Monocular Depth Estimation for 3D Scene Completion in Autonomous Racing

Wei Xun Lai

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-109
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-109.html

May 16, 2025

Acknowledgement

I acknowledge Edward Tregor Neo Lee for his direct contributions towards this work, as well as Kaushik Kunal Singh for his outstanding leadership of the AI Racing Tech (ART) Perception Team, where this research unfolds. I am grateful to work alongside the on-track ART crew, co-led formerly by Siddarth Saha and Haoru Xue, now by Moises Lopez. I thank my first ART project mentor, Aman Saraf, who illuminated this research path; Caitlin K. Wolfe for her unwavering support throughout this journey; and the rest of the ART Perception team for their dedication, technical expertise, and camaraderie. Honorable mentions go to Ashwin Dara and Hao Kun for their brief but valuable participation in this research, and to the friends and family who were always there by my side.

Monocular Depth Estimation for 3D Scene Completion in Autonomous Racing

by Wei Xun Lai

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor S. Shankar Sastry
Research Advisor

5-16-2025

(Date)

* * * * * * *

Professor Allen Y. Yang
Second Reader

5-16-2025

(Date)

Monocular Depth Estimation for 3D Scene Completion in Autonomous Racing

by

Wei Xun Lai

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor S. Shankar Sastry, Chair
Professor Allen Y. Yang, Co-chair

Spring 2025

Monocular Depth Estimation for 3D Scene Completion in Autonomous Racing

Abstract

Monocular Depth Estimation for 3D Scene Completion in Autonomous Racing

by

Wei Xun Lai

Master of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor S. Shankar Sastry, Chair

Professor Allen Y. Yang, Co-chair

This thesis presents an approach to 3D scene completion for high-speed autonomous racing through monocular depth estimation. The research addresses a 22-degree LiDAR blindspot in the Indy Autonomous Challenge (IAC) racing platform, where vehicles operate at speeds exceeding 180 mph (290 km/h). At such speeds, comprehensive 360 degree environmental perception is paramount for safety and competitive performance, yet, due to engineering challenges, sensor configurations present significant gaps in coverage. To bridge this perceptual gap, we develop a transformer-based monocular depth estimation pipeline capable of cross-view generalization from front-facing to rear-facing perspectives without direct rear-view training data. Our approach extends reliable depth estimation from the conventional 40-60 meter range to 100 meters—providing an 80% increase in reaction time for tactical decision-making at racing speeds. Extensive evaluation on the Las Vegas Road Course demonstrates robust cross-track generalization with a 28% reduction in error variance compared to baseline approaches. The implementation is capable of achieving 42Hz inference throughput on the compute-constrained racing platform through TensorRT optimization and FP16 quantization. This work establishes monocular depth estimation as a viable complement to existing perception systems in autonomous racing, addressing sensor blindspots while maintaining the computational efficiency essential for real-time operation in competitive racing environments.

Dedicated to the timeless journey of engineering that flows through generations

Contents

Acknowledgments

I extend my sincere thanks to Professor Allen Yang and AI Racing Tech Team Principal Gary Passon for their generous support and the opportunity to pursue this work, and express my heartfelt gratitude to Professor Shankar Sastry for his invaluable guidance and wisdom.

I acknowledge Edward Tregor Neo Lee for his direct contributions towards this work, as well as Kaushik Kunal Singh for his outstanding leadership of the AI Racing Tech (ART) Perception Team, where this research unfolds. I am grateful to work alongside the on-track ART crew, co-led formerly by Siddarth Saha and Haoru Xue, now by Moises Lopez. I thank my first ART project mentor, Aman Saraf, who illuminated this research path; Caitlin K. Wolfe for her unwavering support throughout this journey; and the rest of the ART Perception team for their dedication, technical expertise, and camaraderie. Honorable mentions go to Ashwin Dara and Hao Kun for their brief but valuable participation in this research, and to the friends and family who were always there by my side.

This journey would not have been possible without the collective efforts, knowledge sharing, and passion of everyone involved. Their contributions have not only shaped this research but have also profoundly influenced my academic and professional development.

Chapter 1

Introduction

1.1 Indy Autonomous Challenge

Competition Overview

As part of the AI Racing Tech (ART) consortium, our team develop a comprehensive software stack for the Dallara AV-24, a purpose-built autonomous racecar based on the Dallara IL-15 chassis and engineered for operation at speeds exceeding 180 mph (290 km/h). The primary objective of this research initiative is to create a fully autonomous racing system capable of executing complex maneuvers, implementing strategic overtaking, and maintaining vehicle stability at these speeds during competitive racing scenarios. This thesis focuses specifically

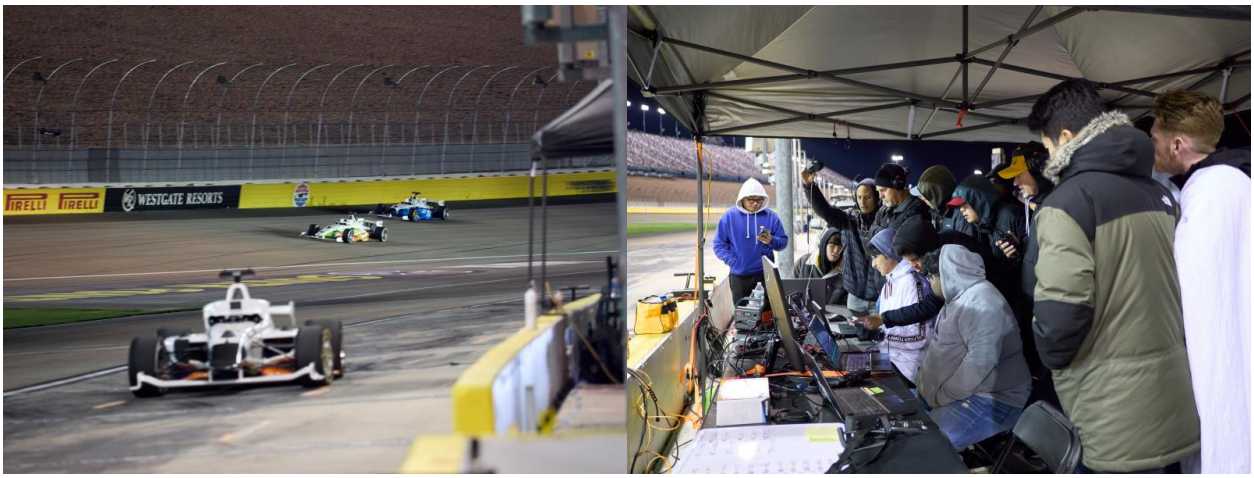


Figure 1.1: Left: Autonomous Dallara AV-24 vehicles competing at high speeds during an IAC racing event, demonstrating the extreme conditions under which our perception systems operate. Right: Team members of AI Racing Tech (ART) in the basestation monitoring real-time telemetry data from the racing vehicle, illustrating the collaborative human-AI partnership essential for autonomous racing competition.

on the unique perception challenges encountered in such extreme environments, detailing our approach to perception, and ultimately, a monocular depth estimation and evaluating its efficacy in the detection of opponent vehicles. In this racing context, "opponent cars" or "opponent vehicles" refer to other autonomous racing vehicles that share the same track with the ego vehicle, operating either as "defenders" positioned ahead of the ego car or "attackers" trailing behind. These opponent vehicles operate independently, without collaboration with the ego car, and do not share live vehicle data during competition—creating a true adversarial racing environment that closely mirrors traditional human motorsport competitions.

1.2 Problem Statement and Research Context

Monocular depth estimation has emerged as an important technology in autonomous driving systems, offering a cost-effective alternative to active sensing modalities. LiDAR and radar systems require specialized hardware with high power consumption and cost implications. On the other hand, camera sensors consume minimal power compared to active sensors. By utilizing well-established monocular depth estimation pipelines with cameras, these affordable sensors can extract three-dimensional environmental information and provide rich semantic information beyond pure geometric measurements. These advantages have driven significant research investment and widespread adoption of monocular depth estimation across industries such as autonomous vehicles, with manufacturers increasingly relying on vision-based perception to reduce system complexity and hardware costs while maintaining robust environmental understanding.

Autonomous racing at extreme speeds introduces unique perception challenges that push the boundaries of existing monocular depth estimation methods:

- **Extended Range Requirements:** While conventional autonomous driving typically operates with perception ranges of 40-60 meters, racing applications could demand reliable, real-time, depth estimation for distances exceeding 100 meters. At speeds of 60-80 meters per second (135-180 mph), this extended range becomes crucial for maintaining safe operation and competitive performance. Algorithms must achieve real-time performance (minimum 40Hz) while consuming minimal computational resources within this isolated computing environment.

- **Compute-Constrained Environment:** Racing platforms impose limitations on computational resources, with processing power shared across perception, control, planning, and vehicle dynamics systems. Unlike conventional autonomous driving systems that can leverage cloud computing or distributed processing, racing applications require all computation to be performed entirely onboard due to latency, reliability constraints, and competition fairness requirements.

- **Limited Training Data:** The specialized nature of racing environments constrains data acquisition, with track testing opportunities occurring infrequently and proper

sensor calibration requiring substantial setup time. Additionally, the significant expense of deploying racing vehicles for data collection sessions—including transport, crew, and operational costs—further limits data gathering opportunities. This scarcity of domain-specific training data necessitates robust zero-shot approaches.

Research Objective

Our research goal is to develop a monocular depth estimation system optimized for high-speed autonomous racing. This thesis establishes several specific research objectives to address the identified challenges:

1. **Development of a Cross-View Depth Estimation pipeline**:
   - Create a monocular depth estimation model capable of generalizing effectively from front/side camera perspectives to the rear-view perspective without direct rear-view training
   - Establish methodologies for evaluating the transferability of depth features across significantly different viewpoints
   - Monocular Dense Depth Model Checkpoint: A trained depth estimation pipeline optimized for autonomous racing environments, capable of cross-view generalization and extended range estimation.
2. **Extension of Effective Depth Range**:
   - Investigate techniques to extend reliable monocular depth estimation beyond the conventional 40-60m range for regular autonomous driving to at least 100 meters
   - Analyze the accuracy degradation curve as a function of distance to establish confidence thresholds for safety-critical applications
3. **Optimization for Computational Efficiency**:
   - Develop a lightweight implementation capable of operating within the strict computational constraints of the racing platform
   - Achieve real-time performance (minimum 20Hz) on the NVIDIA A5000 GPU while maintaining necessary accuracy thresholds
4. **Creation of a Specialized Racing Dataset**:
   - Leading student engineers to efficiently leverage limited track testing opportunities to collect training datasets with accurate ground truths of at least 100 meters
   - Develop techniques for generating synthetic training data to supplement limited real-world collection opportunities
   - Implement data augmentation strategies to simulate unseen racing environments

5. **Validation Framework for Zero-Shot Deployment**:

    - Design a robust validation methodology leveraging complementary sensor data, such as opponent GPS positioning data from transmitters on opposing vehicles or active radar sensors as ground truth for distance estimation accuracy

The contributions resulting from these research objectives advance monocular depth estimation for high-speed autonomous racing applications. This work delivers a method for training depth estimation models, demonstrating robust cross-view generalization and extending reliable depth estimation from the conventional 40-60 meter range to 100 meters while maintaining real-time performance on resource-constrained platforms.

Alongside the model, this work presents a specialized racing dataset that combines limited real-world track data with synthetic augmentations, accompanied by an optimized ROS2 inference pipeline achieving 40Hz performance through quantization and acceleration techniques. To report the model's robustness, a multi-modal evaluation framework that leverages GPS data validates depth estimation accuracy in sensor blindspot regions without requiring additional hardware or extensive track testing. Through these integrated contributions, this thesis bridges the gap in autonomous racing perception by enabling depth estimation from monocular cameras for 3D scene completion, establishing it as a viable complement to existing perception systems while maintaining the computational efficiency essential for competitive racing.

Chapter 2

The Autonomous Driving System in Indy Autonomous Challenge (IAC)

The Indy Autonomous Challenge (IAC) represents a collaborative engineering endeavor that brings together expertise from engineering, perception, strategic planning, controls and localization to push the boundaries of autonomous racing technology. The autonomous driving system is implemented based on a Dallara IL-15 racing vehicle. The overall autonomous driving hardware system, is the product of extensive collaboration between the IAC mechanical and electrical engineering teams, with significant input from academic research groups such as UC Berkeley on sensor placement optimization and configuration. This iterative design process allows for strategic hardware refinements, resulting in a comprehensive sensor suite that effectively balances coverage, redundancy, and computational constraints.

In accordance with autonomous vehicle design principles, the driver compartment has been reconfigured to house the computational and sensing infrastructure. The central processing unit consists of a dSPACE AUTERA AutoBox featuring a 12-core processor coupled with an NVIDIA A5000 GPU—an industrial-grade computing platform designed to withstand the extreme vibration and acceleration forces experienced during high-speed racing. Additional hardware components include an enterprise-class Cisco network switch for high-bandwidth sensor data routing, a sophisticated drive-by-wire system for actuation control, and an array of perception sensors. While the complete autonomous system encompasses perception, planning, control, and actuation subsystems, we would focus primarily on the perception architecture given its relevance to the relevant works in the upcoming chapters.

2.1 Hardware Architecture and Sensor Suite

The IAC establishes strategic partnerships with industry-leading sensor manufacturers, refreshing these collaborations periodically to incorporate technological advancements onto the Dallara Autonomous Vehicle (AV24, for the 2024 edition). The work presented in this thesis utilize sensory data components from these manufacturers including Luminar, Continental,

Delphi, Novatel, ZF, and Allied Vision, each contributing specialized sensing technology to the autonomous platform. Of particular significance to our perception system were the Luminar Hydra and Iris LiDAR units, Continental's high-resolution ARS548 radar array, and Allied Vision's Mako G-319 cameras, which provides the primary visual data for our monocular depth estimation algorithms.

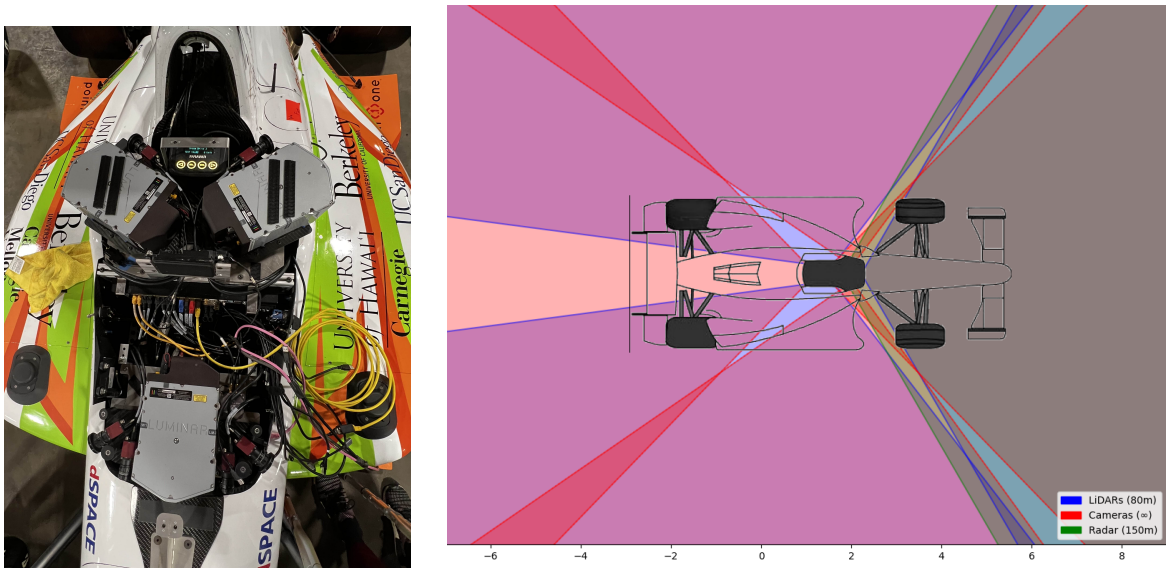


Figure 2.1: Sensor configuration of the Dallara AV21 autonomous racing platform. **Left:** Top-down layout showing the strategic placement and coverage areas of LiDAR, camera, and radar sensors to achieve 360° environmental perception. **Right:** All sensor field of view covering 360°, visualized.

Figure 2.1 illustrates the sensor configuration implemented on the AV24 platform, highlighting the strategic placement of LiDAR units, cameras, and radar sensors. This configuration is designed to replicate and enhance the situational awareness capabilities of professional racing drivers while providing redundant coverage through multiple sensing modalities for robust perception in high-speed racing environments. The spatial arrangement of these sensors is meticulously designed to optimize several factors:

1. **Field-of-View Coverage**: Ensuring comprehensive environmental perception with minimal sensing gaps
2. **Multi-Modal Redundancy**: Facilitating sensor fusion by providing overlapping coverage between different modalities
3. **Structural Rigidity**: Minimizing vibration-induced noise by securing sensors to rigid mounting points
4. **Thermal Management**: Ensuring reliable operation under extreme conditions

The implemented configuration features three Luminar LiDAR units, each providing a 120° field of view, strategically positioned to deliver full 360° environmental coverage with minimal gaps, apart from a 20° blind spot in the rear. The camera system comprises four wide-angled pinhole cameras for peripheral awareness and two forward-facing cameras mounted with precise inter-ocular distance to enable stereo depth calculation when required. Additionally, a forward-facing radar unit provides complementary long-range detection capabilities. This multi-modal approach creates redundancy for safe operation at racing speeds exceeding 180 mph (290 km/h).

2.2 AV24 Sensor Modalities: Capabilities and Limitations

Each sensing modality deployed on the autonomous racing platform offers distinct advantages and presents unique challenges that influenced our perception system architecture. Table 2.1 provides a comprehensive comparison of the technical specifications for each sensor type, including detection range, angular coverage, measurement tolerance, and spatial resolution.

	Range(m)	**Azimuth(°)**	**Elevation(°)**	**Tolerance(m)**	**Resolution**
Radars					
Delphi ESR	1.0 - 175	± 10	± 0	±0.5	Low
ARS 548	0.2 - 301	± 60	± 14	±0.15	Low
ZF-AC1000T	0.2 - 200	± 35	± 0	N/A	Low
Lidars					
Hydra	2 - 250	± 60	± 15	0.01	Medium
Iris	2 - 250	± 60	± 13	0.01	Medium
Camera					
Mako G319	Theoretically Unbounded	± 55*	± 38*	N/A	High

Table 2.1: Technical Specifications of the Multi-Modal Sensor Suite Deployed on the IAC Autonomous Racing Platform. Data are derived from manufacturer-provided technical datasheets available in the public domain, with values marked with an asterisk (*) representing estimated parameters based on calibration / practical testing in racing conditions.

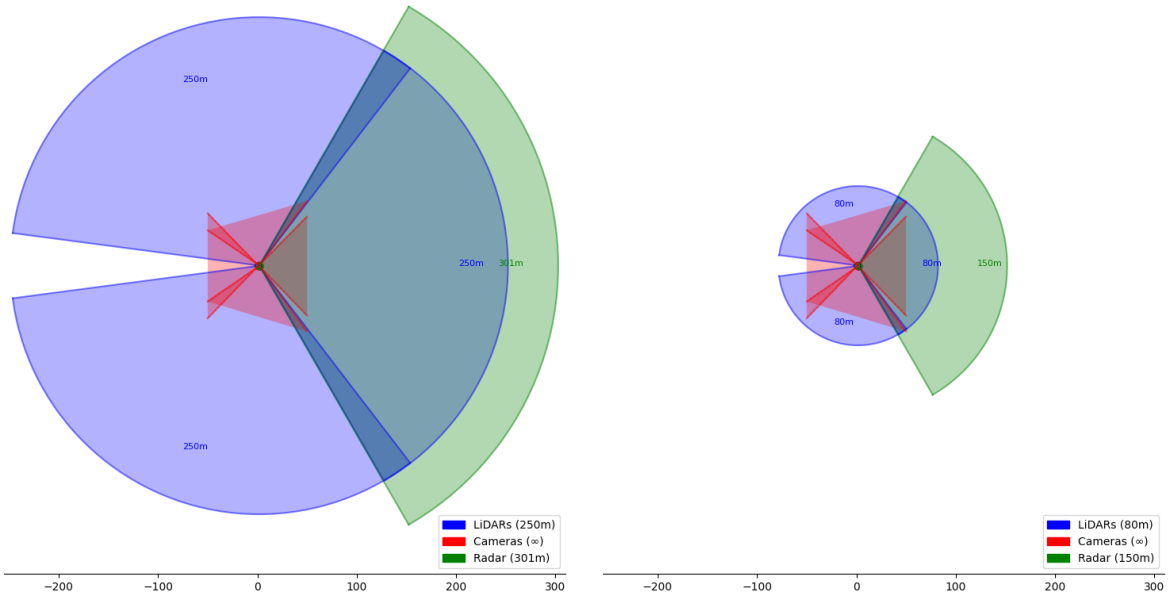


Figure 2.2: Comparison of theoretical and practical sensing ranges we utilize for the AV24 multi-modal sensor suite. **Left:** Theoretical detection ranges as specified in manufacturer datasheets, showing LiDAR (green, 250m), radar (blue, up to 300m), and camera (red, unlimited). **Right:** Practical operational ranges for our software stack, illustrating a reduction in effective LiDAR range (80m) and radar reliability (150m) due to point cloud sparsity, signal degradation, and target object characteristics. Camera range remains theoretically unlimited but depends on object size, contrast, and environmental conditions.

LiDAR Sensor

LiDAR sensors represent the primary source of high-fidelity environmental data, providing accurate three-dimensional point clouds that serve as ground truth for many perception algorithms. The Luminar Hydra (AV21) and Iris units deployed on the AV24 platform, deliver exceptional precision with measurement tolerances rated ± 1 cm throughout their operational range. This accuracy makes LiDAR data invaluable for training and validating camera-based depth estimation algorithms.

However, LiDAR technology exhibits diminishing point cloud density as a function of distance, creating challenges for long-range perception. Through empirical testing, we determine that beyond approximately 80-100 meters, the point density decreases to a level where reliable detection of a standard Dallara AV vehicle (with dimensions of 5m $\times$ 2m $\times$ 1.5m) becomes problematic for conventional lidar clustering algorithms. At this threshold distance, our Luminar units operating at 20 Hz with concentrated elevation scanning patterns ($\pm$3-5°) typically generate only 20-30 points on a target vehicle—sufficient for detection but approaching the lower limits of reliable tracking and classification.

Radar Systems

Radar technology is capable of complementing LiDAR by extending the perception horizon significantly, with demonstrated detection capabilities exceeding 80 meters for moving vehicles. From our experiments, we find that the Continental ARS548 unit is capable of providing long-range awareness up to 200m, visualized in Figure 2.2 for strategic decision-making at IAC racing speeds as the vehicles cover substantial distances in fractions of a second.

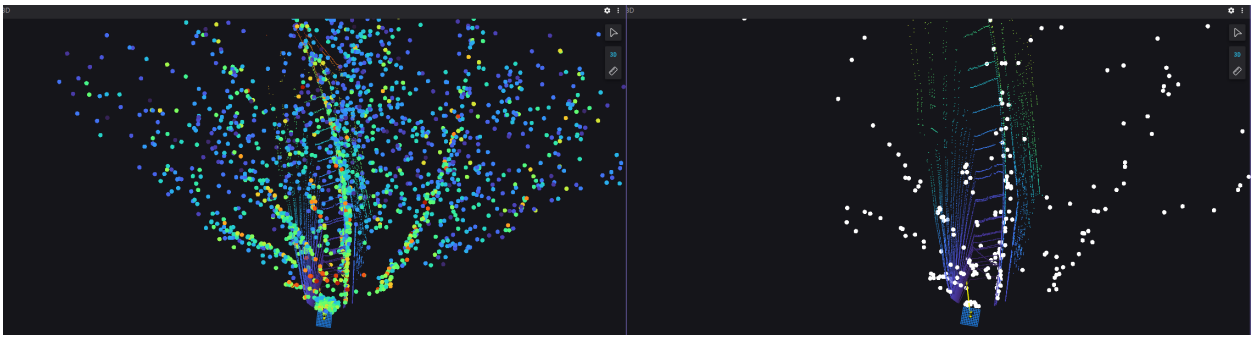


Figure 2.3: **Left**: Raw radar returns (colored by signal-to-noise ratio) overlaid on LiDAR point cloud data, demonstrating the limited number of true positive detections. The track walls appear as regions of higher point density, while spurious reflections create false positives throughout the scene. **Right**: SNR-filtered radar detections showing the persistence of multipath reflections from track walls

Despite these advantages, radar systems present several challenges that limit their effectiveness as a primary perception source. Operating at approximately 15 Hz, radar sensors have lower temporal resolution than other modalities. Additionally, the propagation characteristics of radio waves lead to multipath reflections from track surfaces and barriers shown in Figure 2.3, introducing false positive detections that must be filtered algorithmically. The spatial resolution of radar is also substantially lower than LiDAR or camera systems, with typical detections of a racing vehicle at 20 meters consisting of only 1-3 distinct reflection points. Each of these points would also have higher rated errors as shown in Table 2.1.

A phenomenon of particular concern in racing applications is the generation of anomalous speed readings caused by the highly reflective rotating wheel spokes of racing vehicles. These components can produce radar returns suggesting speeds significantly higher than the vehicle's actual speed, creating potential detection challenges. Consequently, our perception architecture relies primarily on LiDAR and camera data for safety-critical maneuvers within a 40-meter radius, reserving radar data for long-range awareness and speed estimation.

Camera Systems

The Allied Vision Mako G-319 cameras represent the most semantically-rich perception modality in our sensor suite, providing high-resolution visual information that captures texture, color, and structural features unavailable to other sensors. These cameras offer the-

oretically unlimited detection range for sufficiently large or distinctive objects, constrained only by optical factors such as lens quality, atmospheric conditions, and image resolution.

Camera data serves as the primary input for our monocular depth estimation research, leveraging recent advances in computer vision and neural network architectures to extract three-dimensional information from two-dimensional images. This approach provides a computationally efficient alternative to more resource-intensive sensor fusion techniques, particularly valuable given the processing constraints of our onboard computing platform.

Figure 2.4: Composite view of the four primary camera perspectives from the Dallara AV-24 autonomous racing vehicle. Images are arranged clockwise starting from top-left: left-side camera view showing adjacent track area; front camera view capturing the forward racing line; right-side camera view monitoring peripheral activity; and rear camera view displaying following vehicles.

2.3 Real-time Computational Constraints

The extreme operating environment of high-speed autonomous racing in the IAC imposes stringent computational limitations that significantly influence algorithm design and implementation strategies. To address these challenges, our system implements optimized software frameworks tailored to the hardware coupled with efficient data processing practices and algorithms on the racing platform.

Software Architecture and Middleware

Our autonomous racing platform implements the Robot Operating System 2 (ROS2) middleware framework, selected for its robust message-passing architecture, standardized communication protocols, and modular design principles. ROS2 provides significant advantages over conventional monolithic software architectures, including:

- Standardized interfaces facilitate the integration of heterogeneous software components
- Isolating faults within individual modules to enhance system resilience
- Supporting concurrent development modularity by multiple research teams
- Providing comprehensive monitoring and debugging capabilities via open-source tools

- Enabling zero-copy image processing through shared memory and intra-process communication.

Within this framework, dedicated sensor driver nodes we implement in C++ interface directly with hardware-specific APIs, publishing standardized message types to the ROS2 communication infrastructure. Subscriber nodes, including our perception algorithms, consume these messages and perform specialized processing tasks such as object detection, tracking, and depth estimation. This loose coupling between system components enhances maintainability and facilitates the experimental deployment of novel algorithms without compromising core functionality.

To optimize performance-critical perception pipelines, we leverage ROS2's Composable Nodes architecture, which allows multiple nodes to run within a single process. This approach eliminates the inter-process communication overhead traditionally associated with distributed systems. For image processing specifically, we implement zero-copy communication using shared memory and intra-process messaging. By maintaining perception data in shared memory regions accessible by multiple nodes, we avoid costly serialization, copying, and deserialization operations typically required when transferring high-bandwidth visual data between conventional ROS2 nodes.

Hardware Constraints and Optimization Strategies

The DSpace AUTERA AutoBox represents the primary computational platform for the autonomous racing system, featuring 12 physical CPU cores (expandable to 24 virtual cores through Intel Xeon hyperthreading) and 32GB of system memory. While substantial for embedded applications, these resources must simultaneously support the entire autonomous stack including perception, localization, planning, control, and system monitoring processes.

These hardware limitations necessitate careful optimization of our perception algorithms, particularly for computationally intensive tasks such as monocular depth estimation. Our implementation strategy prioritizes:

1. **Algorithmic Efficiency**: Selecting approaches that balance accuracy with computational complexity

2. **Real-time Operating Requirements**: At racing speeds approaching 180 mph (80 m/s), the perception system must maintain sub-50ms or 20Hz end-to-end latency from sensor acquisition to control output to ensure stable vehicle dynamics. For perception systems, models must be fully optimized to reduce throughput and reduce latency.

3. **Selective Processing**: With multiple high-bandwidth sensors generating data concurrently, careful resource allocation is essential to prevent processing bottlenecks. The Mako G-319 cameras alone produce approximately 1.2 GB/minute of raw image data at racing frame rates. We apply full-resolution analysis only to regions of interest.

4. **Memory Bandwidth Constraints**: Data transfer between ROS2 Node processes, system memory and the NVIDIA A5000 GPU introduces latency that must be minimized through efficient buffer management and in-place processing where possible.

These constraints necessitates several optimization strategies in our monocular depth estimation implementation. Rather than applying computationally intensive algorithms to full-resolution images, we employ a selective processing of regions of interest identified through lightweight segmentation techniques. Our neural network architectures were specifically designed for efficient inference on the NVIDIA A5000 GPU, utilizing TensorRT optimization and INT8/FP16 quantization to maximize throughput while maintaining acceptable accuracy.

We strictly manage memory, using zero-copy coding practices wherever possible to reduce system bus contention. This approach minimizes redundant data transfers between CPU and GPU memory spaces, significantly reducing processing latency for image-based depth estimation. Additionally, we implement adaptive processing rates for different perception components based on their relative importance to immediate vehicle safety, allocating greater computational resources to near-field perception versus long-range awareness.

These computational constraints directly informed our research focus on lightweight monocular depth estimation techniques as an alternative to more resource-intensive stereo vision or sensor fusion approaches. By deriving depth information directly from monocular camera inputs, our system reduces both the sensor hardware requirements and the computational overhead associated with multi-sensor calibration and synchronization.

Chapter 3

Monocular Depth Estimation

3.1 Motivation

Sensor Coverage Blindspots: A Safety Vulnerability

Achieving comprehensive 360-degree sensor coverage with active sensors represents a significant financial and technical challenge for autonomous vehicle manufacturers. The high cost of individual LiDAR units—ranging from $10,000 to $100,000 per sensor, combined with their substantial power requirements and integration complexity, often necessitates compromises in coverage design. While some high-end autonomous platforms deploy multiple overlapping LiDAR systems to eliminate blindspots, the cost implications of such redundancy can be prohibitive, particularly for competitive racing applications where weight and aerodynamic considerations further constrain sensor placement.

In the sensor configuration of the IAC AV-24 racing platform, these practical constraints manifest as a significant perception challenge due to the non-overlapping fields of view between the left and right Luminar Iris LiDAR units. This configuration creates a substantial 22-degree blindspot directly behind the vehicle, as illustrated in Figure 3.1. This blindspot represents a safety vulnerability, particularly in racing scenarios where competitors may rapidly approach from the rear during or after overtaking maneuvers.

At racing speeds exceeding 160 mph (72 m/s), a vehicle in this blindspot can close the gap from a safe following distance to a collision scenario in less than one second. Conventional perception pipelines based on LiDAR-camera fusion, including our previous work utilizing YOLO-based detection with depth projection, cannot address this fundamental limitation as they rely on LiDAR data that is simply unavailable in this region. The implementation of a monocular depth estimation approach offers a promising solution to this blindspot problem by leveraging the rear-facing camera to provide depth information without corresponding LiDAR points. However, this approach presents unique challenges in cross-view generalization, as the model must effectively transfer knowledge from primarily front-facing perspective, where training data is semantically similar and available, to the rear-facing view without direct supervision.

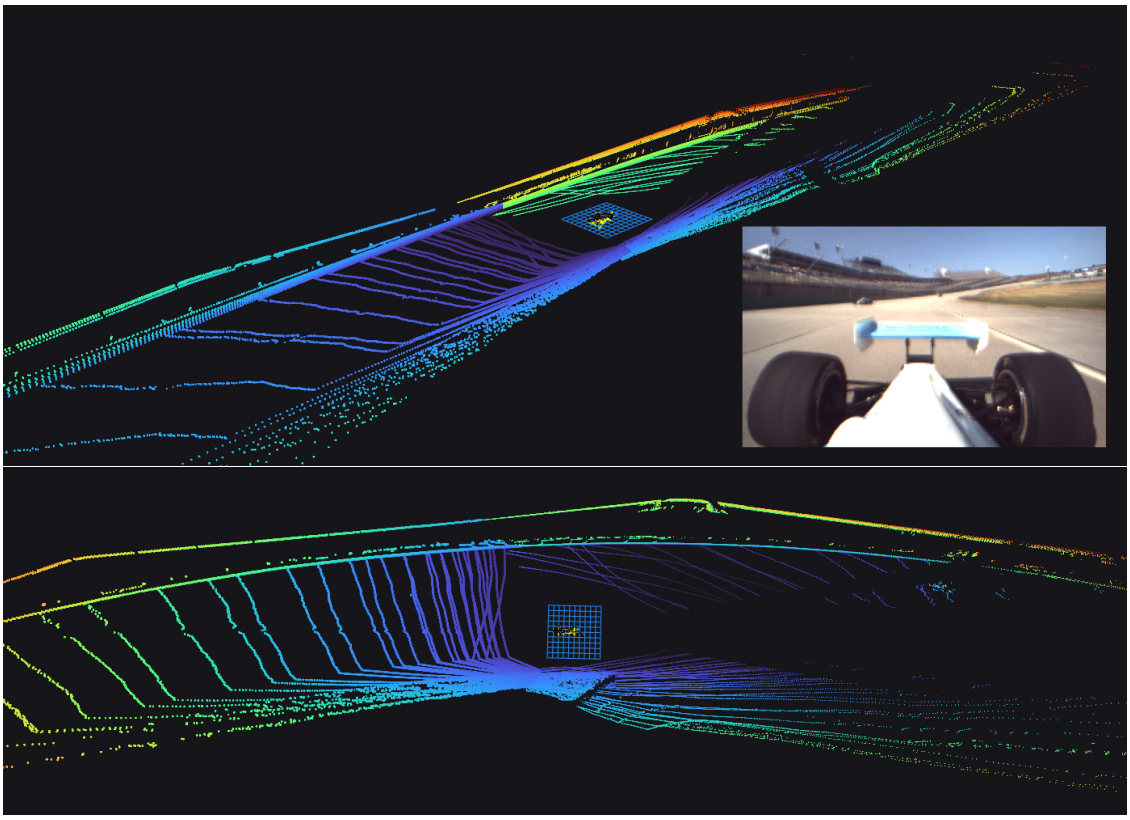


Figure 3.1: Visualization of the 22-degree LiDAR blindspot in the AV-24 racing platform. **Top:** Elevated 45-degree perspective showing the rearward view from the vehicle with an opponent car visible in the camera feed (bottom-right inset) but absent from the LiDAR point cloud. **Bottom:** Bird's-eye view (BEV) representation of the scenario

Calibration Challenges in High-Performance Racing

The maintenance of precise sensor calibration represents a significant operational challenge in racing environments. Based on our prior research experience, the extrinsic calibration process for each camera-LiDAR pair requires approximately 20 minutes with a two-person crew. To achieve comprehensive 360-degree environmental perception, eight separate calibrations must be performed, resulting in a three-hour procedure that is impractical to execute during limited track testing windows. Racing conditions introduce additional complications through extreme vibrations and mechanical stresses that progressively degrade calibration accuracy. Ideally, recalibration would be performed prior to each competitive session involving multiple vehicles to ensure maximum perception reliability. However, this approach is operationally unfeasible given the limited trackside access time and the rapid turnaround requirements between racing sessions.

3.2 Related Works

Monocular Image Depth Estimation

The field of monocular depth estimation has evolved from early learning-based approaches to large foundation models. Eigen et al. [13] pioneered learning-based methods using multiscale convolutional neural networks with scale-invariant loss functions. The field has since diverged into regression-based approaches such as [13, 27, 39] that predict continuous depth values and classification-based methods [17, 30] that estimate discrete depth bins. Recent works have combined these approaches, with AdaBins and LocalBins [1, 15] implementing classification-regression frameworks that first compute depth bins and then perform pixel-wise classification. ZoeDepth [2] further refines this approach by using relative depth models as input and applying metric bin modules to generate final depth estimates. These methods have demonstrated improved performance on standard benchmarks like KITTI and BDD100K [20, 60].

Semi-Supervised and Foundation Models

Recent advances have leveraged large-scale semi-supervised training paradigms inspired by language models. Depth Anything and Depth Anything V2 [53, 54] employ pseudo-labeling on unlabeled datasets combined with extensive augmentations. DepthPro [4] estimates focal lengths on top of their relative depth estimation to produce metric depth estimations. MiDaS [3] introduced robust relative depth estimation through training on diverse datasets. These foundation models have shown significant improvements in generalization and robustness across different domains, though their application to extreme environments like high-speed racing remains largely unexplored.

Geometry-Based Depth Estimation

Self-supervised monocular depth estimation leverages geometric consistency between video frames. SfMLearner [70] established the joint training framework of depth nets and PoseNet [29] using photometric loss. Subsequent works have enhanced this approach through feature-level reconstruction losses [45,62]. However, these methods depend on geometric consistency, in which in racing situations where we would require high accuracy estimations of opponent vehicles, these methods fall out of favor. Some authors have also proposed the handling of dynamic objects [41, 69], and additional geometric constraints [33, 55, 57], while works like [46, 57] incorporate motion segmentation.

Depth Range Extensions

Most existing depth estimation datasets and models are limited to ranges suitable for conventional autonomous driving. KITTI [22] and BDD100K datasets typically constrain depth

estimation to 80 meters, which provides only a one-second reaction window at racing speeds. Recent works have attempted to extend effective depth ranges: HR-Depth [47] focuses on high-resolution depth estimation, while works like [40] explore normal-distance assisted approaches. However, extending reliable depth estimation beyond 100 meters while maintaining accuracy remains an open challenge, particularly for monocular methods.

3.3 Problem Setup & Challenges

Given that the on-track racing team is capable of performing accurate camera intrinsic calibrations quickly, the primary challenge in three-dimensional perception becomes estimating the scale of each point—the depth of each pixel, perpendicular to the optical axis. Monocular depth estimation presents a compelling alternative by eliminating the need for continuous cross-sensor calibration. By deriving depth information directly from the camera's frame of reference, this approach also significantly reduces operational complexity for downstream processing, such as 3D to 2D projection.

Challenges

The principal challenges with monocular estimation in our application include:

1. Zero-shot learning, as we must leverage data where LiDAR-camera overlap is sufficient (e.g., front, left, right views) but deploy in the rear blindspot.
2. The model must generalize to previously unseen race tracks.
3. Extended range accuracy, particularly up to 100 meters, the critical distance required for adequate reaction time at racing speeds exceeding 180 mph (290 km/h).

3.4 Dense Prediction Transformers (DPT) as a Zero-Shot Range Estimator

We employ Depth Prediction Transformers (DPT) [39] for monocular depth estimation as a fundamental architectural component to overcome the inherent limitations of traditional convolution-based networks (CNNs) in dense prediction tasks. The DPT architecture offers several significant advantages over CNNs, enabling superior performance in tasks requiring fine-grained, globally consistent depth predictions:

1. **Global Receptive Field:** Vision transformers maintain a global receptive field at every stage, which allows long-range dependencies and capturing global image context to resolve ambiguities in depth estimation. This contrasts CNNs, which progressively expand their receptive field through stacked layers.

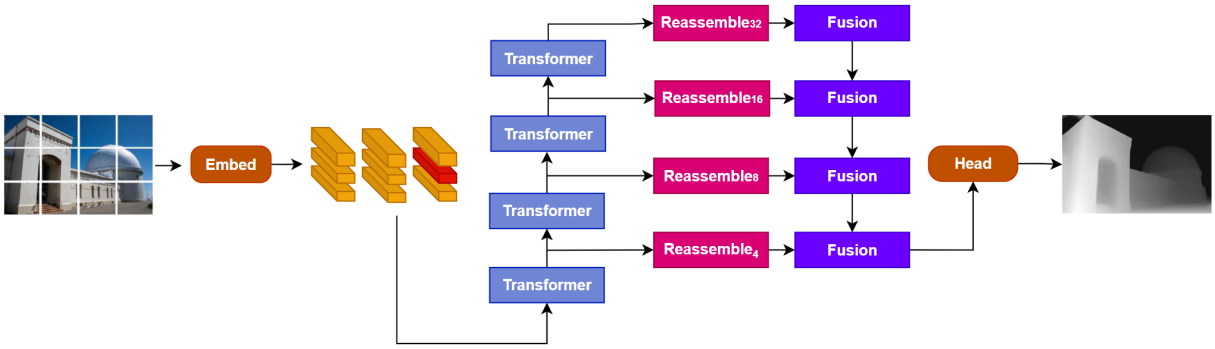


Figure 3.2: Architecture overview of the Dense Prediction Transformer. [39]. The input image is transformed into tokens (orange) using one of two methods: by extracting non-overlapping patches followed by a linear projection of their flattened representations (as in DPT-Base and DPT-Large), or by applying a ResNet-50 feature extractor (as in DPT-Hybrid). The resulting image embedding is then augmented with a positional embedding, and a patch-independent readout token (red) is added. These tokens are passed through multiple transformer stages. Tokens from different stages are reassembled into image-like representations at multiple resolutions (pink). Fusion modules (purple) progressively fuse and upsample these representations to produce a fine-grained prediction.

2. **Preservation of Fine-Grained Features:** CNNs rely on aggressive downsampling to reduce memory and computational requirements, often losing spatial granularity in the process. However, DPT operates at a constant resolution in the transformer encoder, preserving the fine-grained image features essential for dense prediction tasks.

3. **Strong Results Across Tasks:** DPT demonstrates state-of-the-art performance across dense prediction tasks, including monocular depth estimation and semantic segmentation. Experiments show a **28% improvement in relative performance** [53] compared to leading CNNs on large-scale depth datasets.

Based on these substantial advantages, we adopted the DPT architecture to achieve accurate, robust monocular depth predictions as a sensor completion method, supplementing the existing lidar pipeline.

Among the DPT models, we utilized the Depth Anything checkpoint [53]which incorporates a DINOV2 encoder [37] with the DPT decoder. Depth Anything model's comprehensive open-source support, including optimized TensorRT implementations, facilitated seamless integration with our perception stack. The Depth Anything DPT has also been trained on an extensive dataset comprising over 62 million images, ensuring strong generalization capabilities, and distilled to smaller versions of the architecture.

Depth Anything

The authors of *Depth Anything* [53] introduced a powerful framework, involving a dataset construction and augmentation process for the monocular, in-the-wild, relative depth estimation problem. Starting with a limited labeled dataset $\mathcal{D}^l = \{(x_i, d_i)\}_{i\in[M]}$ of high-quality images , the authors trained a teacher model T to generate pseudo-labels $\hat{d}_i = T(x_i, \mathcal{D}^l)$ for a second, diverse unlabeled dataset $\mathcal{D}^u$. These pseudo-labeled samples were then combined into $\hat{\mathcal{D}}^u = \{(x_i, \hat{d}_i)\}_{i\in[M']}$, improving the model's generalization [53]. To further enhance robustness, the authors created an augmented dataset $\tilde{\mathcal{D}}^u$ with techniques such as :

1. **Color Distortions:** Including color jittering and Gaussian blurring to simulate varying lighting conditions and sensor noise
2. **Spatial Transformations:** Implementing techniques such as CutMix [54] to enhance the model's invariance to occlusions and partial views

This comprehensive data augmentation strategy resulted in a training regime that systematically exposed the model to a wide spectrum of visual challenges, significantly improving its generalization capabilities. Following this data preparation process, a student model S was trained on the augmented dataset, strategically retaining the teacher's encoder weights while fine-tuning only the decoder components for relative depth estimation.

Metric Depth Estimation The authors of Depth Anything utilized Zoedepth [2], a post-processing pipeline that converts relative depth estimates to absolute metric values through adaptive bin calculation. In our initial investigations, we attempted to fine-tune their pre-trained large-scale metric depth model using conservative learning rates(1.0×10^{-8})–a process we term "model calibration," drawing inspiration from parameter-efficient fine-tuning techniques commonly employed in large language models (LLMs). However, our experiments revealed performance limitations in this approach when applied to our highly specialized racing dataset.

Instead, we develop a streamlined calibration methodology that directly transforms the relative depth outputs from the DPT model into accurate metric measurements using a computationally efficient linear transformation. This approach provides several crucial advantages over more complex post-processing pipelines: (1) significantly reduced inference latency critical for high-speed racing scenarios where reaction windows are measured in milliseconds; (2) lower computational overhead, enabling deployment on resource-constrained embedded platforms; and (3) robust performance across diverse racing environments through our distance-normalized calibration procedure.

Chapter 4

Implementation: Gathering Data, Training, and Deployment

We demonstrate that with appropriate domain-specific fine-tuning, the Dense Prediction Transformer (DPT) architecture can directly produce accurate metric depth estimations without requiring computationally expensive real-time deployment post-processing methods such as ZoeDepth.

Our implementation framework consists of four integrated components: (1) a Ground Truth Acquisition Pipeline for collecting and processing training data, (2) a Model Training Pipeline with specialized loss functions for transferrable camera intrinsics and viewpoints, (3) a comprehensive Dataset Preprocessing System addressing normalization, augmentation, and cross-dataset alignment challenges, and (4) a Real-time Inference Pipeline, integrated with YOLOv8 and optimized for deployment on autonomous racing platforms. This integrated approach enables high-performance depth estimation while maintaining computational efficiency.

4.1 Ground Truth Collection Pipeline

To fine-tune the model for the racecar, we collect race data from test runs on the track in addition to using the KITTI dataset [19] for pretraining. We then extract camera and LiDAR data from recorded ROS2 data bags of runs with opponent cars. We also utilize the LiDAR-camera calibration data for the car to transform the LiDAR data into the camera frame and project it into black and white images with the camera intrinsics. The white intensity of each pixel corresponds to the depth of the LiDAR points in the z direction. The pairs of the camera images and the LiDAR-projected images are then used to evaluate the model on real racing images and further finetuning with the LiDAR data as the ground truth along with the KITTI dataset (which is already in the same format).

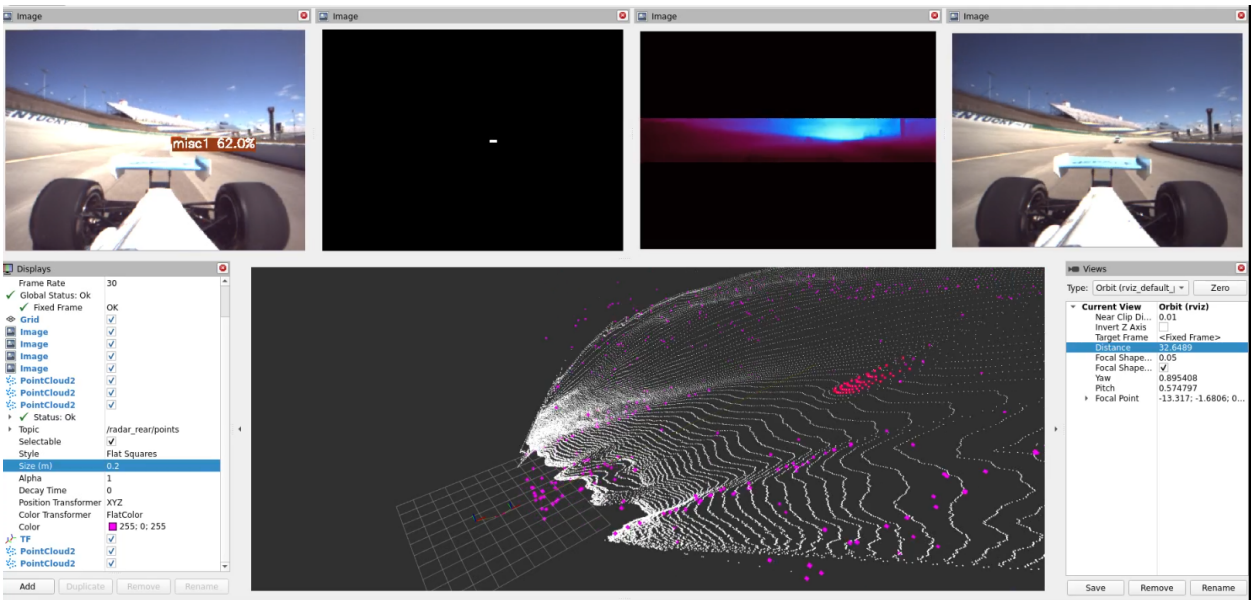


Figure 4.1: Real-time finalized Depth Estimation pipeline deployed on rear cameras at Kentucky Speedway, visualized in RVIZ2. Visualization showing: (**top row, left to right**) rear camera feed with YOLO object detection overlays, YOLOv8 segmentation binary mask, depth estimation heatmap, and original rear camera image. The bottom 3D visualization displays raw, uncalibrated dense white point clouds that we generate using our depth estimation model (trained exclusively on Indianapolis Motor Speedway data), magenta points representing radar detections, and fluorescent magenta/red points showing the final perception pipeline output combining YOLOv8 detection with depth estimation. The alignment of the camera and radar points demonstrate successful cross-track generalization from training on IMS to deployment on Kentucky Speedway.

4.2 Model Training Pipeline

We employ a composite loss function that integrates multiple optimization objectives to address the specific challenges of depth estimation in high-speed racing environments. The loss function incorporates both global accuracy metrics and specialized components that prioritize precise depth estimation for safety-critical objects such as opponent vehicles.

The foundation of our training approach combines the Mean Absolute Error (MAE) and Scale-Invariant Logarithm Loss (SIlog) [13], with additional regularization on regions of interest extracted from segmentation models, such as opponent cars on-track. The SILog is an important component of our loss, enabling high-accuracy cross-view adaptations. Formally,

the Scale-Invariant Logarithmic Error is defined as shown:

$$\text{Scale Invariant Log Loss} = \frac{1}{N}\sum_{i=1}^{N}\left(\log(\hat{d}_i) - \log(d_i)\right)^2 - \frac{1}{N^2}\left(\sum_{i=1}^{N}\left(\log(\hat{d}_i) - \log(d_i)\right)\right)^2$$

This formulation comprises two principal components: the first term quantifies the variance of logarithmic errors, while the second term compensates for global scale inconsistencies. This mathematical structure enables the model to focus on preserving relative depth relationships rather than absolute values, which is essential when generalizing across diverse racing environments with varying lighting conditions and track geometries.

Building upon this foundation, our loss function incorporates additional regularization focused on regions of interest identified through semantic segmentation. The complete formulation is expressed as:

$$\begin{aligned}\mathcal{L}(d, \hat{d}) = &\frac{\alpha_1}{n}||d - \hat{d}||_1 \\ &+ \frac{\alpha_2}{n}||\ln \hat{d} - \ln d||_2^2 - \frac{\alpha_2\lambda}{n^2}||\mathbf{1}^T(\ln \hat{d} - \ln d)||_2^2 \\ &+ \alpha_3\mathcal{L}(d_{\text{seg}}, \hat{d}_{\text{seg}})\end{aligned}$$

Where:

- $d, \hat{d} \in \mathbb{R}^n$ represent the ground truth and predicted depth values, respectively. The depth is defined to be the distance along the optical axis to the point.
- $\alpha_1, \alpha_2, \alpha_3$ are hyperparameters that control the relative contribution of each loss component.
- λ is a balancing coefficient for the scale-invariant term
- $\mathbf{1}^T$ denotes a vector of ones, implementing the summation operation
- $d_{\text{seg}}, \hat{d}_{\text{seg}}$ represent depth values specifically within segmented regions of interest

The third component, $\mathcal{L}(d_{\text{seg}}, \hat{d}_{\text{seg}})$, applies heightened optimization pressure on areas identified as opponent vehicles through our segmentation pipeline. This targeted regularization improves depth estimation accuracy for safety-critical objects. We optimize this composite loss function using stochastic gradient descent with momentum, implementing an adaptive learning rate schedule that begins with an initial rate of 2.8×10^{-7} and decays by a factor of 0.1 every epoch.

Hyperparameter Tuning With tuning, we find $\alpha_1 = 0.25, \alpha_2 = 0.75, \alpha_3 = 0.3$ gives us the lowest absolute error on important regions of our validation dataset. For each hyperparameter configuration, we evaluated the models using:

$$\text{Safety Penalty} = \sum_{b \in \text{bins}} w_b \cdot \text{AbsRel}(\theta, b)$$

Where θ represents the hyperparameters, w_b is the weight for distance bin b, and $\text{AbsRel}(\theta, b)$ is the absolute relative error in that bin. Our method for tuning is as follows: For the first stage, we evaluate combinations of α_1 and α_2 (the MAE and SIlog weights) in increments of 0.05, where $\alpha_1 \in [0, 1]$ and $\alpha_2 = 1 - \alpha_1$. After establishing optimal values for α_1 and α_2, we conduct an extensive sweep of the segmentation weight parameter α_3. This tuning technique aims to first have the model attain the best absolute and relative loss ratio that would result in the best estimation of safety-critical objects, and then find the optimal weight that would emphasize the penalties on those regions.

4.3 Training Dataset Preprocessing

To maximize our limited dataset's richness, model performance and generalization capabilities, we implement several preprocessing techniques addressing normalization, augmentation, cross-dataset alignment challenges, and finding the most suitable Region of Interest for the problem.

Camera Region of Interest Selection

A consideration in our preprocessing pipeline involves determining the optimal camera field of view (FOV) for depth estimation. Our extensive experimental analysis, detailed in Appendix A, demonstrates that strategically cropping the input frames yields substantial improvements in depth estimation consistency and accuracy on a pretrained relative depth foundation model, namely the Depth Anything dense prediction transformer.

To test out different crops, namely the original image, partial crop, and full crop, detailed in Figure A.3, we obtain the depth maps to explore whether it is possible to find a generalizable transformation to convert relative depth to metrically accurate depth maps. We do so first by looking for simple linear transformation parameters to match the depth maps with respective metric depth slices of a lidar scan taken during the same moment. With regards to the pixel intensity values of the depth maps $\text{depth}_{\text{relative}}$, the transformation can be described with a scale factor s and bias offset b as follows:

$$\text{depth}_{\text{metric}} = s \cdot \text{depth}_{\text{relative}} + b \tag{4.1}$$

Through Through our analysis of the pinhole camera model across 71 frames spanning a complete racing lap as detailed in table 4.1, we identify that the car's nose and sky regions

Crop	Mean s	Std s	CV (%)	Mean b	Std b	CV (%)
None	-477.55	115.4	25.17	68.59	12.05	17.57
Partial	-308.81	92.25	29.87	75.56	8.37	11.08
Full	-123.65	19.13	15.47	78.75	8.63	10.96

Table 4.1: Transformation Parameters for different cropping approaches

introduce significant depth estimation inconsistencies. Fully documented in Appendix A, the original image depth estimation results in transformation parameter coefficients of variation (CV) of 25.17% for scale factors and 17.57% for bias offsets—indicating substantial frame-to-frame instability. By implementing an aggressive cropping strategy that completely removes the car's nose from the frame while retaining the track and racing environment, we reduce the CV to 15.47% for scale factors and 10.96% for bias offsets. This improvement in scale factor and bias offset stability translates directly to more reliable depth estimations across varying conditions.

Normalization

A fundamental preprocessing step in our methodology involves the normalization of depth values derived from LiDAR point clouds. The input data is initially stored in a projected format corresponding to the camera's field of view (FOV), ensuring spatial alignment between depth measurements and image pixels. Then, for each ground truth point cloud referenced in Section 4.1, we perform a two-stage normalization process:

1. **Range Clipping**: Point cloud depth values are constrained to the practical operating range of 0.1m to 120m. This boundary establishment serves multiple purposes:
    - Eliminating physiologically implausible near-field values that may result from sensor noise
    - Establishing a maximum effective range aligned with the racing environment's perceptual requirements
    - Reducing the dynamic range that must be learned by the model, thereby improving training stability
2. **Inverse Mapping Transformation**: Following range clipping, we apply an inverse mapping function where:
    - Distances of 120m are transformed to 0.0
    - Distances of 0.1m are transformed to 1.0
    - Intermediate values are linearly interpolated between these extremes

This transformation encodes a crucial semantic constraint wherein a depth value of 0 serves as a specialized token representing infinity or regions with invalid measurements. This encoding approach substantially enhances the model's capacity to differentiate between actual depth measurements and unmeasurable regions such as the sky, distant background, or areas outside the sensor's effective range.

Data Augmentation

To enhance the model's robustness and generalization capabilities across varied racing conditions, we implement image augmentations on our training dataset. These augmentations, visualized in 4.2, are to simulate the range of cross-track variations, including differences in track surface coloration, background infrastructure (grandstands, barriers, etc.) and ambient outdoor environmental conditions.

Original	
Augmented	

Table 4.2: Comparison between original and augmented training images from the Indianapolis Motor Speedway dataset.

Photometrically, we implement random brightness adjustments (± 0.2), contrast variations (0.8-1.2), hue modifications (± 0.05), and saturation alterations (0.8-1.2) to simulate variable lighting conditions encountered across different tracks and weather conditions. Geometrically, we apply horizontal flips with corresponding camera parameter adjustments.

Cross-Dataset Alignment

We pretrain our model with a Kitti dataset we align. When deploying depth estimation models across different data distributions involves managing the variation in camera intrinsics between training and deployment settings. This is particularly relevant when combining the KITTI dataset with our ROAR racing dataset, as these datasets are captured using cameras with different intrinsic parameters.

The depth estimation problem is fundamentally linked to focal length. Specifically, for a given pixel size δ, variations in δ should not affect metric depth information when properly accounted for in the model. More importantly, images captured with different focal lengths but proportionally scaled distances appear visually identical. For instance, if $\hat{f}_1 = 2\hat{f}_2$ and correspondingly $d_1 = 2d_2$, the resulting images will be indistinguishable despite the actual depth doubling, because the focal length scales by the same factor. Building on this insight, we align the KITTI and ROAR datasets by applying focal length normalization. This technique involves:

1. Extracting camera intrinsics from both datasets
2. Computing the focal length ratio between source dataset and target dataset domains
3. Applying appropriate scaling transforms to depth ground truth values

This preprocessing approach ensures that our model learns approximately consistent depth estimation patterns regardless of the source dataset's camera parameters, enabling effective knowledge transfer between domains.

Segmentation of Dynamic Objects

The segmentation of dynamic objects is used in our training objective as binary masks indicating importance regions for the model. To find the best zero-shot instance segmentation model that works well with our training datasets, we conduct a thorough sweep of existing state-of-the-art segmentation models and checkpoints, detailed in Appendix B. The summarized findings in Table 4.3 demonstrate that transformer-based architectures show exceptional generalization capabilities, particularly in the task of segmentation in unseen environments, such as our racing dataset. Among the models we evaluate, Mask2Former [8] and SegFormer [52] consistently outperforms traditional CNN-based approaches. All models in our evaluation is done on models pre-trained on the Cityscapes dataset [10, 11], achieves remarkable zero-shot segmentation results on our racing dataset, significantly surpassing conventional CNN architectures such as MobileNetV3 [24] and ResNet [22].

Mask2Former emerges as the most effective architecture for our application, with its ability to maintain precise instance boundaries and semantic consistency across diverse racing environments. However, implementing this model presents technical challenges due to dimensional constraints—while Mask2Former is optimized for a native resolution of 756 $\times$

	Kentucky Speedway	Indianapolis Motor Speedway
Original		
MobileNet		
ResNet		
Mask2Former		

Table 4.3: This table shows the performances of zero shot both CNN-based Segmentation models vs Transformer based segmentation models. The labels, colored, blue is "car", light pink is "road".

756, our racing platform requires an input resolution of 156 × 1008 to balance computational efficiency with real-time performance requirements. To address this, we implement a straightforward yet effective solution: padding the dataset images where they fall short and then running the model across the modified images. Notably, despite the introduction of non-semantic border regions created by padding, the model maintains exceptional segmentation quality, as evidenced in Table 4.3. The robust performance demonstrated by Mask2Former even under these suboptimal input conditions highlights its strong generalization capabilities.

For integration into our training pipeline, we develop a workflow. Segmentation masks are pre-computed for the entire dataset alongside the corresponding images and dynamically loaded during training. During model training, these segmentation masks enabled targeted loss weighting for crucial objects (For example, opponent vehicles labeled as "car", or "road"), allowing the depth estimation network to prioritize accuracy in high-importance regions. Similarly, during evaluation, these masks facilitated focused performance analysis on dynamic objects, providing insight into model reliability.

This segmentation-aware approach provides two key benefits: (1) it enhances depth estimation accuracy for dynamic objects—for collision avoidance in racing scenarios—and (2) it establishes a streamlined mechanism for evaluating perception performance specifically on these safety-critical elements rather than on less relevant background features.

4.4 Inference Pipeline

Our inference framework implements a hybrid architecture that integrates YOLOv8 [28] segmentation with our fine-tuned monocular depth estimation model. The pipeline operates through selective depth inference within semantically meaningful regions.

Real-Time Object Detection and Optimization Considerations

We deploy a YOLOv8 instance segmentation model to identify opponent vehicles, followed by targeted depth estimation within these regions. For each detected instance, statistical aggregation (primarily median filtering for robustness) derives a representative depth value. To meet the stringent requirements of the DSpace Autera AutoBox with NVIDIA A5000 GPU, we implement platform-specific optimizations using TensorRT with 16-bit quantization. This approach achieves 50Hz inference speeds—a latency sufficient for strategic decision-making.

Model Calibration process

The calibration process is essential for adapting our depth estimation model trained on frontal camera perspective to the rear camera perspective, while maintaining metric accuracy. This procedure establishes the relationship between the model's normalized depth predictions and absolute metric distances through a linear transformation.

For any given camera installation, we collect a calibration dataset consisting of paired samples $(d_i, \hat{d}_i)$, where d_i represents the ground truth depth value for sample i and $\hat{d}_i$ represents the corresponding model prediction after normalization. We collect these sample pairs specifically for crucial objects such as opponent vehicles. Ground truth depth values are obtained using GPS positioning data, track mapping from Structure from Motion pipelines such as COLMAP [44], or lidar euclidean clustering to establish an estimate of the spatial relationships between the ego vehicle and detected objects.

Distance-Balanced Calibration An important consideration in the calibration process is the inherent distance distribution imbalance in real-world datasets. In racing scenarios, nearby objects (around 40m) are typically overrepresented compared to distant objects, potentially biasing the calibration parameters toward short-range accuracy at the expense of long-range performance. To mitigate this imbalance, we implement a distance-normalized calibration approach. The distance normalization procedure operates as follows:

1. The complete range of operational depths (0.1, 120) meters is divided into discrete one meter sized bins $B = \{b_1, b_2, ..., b_k\}$
2. For each bin b_j, we collect all calibration pairs $(d_i, \hat{d}_i)$ where $d_i \in b_j$
3. If a bin contains more than three samples, we compute the representative value for that bin as:
$$d_{rep,j} = \text{median}(\{d_i | d_i \in b_j\})$$
$$\hat{d}_{rep,j} = \text{median}(\{\hat{d}_i | d_i \in b_j\})$$
4. The final calibration dataset consists of these representative bin values: $\{(d_{rep,j}, \hat{d}_{rep,j})\}_{j=1}^{k}$

This binning and aggregation strategy ensures that each distance range contributes equally to the calibration process, preventing the dominance of frequently observed distances. The use of median values within each bin further enhances robustness against outliers.

Calibration Parameter Estimation The calibration parameters α and β are determined by minimizing the following objective function using the distance-normalized dataset:

$$\min_{\alpha,\beta} \sum_{j=1}^{k} \|d_{rep,j} - (\alpha \cdot \hat{d}_{rep,j} + \beta)\|^2$$

This ordinary least squares (OLS) regression problem can be solved through various methods. In our implementation, we employ Singular Value Decomposition (SVD) for its numerical stability and computational efficiency. The closed-form solution using the SVD approach can be expressed as:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = (X^T X)^{-1} X^T Y$$

Where X is the design matrix containing the normalized depth predictions augmented with a column of ones and Y is the vector of ground truth depth values. For increased robustness, particularly when dealing with potential outliers in the calibration dataset, an iteratively reweighted least squares (IRLS) or RANSAC [16] variant that progressively down-weights the influence of samples with high residual errors:

$$\min_{\alpha,\beta} \sum_{j=1}^{k} w_j \cdot \|d_{rep,j} - (\alpha \cdot \hat{d}_{rep,j} + \beta)\|^2$$

Where weights w_j are iteratively updated based on residual errors using Huber or Tukey bisquare weighting functions. This approach proves to be particularly valuable in scenarios with occasional sensor fusion errors in the ground truth data.

Depth Normalization and Post-Processing

The model output undergoes an inverse transformation to convert normalized values back to metric depth measurements within the operational range of up to 120 meters. This transformation is mathematically represented as:

$$d_{metric} = \alpha \cdot (120.0 - d_{normalized}) + \beta$$

Where $d_{normalized}$ is constrained to the range $(0.1, 120)$, d_{metric} represents the final metric depth value in meters, and α and β are calibration parameters derived through the calibration process

This transformation effectively reverses the preprocessing normalization applied during training, where distances of 120m were mapped to 0, and distances of 0.1m were mapped to 1.0. The inverse mapping reestablishes the metric scale while the calibration parameters preserving the semantic relationship between pixel intensities and physical distances.

Temporal Aggregation

To enhance estimation stability, we implement temporal aggregation across sequential frames. This approach introduces a trade-off between latency and accuracy that must be carefully balanced in high-speed racing scenarios. Given that our camera feed is 20Hz and our pipeline is capable of 50Hz depth estimation inference outputs, our analysis indicates that temporal windows of t=3 or t=5 frames provide optimal performance, with the following latency characteristics:

- *t=3 frames*: Initial latency of 0.17s, followed by average processing latency of 0.095s
- *t=5 frames*: Initial latency of 0.27s, followed by average processing latency of 0.145s

Given our model's primary application for gap-closing during overtakes, the perception problem starts out with long-range perception (80+ meters). These latency values are an acceptable compromise for enhanced accuracy. We implement a median temporal aggregation for robustness.

Chapter 5

Experiments and Results

Our experimental evaluation focuses on assessing the efficacy of our monocular depth estimation approach across multiple dimensions: Bridging the gap: cross-view transfer learning capabilities, model training performance, model calibration effectiveness, cross-track generalization, and temporal aggregation benifits. These experiments provide a comprehensive analysis of the system's deployment readiness for high-speed autonomous racing scenarios.

5.1 Experimental Setup and Datasets

The primary foundation for our experiments is a Dense Prediction Transformer (DPT) model trained on the ROAR Indianapolis Motor Speedway (IMS) dataset after pre-training with an aligned KITTI dataset as described in Section 4.3. Model selection was performed using performance on a held-out ROAR Kentucky Speedway (KS) validation set.

For comprehensive evaluation, we employ the Las Vegas Road Course (LVRC) dataset Figure 5.2, which presents substantially greater challenges than the training environment. Unlike the relatively uniform oval track at IMS with consistent banking angles and predictable visual characteristics, LVRC features complex track variations including chicanes, U-turns, flat sections, and multiple elevation changes as shown in Figure 5.2. These features introduces significant perspective shifts during cornering maneuvers. On top of that, the dataset introduces complex background variations including infrastructure and terrain changes, including ill defined road lanes Figure 5.1. These factors collectively create a rigorous evaluation environment that presents a substantial domain shift from training data, serving as an ideal testbed for assessing generalization capabilities across dramatically different visual domains.

The size of this LVRC evaluation dataset contains 450 frontal opponent vehicle ground truths over distances up to 100 meters, and 1800 rear ground truths, where we intend to deploy the models. The ground truths are extracted using opposing vehicle GPS data, providing precise 4D (spatial and yaw) positioning information to evaluate the accuracy of the depth estimation across varying distances.

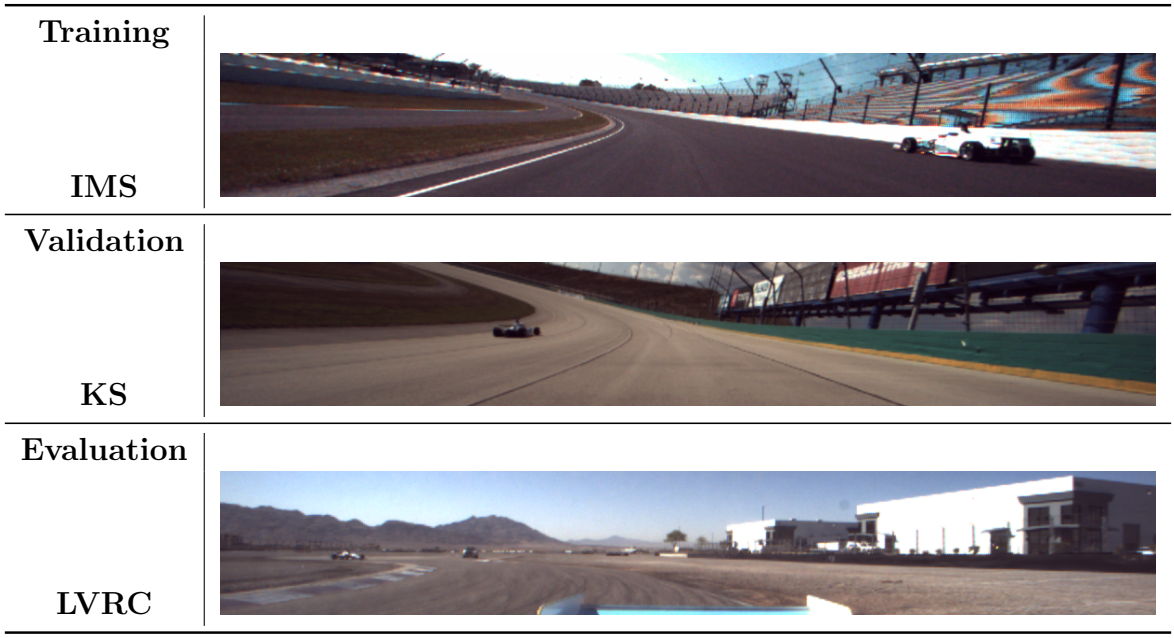


Figure 5.1: Dataset comparisons across different racing environments: Training on Indianapolis Motor Speedway (IMS Oval Track, top), validation on Kentucky Speedway (KS Oval Track, middle), and evaluation on Las Vegas Road Course (LVRC, bottom). Each dataset presents increasing visual complexity and domain shift challenges.

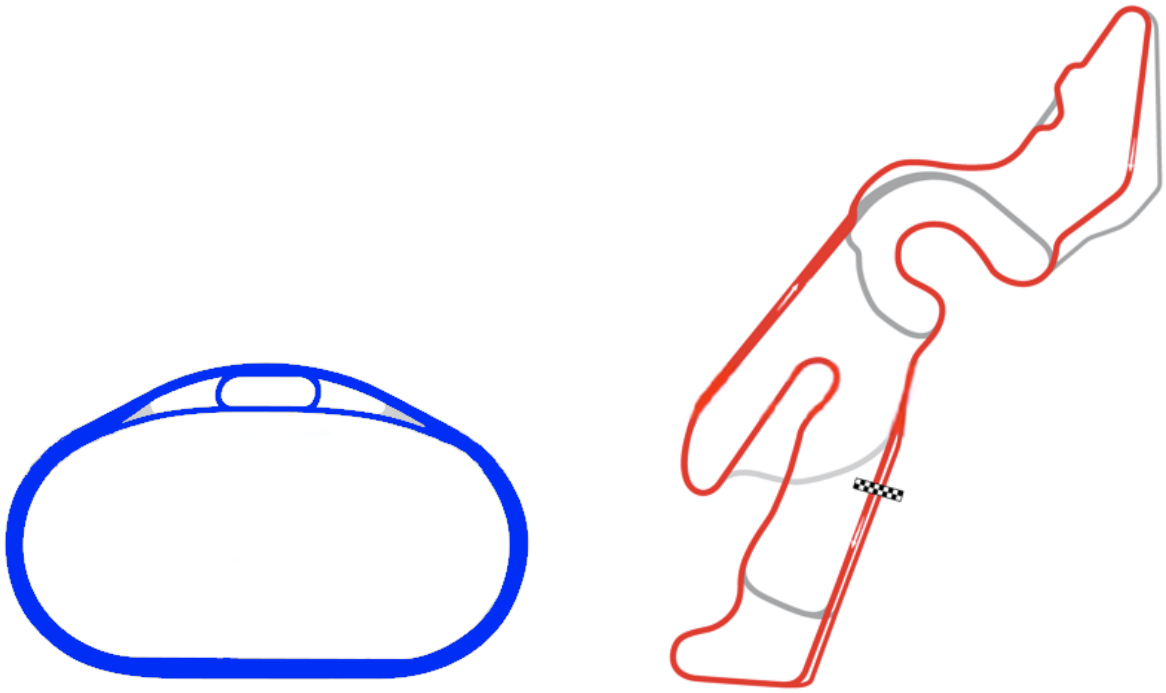
Baseline and Comparative Evaluation Framework

Our evaluation compares our proposed method against a baseline model that is trained using only Mean Absolute Error (MAE) loss without dynamic object segmentation, pretrained on the standard KITTI dataset without alignment techniques, and finetuned on the IMS training dataset. In contrast, our proposed pipeline incorporates the tuned objective (described in Section 4.2), pretraining on an aligned KITTI dataset, dynamic object segmentation-aware loss functions, and cross-view adaptation with calibration due to the scale invariant loss component used. We then evaluate the performance of both models before and after model calibration to demonstrate the effectiveness of our proposed cross-domain and cross-view adaptation techniques.

For quantitative assessment, we use the Relative Error Metric on selected dynamic objects, binned by distance in 10-meter intervals (e.g., [0, 10), [10, 20), etc.). For each depth measurement pair, we calculate the difference between the predicted and actual depths, then divide by the actual depth and average this metric over all pairs:

$$Relative\ Error(\%) = \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{d}_i - d_i}{d_i}$$

This metric choice provides an intuitive understanding of whether depths are being un-



(a) Las Vegas Superspeedway)

(b) Las Vegas Road Course (LVRC)

Figure 5.2: Las Vegas Motor Speedway configurations to scale: (a) oval track similar to Indianapolis Motor Speedway (IMS) and Kentucky Speedway (KS), used for IMS training and KS validation datasets; (b) complex road course used for evaluation, demonstrating the significant domain shift between training, validation, and evaluation environments.

derestimated or overestimated, allowing us to assess overall model safety and reliability in autonomous racing contexts.

5.2 Bridging the Gap: Cross-View Transfer Gap Analysis

One of the fundamental challenges mentioned in the problem statement is to achieve consistent depth estimation across camera viewpoints by only training on a single viewpoint. Our analysis quantifies the performance gap when transferring a model trained on frontal camera views to rear-facing perspectives without overly expensive adaptations.

In the baseline model trained with absolute objectives only, we observed a mean absolute relative error of 15.7% per opponent vehicle detection for frontal views on the LVRC evaluation dataset. However, this error increases significantly to 23% per detection when applied to rear-view data, representing a substantial degradation in estimation accuracy.

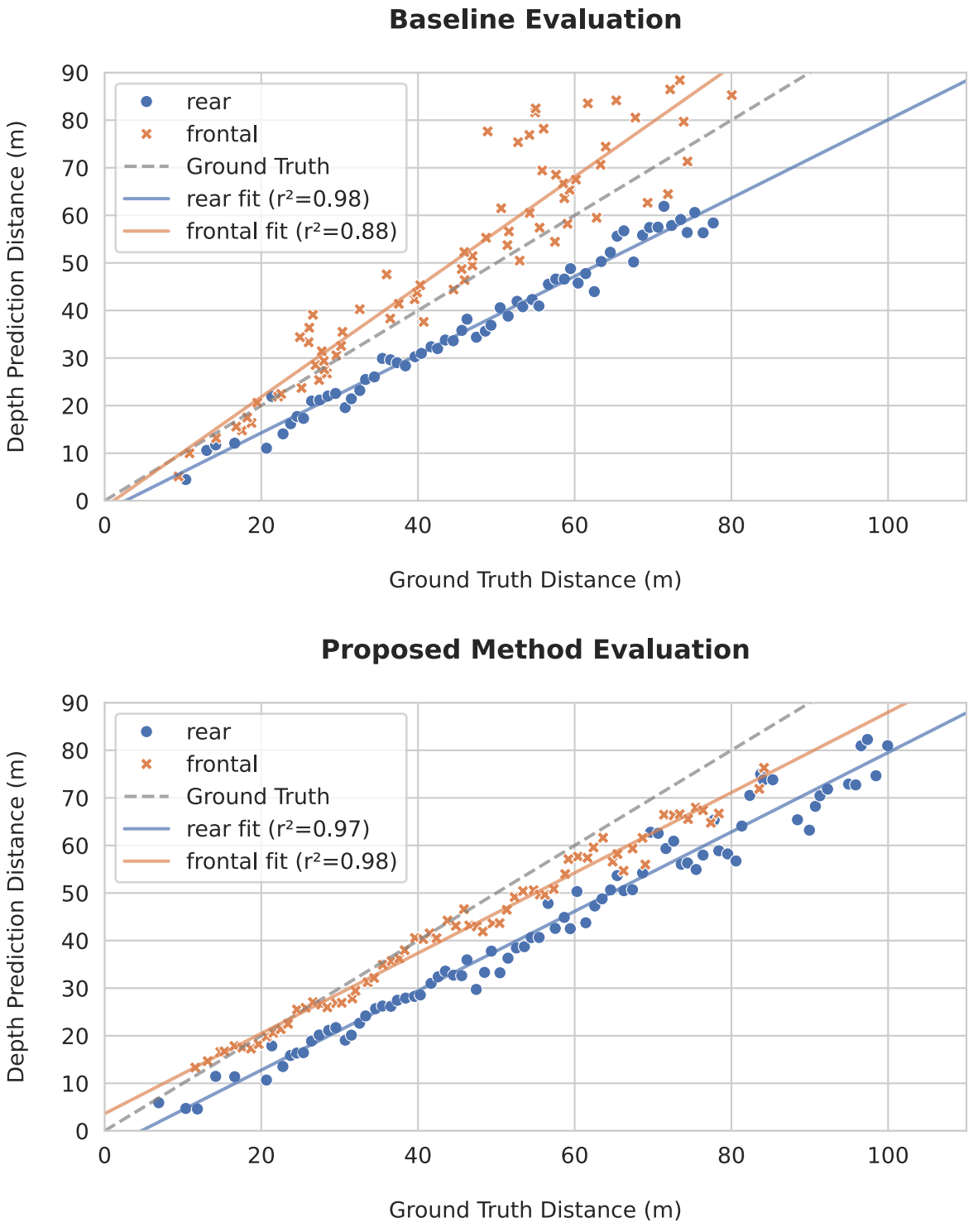


Figure 5.3: Comparison of per-meter-bin median values of GPS ground truth depth versus estimated depth evaluated on the LVRC dataset. **Top:** The Baseline method shows weaker correlation with ground truth and restricted effective range ($< 80m$). **Bottom:** Proposed method demonstrating stronger correlation with ground truth and extended detection capabilities up to 100m. Both methods exhibit systematic deviation in rear-view estimations, indicating the cross-view transfer challenge that necessitates calibration.

Our proposed model trained on the IMS Oval dataset demonstrates marked improvement with a mean absolute relative error of 8.6% per opponent vehicle detection for frontal views on the LVRC evaluation dataset. Despite this improvement, the error still increases significantly to 20% per detection when applied to rear-view data—representing a greater than 2× performance degradation. When normalized across meter-binned detections, the disparity becomes even more pronounced: 7% per aggregated bin for frontal views versus 26% for rear views.

This substantial performance differential confirms the existence of a non-trivial front-to-rear deployment gap. Figure 5.3 visually demonstrates this phenomenon through a comparative plot of ground truth versus estimated depth values across different camera perspectives. For frontal cameras, the median predictions (orange) closely approximate the ground truth values (blue), indicating strong generalization across tracks within the same camera perspective. Conversely, the results for both methods' rear camera estimate exhibits systematic deviation from ground truth, clearly demonstrating the need for targeted calibration techniques to bridge this cross-view adaptation gap. In the next section, we focus on the accuracy and stability of the post-calibration depth estimations.

Calibration Effectiveness Across Viewpoints

The application of our distance-balanced calibration methodology Section 4.4 demonstrates remarkable effectiveness in bridging the cross-view adaptation gap. Figure 5.4 presents a comprehensive error analysis through box plots comparing uncalibrated and calibrated performance across both front and rear camera perspectives. Several key insights emerge from this analysis:

1. **Frontal Camera Robustness**: The top row of Figure 5.4 shows a slight performance degradation when needlessly performing calibration on a model applied to tracks within the training camera perspective, as evidenced by increased variance after frontal view calibration. This showcases the model's robust cross-track performance for perspectives similar to training viewpoints. This robustness is particularly noteworthy given the substantial domain shift between the IMS oval and LVRC road course environments. The consistency of the model's performance post-calibration indicates the model has successfully maintained near-identity parameters when applied to new tracks while using the same camera perspective.

2. **Systematic Underestimation Bias**: Uncalibrated estimations across both perspectives exhibit a consistent negative error trend, indicating systematic distance underestimation. Apart from the camera perspective shift, one might argue that the DPT model's global context integration mechanism may associate visually complex backgrounds with closer depth planes—a potential limitation of global-image depth estimation when generalizing to visually diverse environments. However, we hypothesize that calibration parameters should remain consistent for a given model within each

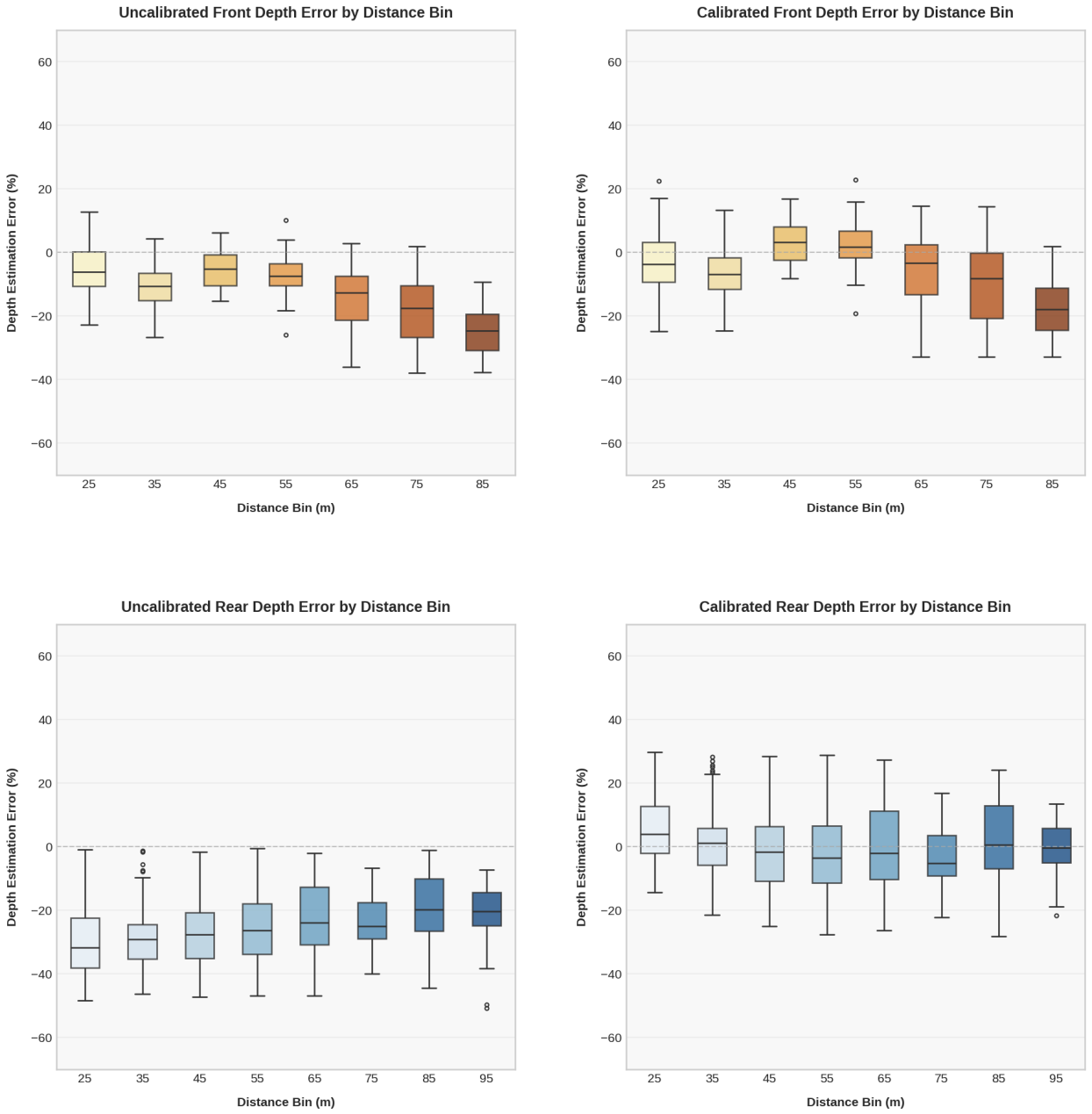


Figure 5.4: Comparison of depth estimation error distributions for the model trained on Indianapolis Motor Speedway Oval (IMS) and evaluated on the Las Vegas Road Course (LVRC) dataset, across different camera viewpoints with and without model-viewpoint calibrations applied. Error distribution comparison between uncalibrated (left) and calibrated (right) models for front (top) and rear (bottom) camera perspectives across different distance ranges. Note the dramatic error reduction in rear camera estimation following calibration.

perspective, as evidenced by the model's robust calibration parameters in the frontal view.

3. **Rear-View Correction**: The bottom row of Figure 5.4 demonstrates remarkable correction results in rear camera estimation following calibration. The calibration process substantially reduces the median error across all distance ranges, confirming the effectiveness of our distance-balanced calibration methodology.

The successful calibration of rear camera estimations using parameters provides strong evidence that our approach effectively captures and compensates for the fundamental geometric differences between camera perspectives, rather than merely overfitting to track-specific visual characteristics. These findings collectively validate that while transformer-based depth estimation models excel at capturing generalizable depth relationships, view-specific calibration remains essential for deploying these models across multiple camera perspectives.

5.3 Model Performance Results

This section examines the impact of our design choices on model performance, with particular focus on their effectiveness in extending the reliable depth estimation range. We analyze how our training objective formulation and segmentation-aware learning contribute to performance improvements across diverse camera perspectives and distance ranges.

Table 5.1 presents a visual comparison between our baseline and proposed methods across both front and rear camera perspectives. The box plots illustrate error distributions across different distance ranges, providing comprehensive visualization of estimation accuracy and consistency. Our analysis reveals that the baseline model training approach fails to transfer effectively across tracks, exhibiting large variance swings particularly for objects at distances exceeding 30 meters. In contrast, our proposed method demonstrates a 28% reduction in error variance across all distance ranges, indicating significantly improved estimation consistency. This improvement is evident in both front and rear camera perspectives, confirming the effectiveness of our cross-view adaptation approach. The performance enhancement can be attributed primarily to our modified depth estimation objective. By incorporating segmentation awareness, our model maintains focus on relevant objects regardless of distance, extending reliable estimation capability up to 100 meters, a suitable threshold for high-speed racing applications.

It is noteworthy that for near-field estimations (distances under 30 meters) in non-cross-view scenarios (frontal perspective), the baseline model outperforms our proposed method. This observation suggests potential benefits from implementing objective-switching training strategies that could optimize performance across different distance ranges by dynamically adjusting loss function weights.

Real-Time Performance and Deployment Considerations With software engineering efforts, this pipeline achieves excellent computational efficiency with an average inference

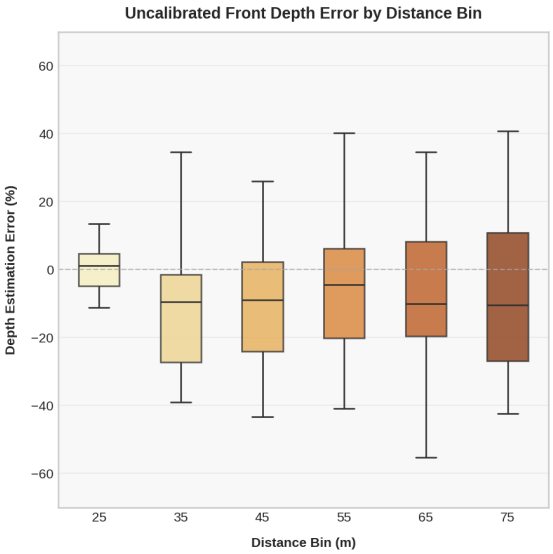
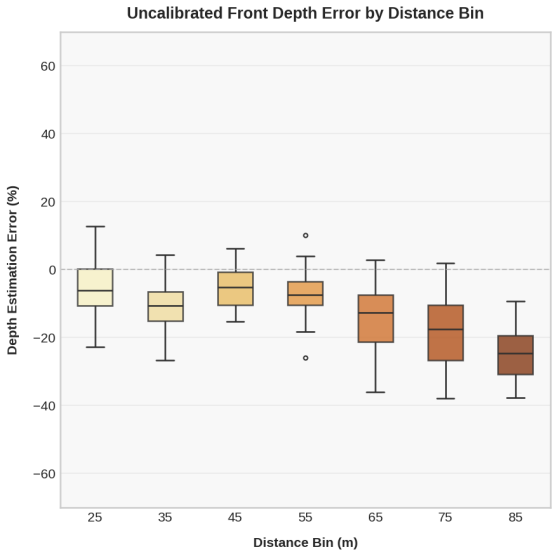
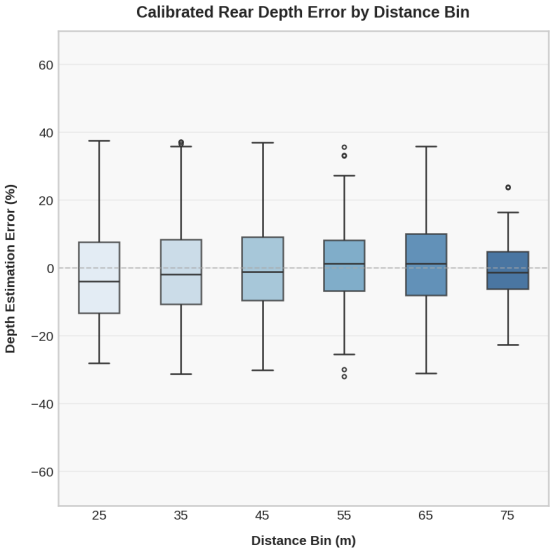
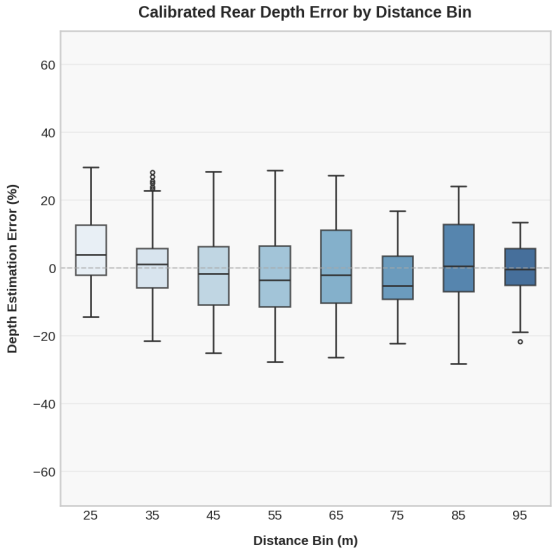
Baseline	Proposed Method
	
	

Table 5.1: Comparison of baseline and proposed method performance for front and rear camera views.

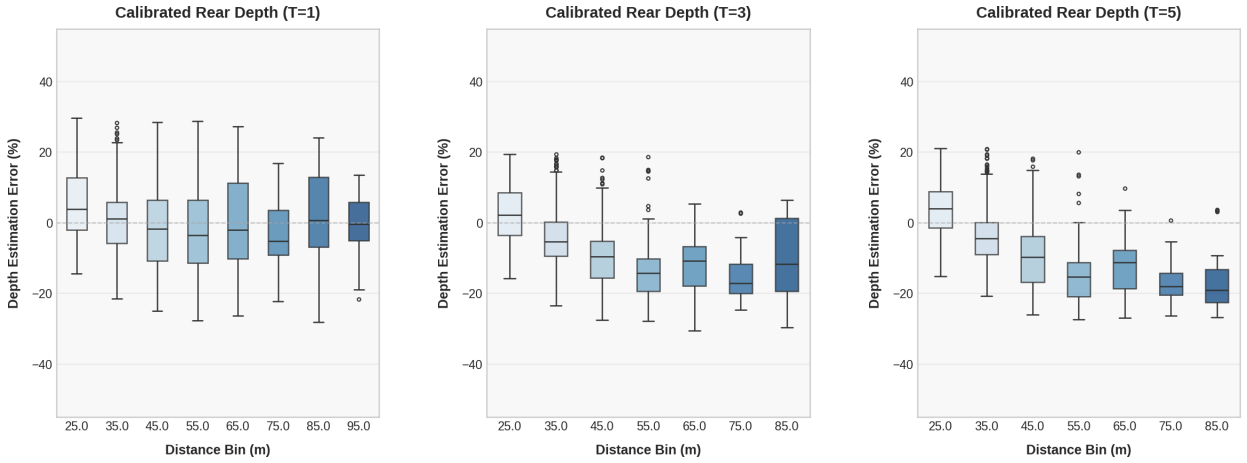
throughput of 42.3Hz when integrated with YOLOv8 instance segmentation model. This performance is achieved through post-training quantization techniques and running models in the Nvidia TensorRT framework, ensuring the system meets the real-time requirements mentioned in previous sections.

While our current implementation demonstrates robust generalization capabilities, further cross-track fine-tuning could potentially enhance performance. This remains an area for

future investigation, particularly for racing environments with extreme visual characteristic variations beyond those represented in our training data.

5.4 Temporal Aggregation Results

To enhance estimation stability and reduce prediction variance, we proposed to apply a temporal aggregation approach that combines predictions across sequential frames. This methodology employs a median-based aggregation over a detection buffer window, theoretically improving accuracy for objects in consistent motion patterns. Figure 5.5 presents a comparative analysis of different temporal buffer sizes (T=1, T=3, and T=5) and their impact on depth estimation performance.



(a) Performance with buffer size 1 (b) Performance with buffer size 3 (c) Performance with buffer size 5

Figure 5.5: Comparison of depth estimation error distributions across different temporal buffer sizes for the model trained on Indianapolis Motor Speedway Oval (IMS) and evaluated on the Las Vegas Road Course (LVRC) dataset. From left to right: Performance metrics for single-frame estimation (T=1), three-frame buffer aggregation (T=3), and five-frame buffer aggregation (T=5).

Our experiments show that temporal aggregation consistently reduces error variance across all distance ranges, with particularly notable improvements in minimum and maximum error bounds. This variance reduction is essential for ensuring stable control inputs to the autonomous racing system. However, the results also reveal several important limitations:

1. **Proximity-Dependent Benefits**: The performance improvements from temporal aggregation are primarily concentrated in the nearest distance bin (0-30m). At T=3, we observe optimal performance for close-range depth estimation, while T=5 introduces increasing drift in mean error values.

2. **Diminishing Returns for Distant Objects**: For objects beyond the immediate proximity range, temporal aggregation benefits diminish and eventually become counterproductive at larger buffer sizes. This pattern aligns with our expectation that for dynamic racing scenarios with significant relative motion, historical detections quickly become irrelevant for distant objects.

These findings inform our final implementation, which employs distance-adaptive temporal aggregation—applying longer buffer windows to close-range detections while maintaining minimal or no aggregation for distant objects. This approach optimizes the trade-off between estimation stability and responsiveness to rapid environmental changes in high-speed racing scenarios.

5.5 Discussion and Implications

Our comprehensive experimental evaluation demonstrates the viability of transformer-based monocular depth estimation for autonomous racing applications, while highlighting several key considerations for practical deployment:

1. **Cross-Track Generalization**: The robust performance across dramatically different track environments confirms generalization capabilities of our model choice and training methods, even in the absence of track-specific fine-tuning. This generalization is particularly valuable for racing applications where comprehensive data collection across all potential venues may be impractical.

2. **View-Specific Calibration**: While the base model demonstrates strong generalization across tracks for the same camera perspective, the significant performance gap between calibrated and uncalibrated cross-view estimations emphasizes the necessity of view-specific calibration processes. Fortunately, our distance-balanced calibration methodology effectively bridges this gap with minimal data requirements. Although we currently lack a dataset to perform a cross-track validation of the calibration parameters, there is evidence showing that the derived parameters should demonstrate consistency across different tracks for the same camera perspective. The calibration process' goal is to capture fundamental projection properties rather than scene-specific characteristics, enabling reliable cross-track generalization.

3. **Temporal Aggregation Trade-offs**: The performance improvements from temporal aggregation must be carefully balanced against increased latency in dynamic racing scenarios. Our findings suggest that moderate buffer sizes (T=3) offer an optimal compromise for close-range perception, while minimal aggregation is preferable for distant objects.

These findings collectively establish a foundation for deploying monocular depth estimation in autonomous racing, particularly addressing the challenge 3D Scene Gaps. The

demonstrated cross-track generalization, calibration methodology, and temporal stability optimizations enable reliable depth perception in challenging racing environments without requiring extensive recalibration or track-specific fine-tuning adaptations. Ultimately, this work bridges the gap in autonomous racing perception through depth estimation from monocular cameras for 3D scene completion. However, it is important to note that the safety implications of integrating this model with planning modules remain to be thoroughly tested in future work, a next step before full deployment in competitive racing scenarios.

Chapter 6

Future Directions

The advancements in monocular depth estimation presented in this thesis open several avenues for future research. In this chapter, we outline promising directions that could further enhance the reliability, accuracy, and applicability of our approach in high-speed autonomous racing environments and beyond.

6.1 Advanced 3D Point Cloud Estimation

Recent developments in 3D point cloud estimation offer compelling approaches to mitigate common limitations in monocular depth prediction, particularly the "hedging artifacts" that occur when models predict intermediate depth values at object boundaries due to uncertainty. The Monocular Geometry Estimation (MoGe) framework proposed by Wang et al. [47] represents a significant advancement in this domain, directly predicting affine-invariant 3D point maps rather than 2D depth maps.

Unlike traditional approaches that infer depth and then back-project to 3D coordinates, MoGe's architecture directly maps image pixels to 3D spatial coordinates, preserving geometric consistency without requiring precise camera intrinsic knowledge. This approach offers several advantages relevant to our autonomous racing application:

1. **Enhanced Boundary Precision**: By operating in the 3D coordinate space rather than depth space, MoGe significantly reduces the boundary bleeding artifacts that plague traditional monocular depth estimation methods, potentially improving the precision of opponent vehicle localization.

2. **Affine-Invariant Representation**: Similar to our calibration approach, MoGe employs an affine-invariant representation that eliminates scale and shift ambiguities, offering a more mathematically principled foundation for cross-view adaptation.

3. **Multi-scale Local Geometry Supervision**: The implementation of spherical region supervision at multiple scales closely aligns with our finding that segmentation-aware depth estimation improves safety-critical object detection accuracy.

Integrating these advancements with our racing-specific implementations could yield substantial improvements in depth estimation accuracy, particularly at object boundaries where precise perception and localization is essential for safe high-speed maneuvering.

6.2 Groundtruth-Independent Calibration Techniques

While our current calibration approach has demonstrated impressive results in bridging the front-to-rear viewpoint gap, it remains dependent on ground truth data for parameter estimation. Future work could explore more flexible calibration methods that reduce or eliminate this dependency.

6.2.1 Structure from Motion with COLMAP

COLMAP (Schönberger et al., [44]) represents a mature Structure from Motion (SfM) pipeline capable of reconstructing sparse 3D scene geometry and camera parameters from image collections without requiring pre-calibrated sensors. This approach could be particularly valuable for deploying our system on new tracks where extensive ground truth data may be unavailable.

By collecting multi-view image sequences during preliminary track testing runs, COLMAP could generate sparse 3D reconstructions sufficient for:

1. **Camera Intrinsic Estimation**: Recovering consistent focal length and principal point parameters for all onboard cameras without requiring controlled calibration procedures.

2. **Cross-View Extrinsic Calibration**: Establishing precise geometric relationships between different camera viewpoints, potentially improving the consistency of depth estimates across the vehicle's full sensing envelope.

3. **Sparse Depth Supervision**: Generating sparse but highly accurate depth points that could serve as calibration anchors for our monocular estimation system.

The primary challenge in this approach involves ensuring sufficient feature correspondence in racing environments, where high speeds and motion blur can degrade feature matching quality. Future research could explore specialized feature extraction and matching techniques optimized for these challenging conditions.

6.2.2 Sim-to-Real Dataset Calibration

Building on the work of previous ART team [9], sim-to-real adaptation techniques offer another promising pathway for groundtruth-independent calibration. The approach addresses the fundamental "reality gap" challenge in autonomous racing, where simulation

data—though plentiful—exhibits visual and physical discrepancies compared to real-world racing scenarios.

A potential implementation could involve:

1. **Domain-Adaptive Depth Estimation**: Training depth estimation models on abundant simulation data where perfect ground truth is available, then employing domain adaptation techniques to transfer this knowledge to real-world camera images.

2. **Few-Shot Calibration**: Leveraging limited real-world depth measurements to learn transformation functions that map simulation-trained models to real-world depth distributions.

3. **Cycle-Consistent Adaptation**: Implementing cycle-consistency constraints between simulation and real domains to ensure geometric consistency without requiring paired examples.

This approach is particularly relevant for multi-car racing scenarios, where collecting real-world training data with opponent vehicles presents logistical and safety challenges. By generating unlimited simulated multi-car racing scenarios and developing robust sim-to-real adaptation techniques, we could train models capable of accurate depth estimation in complex racing scenarios without requiring extensive real-world data collection.

Bibliography

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions, 2022.

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023.

[3] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023.

[4] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second, 2025.

[5] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond, 2019.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[8] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation, 2022.

[9] Ashwat Chidambaram. Leveraging zero-shot sim2real learning to improve autonomous vehicle perception. Master's thesis, EECS Department, University of California, Berkeley, May 2024.

[10] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[13] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network, 2014.

[14] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation, 2021.

[15] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4008–4017. IEEE, June 2021.

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[17] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation, 2018.

[18] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation, 2019.

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[21] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[23] Yuanduo Hong, Huihui Pan, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes, 2021.

[24] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[25] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation, 2019.

[26] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. Ccnet: Criss-cross attention for semantic segmentation, 2020.

[27] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkila. Guiding monocular depth estimation using depth-attention volume, 2020.

[28] Glenn Jocher and Ultralytics. Yolov8: Real-time object detection and image segmentation model, 2023.

[29] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization, 2016.

[30] Daehwan Kim, Haejun Chung, and Ikbeom Jang. Calibration of ordinal regression networks, 2025.

[31] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks, 2019.

[32] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering, 2020.

[33] Mykola Lavreniuk. Spidepth: Strengthened pose information for self-supervised monocular depth estimation, 2024.

[34] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation, 2019.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[38] Rudra P K Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network, 2019.

[39] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

[40] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2020.

[41] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation, 2019.

[42] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2018.

[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019.

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[45] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion, 2020.

[46] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video, 2017.

[47] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision, 2025.

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks, 2018.

[49] Huikai Wu, Junge Zhang, Kaiqi Huang, Kongming Liang, and Yizhou Yu. Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation, 2019.

[50] Tianyi Wu, Sheng Tang, Rui Zhang, and Yongdong Zhang. Cgnet: A light-weight context guided network for semantic segmentation, 2019.

[51] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding, 2018.

[52] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021.

[53] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[54] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[55] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency, 2017.

[56] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks, 2020.

[57] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose, 2018.

[58] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation, 2020.

[59] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation, 2018.

[60] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[61] Yuhui Yuan, Xiaokang Chen, Xilin Chen, and Jingdong Wang. Segmentation transformer: Object-contextual representations for semantic segmentation, 2021.

[62] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction, 2018.

[63] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation, 2018.

[64] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks, 2020.

[65] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images, 2018.

[66] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2017.

[67] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[68] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, 2021.

[69] Kaichen Zhou, Jia-Wang Bian, Jian-Qing Zheng, Jiaxing Zhong, Qian Xie, Niki Trigoni, and Andrew Markham. Manydepth2: Motion-aware self-supervised monocular depth estimation in dynamic scenes, 2025.

[70] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video, 2017.

[71] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation, 2019.

Appendix A

Finding the Best Transferrable Region of Interest

When applying monocular depth estimation models across different viewpoints and conditions, identifying the most transferrable region of interest (ROI) within camera frames is important for achieving consistent and reliable depth predictions. We investigate which portions of the camera view provide the most stable and generalizable depth estimation results when using pre-trained models like DPT-Large [39].

Our objective is to determine the optimal ROI that minimizes transformation parameter variance across frames, thereby enhancing the transferability of depth estimation models to new viewpoints—particularly important for zero-shot deployment scenarios in autonomous racing where training data from all camera perspectives may not be available. Figure A.1 shows the initial depth map results generated by the DPT-Large model.

Initial experiments with a pretrained DPT-Large on full camera frames revealed that certain regions consistently produced less reliable depth estimates. Specifically, areas containing the vehicle's nose and sky regions exhibited higher depth prediction variance, suggesting these regions negatively impact the model's transferability across different frames and potentially across different viewpoints. To quantify the transferability of different ROIs, we would use the coefficient of variation (CV) of these transformation parameters across frames serves as our transferability metric—lower CV values indicate more consistent and transferrable depth estimation performance. Surprisingly, the depth maps generated by the DPT-Large model already provided a good approximation of the actual depth of the objects in the scene without having it having seen any images from the race car's point of view before.

Occasionally, the depth maps generated by the DPT-Large model were not as accurate as desired, especially in areas of the car's nose or in frames where the sky was taking up a large portion of the image, providing little detail for depth inference. Since the overall depth map quality was still usable, we decided that this approach is worth further investigation and evaluation.

One inherent downside of in-the-wild monocular depth estimation is that the depth maps generated by the model are not metrically accurate, meaning that the depth values are only

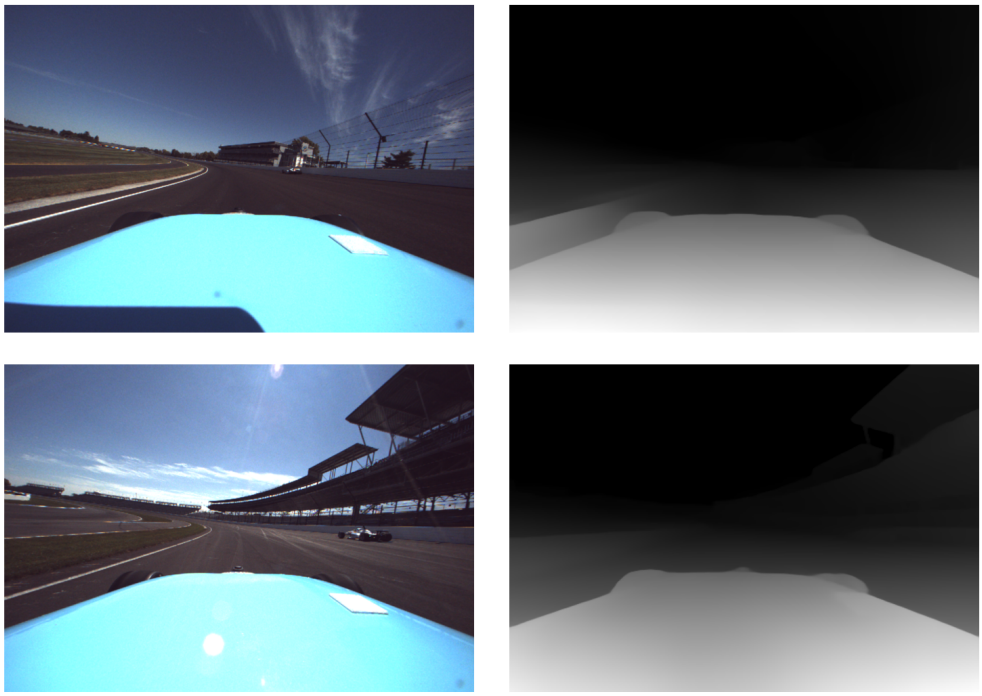


Figure A.1: Depth maps generated by DPT-Large model

relative to each other and not to the real world. Since our goal is to obtain metrically accurate depth maps, we decided to apply a post-processing step to the obtained depth maps to explore, whether it is possible to find a generalizable transformation to convert relative depth to metrically accurate depth maps. We did so first by looking for simple linear transformation parameters to match the depth maps with respective metric depth slices of a lidar scan taken during the same moment. With regards to the pixel intensity values of the depth maps $\text{depth}_{\text{relative}}$, the transformation can be described with a scale factor s and bias offset b as follows:

$$\text{depth}_{\text{metric}} = s \cdot \text{depth}_{\text{relative}} + b \tag{A.1}$$

Figure A.2 shows the exemplary matching accuracy for a few depth map frames when applying the linear transformation to our DPT-Large obtained depth maps.

The linear transformation approach showed promising results, however, we found that the transformation parameters were not consistent across different frames and that applying the aggregated transformation to the entire dataset resulted in a significant loss of accuracy for many depth maps. Over a set of 71 images that represent a full racing lap, we get a mean scale factor s of -477.55 and mean bias offset b of 68.59, however most importantly also a standard deviation of 115.4 for the scale factor and 12.05 for the bias offset, indicating that the required transformation parameters fluctuate in between frames.

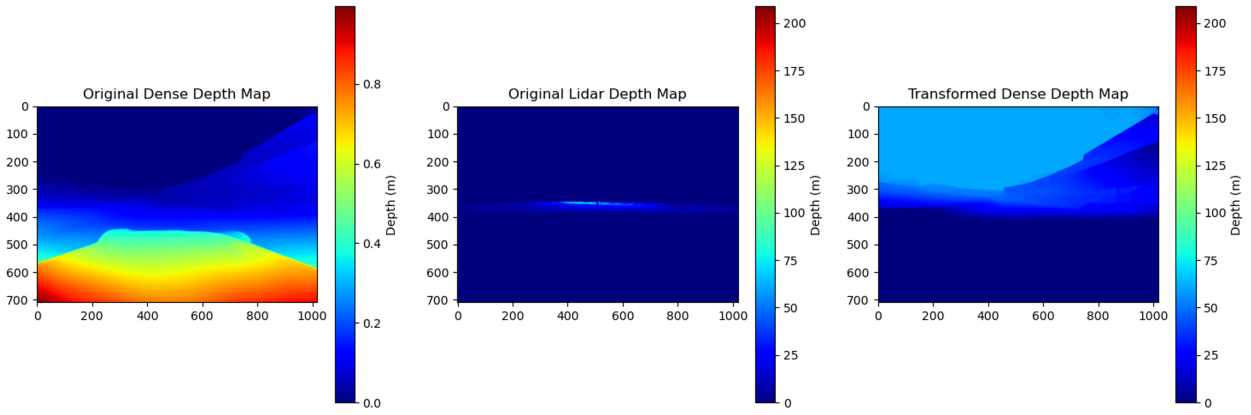


Figure A.2: Depth map transformation from DPT-Large inference (left) to metric depth (right) using lidar ground truth (middle).

Crop	Mean s	Std s	CV (%)	Mean b	Std b	CV (%)
None	-477.55	115.4	25.17	68.59	12.05	17.57
Partial	-308.81	92.25	29.87	75.56	8.37	11.08
Full	-123.65	19.13	15.47	78.75	8.63	10.96

Table A.1: Transformation Parameters for different cropping approaches

In order to curb this issue, we decide to limit the camera frame to a cutout of the track that removes the majority of the sky and the car's nose, as these areas are the most problematic for the depth estimation model. The resulting DPT-Large depth maps and transformation results can be seen in figure A.3.

We distinguish in between two cropped versions: a) One in which we leave a little bit of the car's front wing and nose in the frame and b) an even more aggressive crop in which we eliminate the car's nose from the frame entirely in hopes to reduce the depth variance even further. Table A.1 shows the obtained required transformation parameters, their standard deviations as well as the coefficient of variation (CV) for the different cropping approaches.

When comparing the transformation parameters with varying camera frame crops, the coefficient of variation (CV) for the scale factor and bias offset is most important to gauge fluctuation in the transformation parameters. While the partially cropped frames actually have an increased variance in the scale factors s with a CV of 29.87%, the frames in which the car's nose is fully cropped out have a CV of 15.47%, indicating that the transformation parameters are far more consistent across frames when the car's nose is removed from the frame. The bias b is consistently better and stable across all cropping approaches, with a CV of 17.57% for the full frame, 11.08% for the partial crop, and 10.96% for the full crop. Figure A.4 shows the exemplary transformation of a fully cropped depth map to metric depth.

Removing the car's nose from the frame seems to be the most promising approach to

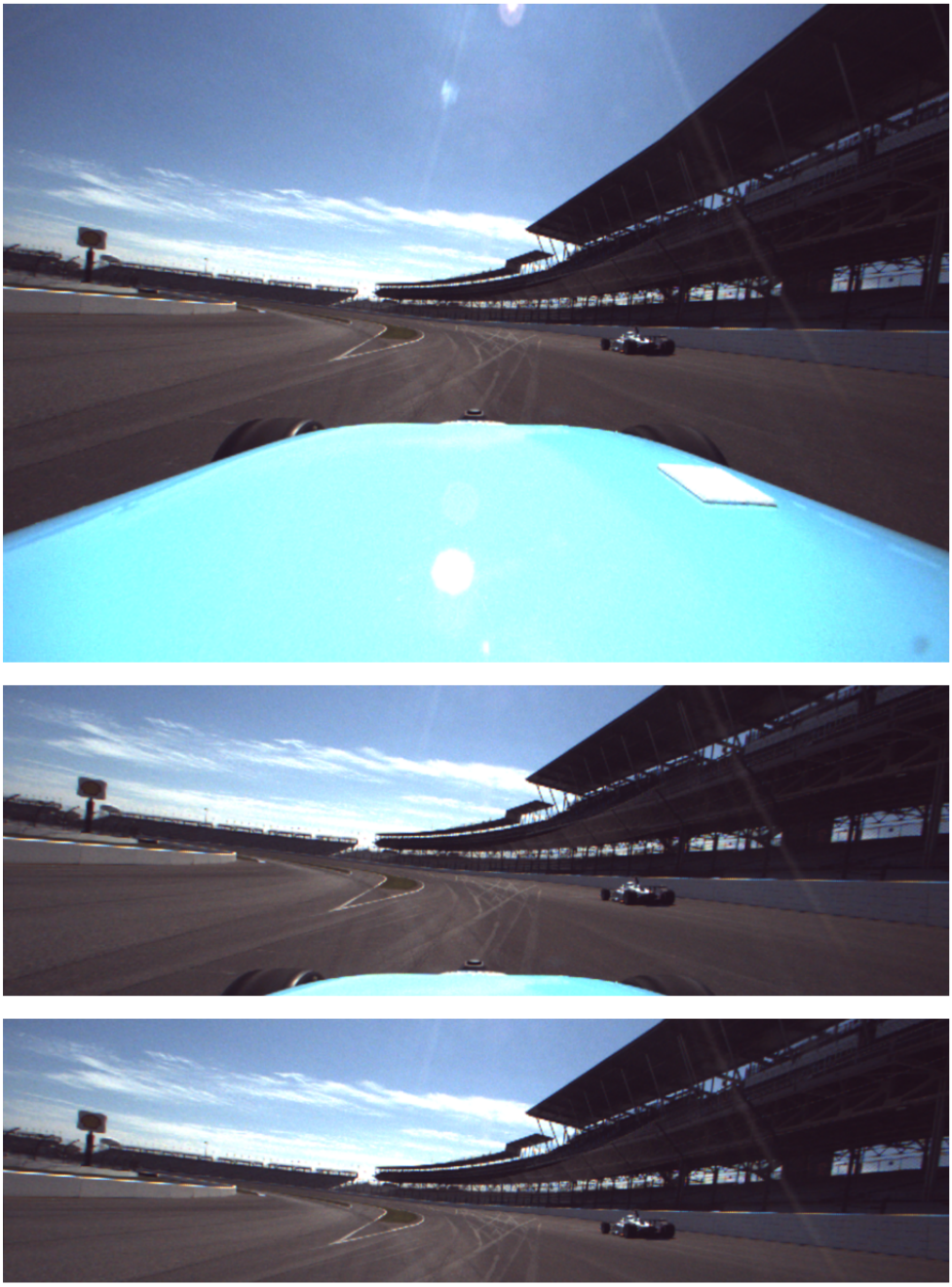


Figure A.3: Image crop variations: Original (top), partial nose crop (middle) and full (bottom).

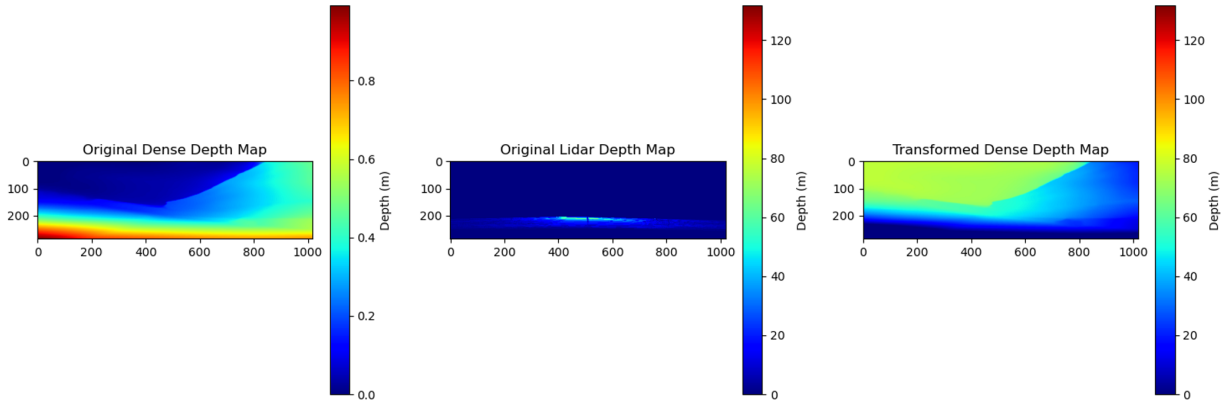


Figure A.4: Depth map transformation from DPT-Large inference (left) to metric depth (right) using lidar ground truth (middle) on fully cropped input image.

obtain metrically accurate depth maps, as the transformation parameters are most stable across frames and the depth maps are most accurate.

Using above approaches has yielded a quick and effective method to leverage large state-of-the-art depth estimation models to obtain metric ground truth depth with mostly acceptable accuracy. Depending on the required ground truth data accuracy and quality, however, more sophisticated transformation methods such as RANSAC are required, which can be examined in the remaining project runtime.

Appendix B

Comparative Analysis of Segmentation Models

B.1 Evaluation Protocol

To determine the most suitable segmentation model for cross-track generalization, we conducted a comprehensive evaluation of state-of-the-art segmentation architectures using the MMSegmentation framework [10]. The evaluation specifically focused on zero-shot performance, testing models trained exclusively on the Cityscapes dataset [11] without any fine-tuning on racing track data. This experimental design was chosen to assess the models' ability to generalize to previously unseen racing environments.

B.2 Model Selection and Methodology

Our comparative analysis encompassed both convolutional neural network (CNN) based [5–7, 14, 18, 21, 23–26, 31, 32, 34, 36, 38, 42, 43, 48–51, 56, 58, 59, 61, 63–67, 71], and architectures and transformer-based architectures [8, 35, 40, 52, 53, 68].

Two representative scenes from the Indianapolis Motor Speedway dataset were selected to demonstrate the qualitative performance differences between architectural paradigms. These scenes were chosen to capture the diversity of visual challenges present in racing environments, including varying lighting conditions, track surface textures, and background complexity.

B.3 Results and Analysis

The experimental results revealed a marked performance advantage for transformer-based architectures in zero-shot racing track segmentation. Figures B.1 and B.2 illustrate the

segmentation outputs from CNN-based models, while Figures B.3 and B.4 present the corresponding results from transformer architectures.

B.4 Key Findings

The superior performance of transformer-based models in this zero-shot evaluation can be attributed to several factors:

1. **Global Context Modeling**: Transformer architectures' self-attention mechanisms enable superior modeling of long-range dependencies, crucial for understanding the spatial relationships in racing track environments [12].

2. **Robust Feature Representations**: The hierarchical feature representations learned by transformers appear to generalize more effectively to previously unseen domains, particularly in scenarios with significant domain shift from urban scenes to racing tracks.

3. **Scale Invariance**: Transformer models demonstrated better handling of scale variations, particularly important for detecting vehicles at varying distances on the track.

B.5 Implications for Racing Applications

These findings support our decision to employ Mask2Former as the primary segmentation backbone in our perception pipeline. The model's robust zero-shot performance eliminates the need for extensive track-specific data collection and annotation, significantly reducing deployment costs and enabling rapid adaptation to new racing venues.

B.6 Limitations and Future Work

While our evaluation focused on qualitative assessment, future work should incorporate quantitative metrics such as intersection over union (IoU) and pixel accuracy. Additionally, evaluating performance under varying weather conditions and times of day would provide a more comprehensive understanding of model robustness in racing applications.

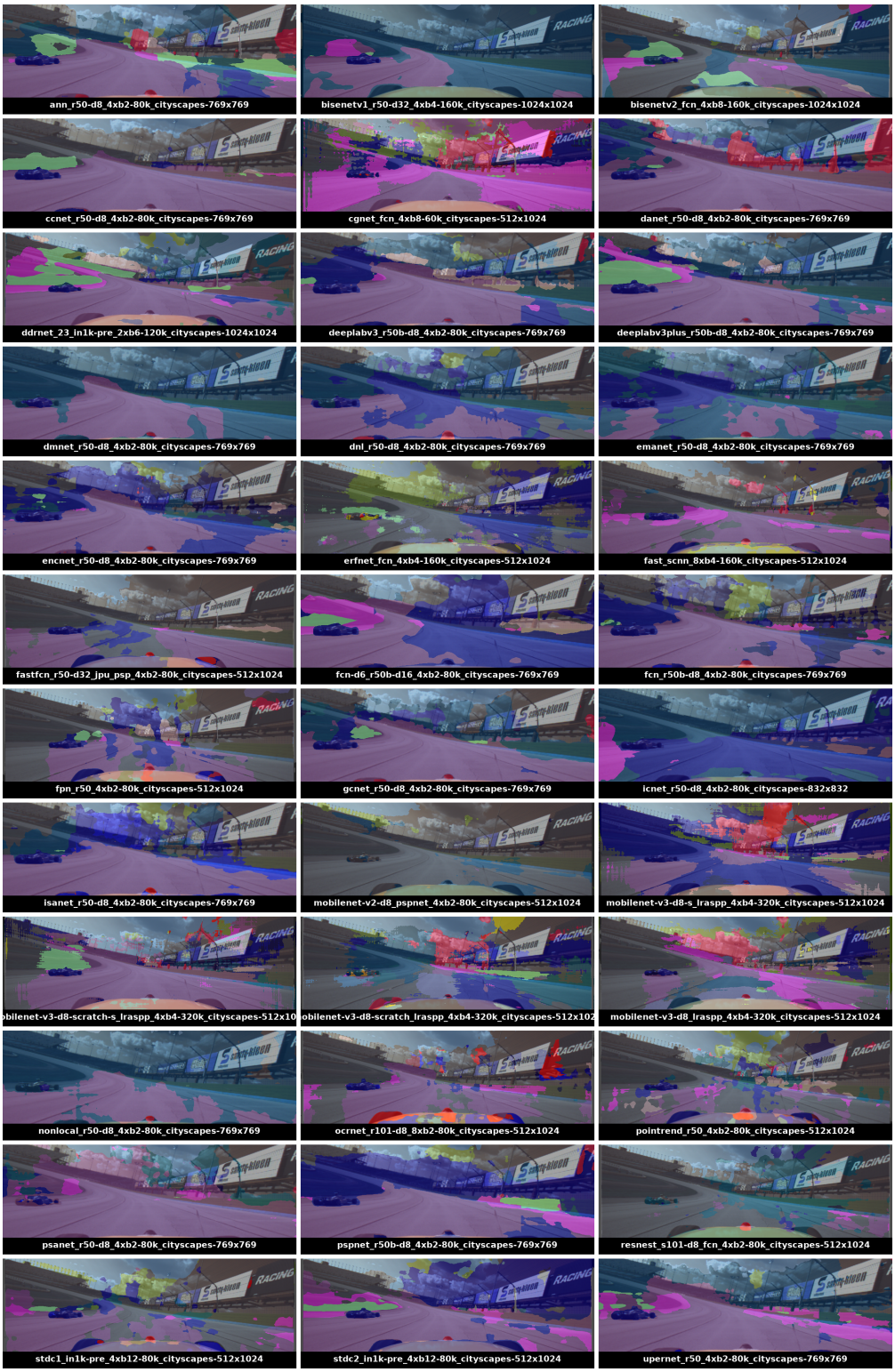


Figure B.1: Segmentation results from CNN-based models on Scene 1 from the Indianapolis Motor Speedway dataset. Models shown (left to right, top to bottom): MobileNetV3, ResNet-50, ResNet-101, and PSPNet.

Figure B.2: Segmentation results from CNN-based models on Scene 2 from the Indianapolis Motor Speedway dataset, demonstrating performance under different lighting and track conditions.

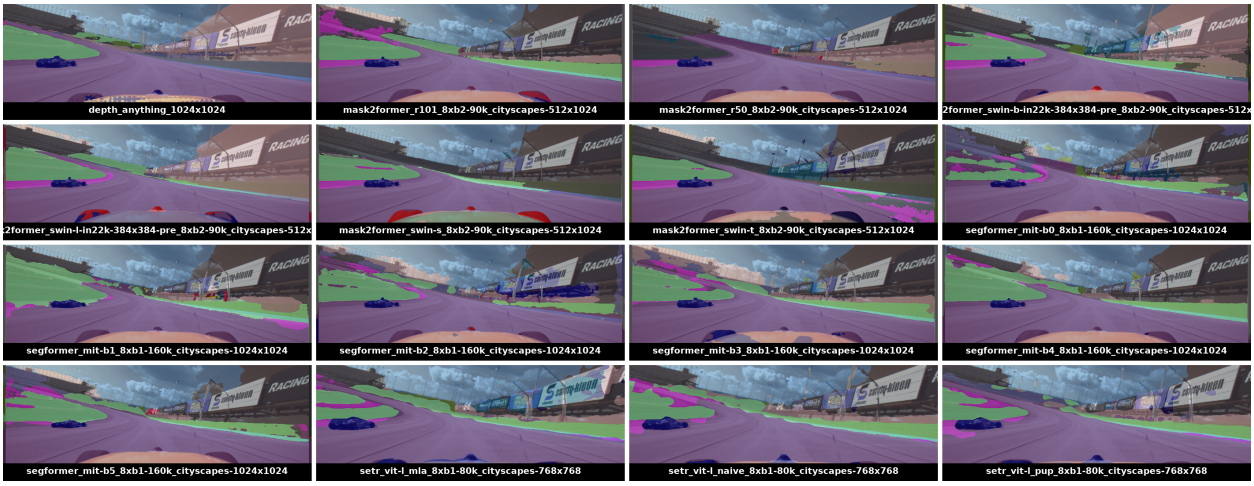


Figure B.3: Segmentation results from transformer-based models on Scene 1 from the Indianapolis Motor Speedway dataset. Models shown (left to right, top to bottom): SegFormer, Mask2Former, and Swin Transformer variants.

Figure B.4: Segmentation results from transformer-based models on Scene 2 from the Indianapolis Motor Speedway dataset, illustrating superior generalization capabilities compared to CNN-based approaches.