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems* 25 (2012).

[135] J. Kruger and D. Dunning. “Unskilled and Unaware of It: How Difficulties in Recognizing One’s Own Incompetence Lead to Inflated Self-assessments.” In: *Journal of personality and social psychology* 77.6 (1999), p. 1121.

[136] J. Kukačka, V. Golkov, and D. Cremers. “Regularization for Deep Learning: a Taxonomy”. In: *arXiv preprint arXiv:1710.10686* (2017).

[137] T. LaBonte, C. Martinez, and S. A. Roberts. “We Know Where We Don’t Know: 3d Bayesian CNNs for Credible Geometric Uncertainty”. In: *arXiv preprint arXiv:1910.10793* (2019).

[138] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[139] S. Landgraf, M. Hilleman, M. Aberle, V. Jung, and M. Ulrich. “Segmentation of Industrial Burner Flames: A Comparative Study From Traditional Image Processing to Machine Learning and Deep Learning”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 10 (2023), pp. 953–960.

[140] S. Landgraf, M. Hilleman, T. Kapler, and M. Ulrich. “Evaluation of Multi-task Uncertainties in Joint Semantic Segmentation and Monocular Depth Estimation”. In: *Forum Bildverarbeitung 2024*. 2024, p. 147.

[141] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich. “A Comparative Study on Multi-task Uncertainty Quantification in Semantic Segmentation and Monocular Depth Estimation”. In: *tm-Technisches Messen* (2025).

[142] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich. “Efficient Multi-task Uncertainties for Joint Semantic Segmentation and Monocular Depth Estimation”. In: *DAGM German Conference on Pattern Recognition*. Springer. 2024, pp. 348–364.

[143] S. Landgraf, M. Hillemann, K. Wursthorn, and M. Ulrich. “Uncertainty-aware Cross-Entropy for Semantic Segmentation”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 10 (2024), pp. 129–136.

[144] S. Landgraf, J. Huber, M. Hilleman, and M. Ulrich. “Evaluation of Semi-supervised Semantic Segmentation for Remote Sensing, Medical Imaging, and Machine Vision Settings”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* ... (2025), p. ...

[145] S. Landgraf, L. Kühnlein, M. Hoyer, S. Keller, and M. Ulrich. “Evaluation of Self-Supervised Learning Approaches for Semantic Segmentation of Industrial Burner Flames”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 43 (2022), pp. 601–607.

[146] S. Landgraf, R. Qin, and M. Ulrich. “A Critical Synthesis of Uncertainty Quantification and Foundation Models in Monocular Depth Estimation”. In: *arXiv preprint arXiv:2501.08188* (2025).

[147] S. Landgraf, K. Wursthorn, M. Hillemann, and M. Ulrich. “DUDES: Deep Uncertainty Distillation Using Ensembles for Semantic Segmentation”. In: *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 92.2 (2024), pp. 101–114.

[148] F. Lateef and Y. Ruichek. “Survey on Semantic Segmentation Using Deep Learning Techniques”. In: *Neurocomputing* 338 (2019), pp. 321–348.

[149] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551.

[150] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[151] E. Ledda, D. Angioni, G. Piras, G. Fumera, B. Biggio, and F. Roli. “Adversarial Attacks Against Uncertainty Quantification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4599–4608.

[152] H.-y. Lee. *Self-attention*. https://speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/self_v7.pdf. Accessed: 2025-02-03.

[153] J. Lee, J. Feng, M. Humt, M. G. Müller, and R. Triebel. “Trust Your Robots! Predictive Uncertainty Estimation of Neural Networks with Sparse Gaussian Processes”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 1168–1179.

[154] K. Lee, H. Lee, K. Lee, and J. Shin. “Training Confidence-calibrated Classifiers for Detecting Out-of-distribution Samples”. In: *International Conference on Learning Representations*. 2018.

[155] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl. “Leveraging Uncertainty Information From Deep Neural Networks for Disease Detection”. In: *Scientific Reports* 7.1 (2017), p. 17816. ISSN: 2045-2322. DOI: 10.1038/s41598-017-17876-z.

[156] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. “Deep Learning of the Tissue-regulated Splicing Code”. In: *Bioinformatics* 30.12 (2014), pp. i121–i129.

[157] B. Li, Y. Shi, Z. Qi, and Z. Chen. “A Survey on Semantic Segmentation”. In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 1233–1240.

[158] X. Li and D. Chen. “A Survey on Deep Learning-based Panoptic Segmentation”. In: *Digital Signal Processing* 120 (2022), p. 103283.

[159] Z. Li and N. Snavely. “MegaDepth: Learning Single-view Depth Prediction From Internet Photos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2041–2050.

[160] Z. Li, Z. Chen, X. Liu, and J. Jiang. “Depthformer: Exploiting Long-range Correlation and Local Information for Accurate Monocular Depth Estimation”. In: *Machine Intelligence Research* 20.6 (2023), pp. 837–854.

[161] Z. Li, X. Wang, X. Liu, and J. Jiang. “BinsFormer: Revisiting Adaptive Bins for Monocular Depth Estimation”. In: *IEEE Transactions on Image Processing* (2024).

[162] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. “Focal Loss for Dense Object Detection”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2980–2988.

[163] X. Lin, D. Sánchez-Escobedo, J. R. Casas, and M. Pardàs. “Depth Estimation and Semantic Segmentation From a Single RGB Image Using a Hybrid Convolutional Neural Network”. In: *Sensors* 19.8 (2019), p. 1795.

[164] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. “Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7498–7512.

[165] J. Liu, Y. Wang, Y. Li, J. Fu, J. Li, and H. Lu. “Collaborative Deconvolutional Neural Networks for Joint Depth Estimation and Semantic Segmentation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 29.11 (2018), pp. 5655–5666.

[166] S. Liu, E. Johns, and A. J. Davison. “End-to-end Multi-task Learning with Attention”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1871–1880.

[167] J. Long, E. Shelhamer, and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[168] A. Loquercio, M. Segu, and D. Scaramuzza. “A General Framework for Uncertainty Estimation in Deep Learning”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3153–3160.

[169] I. Loshchilov and F. Hutter. “Decoupled Weight Decay Regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[170] C. Louizos and M. Welling. “Multiplicative Normalizing Flows for Variational Bayesian Neural Networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2218–2227.

[171] B. Lütjens, M. Everett, and J. P. How. “Safe Reinforcement Learning with Model Uncertainty Estimates”. In: *International Conference on Robotics and Automation*. IEEE. 2019, pp. 8662–8668.

[172] D. J. C. MacKay. “A Practical Bayesian Framework for Backpropagation Networks ”. In: *Neural Computation* 4.3 (1992), pp. 448–472. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/neco.1992.4.3.448.

[173] A. Malinin and M. Gales. “Predictive Uncertainty Estimation via Prior Networks”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[174] A. Malinin, B. Mlodozeniec, and M. Gales. “Ensemble Distribution Distillation ”. In: *arXiv preprint arXiv:1905.00076* (2019).

[175] T. Mallick, P. Balaprakash, and J. Macfarlane. “Deep-ensemble-based Uncertainty Quantification in Spatiotemporal Graph Neural Networks for Traffic Forecasting”. In: *arXiv preprint arXiv:2204.01618* (2022).

[176] S. Marcel and Y. Rodriguez. “Torchvision the Machine-vision Package of Torch”. In: *Proceedings of the 18th ACM International Conference on Multimedia*. 2010, pp. 1485–1488.

[177] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig. “Monocular Depth Estimation Using Deep Learning: A Review”. In: *Sensors* 22.14 (2022), p. 5353.

[178] R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, and A. Weller. “Concrete Problems for Autonomous Vehicle Safety : Advantages of Bayesian Deep Learning ”. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2017, pp. 4745–4753. DOI: 10.24963/ijcai.2017/661.

[179] W. S. McCulloch and W. Pitts. “A Logical Calculus of the Ideas Immanent in Nervous Activity”. In: *The Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133.

[180] A. Mehrtash, W. M. Wells, C. M. Tempany, P. Abolmaesumi, and T. Kapur. “Confidence Calibration and Predictive Uncertainty Estimation for Deep Medical Image Segmentation”. In: *IEEE Transactions on Medical Imaging* 39.12 (2020), pp. 3868–3878.

[181] J. Mena, O. Pujol, and J. Vitrià. “A Survey on Uncertainty Estimation in Deep Learning Classification Systems From a Bayesian Perspective”. In: *ACM Computing Surveys (CSUR)* 54.9 (2021), pp. 1–35.

[182] L. Mi, H. Wang, Y. Tian, H. He, and N. N. Shavit. “Training-free Uncertainty Estimation for Dense Regression: Sensitivity as a Surrogate”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 9. 2022, pp. 10042–10050.

[183] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. “Mixed Precision Training”. In: *arXiv preprint arXiv:1710.03740* (2017).

[184] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černockỳ. “Strategies for Training Large Scale Neural Network Language Models”. In: *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE. 2011, pp. 196–201.

[185] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. “Image Segmentation Using Deep Learning: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2021), pp. 3523–3542.

[186] Y. Ming, X. Meng, C. Fan, and H. Yu. “Deep Learning for Monocular Depth Estimation: A Review”. In: *Neurocomputing* 438 (2021), pp. 14–33.

[187] J. Mitros and B. Mac Namee. “On the Validity of Bayesian Neural Networks for Uncertainty Estimation”. In: *arXiv preprint arXiv:1912.01530* (2019).

[188] A. Mousavian, H. Pirsiavash, and J. Košecká. “Joint Semantic Segmentation and Depth Estimation with Deep Convolutional Networks”. In: *Fourth International Conference on 3d Vision*. IEEE. 2016, pp. 611–619.

[189] B. Mucsányi, M. Kirchhof, and S. J. Oh. “Benchmarking Uncertainty Disentanglement: Specialized Uncertainties for Specialized Tasks”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 50972–51038.

[190] J. Mukhoti and Y. Gal. “Evaluating Bayesian Deep Learning Methods for Semantic Segmentation”. In: *arXiv preprint arXiv:1811.12709* (2018).

[191] J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal. “Deep Deterministic Uncertainty: A New Simple Baseline”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 24384–24394.

[192] M. P. Naeini, G. Cooper, and M. Hauskrecht. “Obtaining Well Calibrated Probabilities Using Bayesian Binning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015.

[193] T. Nair, D. Precup, D. L. Arnold, and T. Arbel. “Exploring Uncertainty Measures in Deep Networks for Multiple Sclerosis Lesion Detection and Segmentation”. In: *Medical Image Analysis* 59 (2020), p. 101557.

[194] V. Nair and G. E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *International Conference on Machine Learning*. 2010, pp. 807–814.

[195] R. M. Neal. *Bayesian Learning for Neural Networks*. Vol. 118. Springer Science & Business Media, 2012.

[196] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid. “Real-time Joint Semantic Segmentation and Depth Estimation Using Asymmetric Annotations”. In: *International Conference on Robotics and Automation*. IEEE. 2019, pp. 7101–7107.

[197] F. Nex, E. Stathopoulou, F. Remondino, M. Yang, L. Madhuanand, Y. Yogender, B. Alsadik, M. Weinmann, B. Jutzi, and R. Qin. “UseGeo-A UAV-based Multi-sensor Dataset for Geospatial Research”. In: *ISPRS Open Journal of Photogrammetry and Remote Sensing* (2024), p. 100070.

[198] V.-L. Nguyen, S. Destercke, and E. Hüllermeier. “Epistemic Uncertainty Sampling”. In: *Discovery Science: 22nd International Conference, Proceedings 22*. Springer. 2019, pp. 72–86.

[199] X. Nie, D. Shi, R. Li, Z. Liu, and X. Chen. “Uncertainty-aware Self-improving Framework for Depth Estimation”. In: *IEEE Robotics and Automation Letters* 7.1 (2021), pp. 41–48.

[200] J. Ning, C. Li, Z. Zhang, C. Wang, Z. Geng, Q. Dai, K. He, and H. Hu. “All in Tokens: Unifying Output Space of Visual Tasks via Soft Token”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 19900–19910.

[201] D. A. Nix and A. S. Weigend. “Estimating the Mean and Variance of the Target Probability Distribution”. In: *Proceedings of 1994 Ieee International Conference on Neural Networks (ICNN’94)*. Vol. 1. IEEE. 1994, pp. 55–60.

[202] P. Oberdiek, G. Fink, and M. Rottmann. “UQGAN: A Unified Model for Uncertainty Quantification of Deep Classifiers Trained via Conditional Gans”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21371–21385.

[203] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. “DINOv2: Learning Robust Visual Features Without Supervision”. In: *Transactions on Machine Learning Research Journal* (2024), pp. 1–31.

[204] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. “Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[205] Y. Park and M. Kellis. “Deep Learning for Regulatory Genomics”. In: *Nature Biotechnology* 33.8 (2015), pp. 825–826.

[206] V. Patil, C. Sakaridis, A. Liniger, and L. Van Gool. “P3Depth: Monocular Depth Estimation with a Piecewise Planarity Prior”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1610–1621.

[207] S. Patni, A. Agarwal, and C. Arora. “ECoDepth: Effective Conditioning of Diffusion Models for Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 28285–28295.

[208] T. Pearce, A. Brintrup, M. Zaki, and A. Neely. “High-quality Prediction Intervals for Deep Learning: A Distribution-free, Ensembled Approach”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4075–4084.

[209] X. Peng, F. Qiao, and L. Zhao. “Out-of-domain Generalization From a Single Source: An Uncertainty Quantification Approach”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.3 (2022), pp. 1775–1787.

[210] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean. “Efficient Neural Architecture Search via Parameters Sharing”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4095–4104.

[211] L. Piccinelli, C. Sakaridis, and F. Yu. “iDisc: Internal Discretization for Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 21477–21487.

[212] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, and F. Yu. “UniDepth: Universal Monocular Metric Depth Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 10106–10116.

[213] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia. “On the Uncertainty of Self-supervised Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3227–3237.

[214] K. Posch, J. Steinbrener, and J. Pilz. “Variational Inference to Measure Model Uncertainty in Deep Neural Networks”. In: *arXiv preprint arXiv:1902.10189* (2019).

[215] S. Prokudin, P. Gehler, and S. Nowozin. “Deep Directional Statistics: Pose Estimation with Uncertainty Quantification”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 534–551.

[216] R. Ranftl, A. Bochkovskiy, and V. Koltun. “Vision Transformers for Dense Prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12179–12188.

[217] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3 (2020), pp. 1623–1637.

[218] M. Ranzato, Y.-L. Boureau, and Y. Cun. “Sparse Feature Learning for Deep Belief Networks”. In: *Advances in Neural Information Processing Systems* 20 (2007).

[219] M. Rawat, M. Wistuba, and M.-I. Nicolae. “Harnessing Model Uncertainty for Detecting Adversarial Examples”. In: *NIPS Workshop on Bayesian Deep Learning*. 2017.

[220] J. C. Reinhold, Y. He, S. Han, Y. Chen, D. Gao, J. Lee, J. L. Prince, and A. Carass. “Validating Uncertainty in Medical Image Translation”. In: *2020 IEEE 17th International Symposium on Biomedical Imaging*. IEEE. 2020, pp. 95–98.

[221] H. Ritter, A. Botev, and D. Barber. “A Scalable Laplace Approximation for Neural Networks”. In: *International Conference on Learning Representations*. Vol. 6. International Conference on Representation Learning. 2018.

[222] H. Ritter, A. Botev, and D. Barber. “Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[223] H. Robbins and S. Monro. “A Stochastic Approximation Method ”. In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729586.

[224] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. “FitNets: Hints for Thin Deep Nets”. In: *arXiv preprint arXiv:1412.6550* (2015).

[225] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[226] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. 1961.

[227] F. Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” In: *Psychological Review* 65.6 (1958), p. 386.

[228] N. Rosenfeld, Y. Mansour, and E. Yom-Tov. “Discriminative Learning of Prediction Intervals”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 347–355.

[229] T. Roussel, L. Van Eycken, and T. Tuytelaars. “Monocular Depth Estimation in New Environments with Absolute Scale”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2019, pp. 1735–1741.

[230] A. G. Roy, S. Conjeti, N. Navab, C. Wachinger, and A. D. N. Initiative. “Bayesian QuickNAT: Model Uncertainty in Deep Whole-brain Segmentation for Structure-wise Quality Control”. In: *NeuroImage* 195 (2019), pp. 11–22.

[231] S. Ruder. “An Overview of Multi-Task Learning in Deep Neural Networks”. In: *arXiv preprint arXiv:1706.05098* (2017).

[232] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Representations by Back-propagating Errors”. In: *Nature* 323.6088 (1986), pp. 533–536.

[233] M. Rußwurm, M. Ali, X. X. Zhu, Y. Gal, and M. Körner. “Model and Data Uncertainty for Satellite Time Series Forecasting with Deep Recurrent Models”. In: *International Geoscience and Remote Sensing Symposium*. IEEE. 2020, pp. 7025–7028.

[234] C. Sakaridis, D. Dai, and L. Van Gool. “ACDC : the Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

[235] C. Sakaridis, D. Dai, and L. Van Gool. “Semantic Foggy Scene Understanding with Synthetic Data”. In: *International Journal of Computer Vision* 126 (2018), pp. 973–992.

[236] T. Salimans, D. Kingma, and M. Welling. “Markov Chain Monte Carlo and Variational Inference: Bridging the Gap”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1218–1226.

[237] S. Saxena, C. Herrmann, J. Hur, A. Kar, M. Norouzi, D. Sun, and D. J. Fleet. “The Surprising Effectiveness of Diffusion Models for Optical Flow and Monocular Depth Estimation”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[238] S. Saxena, A. Kar, M. Norouzi, and D. J. Fleet. “Monocular Depth Estimation Using Diffusion Models”. In: *arXiv preprint arXiv:2302.14816* (2023).

[239] P. Seeböck, J. I. Orlando, T. Schlegl, S. M. Waldstein, H. Bogunović, S. Klimscha, G. Langs, and U. Schmidt-Erfurth. “Exploiting Epistemic Uncertainty of Anatomy Segmentation for Anomaly Detection in Retinal OCT”. In: *IEEE Transactions on Medical Imaging* 39.1 (2019), pp. 87–98.

[240] M. Sensoy, L. Kaplan, and M. Kandemir. “Evidential Deep Learning to Quantify Classification Uncertainty”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[241] A. C. Serban, E. Poll, and J. Visser. “Adversarial Examples-a Complete Characterisation of the Phenomenon”. In: *arXiv preprint arXiv:1810.01185* (2018).

[242] R. Sharma, M. Saqib, C.-T. Lin, and M. Blumenstein. “A Survey on Object Instance Segmentation”. In: *SN Computer Science* 3.6 (2022), p. 499.

[243] Y. Shen, Z. Zhang, M. R. Sabuncu, and L. Sun. “Real-Time Uncertainty Estimation in Computer Vision via Uncertainty-Aware Distribution Distillation”. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. 2021, pp. 707–716.

[244] A. Shrivastava, A. Gupta, and R. Girshick. “Training Region-based Object Detectors with Online Hard Example Mining”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 761–769.

[245] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. “Indoor Segmentation and Support Inference From RGB-D Images”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2012, pp. 746–760.

[246] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-scale Image Recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[247] I. J. A. Simpson, S. Vicente, and N. D. F. Campbell. “Learning Structured Gaussians to Approximate Deep Ensembles”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 366–374.

[248] V. Sindhwani, T. Sainath, and S. Kumar. “Structured Transforms for Small-footprint Deep Learning”. In: *Advances in Neural Information Processing Systems* 28 (2015).

[249] L. Smith and Y. Gal. “Understanding Measures of Uncertainty for Adversarial Example Detection”. In: *arXiv preprint arXiv:1803.08533* (2018).

[250] S. Song, S. P. Lichtenberg, and J. Xiao. “SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 567–576.

[251] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks From Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[252] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese. “Which Tasks Should Be Learned Together in Multi-task Learning?” In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9120–9132.

[253] C. Steger, M. Ulrich, and C. Wiedemann. *Machine Vision Algorithms and Applications*. John Wiley & Sons, 2018.

[254] R. S. Stone, N. Ravikumar, A. J. Bulpitt, and D. C. Hogg. “Epistemic Uncertainty-weighted Loss for Visual Bias Mitigation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2898–2905.

[255] L. Sun, J. Gou, B. Yu, L. Du, and D. Tao. “Collaborative Teacher-student Learning via Multiple Knowledge Transfer”. In: *arXiv preprint arXiv:2101.08471* (2021).

[256] M. Sundermeyer, T. Hodaň, Y. Labbe, G. Wang, E. Brachmann, B. Drost, C. Rother, and J. Matas. “Bop Challenge 2022 on Detection, Segmentation and Pose Estimation of Specific Rigid Objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2785–2794.

[257] I. Sutskever, O. Vinyals, and Q. V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *Advances in Neural Information Processing Systems* 27 (2014).

[258] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. “Going Deeper with Convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

[259] K.-H. Thung and C.-Y. Wee. “A Brief Review on Multi-task Learning”. In: *Multimedia Tools and Applications* 77.22 (2018), pp. 29705–29725.

[260] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. “Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation”. In: *Advances in Neural Information Processing Systems* 27 (2014).

[261] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield. “6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark”. In: *International Conference on Intelligent Robots and Systems (IROS)*. 2022.

[262] M. Ulrich and M. Hillemann. “Uncertainty-aware Hand–eye Calibration”. In: *IEEE Transactions on Robotics* (2023).

[263] M. Valdenegro-Toro. “Sub-Ensembles for Fast Uncertainty Estimation in Neural Networks”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4119–4127.

[264] J. Van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal. “On Feature Collapse and Deep Kernel Learning for Single Forward Pass Uncertainty”. In: *arXiv preprint arXiv:2102.11409* (2021).

[265] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. “Uncertainty Estimation Using a Single Deep Deterministic Neural Network”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9690–9700.

[266] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool. “Multi-task Learning for Dense Prediction Tasks: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2021), pp. 3614–3633.

[267] S. Vandenhende, S. Georgoulis, and L. Van Gool. “MTI-Net: Multi-scale Task Interaction Networks for Multi-task Learning”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 527–543.

[268] A. Vaswani. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems* (2017).

[269] S. Wan, T.-Y. Wu, W. H. Wong, and C.-Y. Lee. “ConfNet: Predict with Confidence”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2018, pp. 2921–2925.

[270] C. Wang, C. Wang, W. Li, and H. Wang. “A Brief Survey on RGB-D Semantic Segmentation Using Deep Learning”. In: *Displays* 70 (2021), p. 102080.

[271] L. Wang, X. Zhang, H. Su, and J. Zhu. “A Comprehensive Survey of Continual Learning: Theory, Method and Application”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[272] M. Wang and W. Deng. “Deep Visual Domain Adaptation: A Survey”. In: *Neurocomputing* 312 (2018), pp. 135–153.

[273] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. “Towards Unified Depth and Semantic Prediction From a Single Image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2800–2809.

[274] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. “DUSt3R: Geometric 3d Vision Made Easy”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 20697–20709.

[275] Y. Wang, H. Wang, Y. Shen, J. Fei, W. Li, G. Jin, L. Wu, R. Zhao, and X. Le. “Semi-Supervised Semantic Segmentation Using Unreliable Pseudo-Labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4248–4257.

[276] F. Wenzel, J. Snoek, D. Tran, and R. Jenatton. “Hyperparameter Ensembles for Robustness and Uncertainty Quantification”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6514–6527.

[277] A. G. Wilson and P. Izmailov. “Bayesian Deep Learning and a Probabilistic Perspective of Generalization”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4697–4708.

[278] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. “Deep Kernel Learning”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2016, pp. 370–378.

[279] C. Wolf, M. Karl, and P. van der Smagt. “Variational Inference with Hamiltonian Monte Carlo”. In: *arXiv preprint arXiv:1609.08203* (2016).

[280] D. Wu, L. Gao, M. Chinazzi, X. Xiong, A. Vespignani, Y.-A. Ma, and R. Yu. “Quantifying Uncertainty in Deep Spatiotemporal Forecasting”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 1841–1851.

[281] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. “Quantized Convolutional Neural Networks for Mobile Devices”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4820–4828.

[282] K. Wursthorn, M. Hillemann, and M. Ulrich. “Comparison of Uncertainty Quantification Methods for CNN -based Regression”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLIII-B2-2022 (2022), pp. 721–728.

[283] J. Xiang, Y. Wang, L. An, H. Liu, Z. Wang, and J. Liu. “Visual Attention-based Self-supervised Absolute Depth Estimation Using Geometric Priors in Autonomous Driving”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11998–12005.

[284] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.

[285] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes, Q. Morris, Y. Barash, A. R. Krainer, N. Jojic, S. W. Scherer, B. Blencowe, and B. J. Frey. “The Human Splicing Code Reveals New Insights Into the Genetic Determinants of Disease”. In: *Science* 347.6218 (2015), p. 1254806.

[286] D. Xu, W. Ouyang, X. Wang, and N. Sebe. “PAD-Net: Multi-tasks Guided Prediction-and-distillation Network for Simultaneous Depth Estimation and Scene Parsing”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 675–684.

[287] X. Xu, H. Zhao, V. Vineet, S.-N. Lim, and A. Torralba. “MTFormer: Multi-task Learning via Transformer and Cross-task Reasoning”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 304–321.

[288] F. Xue, G. Zhuo, Z. Huang, W. Fu, Z. Wu, and M. H. Ang. “Toward Hierarchical Self-supervised Monocular Absolute Depth Estimation for Autonomous Driving Applications”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2020, pp. 2330–2337.

[289] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci. “Transformer-based Attention Networks for Continuous Pixel-wise Prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16269–16279.

[290] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. “Depth Anything: Unleashing the Power of Large-scale Unlabeled Data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 10371–10381.

[291] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. “Depth Anything V2”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 21875–21911.

[292] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen. “Metric3D: Towards Zero-shot Metric 3d Prediction From a Single Image”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 9043–9053.

[293] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen. “Learning to Recover 3d Scene Shape From a Single Image”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 204–213.

[294] X. Yu, G. Franchi, and E. Aldea. “SLURP: Side Learning Uncertainty for Regression Problems”. In: *Procedings of the British Machine Vision Conference*. 2021.

[295] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan. “Neural Window Fully-connected Crfs for Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3916–3925.

[296] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. “CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6023–6032.

[297] M. D. Zeiler and R. Fergus. “Visualizing and Understanding Convolutional Networks”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014, pp. 818–833.

[298] J. Zeng, A. Lesnikowski, and J. M. Alvarez. “The Relevance of Bayesian Layer Positioning to Model Uncertainty in Deep Bayesian Active Learning”. In: *arXiv preprint arXiv:1811.12535* (2018).

[299] R. Zhang, L. Du, Q. Xiao, and J. Liu. “Comparison of Backbones for Semantic Segmentation Network”. In: *Journal of Physics: Conference Series*. Vol. 1544. 1. IOP Publishing. 2020, p. 012196.

[300] Y. Zhang and Q. Yang. “A Survey on Multi-task Learning”. In: *IEEE transactions on Knowledge and Data Engineering* 34.12 (2021), pp. 5586–5609.

[301] Y. Zhang and Q. Yang. “An Overview of Multi-task Learning”. In: *National Science Review* 5.1 (2018), pp. 30–43.

[302] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian. “Monocular Depth Estimation Based on Deep Learning: An Overview”. In: *Science China Technological Sciences* 63.9 (2020), pp. 1612–1627.

[303] C. Zhao, Y. Zhang, M. Poggi, F. Tosi, X. Guo, Z. Zhu, G. Huang, Y. Tang, and S. Mattoccia. “MonoViT: Self-supervised Monocular Depth Estimation with a Vision Transformer”. In: *International Conference on 3d Vision*. IEEE. 2022, pp. 668–678.

List of Figures

List of Tables

List of Abbreviations

AUC Area Under the Curve

CE Cross-Entropy

CNN Convolutional Neural Network

DE Deep Ensemble

DL Deep Learning

DNN Deep Neural Network

DSE Deep Sub-Ensemble

DUDES **D**eep **U**ncertainty **D**istillation using **E**nsembles for **S**egmentation

ECE Expected Calibration Error

EMUFormer **E**fficient **M**ulti-task **U**ncertainty Vision Trans**former**

FL Focal Loss

GNLL Gaussian Negative Log-Likelihood

ID in-domain

IoU Intersection over Union

LC Learned Confidence

MCD Monte Carlo Dropout

MDE Monocular Depth Estimation

mIoU mean Intersection over Union

MSE Mean Squared Error

mUnc mean class-wise Predictive Uncertainty

OOD out-of-domain

ReLU Rectified Linear Unit

RMSE Root Mean Squared Error

RMSLE Root Mean Squared Logarithmic Error

RN101 ResNet-101

RN18 ResNet-18

SS Semantic Segmentation

TTA Test-Time Augmentation

U-CE	**U**ncertainty-aware **C**ross-**E**ntropy
UQ	Uncertainty Quantification
ViT	Vision Transformer

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.