Herd's Eye View: Improving Game AI Agent Learning with Collaborative Perception

by

A Thesis submitted to the

School of Graduate Studies

in partial fulfillment of the

requirements for the degree of

Master of Science

Supervisors: Dr. Andrew Vardy & Dr. David Churchill

Department of Computer Science

Memorial University of Newfoundland

March 2024

St. John's Newfoundland

Abstract

We present a novel perception model named Herd's Eye View that adopts a global perspective derived from multiple agents to boost the decision-making capabilities of reinforcement learning agents in multi-agent environments, specifically in the context of game AI. The Herd's Eye View approach utilizes cooperative perception to empower reinforcement learning agents with global reasoning ability, enhancing their decision-making. We demonstrate the effectiveness of Herd's Eye View within simulated game environments and highlight its superior performance compared to traditional egocentric perception models. This work contributes to cooperative perception and multi-agent reinforcement learning by offering a more realistic and efficient perspective for global coordination and decision-making within game environments. Moreover, our approach promotes broader AI applications beyond gaming by addressing constraints faced by AI in other fields such as robotics.

Contents

List of Figures

List of Tables

List of Acronyms

BEV	Bird's Eye View
CNN	Convolutional Neural Network
CVT	Cross-View Transformer
GAN	Generative Adversarial Networks
GNN	Graph Neural Network
HEV	Herd's Eye View
HEV-CVT	Herd's Eye View Cross-View Transformer
IPM	Inverse Perspective Mapping
IoU	Intersection over Union
LSS	Lift Splat Shoot
MA-POCA	Multi-Agent POsthumous Credit Assignment
MARL	Multi-Agent Reinforcement Learning

MLP	Multilayer Perceptron
OC	Orbital Construction
PON	PyrOccNet
PPO	Proximal Policy Optimization
PV	Perspective View
RNN	Recurrent Neural Network
RL	Reinforcement Learning
TRPO	Trust Region Policy Optimization
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
ViT	Vision Transformer

Chapter 1

Introduction

1.1 Perspective Views

Game environments traditionally grant AI agents access to extensive global information from the game engine. While this configuration assists in efficient decision-making, it does not accurately represent the restrictions encountered by AI applications outside of gaming, where comprehensive access to a system's software or engine is not feasible. Consequently, game AI techniques that rely predominantly on game engine data may limit their potential contribution to broader AI applications, as their dependency on perfect information and global environmental data is often unrealistic in other contexts such as robotics and autonomous vehicles.

In response to these challenges, our work delves into the application of more constrained, realistic perception models for game AI. We take inspiration from pub-

lications like the ViZDoom platform [17] and the Obstacle Tower Challenge [18] that have embraced the shift towards game AI with real-world constraints. ViZDoom and Obstacle Tower have utilized visual data as the primary input for AI agents, enabling them to navigate complex 3D environments. These perception-based game AI agents reinforce the importance of models without access to game engine information.

1.1.1 Bird's Eye View

Research in autonomous vehicles has made extensive strides in AI perception models, particularly using intermediary environmental representations like the Bird's Eye View (BEV) [19]. The BEV model provides an overhead perspective of the environment, often in the form of a semantic obstacle grid, from a single "ego" vehicle's standpoint. This concept has become a key component in many self-driving systems [19] an example of this can be seen in Figure 1.1.

Drawing on these past works, we propose a similar intermediary representation for game AI: the Herd's Eye View (HEV) model. Differing from the BEV's ego-centric perspective, the HEV model offers a shared world-centric perception derived from multiple agents. This shared perception model aligns more closely to real-world AI applications, where multiple systems often work together to understand and navigate their environment.

Figure 1.1: Real-time Bird's Eye View on the Infotainment System of a Tesla [1].

1.1.2 Herd's Eye View

The HEV model presents dual advantages. First, it mirrors the constraints faced by AI outside of gaming, contributing to the development of more believable AI behaviour in games. Second, it alleviates the computational demands associated with the BEV model, where each agent maintains its own unique view of the environment, instead, only a single shared global view is utilized.

We also incorporate Reinforcement Learning (RL) into our approach, enabling us to test the effectiveness of HEV in both low-level control tasks and high-level planning challenges concurrently in complex environments. Importantly, our agents are assessed not solely on their ability to navigate familiar environments, but also on their ability to handle unique variations of these environments. This highlights the

importance of generalization in adapting to novel scenarios within environments.

To assess the effectiveness of the HEV model, we conduct two sets of experiments in three simulated Multi-Agent Reinforcement Learning (MARL) game environments. The first compares the accuracy of HEV world-centric predictions with BEV egocentric predictions. The second experiment evaluates the efficiency of policies learned by RL agents trained on HEV perspective views compared to those trained on BEV perspective views.

1.2 Motivation

Our research responds to the growing interest in intermediary 3D perception representations in computer vision applications [19]. This trend is crucial as AI applications in areas such as autonomous vehicles and robotics, where the luxury of unrestricted access to global information, typical in game environments, is rarely available. Recognizing this, our research pivots towards exploring more grounded and practical perception models in game AI.

The rationale for our focus is evident in recent academic trends. The graph in Figure 1.2 shows a consistent increase in studies mentioning "Bird's Eye View" and "Perception" from 2018 to 2023. This data points to a significant interest in refining AI perception models, aligning well with the direction of our work.

Our motivation is to align game AI with real-world AI challenges and contribute to the expanding knowledge base of advanced perception models. The growing aca-

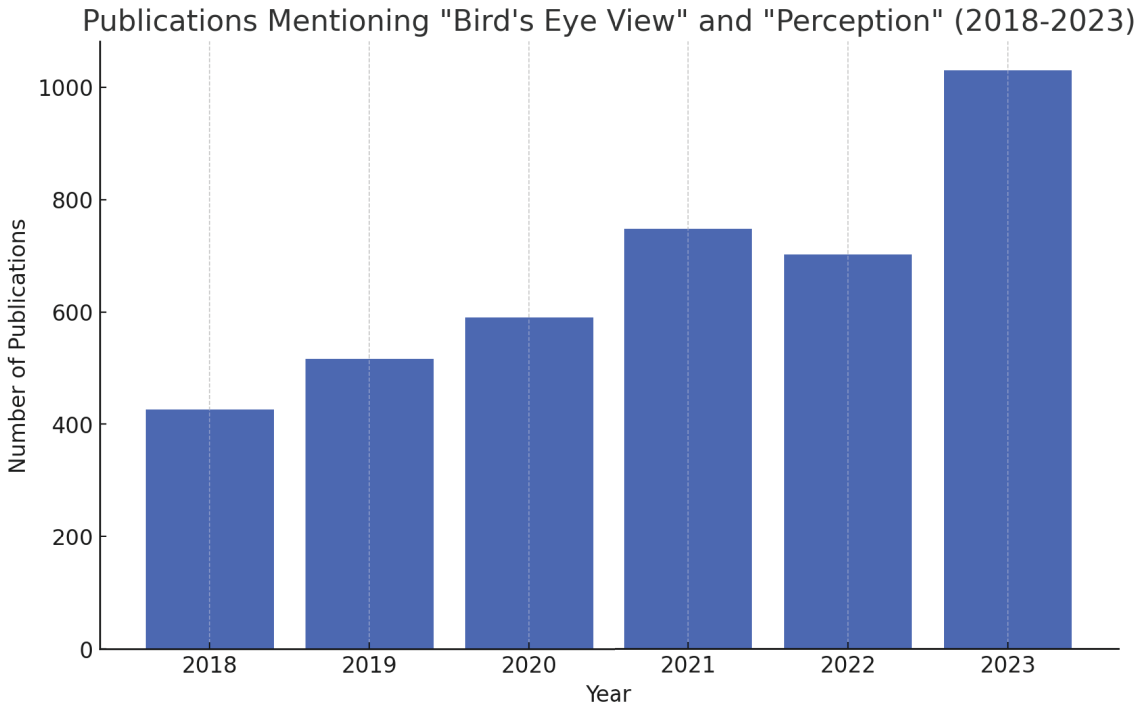

Figure 1.2: Publications mentioning "Bird's Eye View" and "Perception" (2018-2023), indexed by Google Scholar.

demic interest underlines our work's relevance and potential impact in the wider AI landscape. Our research shows the HEV model is effective and viable for creating sophisticated and realistic AI agents. By aligning with current academic trends and addressing the needs of modern AI applications, our work contributes to the evolving 3D perception field in AI and games.

Our exploration of more realistic perception models provides significant insights for game AI development, stressing the wider applicability of these techniques beyond the gaming industry.

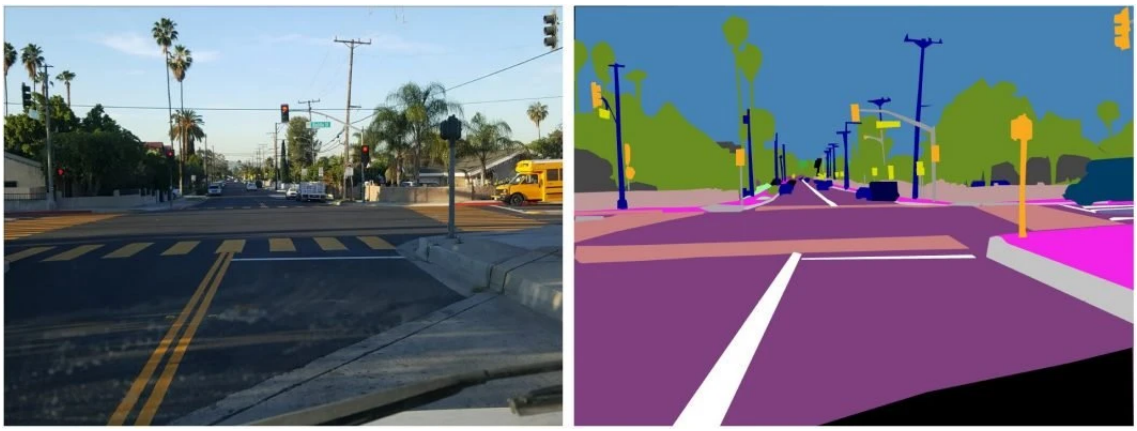

Figure 1.3: Example semantic segmentation (left) original image, (right) semantic segmentation [2].

1.3 Image Segmentation

Rooted in the realm of computer vision, image segmentation is a foundational image processing task. This process involves the task of partitioning an image into constituent segments, typically referred to as pixels or superpixels [20]. The primary goal of image segmentation is to map an image's representation into a more analyzable and informative structure. Specifically, image segmentation assigns a unique label to each pixel in an image, thus ensuring that pixels bearing the same label display similar visual attributes, such as colour or intensity [20]. This transformation effectively morphs an image from a collection of RGB pixels into a topographical map of regions, simplifying further analysis and interpretation. An example of image segmentation can be seen in Figure 1.3.

1.3.1 Segmentation Metrics

The evaluation of image segmentation techniques is a critical facet of their development, ensuring the accuracy and reliability of the methodology. Among the wide array of metrics available for this task, Intersection over Union (IoU), also referred to as the Jaccard Index, is frequently utilized [21].

IoU provides a quantification of the degree of overlap between two bounding boxes or segmented regions. It computes the ratio of the intersecting area between the predicted segmentation and the ground truth, and the union of these two areas [21]. The resulting value falls within the range [0, 1], where 0 indicates no overlap and 1 represents complete congruence between the predicted and actual segmentation.

As illustrated in Figure 1.4, IoU serves as a comprehensive yet intuitive metric of segmentation accuracy. It offers essential quantitative feedback instrumental for the refinement and improvement of image segmentation algorithms [22].

1.4 Reinforcement Learning

Reinforcement Learning is a sub-field of machine learning where agents must learn the highest reward behaviours by interacting with an environment [3]. In this process, an agent first takes an action, which transitions the environment to a new state which usually provides a reward related to the action taken. This flow of action and feedback of the reinforcement learning agent can be seen below in Figure 1.5.

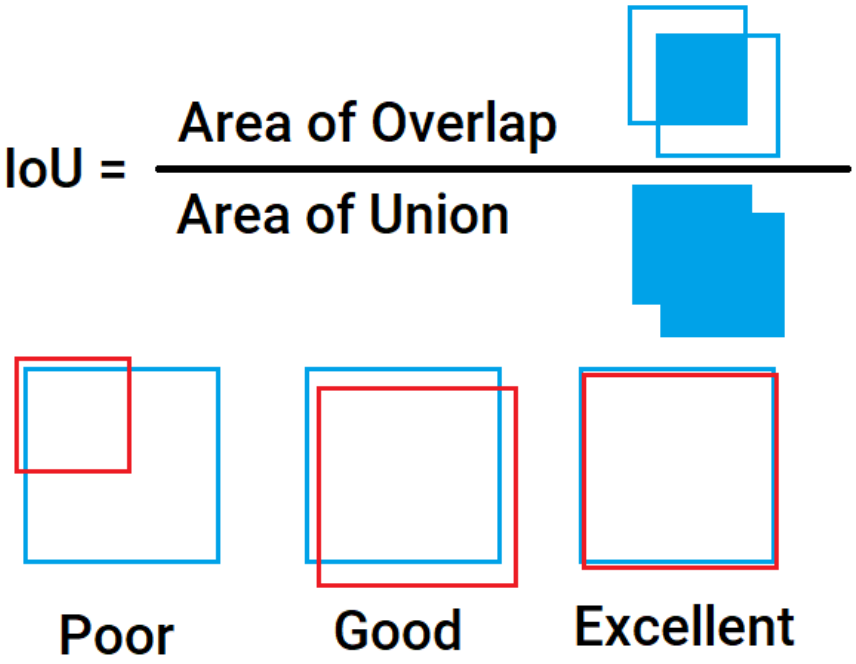

Figure 1.4: Visualization of the Intersection over Union (IoU) metric

Reinforcement learning consists of five key elements, the *environment*, *state*, *reward*, *policy* and *value*, these are more clearly defined below:

Environment: The physical world agent(s) interact with.

State: The agents most recent observations and beliefs of the environment.

Reward: Feedback from the environment about how *good* action(s) taken are.

Policy: The mapping from agent state to action.

Value: Future reward an agent receives taking action in a state.

The structure and the granularity of information encapsulated within the state representation significantly influence an agent's learning trajectory and success within a given environment [3]. The complexity of the environment dictates the required

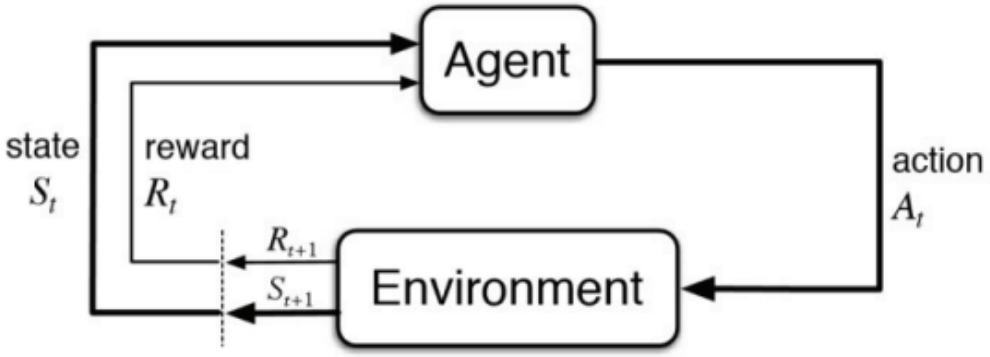

Figure 1.5: The flow of a reinforcement learning model [3].

state representation; in simplistic environments, a few binary variables might suffice to encapsulate the state, whereas, in more complex environments, high-dimensional images may be required to capture details required for the agent to make optimal decisions. Choosing the optimal state representation in RL often encapsulates a trade-off between the richness of information and the speed or size of the model, highlighting the importance of adequate state representation strategies to balance effective learning and judicious decision-making against computational efficiency and model compactness. It should be noted that the described RL framework assumes a fully observable Markov Decision Process (MDP), where the agent's state encapsulates all relevant information from the environment for decision-making. This contrasts with a Partially Observable Markov Decision Process (POMDP), where the agent must make decisions with incomplete information about the current state.

1.4.1 Multi-Agent Reinforcement Learning

In multi-agent reinforcement learning systems, agents cooperate or compete to maximize a reward signal they simultaneously receive [3]. In multi-agent systems, agents often lack knowledge of other agents and their objectives, this is commonly referred to as *social awareness*. It is not always necessary for agents to have social awareness to learn the optimal policy. In our multi-agent environments, social awareness is not necessary when working with agents but becomes increasingly necessary the more agents that exist in the environment [23]. Some multi-agent systems train with no possibility for social awareness, using only a single agent and then later applying the learned policy to multiple agents in the environment. This single-agent training process is beneficial because it shrinks problem dimensionality but often leads to unforeseen errors and inefficiencies between agents [23].

1.4.2 Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a family of policy gradient methods for reinforcement learning. These methods alternate between sampling data through interaction with the environment and optimizing a "surrogate" objective function using stochastic gradient ascent [24]. Unlike previous policy gradient methods, which perform one gradient update per data sample, PPO utilizes an objective function that enables multiple epochs of minibatch updates [24].

PPO has some of the benefits of Trust Region Policy Optimization (TRPO) [25],

but it is much simpler to implement, more general, and has better sample complexity. PPO has been tested on many games and benchmarks, such as robotic locomotion and Atari games, outperforming other online policy gradient methods [24].

The key idea behind PPO is to use a surrogate objective function with clipped probability ratios, which forms a pessimistic estimate (i.e., lower bound) of the performance of the policy. To optimize policies, PPO alternates between sampling data from the policy and performing several epochs of optimization on the sampled data [24].

The surrogate objective function in PPO is defined as follows:

$$L^{CLIP}(\theta) = \hat{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t)]$$

Where θ is the policy parameter, $\hat{E}_t$ denotes the empirical expectation over timesteps, r_t is the ratio of the probability under the new and old policies respectively, $\hat{A}_t$ is the estimated advantage at time t, and ϵ is a hyperparameter, typically set to 0.1 or 0.2 [24]. The first term inside the min is the surrogate objective. The second term, $\text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t$, modifies the surrogate objective by clipping the probability ratio, this removes the incentive for moving r_t outside of the interval $[1-\epsilon, 1+\epsilon]$. The minimum of the clipped and unclipped objective is taken, so the resulting objective is a lower bound of the original objective [24].

1.4.3 Multi-Agent POsthumous Credit Assignment

Agents within a cooperative Multi-Agent Reinforcement learning (MARL) system must collaborate to achieve shared objectives. However, a significant challenge arises when agents terminate before their teammates within an episode, where an agent may "spawn" (i.e., be created) or "die" (i.e., terminate) within a single episode. The core challenge is called the Posthumous Credit Assignment problem, where terminated agents must learn from the team's success or failure that occurs beyond their own existence [26]. Traditional MARL algorithms often handle this by placing terminated agents in an "absorbing state" until the entire group reaches termination Cohen et al.. While this approach allows existing algorithms to manage terminated agents without architectural modifications, it introduces challenges in training efficiency and resource utilization [26].

The Multi-Agent POsthumous Credit Assignment (MA-POCA) architecture introduced by Cohen et al. leverages attention mechanisms instead of fully connected layers with absorbing states. MA-POCA is designed to handle scenarios where agents are created or terminated within an episode but share a reward function. By applying a self-attention layer to active agent information, MA-POCA scales to an arbitrary number of agents. This attention mechanism allows the critic to attribute the future expected value of the group to states with terminated agents without the need for absorbing states [26].

Chapter 2

Related Works

2.1 Transformers

Transformer models represent a significant paradigm shift in the field of machine learning, notably deviating from the Recurrent Neural Network (RNN) based encoder-decoder architectures that were prevalent before their introduction [4]. The primary innovation of Transformer models lies in their utilization of self-attention mechanisms, entirely dispensing with the recurrence and convolutions that previous models relied on. Unlike traditional attention mechanisms that depend on sequence-aligned RNNs or convolutions, the Transformer model computes the representation of sequences by relating different items within the same sequence. This approach, termed self-attention, allows the model to internalize the relationships between the various elements in a sequence [4].

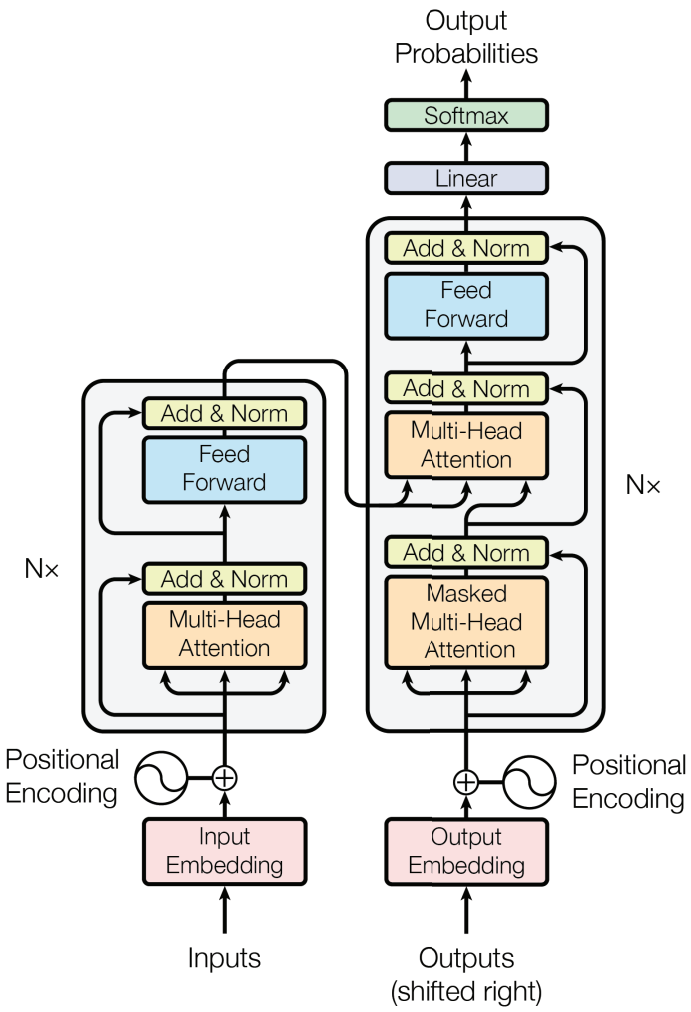

Figure 2.1: Transformer model architecture [4].

2.1.1 Architecture

The Transformer model first proposed by Vaswani et al. comprises two main components: the encoder and the decoder. Each of these components consists of multiple identical layers, with each layer having two sub-layers. The first is a multi-head self-attention mechanism, and the second is a position-wise fully connected feed-forward network. This architecture enables the model to process multiple positions in the

sequence simultaneously in a constant number of steps, improving computational efficiency. Additionally, the Transformer uses position encodings added to the input embeddings at the bottoms of the encoder and decoder stacks. This method allows the model to consider the order of the sequence, which is essential for tasks with ordered data. A full overview of the transformer architecture can be seen in Figure 2.1.

2.1.2 Scaled Dot-Product Attention

In the Transformer model, the attention mechanism is a mapping between a query and a set of key-value pairs to an output. The scaled dot-product attention mechanism is a crucial component of this model, where the output is computed as a weighted sum of the values. The weight assigned to each value is computed by a compatibility function of the query with the corresponding key [4]. This mechanism first computes a dot product for each query with all of the keys, then divides each result by the square root of the dimension of keys ($\sqrt{d_k}$), and then applies a softmax function to obtain the weights that are used to scale the values [4]. This process can be seen in Figure 2.2.

The attention function can be computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V [4]. The matrix of outputs can be computed as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

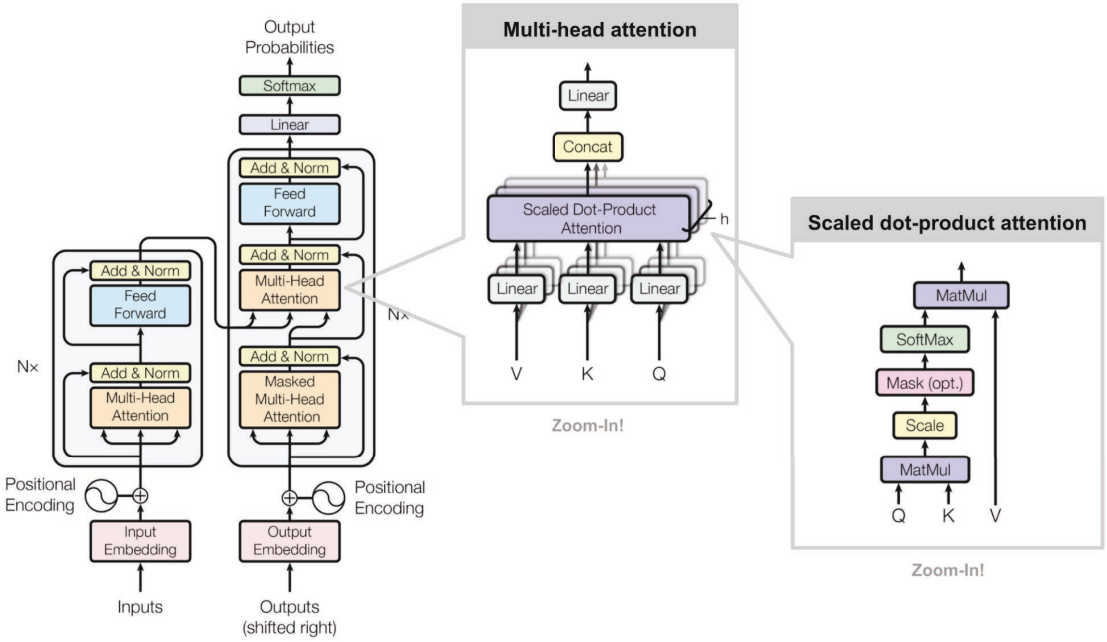

Figure 2.2: Transformer model architecture with zoom in on multi-head attention and scaled dot-product attention [4].

The scaled dot-product attention mechanism in the Transformer model incorporates a scaling factor to mitigate the vanishing gradients problem, a common issue in neural networks that can slow or halt learning. As the dot product of query and key vectors can increase significantly with their dimensionality (d_k), applying the softmax function to these large values could yield extremely small gradients, which is indicative of the vanishing gradient problem. To avoid this, the Transformer model uses a scaling factor of $1/\sqrt{d_k}$ to moderate the magnitude of the dot products before they are passed through the softmax function, ensuring a more balanced gradient flow and promoting effective learning [4].

2.1.3 Multi-Head Attention

The second key innovation in the Transformer model is the Multi-Head Attention mechanism. Vaswani et al. [4] built upon their scaled dot-product attention to propose this mechanism, which provides multiple learned linear projections of the input data. The Multi-Head Attention mechanism allows the model to jointly attend to information from different representation subspaces at different positions. Multiple attention heads allow the model to focus on different positions, capturing various relationships between the input sequences.

The query, keys, and values for each head are computed by applying distinct learned linear transformations (represented by the weight matrices W_i^Q, W_i^K, and W_i^V) to the original queries, keys, and values. The scaled dot-product attention is then applied to these transformed inputs:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

The output of each head i is then concatenated and linearly transformed to result in the final values:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

Here, W^O is the projection matrix for the multi-head output. The concatenated output is thus projected back to the model dimension, mixing information from all the individual attention heads. The Multi-Head Attention mechanism ensures that the model can capture a broader context, attending to different positions and focusing

on different patterns, thereby significantly improving the model's ability to handle complex tasks [4].

2.1.4 Cross Attention

Building upon the foundation of the Transformer's architecture and its self-attention mechanism, an additional attention mechanism, termed cross-attention, augments the model's capability. Cross-attention was originally used within the decoder part of the Transformer model, allowing for a richer interaction between the encoder and decoder, but has since been utilized in many other contexts [5, 12, 27, 28].

As previously described, each layer in the decoder has two sub-layers: the first one is the self-attention mechanism, and the second one is the cross-attention mechanism. In the original transformer architecture, the cross-attention sub-layer takes the output of the encoder stack as its input, in contrast to the self-attention mechanism that only operates over positions in the previous layers of the decoder [4]. This cross-attention mechanism allows each position in the decoder to map attention scores for each position in the input sequence.

Cross-attention serves as a bridge, connecting two different sequences and enabling the Transformer model to understand the relationships between them. For instance, in the originally proposed translation task [4], the cross-attention mechanism helps the model relate each word in the target language (decoder) with each word in the source language (encoder). This mechanism significantly improves the model's performance

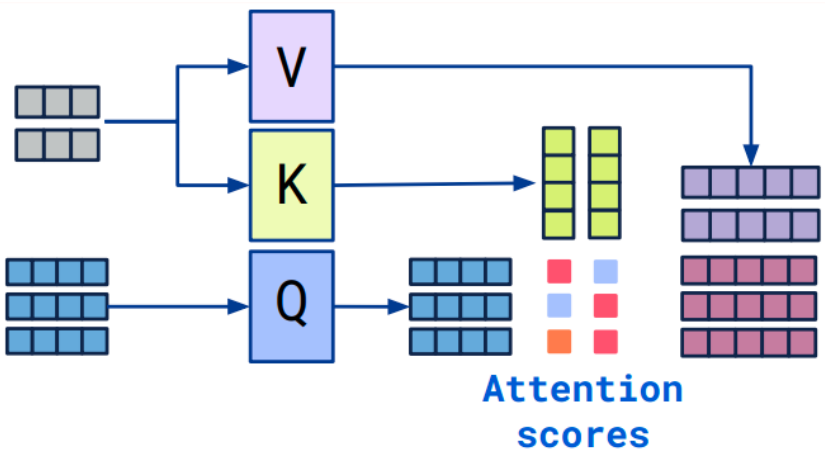

Figure 2.3: Visualization cross attention [5].

in sequence-to-sequence tasks by allowing it to consider the full context of the input sequence while generating each position in the output sequence [4].

The cross-attention mechanism, seen in Figure 2.3, reinforces the Transformer model's effectiveness, allowing it to capture both the internal relationships within a sequence and the relationships between different sequences. This ability to handle complex inter-sequence dependencies is a key factor behind the widespread success and adoption of Transformer models in various machine learning tasks [4].

2.1.5 Positional Embeddings

Positional embeddings are a key addition to Transformer models, enabling the incorporation of sequence order into the model's computations, as the self-attention mechanism in Transformers is inherently position-insensitive [4]. These embeddings are usually added or multiplied with input sequences before input to transformer

models.

The concept of positional embeddings was first introduced in the context of convolutional sequence-to-sequence learning [29], and later adopted in the Transformer model proposed by Vaswani et al. In the first proposed Transformer, a pre-defined sinusoidal function generates a unique embedding for each position:

$$PE(i, 2j) = sin(i/10000^{2j/d_{model}}),$$

$$PE(i, 2j + 1) = cos(i/10000^{2j/d_{model}}),$$

where i is the position index and j is the dimension index. Later research has explored many types of positional embeddings, including learned embeddings [6, 30, 31, 32]. A study by Wang and Chen found that Transformer encoders can learn local position information, while Transformer decoders for autoregressive language modelling can learn about absolute positions. However, the position embeddings of Transformer encoders do not capture much information about their absolute positions, especially in the case of BERT [6].

Figure 2.4 from Wang and Chen visualizes the cosine similarity of different learned position embeddings, with lighter colours denoting higher similarity. The choice of positional encoding function can significantly influence the performance of a transformer model on various tasks [6].

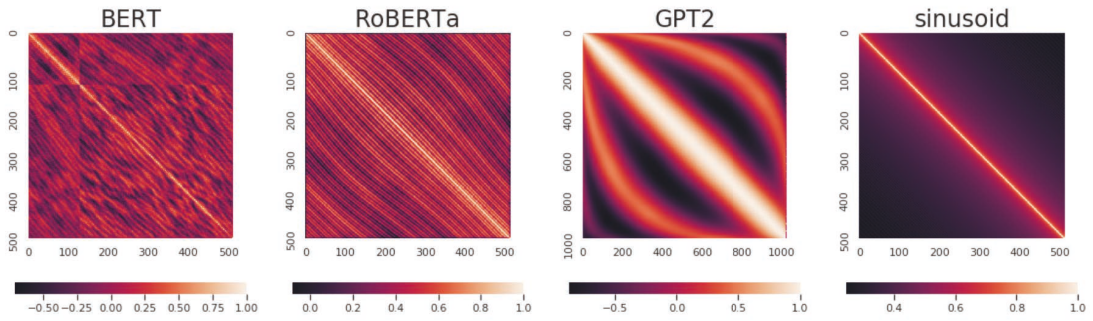

Figure 2.4: Visualization of cosine similarity of different position embeddings [6].

2.1.6 Transformers In Computer Vision

Transformers, which were originally devised for natural language processing tasks, have made a substantial impact in the field of computer vision [7, 27, 33, 34, 35]. The ability of transformers to capture long-range dependencies in data sets them apart from traditional convolutional neural networks (CNNs). This unique feature is particularly beneficial in image analysis tasks, where understanding the relationships between distant pixels can be crucial [7].

The introduction of the Vision Transformer (ViT) by Dosovitskiy et al. marked a significant shift in the application of transformers in the field of computer vision. ViT applies transformer mechanisms to sequences of image patches, effectively capturing dependencies across the entire image. This model architecture can be seen in Figure 2.5. At the time of its release, ViT demonstrated performance on par with, or even surpassing, traditional CNNs on many image classification datasets. Since the release of ViT, transformer-based computer vision models have dominated the leaderboards

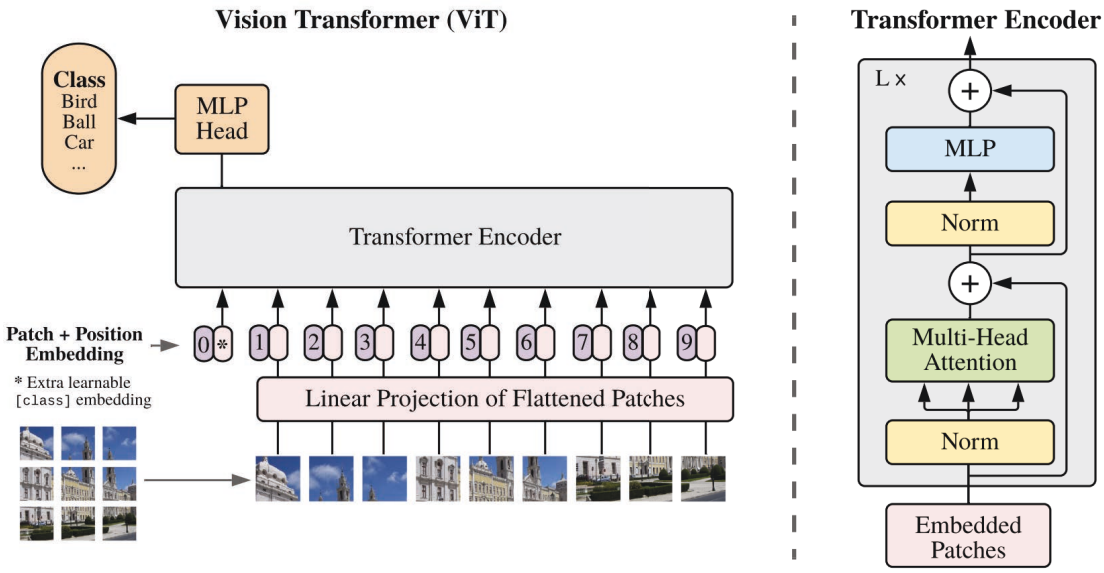

Figure 2.5: ViT Model Architecture [7].

of computer vision benchmarks like ImageNet.

However, the excellent performance of ViT was accompanied by considerable computational costs. To address this, LeViT, a hybrid neural network that combines principles from both CNNs and transformers, was developed [34]. LeViT introduced a pyramid structure, a common feature in traditional convolutional architectures, that uses pooling steps to decrease the resolution of activation maps, thereby enhancing computational efficiency. Furthermore, it incorporated a novel attention bias to integrate positional information, a key element in vision transformers. LeViT significantly improved the trade-off between accuracy and speed, demonstrating that at 80% ImageNet top-1 accuracy, it is five times faster than EfficientNet on a CPU [34].

In the ongoing research of integrating CNNs and transformers, a state-of-the-art

model, Lion, has been introduced [35]. Lion represents another leap forward in effectively leveraging the strengths of both paradigms. Not only does Lion create a unified architecture capable of handling various scales and regions of an image, but it also sets a high standard in performance. It demonstrates a substantial improvement in accuracy on multiple datasets, marking a new milestone in the evolution of vision transformers. This model builds on the foundations laid by earlier models like CrossViT, but goes further by providing a more efficient framework for image analysis [35].

The trajectory of these developments highlights the vast potential and continuing evolution of transformers in computer vision. Researchers continue to innovate, aiming to maximize the strengths of transformer models and mitigate their limitations, thereby enhancing their applicability and performance across a diverse range of visual tasks [35]. This ongoing exploration is set to yield further advancements in the field, laying the groundwork for the next generation of computer vision applications [33].

2.2 Bird's Eye View Perception

An accurate perception of the surrounding environment is critical for the operation of autonomous vehicles. In this regard, bird's eye view (BEV) representations in the 3D perception field have received substantial interest in recent years due to their ability to provide rich semantic information and precise localization [36, 37, 14]. These characteristics allow BEV to be used directly in numerous real-world applications

such as occupancy prediction, 3D object detection, motion planning and behaviour prediction [10, 38, 39, 19].

The BEV provides a straightforward approach for fusing information from different views, sensors, and time series. By representing the scene in a world coordinate system, BEV can seamlessly integrate multiple views of data into a comprehensive representation. It also allows for the accurate and seamless temporal fusion of sequential visual data and the integration of data from other common sensors, such as LiDAR and radar [14, 13, 12]. An example of camera-only BEV perception can be seen in Figure 2.6

However, the transformation from Perspective Views (PV) to BEV, known as inverse perspective mapping, presents substantial technical difficulties [9]. Early strategies relied primarily on geometric computations, which often fell short in complex real-world scenarios. The advent of data-driven techniques in computer vision, particularly deep learning, has heralded a crucial evolution in this field. These advanced methodologies offer significant improvements over traditional approaches, unlocking further possibilities for enhancing vision-centric BEV perception [10, 38, 12].

2.2.1 Homography Based BEV

Homography-based transformation, such as Inverse Perspective Mapping (IPM) [9], is a traditional methodology for converting PV to BEV. Leveraging geometric projection relationships, it is conventionally seen in vehicle backup cameras and dashboard 360°

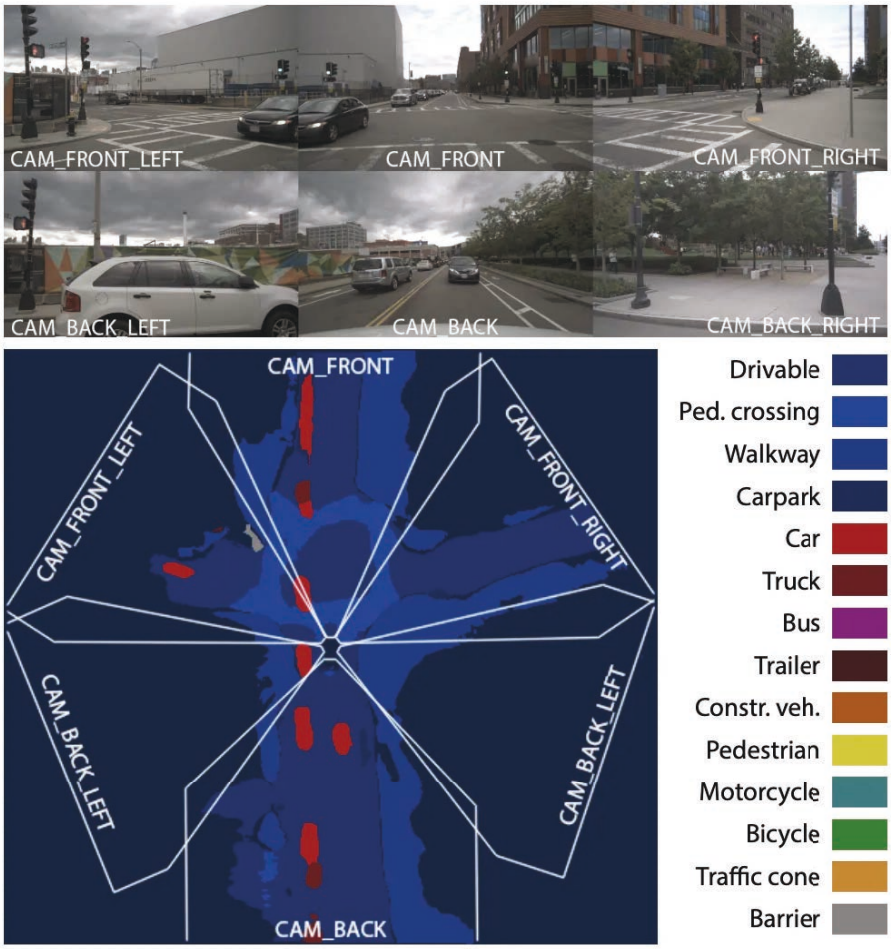

Figure 2.6: Example BEV model camera inputs (top) and prediction (bottom) [8].

views [40, 41]. Four example IPMs on vehicle dash cams can be seen in Figure 2.7.

However, this approach has its limitations. The most glaring is its dependence on the flat-ground assumption, often leading to inaccuracies when detecting objects above ground level, such as buildings or vehicles [42, 43, 44]. Although some techniques employ semantic information [45, 46], and Generative Adversarial Networks (GANs) [19, 47] to mitigate these issues, these solutions only partially resolve the distortions and do not sufficiently address the complexity of real-world 3D scenarios

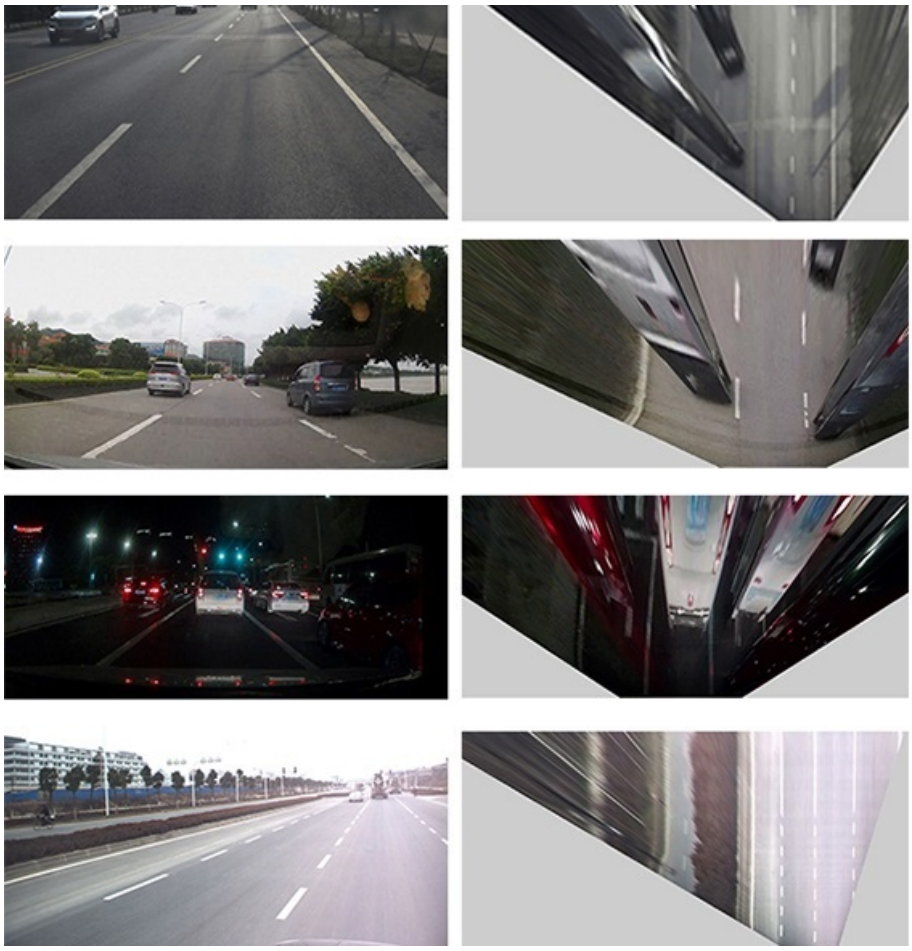

Figure 2.7: Example IPM: (left) input image (right) output IPM [9].

[44, 48]. The homography-based approach's inherent constraints necessitate further improvements to deliver more robust and accurate BEV mapping, despite its utility in creating interpretable BEV features [19].

2.2.2 Depth Based BEV

Depth-based methods for transforming PV to BEV primarily rely on depth predictions to go beyond the limitations of homography methods. Initially, these methods

were established on the concept of point-based view transformation, where depth estimation is directly utilized to convert pixels into point clouds scattered in continuous 3D space [49, 50]. They integrated techniques from monocular depth estimation and LiDAR-based 3D detection [51, 52]. Despite their advantages, these methods encounter several issues, such as data leakage, the complexity of the pipeline, and inaccurate pseudo-LiDAR generation [49, 53, 54, 55, 56]

Lift, Splat, Shoot (LSS) [10], illustrated in Figure 2.8, is a popular example of depth-based PV-to-BEV transformation, this model predicts a categorical distribution of depth alongside a context vector. The outer product of these elements determines the feature at each point along the perspective ray, approximating the real depth distribution [10]. "LSSlifts" each image individually into a frustum of features for every camera, subsequently "splatting" all frustums onto a bird's-eye-view grid Philion and Fidler. Despite sacrificing local spatial precision, these depth-based methods are more effective at covering large-scale scene structure information and are compatible with end-to-end learning paradigms for view transformation [10].

2.2.3 MLP Based BEV

Multilayer Perceptron (MLP) has been used extensively to map inputs to outputs of varying modalities, dimensions, and representations. This architecture has been utilized in various studies to transform between PV and BEV. It has seen application in various forms such as VED [57], VPN [58], FishingNet [59], PON [11], and STA-ST

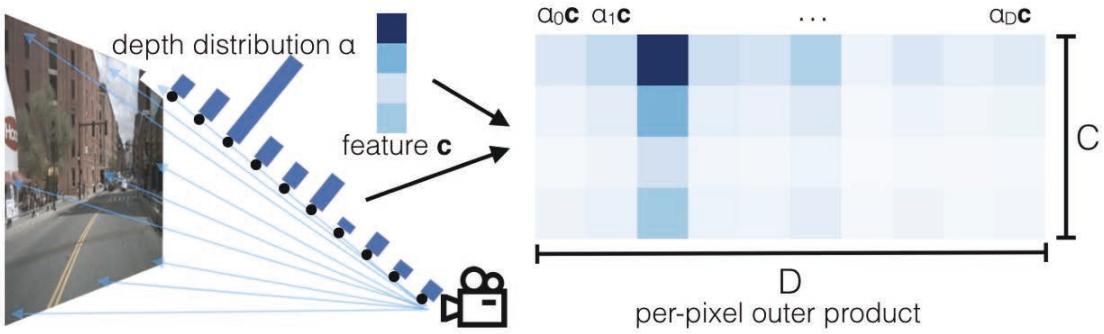

Figure 2.8: The "lift" step in "Lift, Splat, Shoot" [10].

[60], among others. These methods achieve view transformations by harnessing the MLP's capacity to learn implicit representations, thus enabling them to sidestep the inductive biases inherent in camera calibrations [19].

Additionally, certain methods like PON [11] and STA-ST [60] employ a feature pyramid [61] to extract image features at multiple resolutions, thus harnessing more spatial context and allowing for a focus on smaller objects. The PON architecture can be seen in Figure 2.9. Despite their potential, MLP-based PV-to-BEV methods exhibit certain weaknesses. Due to the lack of depth information, occlusion, and the unknown ground topology, the network needs a lot of vertical context to map features to BEV, leading to complications in the view transformation process [11].

The individual transformation and late fusion of multi-view images further hamper these methods from fully leveraging the geometric potential of the overlapping regions [19]. Although the MLP has been a valuable tool for PV to BEV transformations, it is important to note that the approach faces challenges, and has been generally

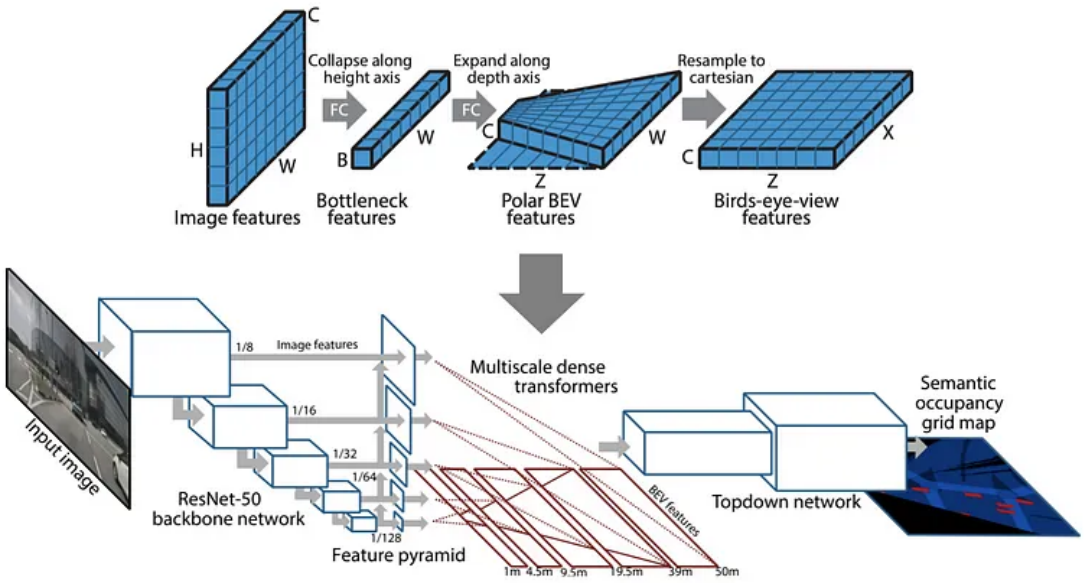

Figure 2.9: Architecture diagram of PyrOccNet (PON) with Dense Transformer [11].

outperformed by recently proposed transformer-based methods [19].

2.2.4 Transformer Based BEV

The application of transformers for PV to BEV conversion was first proposed by Tesla [1] and has since served as a catalyst for subsequent research in the autonomous driving domain [19]. Their pioneering technique crafted BEV queries via BEV positional encodings, this allowed for transformation from camera perspective features to the BEV plane, facilitated by cross-attention.

Building on Tesla's initial concept, the research community has produced a plethora of studies that have adopted and refined the cross-attention mechanism within transformers to effectively model perspective view transformation [14, 12, 62, 63]. Specif-

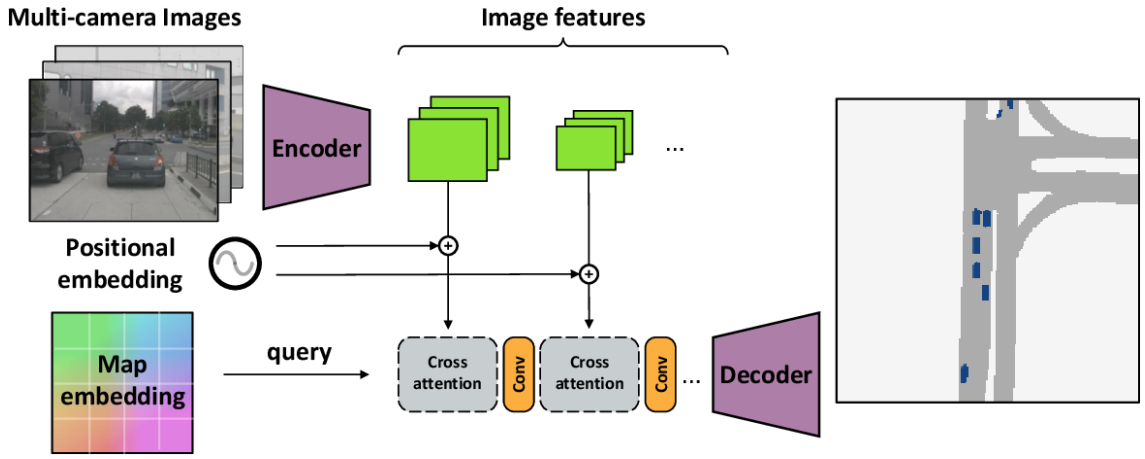

Figure 2.10: Architecture diagram of the Cross-View Transformer [12].

ically, the work of Cross-View Transformer [12] echoes Tesla's original concept, and subsequent studies have built upon its techniques [64, 65, 66], aimed at enhancing its performance and efficiency, pushing the envelope of transformer use in autonomous driving.

One key decision in these transformer-based methods revolves around the use of sparse versus dense queries. Sparse query-based approaches are used in object detection tasks, where each query represents a possible object in the scene; the transformer resolves each query as null or to a bounding box surrounding the object's occupancy. However, sparse query-based methods lack some of the geometric structure in the 3D representation making it challenging for dense prediction tasks like map segmentation [19, 67, 68]. In contrast, dense query-based techniques offer a comprehensive representation of the input space; each dense query resolves to x and y locations in the BEV plane. Dense query-based methods necessitate higher computational resources

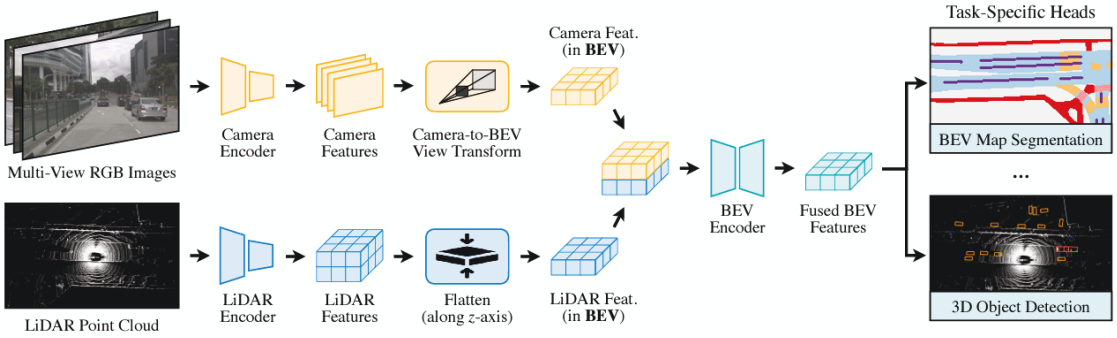

Figure 2.11: Architecture diagram of BEVFusion [13].

because many additional queries are required to represent the scene when compared with sparse query-based methods. To balance the trade-off, many researchers focus on developing more efficient cross-attention mechanisms, especially for high-resolution feature maps [14, 19, 64].

The incorporation of geometric information in the transformer framework is another crucial aspect. Although transformers have the capability to function without geometric priors, the inclusion of geometric data, like calibration matrices and depth information, can considerably enhance performance [12]. Calibration matrices play a pivotal role in defining the geometric projection from the 3D space to the image plane, assisting in the interaction between visual features and queries. Although not essential for PV-to-BEV conversion, depth information aids in geometric reasoning and aids in creating meaningful associations between queries and image features as seen in Figure 2.11 [56, 64, 69, 70].

The current research is trending towards larger, more generalized transformer models. These models have the potential to replace traditional components in the

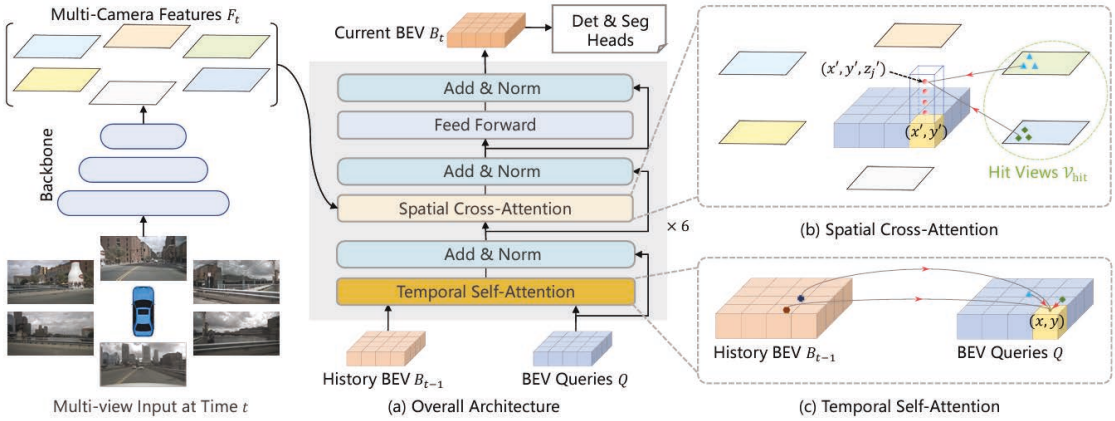

Figure 2.12: Architecture diagram of BEVFormer [14].

perception system, serving as feature extractors and detection heads [62, 69]. Temporal fusion, seen in Figure 2.12 is also becoming a crucial element in transformer-based methodologies. Incorporating temporal information, these methods generally outperform single-frame methods, yielding more accurate perception over time [14, 19, 69].

2.3 Collaborative Perception

The burgeoning field of Collaborative Perception represents a significant frontier in autonomous systems research. This section presents an overview of collaborative perception, focusing on the datasets and methods integral to this research domain. We pay particular attention to how these approaches leverage multiple sensory inputs and agent collaborations to enhance perception and decision-making capabilities in autonomous systems.

2.3.1 Collaborative Perception Datasets

Autonomous vehicle datasets have been widely used in collaborative perception research, comprising various sensory inputs, including cameras, lidar, and GPS [71], from multiple vehicles in a vehicle-to-vehicle (V2V) environment [15, 72]. Some datasets, such as those proposed in [73, 74], include infrastructure sensors, resulting in a vehicle-to-infrastructure (V2I) data model. Others, such as the dataset presented in [75], employ a vehicle-to-everything (V2X) model. The CoPerception-UAVs dataset [76] employs five images from five drones flying in formation. It is worth noting that these datasets are all sourced from CARLA [77] in Unreal Engine, a widely used open-source platform for simulating autonomous vehicles.

The HEV datasets sourced from our simulated environments are uniquely challenging in the field of collaborative perception, as the agents are equipped with only one or two cameras. Unlike previously proposed collaborative perception problems, the HEV task does not provide the agents with the transformation component of their pose. The unknown position of each camera view within the global coordinate frame adds a significant challenge to the semantic segmentation prediction task and other downstream tasks.

2.3.2 Collaborative Perception Methods

Collaborative perception has been explored in recent years, improving the capability of single-agent perception models [78, 76, 79, 80, 81]. In conventional collaborative

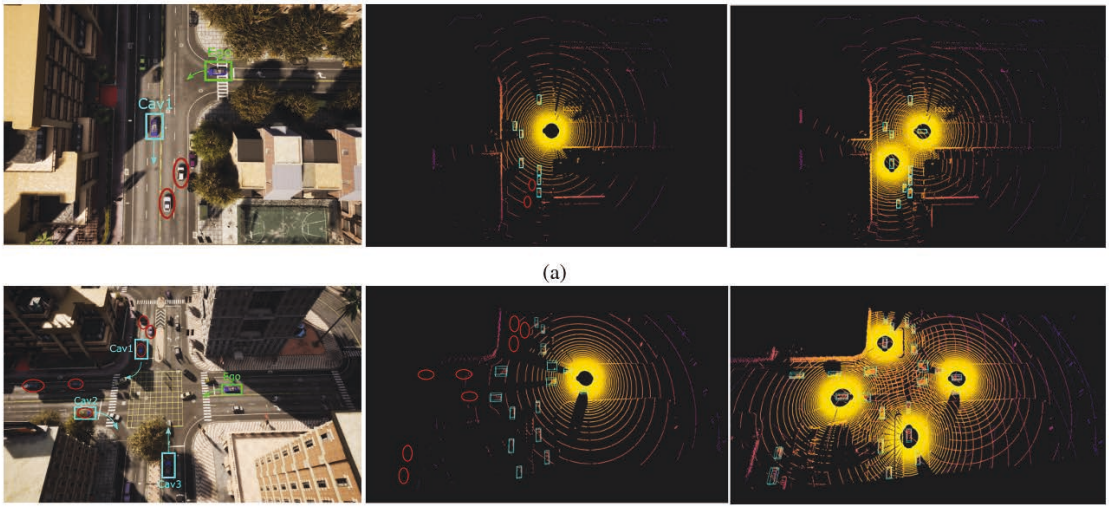

Figure 2.13: Derived from the OPV2V Dataset [15], this illustration provides two representative scenarios. The first (a) depicts a T-intersection, where roadside vehicles obscure the subject vehicle's sightline, shown via LiDAR and connected vehicle point cloud data. The second scenario (b) demonstrates how heavy traffic can cause LiDAR occlusions. The red markers identify vehicles unseen by the subject but visible to other connected vehicles.

perception, intermediate representations produced by sensors or neural networks from multiple viewpoints are propagated among a team of robots, such as a group of vehicles [76, 16] or a swarm of drones [81, 76]. The existing works commonly learn a collaboration module, produced by a Graph Neural Network (GNN) [81, 82], Convolutional Neural Network (CNN) [78, 83], or a Transformer [16, 76] to combine multiple robot intermediate representations. The FuseBEVT collaboration module introduced by Xu et al. can be seen in Figure 2.14.

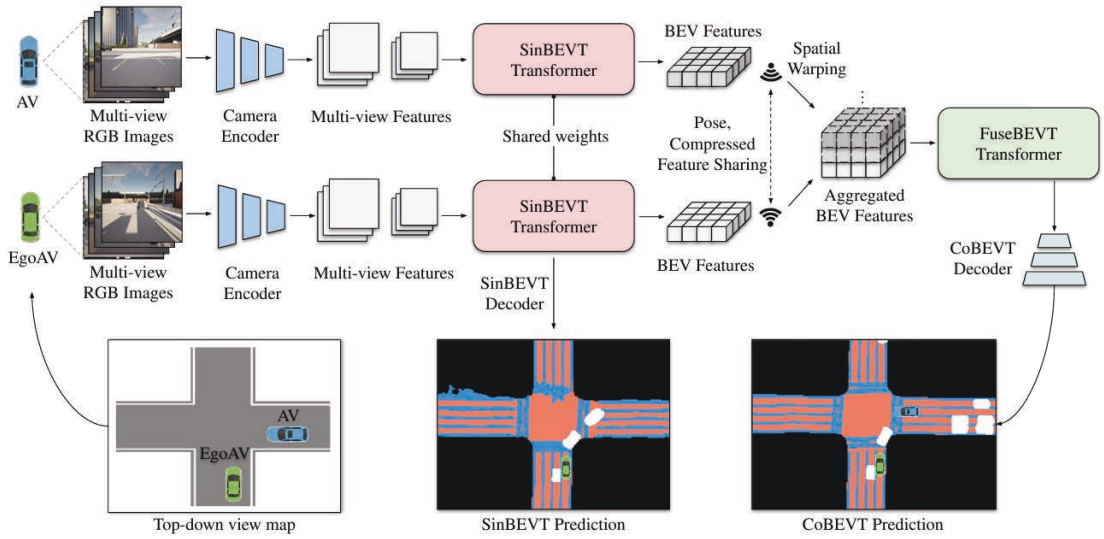

Figure 2.14: Architecture diagram of CoBEVT [16].

Prior research has focused on robots equipped with multiple sensors, requiring sensor data fusion on a per-agent basis before information exchange among agents [71]. However, in this work, we focus on robots with only one or two cameras and no additional sensors, making our approach more amenable to smaller, simpler robot swarms. Since we focus on simpler robots, we do not utilize a collaboration module, and instead fuse all camera views together in a single cross-attention module.

2.4 RL in Game Environments

Reinforcement Learning (RL) has emerged as one of the most influential methods for training intelligent agents in game environments. The basic premise of RL is that agents learn by interacting with an environment, taking actions, and receiving rewards

or penalties based on the consequences of those actions. Over time, agents aim to optimize their actions to maximize cumulative rewards. In gaming scenarios, this often translates to achieving high scores, optimizing in-game objectives, or defeating opponents.

Game environments, with their dynamic nature, have a myriad of possible actions, and the need for strategic planning, provide an ideal sandbox for RL algorithms. The controlled settings of games allow for extensive experimentation and fine-tuning, while the complexity inherent in many games offers rich challenges that push the capabilities of RL methods.

A notable example of RL's application in games is the ViZDoom platform [17]. ViZDoom offers a first-person shooter environment where agents, driven solely by visual inputs, must navigate intricate levels, avoid obstacles, and combat adversaries. By operating purely on pixels, the agents trained in this platform exhibit behaviours that are not only sophisticated but also constrained by real-world-like perceptions. Such settings challenge the RL agents to rely on spatial awareness, quick reflexes, and strategic planning to succeed, emphasizing the potential of vision-based RL methods.

The Obstacle Tower Challenge [18] provides another testament to the efficacy of RL in games. The challenge presents a procedurally generated tower with increasing difficulty levels. The primary task is to ascend the tower, requiring agents to handle a range of tasks from simple platform jumps to puzzle-solving. Like ViZDoom, agents in the Obstacle Tower rely predominantly on visual inputs, emphasizing perception-

driven decision-making. Furthermore, the procedural nature of the environment forces agents to generalize their strategies, moving beyond rote memorization to true adaptability.

2.4.1 ML-Agents

Unity, a real-time 3D development platform, has been widely adopted across various industries, including gaming, architecture, engineering, construction, automotive, and film [84]. The platform comprises both a rendering and physics engine, accompanied by a user-friendly interface known as the Unity Editor. Given Unity's historical emphasis on creating a versatile engine to cater to diverse platforms, developer expertise levels, and game genres, it has emerged as an ideal candidate for AI research simulations. The flexibility of Unity allows for the design of tasks ranging from straightforward 2D gridworld problems to intricate 3D strategy games, physics-based puzzles, or multi-agent competitive games. The Unity Editor further facilitates rapid prototyping and development of games and simulated environments [84].

The Unity ML-Agents Toolkit, an extension of the Unity platform, has been instrumental in advancing AI research. This toolkit provides the necessary tools and architecture to deploy learning algorithms within Unity environments. The Unity engine's generality and the ML-Agents Toolkit's capabilities have already played a pivotal role in fostering innovation within the AI research domain [84].

Cohen et al. introduced multiple Unity environments that serve as standard bench-

marks for evaluating the MA-POCA algorithm in multi-agent reinforcement learning settings. These environments, each with their own unique challenges and dynamics, serve as valuable testbeds for evaluating the adaptability, collaboration, and strategic planning capabilities of reinforcement learning agents in game-like settings.

Chapter 3

Herd's Eye View

This chapter outlines the methodology adopted in our research, focusing on two intertwined tasks: the PV-to-BEV task of Herd's Eye View (HEV) and the reinforcement learning task that benefits from the HEV perspective.

BEV transformation is a crucial task in computer vision, with a wide range of applications in diverse areas such as autonomous vehicles, video surveillance, and advanced robotics. It moves beyond basic object detection and classification, offering a deeper understanding of a scene by precisely delineating and classifying each pixel [20]. In the context of BEV, semantic segmentation proves to be particularly advantageous, paving the way for more effective planning and control in later stages due to its rich representation of the environment [**?**].

Alongside the semantic segmentation task, we also consider RL that utilizes the resulting HEV output as its state representation. This RL task demonstrates that

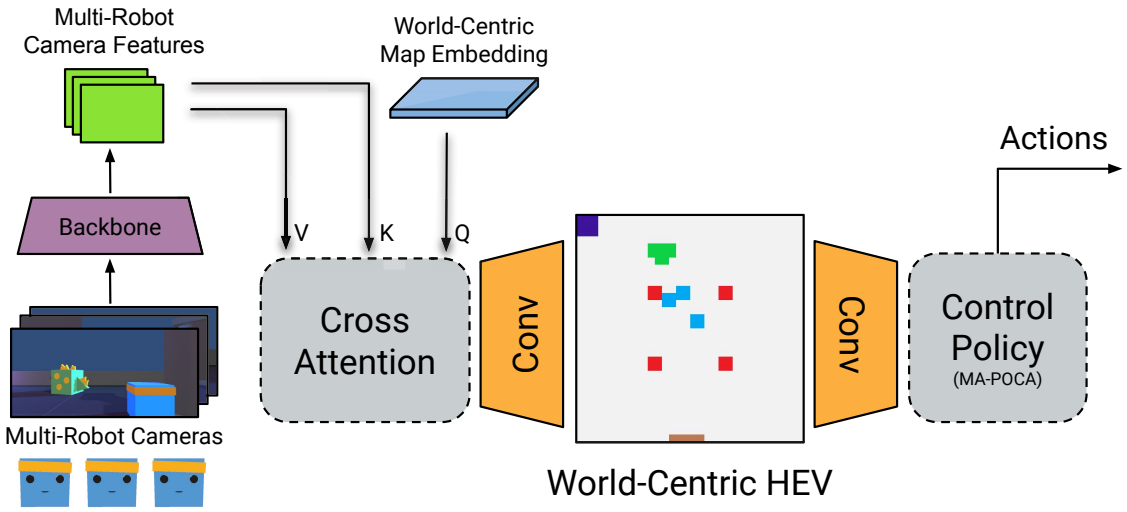

Figure 3.1: A visualization of the proposed HEV approach in the Dungeon Escape environment: Agent camera views are extracted via a backbone model, then combined in a cross-attention module, then decoded into a world-view semantic segmentation. The resulting semantic segmentation can be used as an observation for a swarm of robots.

the utilization of the HEV perspective could lead to more effective decision-making as compared to traditional BEV-based RL.

Our approach, seen in Figure 3.1, follows three steps:

1. Collect multiple views of the environment from robot cameras.

2. Use a collaborative perception model to obtain the HEV, the world-centric semantic segmentation of the environment.

3. Input the HEV to a Reinforcement Learning (RL) control policy to obtain agent control commands.

Through these steps, we aim to illustrate the potential of HEV in semantic segmentation and reinforcement learning tasks, offering a robust and comprehensive understanding of the environment that surpasses traditional BEV limitations. This chapter and the following chapter discuss the methodology and approach used to achieve these goals in detail.

3.1 Perception

In contrast to the commonly used Bird's Eye View (BEV), our research introduces the concept of Herd's Eye View (HEV). The HEV task seeks to go beyond the constraints of a single perspective, instead adopting an approach that benefits from a multiple perspectives – much like the vantage points available to a herd. This concept attempts to address some of the limitations inherent in relying solely on a single perspective, as is the case in the traditional BEV.

The BEV, by definition, presents an overhead, orthographic projection of the environment with respect to the ego-agent. While this offers a simplified view that eliminates the problems of perspective, it also introduces limitations, such as occlusions and limited perceptual data, due to the restriction to a single viewpoint. The HEV model leverages multiple camera views, each with its own unique pose and location, which can mitigate these problems and offer a more robust and comprehensive scene understanding.

In the Herd's Eye View (HEV) semantic segmentation task, we are given a set of n

monocular camera views, $(I_k, K_k, R_k)_{k=1}^{n}$ consisting of an input image $I_k \in \mathbb{R}^{H \times W \times 3}$, camera intrinsics $K_k \in \mathbb{R}^{3 \times 3}$, and extrinsic rotation $R_k \in \mathbb{R}^{3 \times 3}$ with respect to the agent base. The goal of the HEV task is to predict a binary semantic segmentation mask $y \in \{0, 1\}^{h \times w \times C}$ in the global coordinate frame, where C is the desired number of segmentation classes. The HEV task adds additional ambiguity to the well-studied BEV task as each camera view is at an unknown translation and orientation with respect to the global coordinate frame.

The multiple views sourced for the HEV provide a broader perspective, enabling a more complete understanding of the environment, that would be difficult for a single agent to achieve alone. The improved environment perspective could have potential applications in improving the performance of autonomous systems in complex environments.

Figure 3.2 demonstrates the disparity between the ego-centric view and the world-centric view. As seen in the left-hand side of the figure, ego-centric views fluctuate with the translation and rotation of the agent, thereby offering a dynamic perspective. Conversely, the world-centric view, displayed in the center, remains static and offers a more consistent perspective.

The main objective of the perception portion of our research is to establish a baseline HEV perception model to extract information from the multiple camera views and project them onto a fixed world-centric view. We propose a baseline perception model using the Cross-View Transformer (CVT) [12] and use semantic segmentation

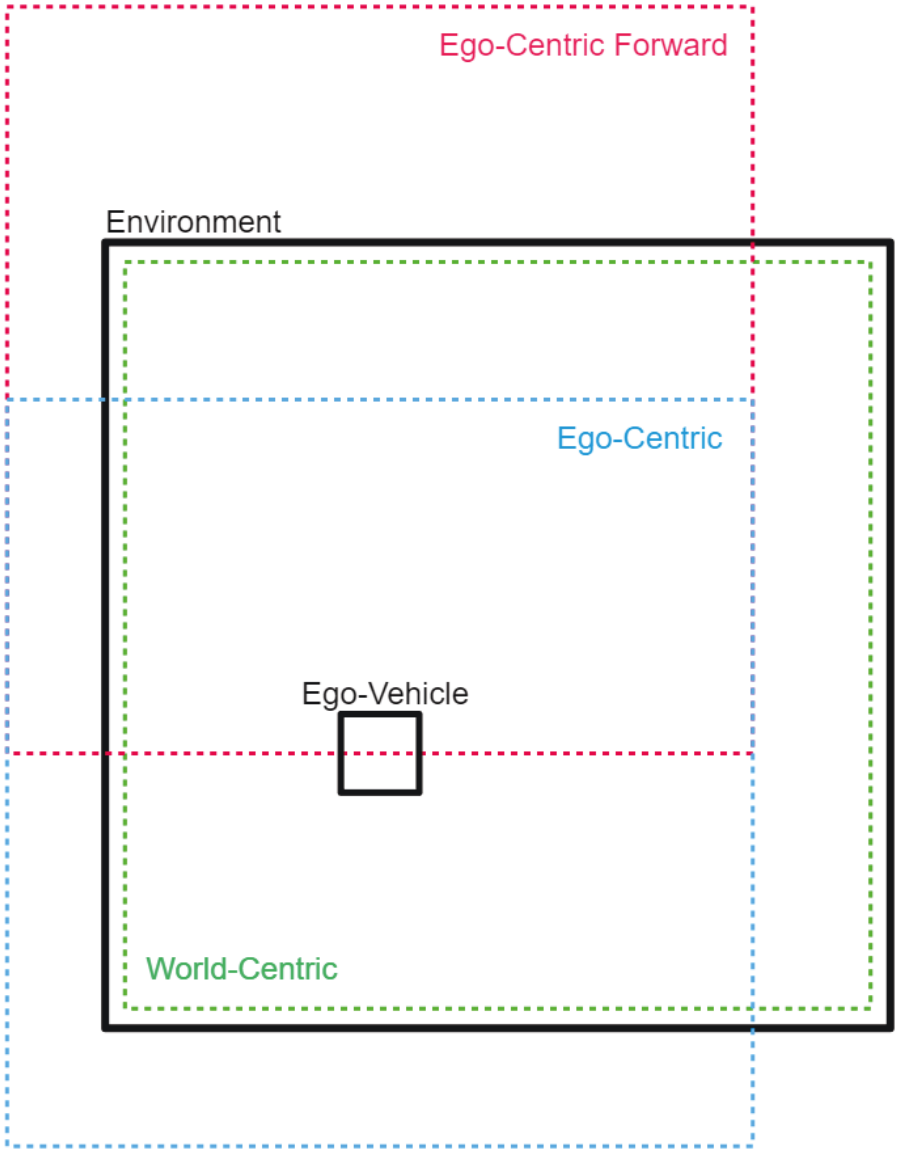

Figure 3.2: A visualization of the different map-view coordinate frames tested, ego-centric views move in accordance to translation and rotation of the agent, whereas the world-centric view does not.

as our downstream task. The Cross-View Transformer is a recent approach that uses a cross-view attention module, first proposed by [12], enabling the agents to reason

about their environment in an ego-centric coordinate frame. We extend the CVT model to further improve its accuracy and speed for the HEV use case. We name our baseline model the Herd's Eye View Cross-View Transformer (HEV-CVT).

3.1.1 Model Architecture

The Cross-View Transformer's (CVT) capability to reason about an environment given multiple camera views makes it a natural fit for our task. For our specific use case, we propose several enhancements to the original CVT model in order to optimize its accuracy and processing speed. Our extended model, the HEV-CVT, incorporates a world-centric map embedding. Furthermore, we tune the positional embeddings, output sizes, and the number of transformer layers to harmonize with our proposed HEV settings.

One advantage of the CVT architecture over other possible models lies in its attention mechanism. Transformer-based methods calculate attention between image pixel features and BEV x,y positions. This attention mechanism facilitates the understanding of associations between different spatial positions in the image perspective view and the BEV. It is this relationship that forms the basis of the mapping from image space to the BEV or HEV plane.

The standard CVT architecture can learn both the map and positional embeddings. This is an advantage over other methods which require explicit map and positional embeddings like GKT [85]. By learning these embeddings, the CVT can

adapt to the unique characteristics of the specific environment and task, providing a more flexible and adaptive approach which is compatible with both the BEV and HEV tasks.

The CVT's global attention mechanism represents another key advantage for our use case. It calculates attention between all image pixel features and all BEV locations, rather than just between corresponding pairs. In contrast to many other transformer methods, which seek to optimize the attention calculation by using known geometric information about the camera perspective with respect to the BEV, the CVT does not require such explicit geometric information. This global attention mechanism is particularly relevant to the HEV problem, where the environment is observed from multiple, potentially unrelated, camera perspectives.

Conventional PV-to-BEV approaches rely heavily on known geometric information about the camera perspective with respect to the BEV plane, which is less suitable for our task. The HEV task introduces an additional level of complexity as the position and orientation of each camera with respect to the HEV are not known. As such, we cannot make the same assumptions about the geometry of the scene, and must instead rely on a model like the CVT, which can handle this inherent uncertainty.

Chapter 4

Experiments and Results

4.1 RL Environments

In this research, we employ Reinforcement Learning (RL) as a practical and effective measure to gauge the complexity of the planning and control problem. Using RL enables us to systematically evaluate the performance differences that arise from various sensor setups. However, it is important to underscore that while RL provides useful insights into these differences, it does not offer an exhaustive evaluation. Consequently, our conclusions are grounded in these RL experiments, but we acknowledge that there may be other aspects to consider for a more comprehensive understanding of the sensor approaches.

To gather our datasets for both the HEV semantic segmentation task and the MARL task, we leverage identical Unity simulation environments. The use of Unity

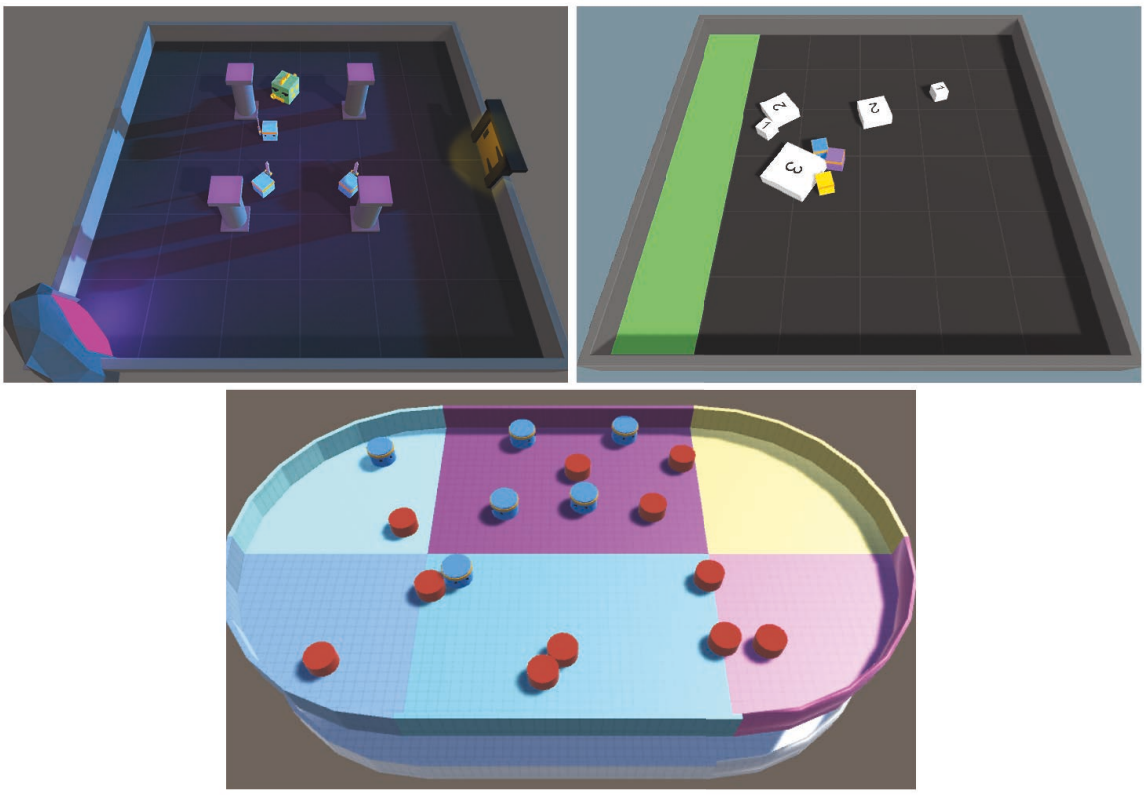

Figure 4.1: Images of the three environments used to test the HEV collaborative perception and reinforcement learning algorithms. The top left is the Dungeon Escape environment. The top right is the Collaborative Push Block environment. The bottom is the Planar Construction environment.

allows us to create versatile and complex virtual environments where agents can learn and be evaluated under controlled yet challenging conditions. We conduct our simulations across three distinct ML-Agents environments, seen in Figure 4.1, each presenting a range of unique challenges and opportunities for testing our HEV approach. These environments, discussed in the following subsections, have been designed and selected to test and validate the capabilities of our method under various

conditions.

4.1.1 Collaborative Push Block

The Collaborative Push Block environment, seen in Figure 4.2, developed within the Unity ML-Agents framework, initially proposed by Cohen et al. [86], establishes an arena for the study of multi-agent reinforcement learning techniques. This environment contains a grid area where the agents and blocks are spawned at random locations outside of the goal area.

The goal area, characterized by a green colour, is randomly assigned to a side of the grid square in each episode, adding to the unpredictability and diversity of the task scenarios. The core objective for the agents is to push white blocks into this goal area. The task emphasizes the need for cooperative behaviour among the agents, as the block sizes vary, requiring different numbers of agents for movement based on the size.

Blocks within the environment are categorized into three distinct sizes, labelled as "size" 1, 2, and 3. The size of the block directly corresponds to the number of agents required to push it, with larger blocks needing more agents. This variability in block size presents a unique challenge for the agents, as they must decide whether to collaboratively push larger blocks or individually maneuver smaller ones, leading to the development of coordinated strategies.

When a block is successfully pushed entirely into the goal area, the agents actively

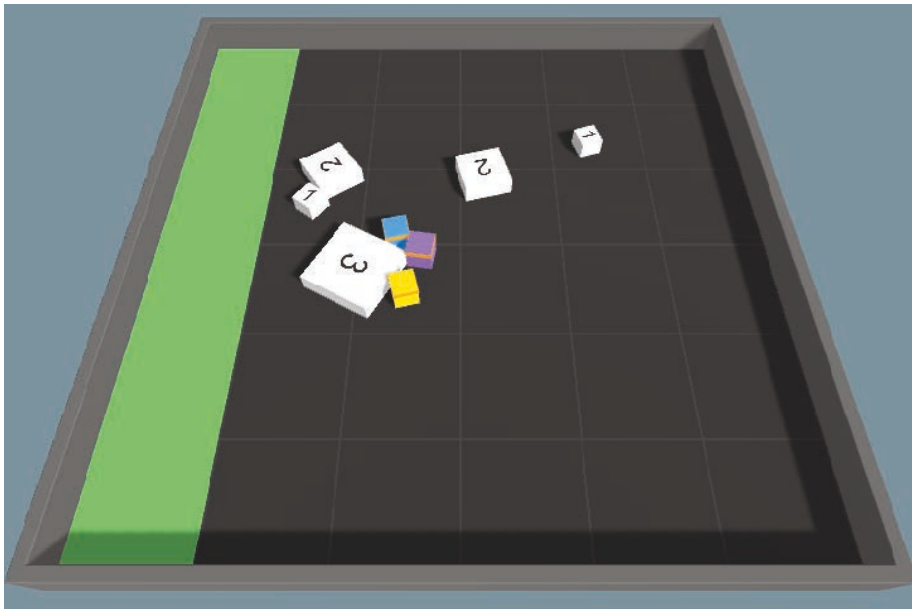

Figure 4.2: Push Block

involved in pushing that particular block are rewarded with a positive reinforcement signal. This reward serves to encourage the cooperative behaviour required to accomplish the task. To ensure efficiency and avert inaction, a slight negative reward is applied at each timestep, thereby promoting rapid completion of the task.

Each agent within this environment operates individually in terms of observation collection and action application. Despite this, all agents train the same MA-POCA policy. This policy allows for the development of synchronized behaviours and cooperative strategies among the agents, despite their independent operation.

The original Collaborative Push Block environment by [86] employed a sophisticated observation framework. Each agent was equipped to cast 21 rays across a 180° arc to perceive various elements in the environment: other agents, walls, the goal,

and block types. The observation system operated on two distinct levels. One set of rays was cast higher, enabling the agents to observe over obstacles and locate the goal or walls. A second set of rays, cast at the agent level, was employed for detecting nearby objects like blocks and other agents.

This dual-level observation system provided a comprehensive perspective for the agents to navigate and strategize effectively. However, it is worth mentioning that we have considerably altered this observation structure in our study, the details of which will be covered in the subsequent Model Training section.

4.1.2 Dungeon Escape

Introduced by Cohen et al. [86], the Dungeon Escape environment, seen in Figure 4.3, is a task within the Unity ML-Agents framework. This environment tests multi-agent reinforcement learning models by enforcing strategic cooperation among the agents. The Dungeon Escape environment is confined within walls, simulating a classic dungeon setup. Within this bounded area, the agents and the dragon are spawned at random positions and orientations at the beginning of each episode, promoting a diverse set of scenarios and an element of unpredictability.

Within the Dungeon Escape environment, there are three agents and a green dragon, the latter being the holder of a key. The environment also features a portal and a door, both randomly placed within the dungeon at the onset of each episode. The primary objective for the agents is to escape the dungeon through the door.

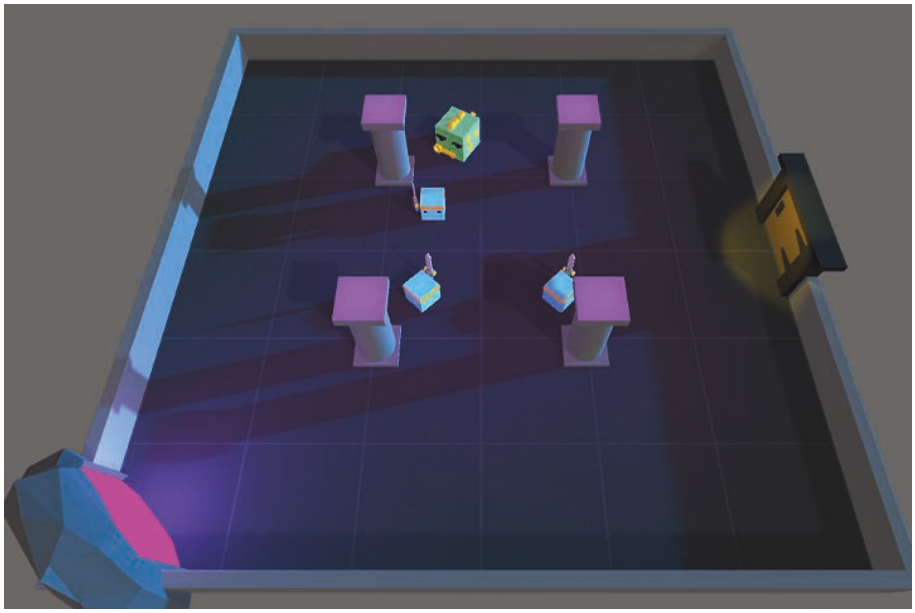

Figure 4.3: Dungeon Escape

However, the door requires a key to open, which is initially held by the green dragon. An agent must sacrifice itself to defeat the green dragon, in which process, both the dragon and the agent disappear from the environment. This action results in the key being dropped, which the remaining agents must then retrieve and use to escape the dungeon.

The episode ends when either an agent with the key successfully reaches the door or the green dragon makes its way to the portal. To receive a positive group reward, at least one agent must make it through the door.

In the original environment, agents collect observations using 15 rays cast in a 120° arc in front of them, allowing them to navigate the dungeon and respond to the presence of other agents and the dragon. However, within the context of our study,

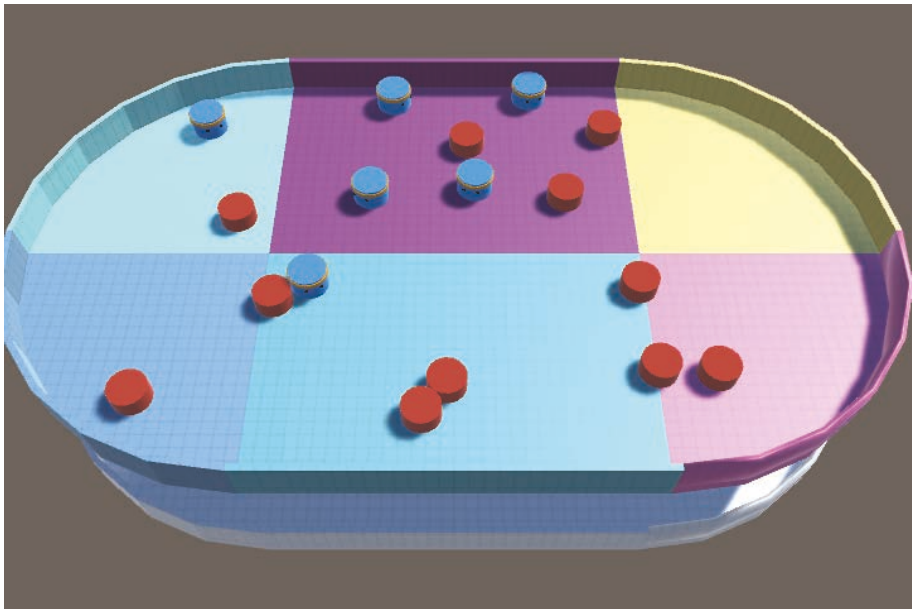

Figure 4.4: Planar Construction

the agent's observations have been significantly altered, which will be discussed in more detail in a subsequent Model Training section.

4.1.3 Planar Construction

In the Planar Construction environment, seen in Figure 4.4 six agents collaborate to reposition red pucks into desired locations or patterns. This environment's foundation rests on the Orbital Construction (OC) problem, which has been studied in multiple prior works, in multiple simulators [87, 88, 89, 90, 91]. We base our planar construction environment specifically on the Strickland et al. implementation in CWaggle.

In the original RL formulation, the environment comprises of a square enclosed

space, filled with circular agents and circular puck objects. The agents, capable of moving freely, can influence the positions of the pucks by colliding with them. This environment also featured a scalar field grid projected onto the floor surface, with values ranging from 0 to 1. This scalar field encoded information relating to the final shape to be constructed [88].

In our recreation of the Planar Construction task within the Unity ML-Agents framework, the scalar field has been replaced with a Grid Sensor. Instead of agents detecting the current scalar field values around them, the desired positions and current positions of pucks are encoded into the grid-sensor. We achieve this by placing a second "goal grid" underneath the planar construction environment. In the goal grid area, the pucks are in the desired positions, so the agents above them can detect where the goal areas are without the physics of the goal pucks interfering.

The recreated Planar Construction environment challenges the six agents to navigate the environment and effectively manipulate the red pucks to reach either random or static designated positions in each episode. This process demands cooperative effort and strategic alignment among the agents, adding to the complexity and challenge of the task. The unpredictability of these positions requires the agents to continually adapt their strategies, making the Planar Construction environment a comprehensive testbed for advancing research in cooperative multi-agent reinforcement learning.

4.2 Model Training

4.2.1 Reinforcement Learning

In order to explore the utility of different agent perspective views for collaborative tasks, we designed a series of Unity MARL environment experiments. The underlying hypothesis is that we want to test whether agents trained with a global perspective view could demonstrate superior efficiency and effectiveness when compared to those trained with ego-centric perspective views. As seen in Figure 4.5, the following configurations of agents where benchmarked:

1. Ego-Centric agent: These agents were equipped with a 32x32 semantic segmentation grid of the objects surrounding the agent at all times. This grid adheres to the translation and rotation of the agent, providing the agent with a coherent perspective that aligns with its motion.

2. Ego-Forward agent: This variation of the BEV-based agents maintains the same 32x32 semantic segmentation grid, but it is offset forward from the agent. This allows the agent to have a more forward-facing view of its environment, which could potentially result in improved anticipation of the objects in its path.

3. World-Centric agent: In contrast to the BEV-based agents, the HEV-based agents utilize a 32x32 semantic segmentation grid centred at the world's origin and scaled to cover the entirety of the world space. The grid remains stationary and does not rotate or translate with the agent. This necessitates an additional

encoding to the semantic segmentation, enabling the reinforcement learning policy to interpret the agent's relative location with respect to obstacles and objectives within the world space.

All the agents in our experiments were trained using the MA-POCA policy. MA-POCA was chosen due to its established efficacy in Unity cooperative multi-agent settings. This algorithm integrates a centralized value function that utilizes the observations and actions of all agents, enhancing the ability to coordinate policies among agents. This value function serves as a critic during policy updates, providing a form of guidance to the decentralized policies. We ensure the total area of observation between each type of agent is equal so they can be compared on an equal playing field. Exact training parameters can be seen in Table 4.1.

By comparing the performance of these different agent configurations, we aim to uncover the relative benefits and drawbacks of each perspective view. This will not only inform our understanding of how the choice of perspective impacts reinforcement learning performance but will also provide a benchmark for future studies in this area.

4.2.2 Grid Sensor

An integral part of our environment setup is the collection of the HEV ground truth. For this purpose, we developed a custom fork of MBaske's Unity Grid-Sensor Library [92] which allows the collection of HEV world-centric grid-sensors. The modified library allows us to create grid-based observations for the agents in the environment,

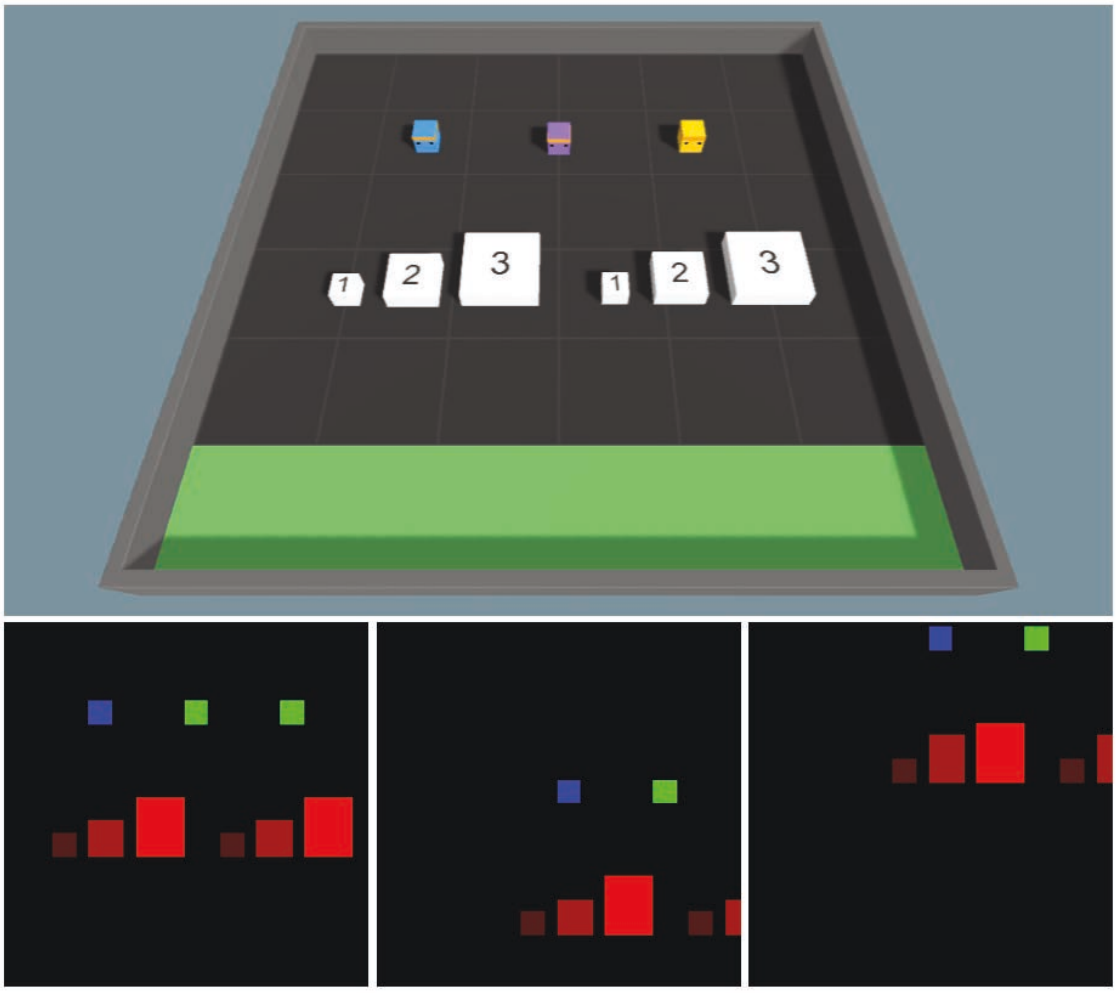

Figure 4.5: Example scene and corresponding agent observations from the Collaborative Push Block environment. The top image shows a debug camera (not available for agent observation). The bottom left shows the HEV world-centric observation of the blue agent. The bottom middle shows the BEV-centric observation of the blue agent. The bottom right shows the BEV-forward observation of the blue agent. Blue is the controller agent, green is ally agents, and red shades are differently-sized push blocks.

Table 4.1: Parameters for the simulated environments with HEV inputs.

	Robots	Robot Cameras	Grid Size
Collaborative Push Block	3	1-left, 1-right	32x32
Dungeon Escape	2-3	1-left, 1-right	32x32
Planar Construction	6	1-forward	32x64

facilitating the conversion of the agent's observations to the required HEV representation.

A notable feature of our custom fork of the grid-sensor library is its capability to differentiate between the controller agent and the other agents. This feature, which we will call the self-perception mechanism, permits the controller agent to perceive itself differently from the surrounding agents. This distinction is crucial in enabling the controller agent to take actions based on its unique pose within the environment.

One key distinction between the ego-centric and world-centric based agents lies in the positioning of their grid-sensor and their action perspectives. For instance, "forward" for a world-centric agent always corresponds to the north, but for an ego-centric agent, "forward" is relative to their current orientation.

All agents are trained using ground-truth sensors, calculated using bounding boxes individually tuned to each object in the environment. The resolution of the grid-sensor

is carefully adjusted to match the complexity and size of the environment, facilitating a balance between perception accuracy and computational efficiency. Detailed specifications can be found in Table 4.1.

4.2.3 Perception

The collection of vision data for the HEV-CVT model forms the backbone of our research. The data is obtained by training a unique reinforcement learning model for each specific observation type ego-centric, ego-forward, and world-centric for each environment. Thus, a total of 9 distinct models are trained, which are then deployed across different environments for our RL experiments and perception data collection.

We utilize an RTX 3090 GPU to train these perception models. Once we have trained an MA-POCA model, we run it in the environment and collect training data every ten time steps, capturing a snapshot of the agent's view, camera intrinsics, and the corresponding ground truth HEV and BEV each time.

To streamline the data collection process and boost the variety of our dataset, our models are concurrently deployed in 8-12 different environments. This simultaneous data collection significantly accelerates the training process, encapsulating a wide range of situational and environmental dynamics.

Our extensive dataset comprises approximately 30,000 instances per environment and agent type, these are collected across different runs, promoting robustness and diversity. To enhance our model's generalization capabilities for unseen scenarios, we

reserve a validation dataset that is 10% the size of the training data, sourced from unique seeds.

4.2.4 Evaluation Metrics

The dual nature of our research—focusing on both vision and reinforcement learning (RL)—requires an evaluation process that measures the effectiveness of our models in these two diverse domains. To this end, we have chosen metrics that are relevant to each task and capable of providing insightful analysis of our model's performance.

On the vision front, the Cross-View Transformer models' performance is quantified using the intersection over union (IoU) score. A widely used metric in the field of semantic segmentation, IoU score offers a reliable measure of the model's accuracy by quantifying the overlap between the predicted and ground truth segmentation. The choice of IoU reflects its ability to provide an understanding of how well our model is performing the semantic segmentation task.

For the RL task, we opt to measure agent performance via the length of an episode. This decision is primarily driven by the characteristics of our RL environments where the speed of task completion differentiates the performance of agents. By focusing on episode length, we emphasize the efficacy and efficiency of our agents in achieving their objectives, hence providing a clear depiction of the RL model's performance.

Although alternative metrics such as cumulative or mean reward are often used in RL, in our case they primarily reflect minor negative rewards assigned per time

step. In our particular context, these metrics would provide less insight into an agent's efficiency over the course of the episode rather than the agent's capability to complete tasks swiftly.

By employing these specific evaluation metrics, we aim to provide a comprehensive and insightful analysis of our model's performance, that encapsulates the complexity and distinct aspects of our research domains.

4.3 Collaborative Perception

We test the HEV-CVT model's ability to accurately localize the position of each agent based on the overlap of six camera frames, which are located at unknown rotations and positions. The cameras are recorded at resolution 480×224, and we use the camera intrinsics of a Raspberry Pi Camera Module 3 with a wide-angle lens. We compare the performance of the baseline CVT model on world-centric, ego-centric, and ego-centric-forward coordinate frames.

4.3.1 Experiment 1: Collaborative Push Block

In the Collaborative Push Block environment, three agents collaborate to push different-sized blocks into a goal area. The three agents are equipped with two forward-facing cameras angled positively 45 degrees and negatively 45 degrees from the center of the agent's front face. The goal for this perception task is to use the six different camera views available from all agents to predict the occupancy of all agents and the goal

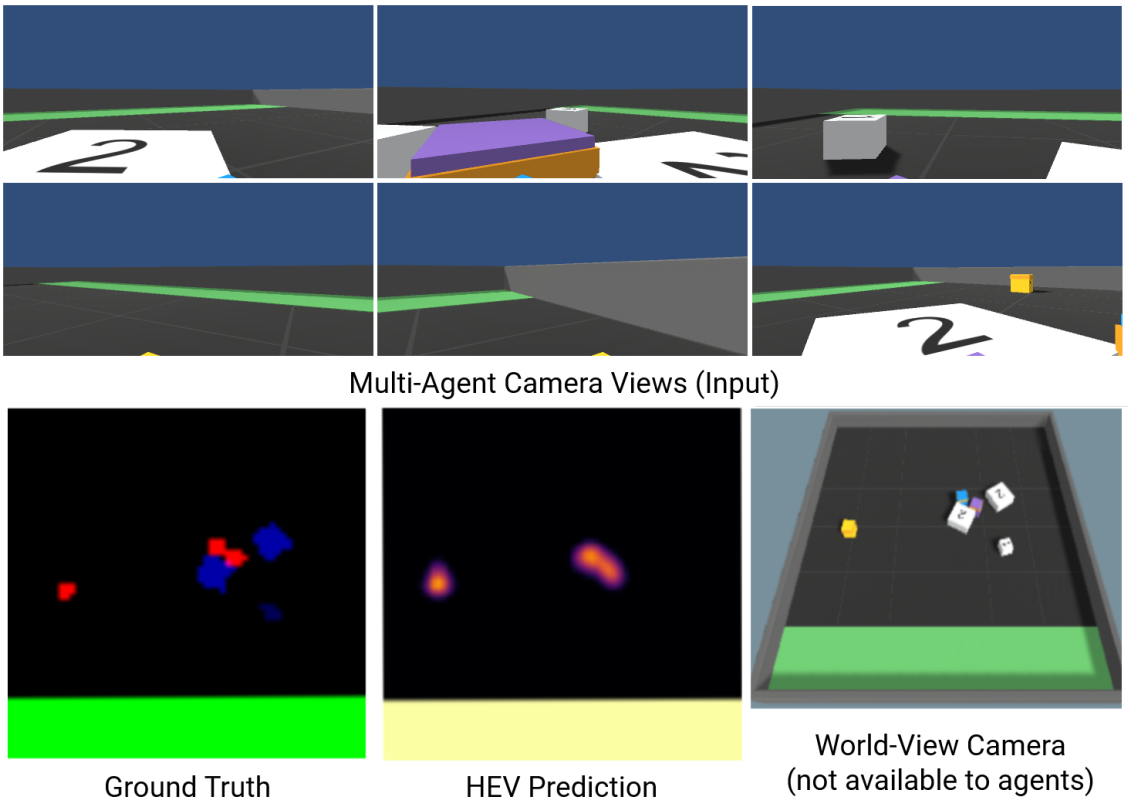

Figure 4.6: (left) example six input images to HEV model, (right) HEV model prediction of agent locations and dragon agent.

area.

We find the HEV-based model is significantly more effective for occupancy prediction in this environment, boasting a 96.94% IoU score. A significant contribution to the HEV prediction advantage is the goal area, which always exists on one side of the predicted area as seen in the sample prediction Figure 4.6. The bright yellow area on the heat map demonstrates the model's high confidence in the goal area's location within the HEV, an easy landmark that the model can use to help orient itself.

The BEV model also successfully predicts occupancy in this environment but at

a lower IoU score of 63.87% and 64.22% for the ego-centric model and ego-forward model respectively. The BEV models must predict occupancy with respect to the agent, meaning they must predict the exact offset and angle of the goal area, which our results demonstrate to be more challenging than the HEV version of the task.

4.3.2 Experiment 2: Dungeon Escape

In the Dungeon Escape environment, three agents are equipped with two angled forward-facing cameras and are tasked with predicting the occupancy of the dragon, agents and key. We choose not to train the model to predict the portal as this information is not necessarily required to complete the RL task. Additionally, the model implicitly predicts the door by aligning the world-view to always place the door on the east side. This environment poses the additional challenge of an agent sometimes being disabled. In order to collect the key needed to escape the environment, an agent must sacrifice itself to take down the dragon, meaning sometimes the HEV or BEV prediction is made from only four images instead of six. Camera dropout has been tested in past autonomous vehicles research [10, 12], and we find similarly the HEV-CVT is resilient to camera dropout.

The ego-centric and ego-forward models perform worst in the Dungeon Escape environment of all the environments with IoU scores of 13.47% and 26.07%. These low IoU scores show the BEV models fail to converge on a robust prediction policy, unlike the HEV model. The BEV models are trained with the same amount of data

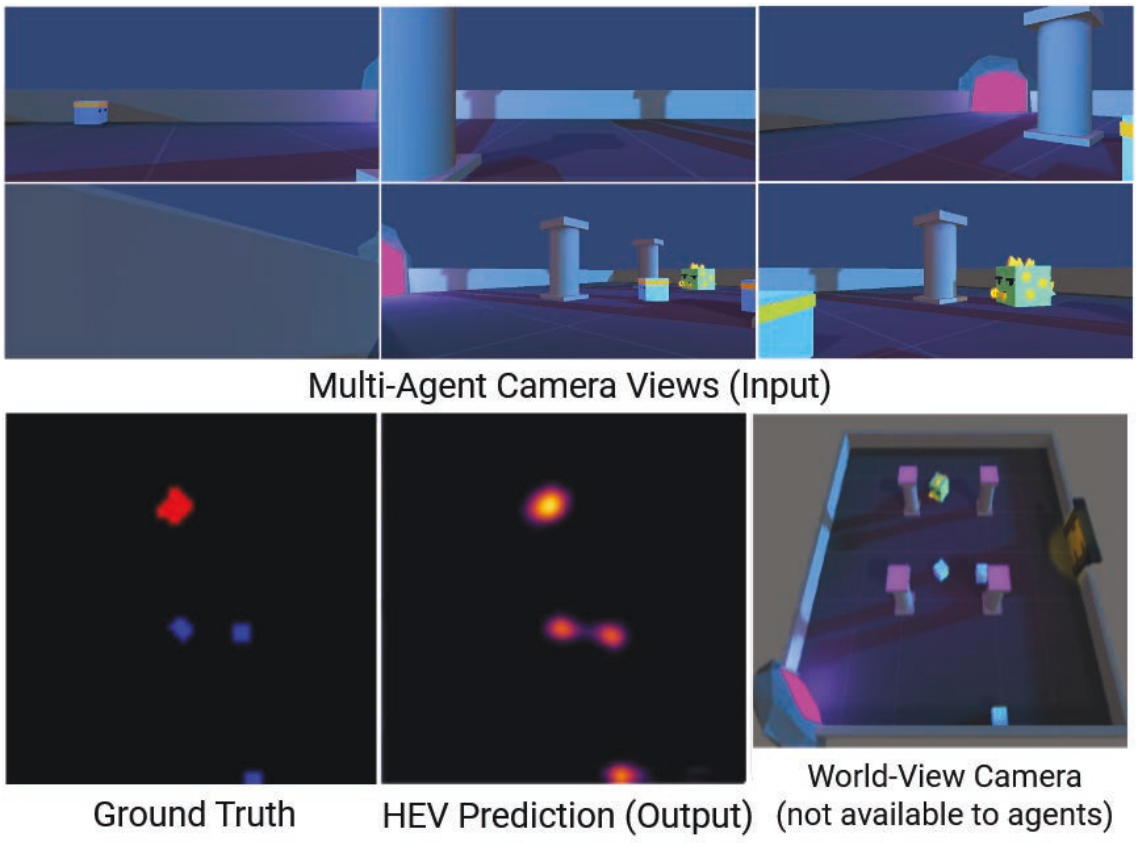

Figure 4.7: (left) example six input images to HEV model, (right) HEV model prediction of agent locations and dragon agent.

as the HEV policy. We additionally train various sized smaller and larger models with multiple hyper-parameters combinations and report the highest achieved IoU scores. These results suggest the Dungeon Escape environment is the hardest for the BEV model due to its lack of landmarks available. If the HEV ego-agent input has some views without landmarks or important information it can choose to ignore them, but if the BEV ego-agent's view does not contain landmarks it can be more difficult to determine where it is located and the contents of its BEV. In other words, having a

poor perspective view is more detrimental to the BEV models compared to the HEV models.

The HEV model consistently makes a good occupancy prediction, but there are some rare scenarios where the input images provided do not contain enough information (e.g. dragon is not in view) to make accurate predictions. This disadvantage occurs significantly less frequently in the BEV datasets due to the camera placement on the ego-agent. Despite this disadvantage, the HEV model converges on a sufficient policy in the Dungeon Escape environment. The HEV-CVT model scores an IoU of 43.53%, an example prediction from this model can be seen in Figure 4.7. The model accurately predicts the location of agents, the dragon and the key, each appearing as a unique size in the HEV.

4.3.3 Experiment 3: Planar Construction

In the Planar Construction environment, six agents are equipped with a single forward-facing camera and are tasked with predicting the occupancy of all pucks. The previous ML-Agents environments were not as challenging for the CVT models due to the availability of objects in predictable locations. The Planar Construction environment presents a more complex challenge as we randomly change the colouring of all six wall and floor components at every time step of the environment during data collection. The wall colours provide a helpful aid for the model to piece together the locations of the pucks as they are the only overlapping feature between images besides the pucks

themselves. We alter the environment to add the additional challenge of random wall colours because it's possible that the model would just learn to associate specific wall colours with their spatial positions, a task that is considerably less complex. Despite this additional challenge the HEV-CVT and ego-centric BEV policies converge.

The Ego-Forward CVT model performs the worst of all tested models on the Planar Construction environment with an IoU score of 10.16%. We think the Ego-forward model performs the worst of all models because the forward viewpoint is less advantageous in the rectangular environment. Often, the extra forward range provided by the shifted forward BEV is wasted on regions outside the environment with no information. In contrast to the forward model the ego-centric BEV CVT model converges on an adequate policy with an IoU score of 35.45%.

Both the HEV and BEV models are imperfect and sometimes hallucinate pucks into the prediction or do not include pucks in view in the prediction. Interestingly, when the model does not have a view of a region of the stadium, it will often predict a low confidence interval for the entire unseen area, demonstrating its understanding of the environment. The HEV-CVT model has the best IoU score of all models tested in the Planar Construction environment, with a score of 48.37%. A sample prediction from the model can be seen in Figure 4.8

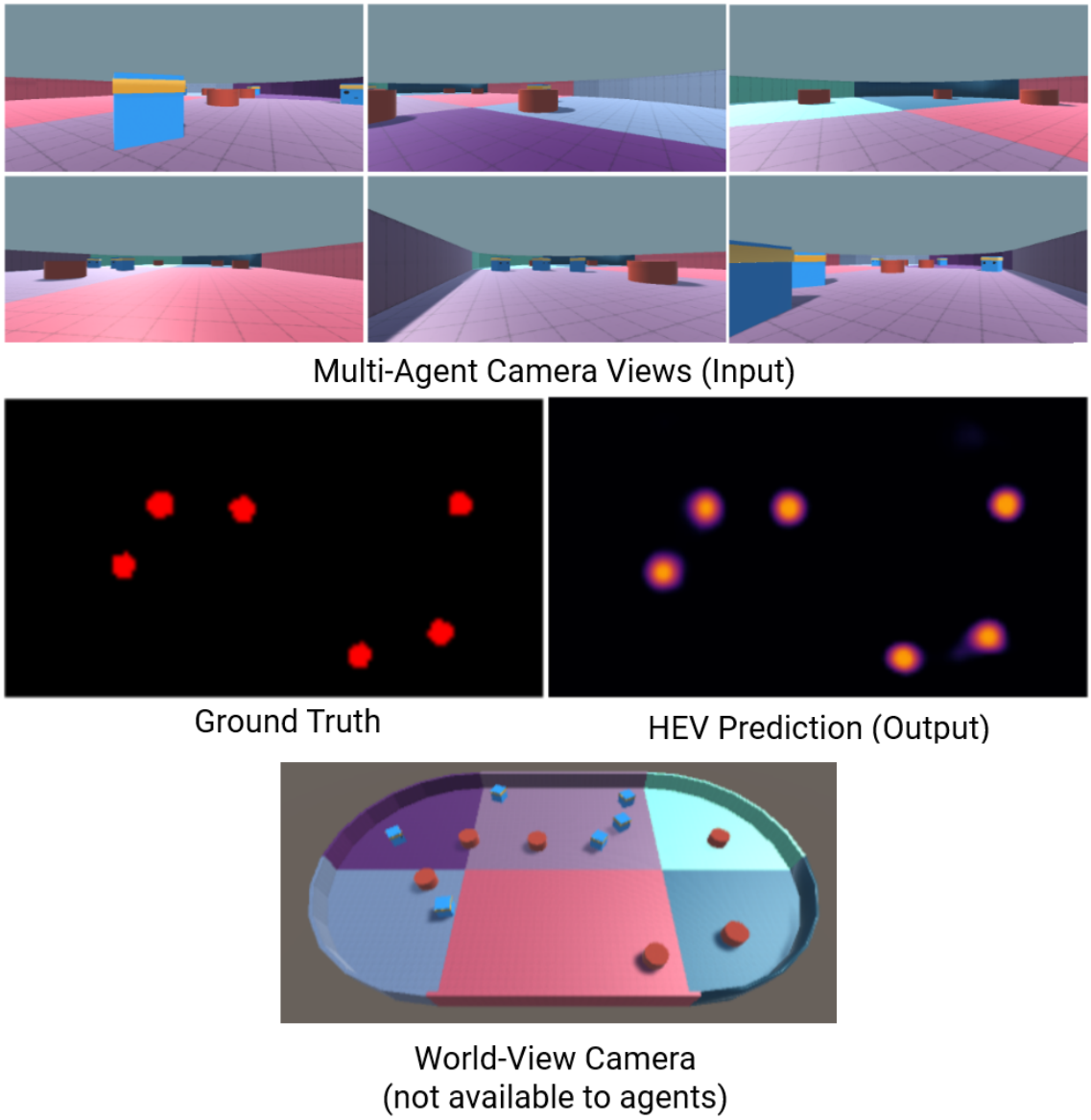

Figure 4.8: (left) example six input images to HEV model, (right) HEV model prediction of agent locations and dragon agent.

4.3.4 Summary of Results

Our results are shown in Table 4.2 and Figure 4.9 demonstrate the world-centric coordinate frame consistently outperforms the ego-centric coordinate frames in all en-

Table 4.2: HEV-CVT validation IoU results per coordinate frame in each environment (higher is better).

	World-Centric	Ego-Centric	Ego-Forward
Collaborative Push Block	**96.94%**	63.87%	64.22%
Dungeon Escape	**43.53%**	13.47%	26.07%
Planar Construction	**48.37%**	35.45%	10.16%

vironments. The Collaborative Push Block and Dungeon Escape environments show the largest performance improvements, with up to 32.72% and 43.71% improvement in IoU, respectively. These results suggest that the world-centric HEV approach is effective in addressing the challenges of collaborative perception in multi-agent environments. This result is especially apparent in the Collaborative Push Block environment, where HEV-CVT model easily localizes itself based on the large goal location seen in most camera views for a near-perfect 96.94% IoU score.

4.4 Multi-Agent Reinforcement Learning

In order to compare the performance of the fixed world-view coordinate frames with other commonly used coordinate frames, we conduct experiments in all three proposed environments. To ensure a fair comparison between the performance of agents using

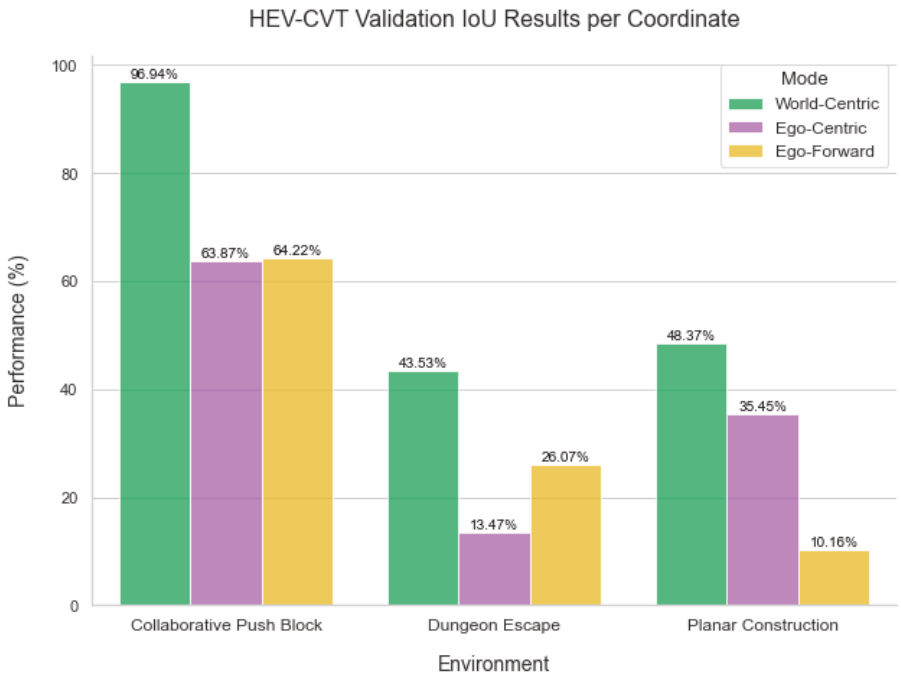

Figure 4.9: HEV-CVT validation IoU results per coordinate frame in each environment (higher is better).

different coordinate frames, we use identical reward functions to each environment's original implementation and identical grid sizes. We train unique RL models for world-centric, ego-centric and ego-forward perspective view models. To evaluate the strength of each trained RL model, after training has converged, we run the agent in the environment 1000 times and report the mean and standard deviation for the agent to complete the task in the environment in Table 4.3.

4.4.1 Experiment 1: Collaborative Push Block

In the Push Block environment, agents collaboratively work to push various-sized blocks into a designated goal area. The distinct feature of this environment is the differential block sizes (1-3), the number of agents required to push a block corresponds to its size, e.g., pushing a size-three block necessitates the collective effort of all three agents.

When agents were trained using the BEV perspective view, several inefficiencies were observed. The most glaring of which was the frequent scenario where one of the three agents did not observe the size three block. This lack of visibility led to a predicament where the two agents, who were aware of the larger block, would position themselves ready for a collaborative push. However, due to the missing third agent, there was a prolonged waiting period, leading to wasted time and decreased task efficiency. This recurrent scenario is not merely an oversight but can be attributed to the inherent limitations of the BEV architecture, which restricts agents' observational capacities and situational awareness. This behaviour occurred in both ego-centric and ego-forward agents but occurred more in ego-centric agents as reflected in Table 4.3

In stark contrast to the BEV agents, those modelled using the HEV perspective view demonstrated significantly superior collaboration. From the onset of the task, HEV agents had a tendency to remain in close proximity. This behaviour intuitively aligned with the task's demands, especially for handling the size three block, which requires all three agents for movement. This strategy allowed for HEV agents to

almost never wait on each other for a collaborative push. Additionally, their strategy consistently prioritized pushing the highest-value blocks first. By addressing the most collaboration-intensive part of the task at the outset, they mitigated the risks of wasted time and uncoordinated efforts. Following this, they would disperse, focusing on the smaller blocks, ensuring maximum task efficiency. This systematic approach by the HEV agents highlights the potential of the HEV architecture in enhancing agent collaboration and overall task performance.

4.4.2 Experiment 2: Dungeon Escape

In the Dungeon Escape environment, one agent must sacrifice itself to the green dragon so one of the remaining two agents can recover the key and escape. The agents have a short amount of time to reach the dragon before it reaches the exit portal, if they don't, they lose. If the agents do reach the dragon, they must pick up the key the dragon drops and then bring it to the exit door random location.

The agents trained in the Dungeon Escape environment showed the smallest change between viewpoint observations. We think this small amount of variance is due to the lack of collaboration required to complete the dungeon escape environment. Only a single agent needs to observe the dragon in order for them to move towards it and sacrifice themselves for the key. Then, only one of the two remaining agents needs to find the key and bring it to the door for all the agents to win.

Despite the Dungeon Escape environment requiring less collaboration than the

other environments, we still find the HEV model outperforms its BEV counterparts. The world-centric HEV approach completes the task in 14.15 time steps on average while the ego-centric completes the task in 16.74 time steps on average.

4.4.3 Experiment 3: Planar Construction

In the Planar Construction environment, six agents must push randomly arranged pucks into desired positions or patterns. Each round a random number of pucks are spawned to be pushed, with the same number of desired destinations available for the pucks. Any puck can be pushed into any of the desired positions, the agents win when all pucks are in desired positions with an overlap greater than a set threshold.

We find that agents trained with ego-centric perspective views prioritize pushing pucks that are near them and easy to travel to, requiring the least turning. This strategy is effective at first but often leads to the last remaining pucks requiring to be pushed at longer distances than required if solved optimally. In contrast, agents trained on world-centric policies account for the actions of other agents and spread out their efforts, resulting in a more efficient policy. These HEV agents will more frequently avoid goals that are targeted by teammates and take paths that avoid collision with other agents.

Table 4.3: MA-POCA mean episode timesteps ± standard deviation per coordinate frame in each environment (lower is better).

	World-Centric	Ego-Centric	Ego-Forward
Collaborative Push Block	**246.5** ± 106.6	280.9 ± 112.6	259.5 ± 109.2
Dungeon Escape	**42.7** ± 44.6	16.74 ± 49.6	23.27 ± 57.6
Planar Construction	**110.5** ± 16.6	120.3 ± 18.2	149.9 ± 26.4

4.4.4 Summary of Results

Our experiments highlight a common challenge faced by BEV agents in all three environments, where sometimes the object necessary to take the optimal action was missing from the agent's view, leading to sub-optimal decision-making and increased episode lengths. The HEV agents were able to leverage the world-centric viewpoint available to them, enabling them to perceive their environment more holistically and take more optimal actions. This issue was particularly evident in the Planar Construction environment, where the improved perception of world-centric agents resulted in significantly lower episode lengths than ego-based agents.

Overall, these findings suggest that the HEV model offers a superior perception perspective view in MARL environments, providing agents with a more comprehensive understanding of their surroundings, leading to improved decision-making and better

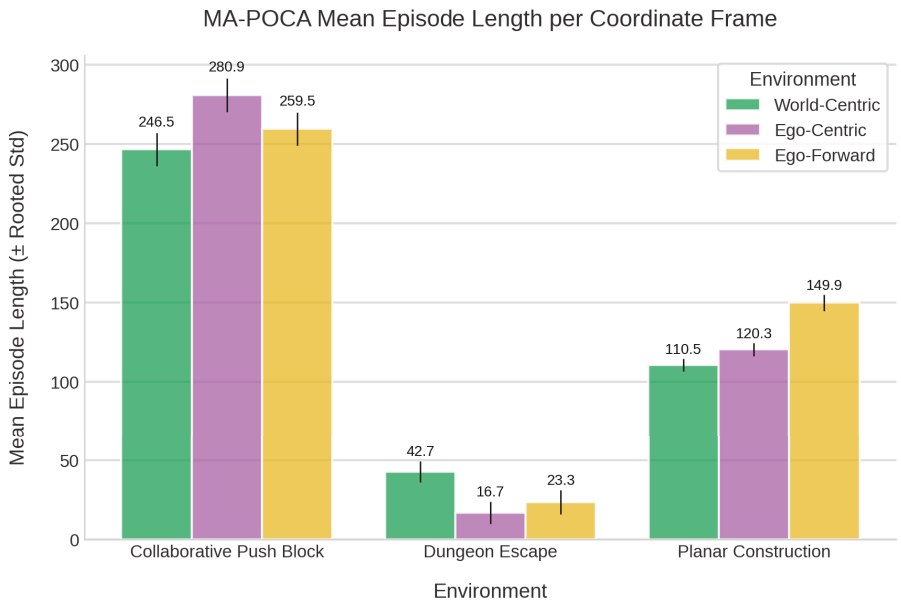

Figure 4.10: MA-POCA mean episode length ± standard deviation per coordinate frame in each environment (lower is better).

overall performance.

Table 4.3 and Figure 4.10 compare agents' performance using different coordinate frames to those using traditional sensor frames in all three proposed environments. We find consistently lower episode lengths with world-centric based agents compared to ego-centric agents.

Chapter 5

Conclusion

We have proposed a new perception model called Herd's Eye View (HEV) that provides a global view of the environment, enabling better global coordination and cooperation in MARL scenarios. We conduct two sets of experiments in three simulated multi-agent environments. Our first experiment focuses on the perception aspect of HEV and shows the same Cross-View Transformer model performs better on the world-centric HEV task than its BEV ego-centric counterpart. Our second experiment focuses on the effectiveness of the HEV perspective view compared to BEV perspective views for MARL agents. We find that RL agents trained on world-centric perspective views learn more efficient policies than those trained on ego-centric perspective views. Our work opens up new possibilities for advanced perception models in MARL game environments, which can greatly enhance the performance of multi-agent systems by enabling better collaboration and coordination.

5.1 Contributions

This thesis presents two primary contributions to the autonomous vehicle perception domain within Multi-Agent Reinforcement Learning settings:

HEV-CVT Model Development: We introduced the HEV-CVT model, a novel transformer-based approach providing a global, world-centric perspective of the environment. This model contrasts with traditional ego-centric BEV models by leveraging Cross-View Transformers to aggregate information across multiple agents, thereby enhancing global coordination and strategic decision-making in MARL scenarios.

HEV vs. BEV Perspective Evaluation: Our experiments demonstrated that MARL agents trained with the HEV perspective exhibit superior policy efficiency compared to those trained with conventional BEV perspectives. This finding highlights the Herd Eye View advantage of adopting a global viewpoint for improving the performance of multi-agent systems.

We additionally open-source the code-base for our Unity simulator and HEV dataset generation. These contributions advance the field by showcasing the potential of transformer-based models for collaborative perception in dynamic multi-agent environments.

5.2 Future Work

We introduce the HEV-CVT model to produce a global semantic segmentation of a game environment. More advanced transformer architectures have been published for Bird's Eye View semantic segmentation since the Cross-View transformer that could inspire improvements in the HEV-CVT architecture. One such improvement could be making the model input video-based; additional frames from previous time steps would likely improve the IoU scores of a HEV model [93, 94]. Improvements such as transformer size, dataset size and additional transformer layers could also increase model IoU scores.

Another avenue for feature research could be unifying the HEV perception and control algorithms. Some recent works directly predict future paths for an agent using recurrent neural networks. These models, often referred to as "end-to-end" models, operate similarly to the CVT model utilizing transformer attention, but instead of using CNNs in the last stage to predict semantic segmentation, they use a recurrent network to predict waypoints that form a future path for the agent. The HEV-CVT or a similar model's semantic segmentation head could be swapped with a recurrent network to predict future paths for an agent, or multiple agents, and our previously developed MA-POCA model could be used to create labels for these paths.

Bibliography

[1] Tesla. Tesla ai day 2021, 08 2021. URL https://www.youtube.com/watch?v=j0z4FweCy4M.

[2] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, 2017. doi: 10.1109/ICCV.2017.534.

[3] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[5] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisser-

man, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. *CoRR*, abs/2107.14795, 2021. URL https://arxiv.org/abs/2107.14795.

[6] Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. *CoRR*, abs/2010.04903, 2020. URL https://arxiv.org/abs/2010.04903.

[7] 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. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[8] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. *CoRR*, abs/2003.13402, 2020. URL https://arxiv.org/abs/2003.13402.

[9] Hanspeter A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biol. Cybern.*, 64(3):177–185, jan 1991. ISSN 0340-1200. doi: 10.1007/BF00201978. URL https://doi.org/10.1007/BF00201978.

[10] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbi-

trary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, 2020.

[11] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. *CoRR*, abs/2003.13402, 2020. URL `https://arxiv.org/abs/2003.13402`.

[12] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022.

[13] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[14] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022.

[15] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[16] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning (CoRL)*, 2022.

[17] Marek Wydmuch, Michal Kempka, and Wojciech Jaskowski. Vizdoom competitions: Playing doom from pixels. *IEEE Transactions on Games*, 11(3):248–259, 2019. doi: 10.1109/TG.2018.2877047.

[18] Arthur Juliani, Ahmed Khalifa, Vincent-Pierre Berges, Jonathan Harper, Ervin Teng, Hunter Henry, Adam Crespi, Julian Togelius, and Danny Lange. Obstacle tower: A generalization challenge in vision, control, and planning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2684–2691. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/373. URL `https://doi.org/10.24963/ijcai.2019/373`.

[19] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Y. Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric bev perception: A survey. 2022.

[20] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson, 2018. ISBN 9780133356724. URL `https://books.google.ca/books?id=0F05vgAACAAJ`.

[21] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes chal-

lenge: A retrospective. *International Journal of Computer Vision*, 111:98–136, 2014.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[23] Caroline Strickland, David Churchill, and Andrew Vardy. A reinforcement learning approach to multi-robot planar construction. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 238–244, 2019. doi: 10.1109/MRS.2019.8901087.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[25] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. URL `http://arxiv.org/abs/1502.05477`.

[26] Andrew Cohen, Ervin Teng, Vincent-Pierre Berges, Ruo-Ping Dong, Hunter Henry, Marwan Mattar, Alexander Zook, and Sujoy Ganguly. On the use and misuse of absorbing states in multi-agent reinforcement learning. *CoRR*, abs/2111.05992, 2021. URL `https://arxiv.org/abs/2111.05992`.

[27] Chun-Fu (Richard) Chen, Quanfu Fan, and Rameswar Panda. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In *International Conference on Computer Vision (ICCV)*, 2021.

[28] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 212–228, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01225-0.

[29] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017. URL `http://arxiv.org/abs/1705.03122`.

[30] Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, Hyung Won Chung, Yin-Wen Chang, and Chun-Sung Ferng. A simple and effective positional encoding for transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2988, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.236. URL `https://aclanthology.org/2021.emnlp-main.236`.

[31] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu.

Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127063. URL https://www.sciencedirect.com/science/article/pii/S0925231223011864.

[32] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position Information in Transformers: An Overview. *Computational Linguistics*, 48(3):733–763, 09 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00445. URL https://doi.org/10.1162/coli_a_00445.

[33] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on visual transformer. *CoRR*, abs/2012.12556, 2020. URL https://arxiv.org/abs/2012.12556.

[34] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *CoRR*, abs/2104.01136, 2021. URL https://arxiv.org/abs/2104.01136.

[35] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023.

[36] Yue Wang, Vitor Campanholo Guizilini, Tianyuan Zhang, Yilun Wang, Hang

Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. *ArXiv*, abs/2110.06922, 2021.

[37] Zhihuang Zhang, Meng Xu, Wenqiang Zhou, Tao Peng, Liang Li, and Stefan Poslad. Bev-locator: An end-to-end visual semantic localization network using multi-view images, 2022.

[38] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: Future instance segmentation in bird's-eye view from surround monocular cameras. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[39] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A. Sophia Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. In *Conference on Robotic Learning (CoRL)*, 2022.

[40] Andrea Palazzi Guido Borghi, Davide Abati, Simone Calderara, and Rita Cucchiara. Learning to map vehicles into bird's eye view. *CoRR*, abs/1706.08442, 2017. URL `http://arxiv.org/abs/1706.08442`.

[41] Minghan Zhu, Songan Zhang, Yuanxin Zhong, Pingping Lu, Huei Peng, and John Lenneman. Monocular 3d vehicle detection using uncalibrated traffic cameras through homography. *CoRR*, abs/2103.15293, 2021. URL `https://arxiv.org/abs/2103.15293`.

[42] L. Reiher, B. Lampe, and L. Eckstein. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020. doi: 10.1109/ITSC45102.2020.9294462.

[43] Yves Grandvalet, Tom Drummond, and You Li. Driving among flatmobiles: Bird-eye-view occupancy grids from a monocular camera for holistic trajectory planning. *CoRR*, abs/2008.04047, 2020. URL `https://arxiv.org/abs/2008.04047`.

[44] Yigit Baran Can, Alexander Liniger, Ozan Unal, Danda Pani Paudel, and Luc Van Gool. Understanding bird's-eye view semantic hd-maps using an onboard monocular camera. *CoRR*, abs/2012.03040, 2020. URL `https://arxiv.org/abs/2012.03040`.

[45] Sunando Sengupta, Paul Sturgess, L'ubor Ladický, and Philip H. S. Torr. Automatic dense visual semantic mapping from street-level imagery. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 857–862, 2012. doi: 10.1109/IROS.2012.6385958.

[46] Jinwoo Hwang, Philipp Benz, and Pete Kim. Booster-shot: Boosting stacked homography transformations for multiview pedestrian detection with attention. In

Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 363–372, January 2024.

[47] Xinge Zhu, Zhichao Yin, Jianping Shi, Hongsheng Li, and Dahua Lin. Generative adversarial frontal view to bird view synthesis. In *2018 International Conference on 3D Vision (3DV)*, pages 454–463, 2018. doi: 10.1109/3DV.2018.00059.

[48] Kaustubh Mani, Swapnil Daga, Shubhika Garg, Narasimhan Sai Shankar, Krishna Murthy Jatavallabhula, and K. Madhava Krishna. Monolayout: Amodal scene layout from a single image. *CoRR*, abs/2002.08394, 2020. URL `https://arxiv.org/abs/2002.08394`.

[49] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. *CoRR*, abs/1812.07179, 2018. URL `http://arxiv.org/abs/1812.07179`.

[50] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *CoRR*, abs/1906.06310, 2019. URL `http://arxiv.org/abs/1906.06310`.

[51] Xinzhu Ma, Zhihui Wang, Haojie Li, Wanli Ouyang, and Pengbo Zhang. Accurate monocular 3d object detection via color-embedded 3d reconstruction for

autonomous driving. *CoRR*, abs/1903.11444, 2019. URL http://arxiv.org/abs/1903.11444.

[52] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. *CoRR*, abs/2008.04582, 2020. URL https://arxiv.org/abs/2008.04582.

[53] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, and Elisa Ricci. Demystifying pseudo-lidar for monocular 3d object detection. *CoRR*, abs/2012.05796, 2020. URL https://arxiv.org/abs/2012.05796.

[54] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge J. Belongie, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. *CoRR*, abs/2004.03080, 2020. URL https://arxiv.org/abs/2004.03080.

[55] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution network for monocular 3d object detection. *CoRR*, abs/2103.01100, 2021. URL https://arxiv.org/abs/2103.01100.

[56] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? *CoRR*, abs/2108.06417, 2021. URL https://arxiv.org/abs/2108.06417.

[57] Chenyang Lu, Gijs Dubbelman, and Marinus Jacobus Gerardus van de Molengraft. Monocular semantic occupancy grid mapping with convolutional varia-

tional auto-encoders. *CoRR*, abs/1804.02176, 2018. URL http://arxiv.org/abs/1804.02176.

[58] Bowen Pan, Jiankai Sun, Alex Andonian, Aude Oliva, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *CoRR*, abs/1906.03560, 2019. URL http://arxiv.org/abs/1906.03560.

[59] Noureldin Hendy, Cooper Sloan, Feng Tian, Pengfei Duan, Nick Charchut, Yuesong Xie, Chuang Wang, and James Philbin. FISHING net: Future inference of semantic heatmaps in grids. *CoRR*, abs/2006.09917, 2020. URL https://arxiv.org/abs/2006.09917.

[60] Avishkar Saha, Oscar Alejandro Mendez Maldonado, Chris Russell, and R. Bowden. Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5133–5139, 2021.

[61] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. URL http://arxiv.org/abs/1612.03144.

[62] Lang Peng, Zhirong Chen, Zhang-Hua Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5924–5932, 2022.

[63] Jiachen Lu, Zheyuan Zhou, Xiatian Zhu, Hang Xu, and Li Zhang. Learning ego 3d representation as ray tracing. In *European Conference on Computer Vision*, 2022.

[64] Florent Bartoccioni, Eloi Zablocki, Andrei Bursuc, Patrick Perez, Matthieu Cord, and Karteek Alahari. Lara: Latents and rays for multi-camera bird's-eye-view semantic segmentation. In *6th Annual Conference on Robot Learning*, 2022.

[65] Yunze Man, Liangyan Gui, and Yu-Xiong Wang. Bev-guided multi-modality fusion for driving perception. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21960–21969, 2023. URL `https://api.semanticscholar.org/CorpusID:260870088`.

[66] Naiyu Fang, Le miao Qiu, Shuyou Zhang, Zili Wang, Kerui Hu, and Kang Wang. A cross-scale hierarchical transformer with correspondence-augmented attention for inferring bird's-eye-view semantic segmentation. *ArXiv*, abs/2304.03650, 2023. URL `https://api.semanticscholar.org/CorpusID:258041225`.

[67] Shaoyu Chen, , Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv:2206.10965*, 2022.

[68] Zitian Wang, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Object as query: Equipping any 2d object detector with 3d detection ability. *arXiv preprint arXiv:2301.02364*, 2023.

[69] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H. Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. In *CVPR*, 2022.

[70] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. Monodetr: Depth-aware transformer for monocular 3d object detection. *arXiv preprint arXiv:2203.13310*, 2022.

[71] Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. Collaborative perception in autonomous driving: Methods, datasets and challenges. *ArXiv*, abs/2301.06262, 2023.

[72] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[73] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7(4): 10914–10921, 2022.

[74] Ruiqing Mao, Jingyu Guo, Yukuan Jia, Yuxuan Sun, Sheng Zhou, and Zhisheng Niu. Dolphins: Dataset for collaborative perception enabled harmonious and

interconnected self-driving. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 4361–4377, December 2022.

[75] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[76] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Thirty-sixth Conference on Neural Information Processing Systems (Neurips)*, November 2022.

[77] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[78] Yiming Li, Juexiao Zhang, Dekun Ma, Yue Wang, and Chen Feng. Multi-robot scene completion: Towards task-agnostic collaborative perception. In *6th Annual Conference on Robot Learning*, 2022.

[79] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. *Latency-Aware Collaborative Perception*, pages 316–332. 11 2022. ISBN 978-3-031-19823-6. doi: 10.1007/978-3-031-19824-3_19.

[80] Sanbao Su, Yiming Li, Sihong He, Songyang Han, Chen Feng, Caiwen Ding, and Fei Miao. Uncertainty quantification of collaborative detection for self-driving. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5588–5594, 05 2023. doi: 10.1109/ICRA48891.2023.10160367.

[81] Yang Zhou, Jiuhong Xiao, Yue Zhou, and Giuseppe Loianno. Multi-robot collaborative perception with graph neural networks. *IEEE Robotics and Automation Letters*, 7(2):2289–2296, 2022.

[82] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

[83] Donghao Qiao and Farhana Zulkernine. Adaptive feature fusion for cooperative perception using lidar point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1195, January 2023.

[84] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2020. URL `https://arxiv.org/pdf/1809.02627.pdf`.

[85] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang,

and Wenyu Liu. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer. *arXiv preprint arXiv:2206.04584*, 2022.

[86] Andrew Cohen, Ervin Teng, Vincent-Pierre Berges, Ruo-Ping Dong, Hunter Henry, Marwan Mattar, Alexander Zook, and Sujoy Ganguly. On the use and misuse of absorbing states in multi-agent reinforcement learning. *RL in Games Workshop AAAI 2022*, 2022. URL `http://aaai-rlg.mlanctot.info/papers/AAAI22-RLG_paper_32.pdf`.

[87] Andrew Vardy. Orbital construction: Swarms of simple robots building enclosures. In *2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 147–153, 09 2018. doi: 10.1109/FAS-W.2018.00040.

[88] C. Strickland, D. Churchill, and A. Vardy. A reinforcement learning approach to multi-robot planar construction. In *IEEE International Symposium on Multi-Robot and Multi-Agent Systems*, pages 238–244, 2019.

[89] Andrew Vardy. Robot distancing: Planar construction with lanes. In Marco Dorigo, Thomas Stützle, Maria J. Blesa, Christian Blum, Heiko Hamann, Mary Katherine Heinrich, and Volker Strobel, editors, *Swarm Intelligence*, pages 229–242, Cham, 2020. Springer. ISBN 978-3-030-60376-2.

[90] Andrew Vardy. The lasso method for multi-robot foraging. In *19th Conference*

on Robots and Vision (CRV), pages 106–113. IEEE Xplore, 2022. doi: 10.1109/CRV55824.2022.00022.

[91] Andrew Vardy. The swarm within the labyrinth: Planar construction by a robot swarm. *Artificial Life and Robotics*, 2023. doi: 10.1007/s10015-022-00849-5. URL `https://rdcu.be/c2M43`.

[92] Mathias Baske. Grid sensors for unity ml-agents - version 2.0. `https://github.com/mbaske/grid-sensor`, 2021.

[93] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[94] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *NeurIPS*, 2022.