Ice Hockey Puck Detection and Tracking from Broadcast Video

by

Liam Salass

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2026

© Liam Salass 2026

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis was written by Liam Salass under the supervision of Dr. John Zelek and Dr. David Clausi. Portions of the research presented in this thesis are based on the following co-authored publications. I was the lead author and primary contributor to both works, responsible for the methodology, implementation, experiments, analysis, and manuscript preparation. The co-authors provided supervision, research guidance, and feedback.

Ice Hockey Puck Localization Using Contextual Cues [49]

L. Salass, J. Bright, A. Nazemi, Y. Chen, J. Zelek, and D. Clausi. This work forms the basis of the PLUCC puck localization methodology and associated experiments presented in this thesis.

Geometrically Aware Ice Hockey Puck Tracking via Stable Planar Representations [50]

L. Salass, J. Zelek, and D. Clausi. This work forms the basis of the stable rink-plane representation, geometry-aware puck tracking methodology, and associated experiments presented in this thesis.

Abstract

Accurate localization and tracking of fast-moving sports objects in broadcast video remain a fundamental problem in computer vision, enabling downstream applications such as tactical analysis, player analytics, automated highlight generation, officiating assistance, and augmented reality broadcasting. In ice hockey, this problem is particularly difficult because the puck occupies only a few pixels, rapidly accelerates, exhibits motion blur, becomes occluded by players and boards, and is observed under continuously changing camera viewpoints. These characteristics substantially degrade the performance of conventional appearance-based detection and tracking methods.

This thesis presents an end-to-end framework for puck detection and tracking that combines contextual reasoning, geometric scene understanding, and temporal motion modelling. First, I propose a context-aware puck detector, termed *Player Localization Using Contextual Cues* (PLUCC). By using player segmentation as an explicit contextual prior, the proposed method exploits the strong spatial relationship between players and the puck to improve localization accuracy over appearance-only detection approaches.

The thesis then assesses the suitability of modern visual object trackers for broadcast hockey footage through a comprehensive evaluation of modern tracking algorithms. Experimental results show that existing appearance-based trackers remain fundamentally limited by the puck's small image footprint, rapid motion, frequent occlusions, and visually similar distractors, motivating an alternative tracking paradigm.

To address these limitations, we introduce a geometry-aware tracking framework in which puck motion is represented in a temporally stable planar coordinate system rather than directly in image space. A monocular 3D reconstruction pipeline estimates the broadcast camera pose and reconstructs the rink surface, while semantic segmentation removes dynamic objects to enable robust plane estimation. Projecting detections into this canonical coordinate system substantially reduces perspective distortion and camera-induced motion, producing geometrically consistent trajectories throughout time. Temporal motion modelling is subsequently performed within the planar representation using sequence-based learning, allowing the model to focus on the puck's underlying physical motion rather than compensating for camera motion.

Collectively, the proposed framework demonstrates that combining contextual detection with geometry-aware motion modelling provides a more robust foundation for puck tracking than conventional appearance-based approaches alone. Although developed for broadcast ice hockey, the suggested approach is largely sport-agnostic and is readily applicable to other sports involving small objects moving on approximately planar playing surfaces, including soccer, tennis, basketball, lacrosse, and baseball.

Acknowledgements

I would like to express my sincere gratitude to everyone who supported me throughout my master's degree and contributed to making this thesis possible.

First and foremost, I would like to thank my supervisors, Professor David Clausi and Professor John Zelek, for their guidance, encouragement, and continual support throughout this research. Their expertise, thoughtful feedback, and willingness to challenge my ideas have greatly shaped both this thesis and my development as a researcher. I am grateful for the opportunity to have learned from two outstanding mentors.

I would also like to thank Stathletes for their collaboration and support throughout this work. Their financial support, industry expertise, and access to broadcast hockey data made this research possible and provided the opportunity to investigate problems with direct real-world impact.

I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), whose funding enabled this research.

I am thankful to the members of the Vision and Image Processing (VIP) Lab and the students within the Systems Design Engineering department. Thank you for the many discussions, ideas, collaborations, and friendships that made graduate school an enjoyable and rewarding experience. The supportive and collaborative environment you fostered made coming into the lab every day something I genuinely looked forward to.

Finally, I would like to thank my family and friends for their unwavering encouragement throughout this journey. To my parents, thank you for always believing in me, supporting my ambitions, and encouraging me to pursue my interests with curiosity and determination. To my friends, thank you for continually challenging me to grow, remain disciplined, and strive to become a better researcher and person. Your support has meant more than you know.

To everyone who has been part of this journey, thank you.

Dedication

This is dedicated to those who have shaped the person I have become:

To my parents, whose unwavering support encouraged me to pursue my interests, nurtured my curiosity, and made this journey possible.

To my friends, who continually challenged me to grow, strive for more, and remain committed to becoming a better version of myself.

Table of Contents

Author's Declaration ii

Statement of Contributions iii

Abstract iv

Acknowledgements v

Dedication vi

List of Figures xi

List of Tables xii

1 Introduction 1

1.1 Motivation 1

1.1.1 AI in Sports Analytics 1

1.1.2 Object Detection and Tracking in Sports 2

1.1.3 Applications of Puck Localization 2

1.2 Challenges in Ice Hockey Puck Tracking 2

1.2.1 Visual Challenges 2

1.2.2 Broadcast and Geometric Challenges 3

1.2.3 Limitations of Existing Approaches . . . 3
1.2.4 Research Gap and Thesis Direction . . . 4
1.3 Thesis Organization . . . 5

2 Background and Prior Works **6**
2.1 Background . . . 6
2.1.1 Deep Learning Fundamentals . . . 6
2.1.2 Object Detection . . . 9
2.1.3 Contextually Constraint Detection . . . 11
2.1.4 Visual Object Tracking . . . 12
2.1.5 Long Short-Term Memory Networks . . . 14
2.1.6 Geometric Vision . . . 16
2.2 Prior Works . . . 18
2.2.1 Ice Hockey Puck Detection . . . 18
2.2.2 Ice Hockey Puck Tracking . . . 18
2.2.3 Ice Rink Registration . . . 19

3 System Overview and Methodology **20**
3.1 Puck Detection Methodology . . . 20
3.1.1 Context in Puck Detection . . . 21
3.1.2 PLUCC Model Architecture . . . 22
3.1.3 Objective Function . . . 26
3.2 Puck Motion Estimation Methodology . . . 27
3.2.1 Temporally Stable Rink Plane Estimation . . . 27
3.2.2 Motion Estimation LSTMs . . . 31

4 Dataset, Baselines, and Metrics **36**
4.1 Dataset . . . 36
4.2 Baselines . . . 39
4.2.1 Detection Baselines . . . 39
4.2.2 Tracking Baselines . . . 39
4.3 Metrics . . . 39
4.3.1 Puck Detection Metrics . . . 39
4.3.2 Puck Tracking Metrics . . . 42

5 Experiments and Results **44**
5.1 Detection Experimental Setup and Results . . . 44
5.1.1 Puck Detection Experiment Implementation Details . . . 44
5.1.2 Detection Results . . . 45
5.2 Puck Tracking Experimental Setup and Results . . . 48
5.2.1 Puck Tracking Experiment Implementation Details . . . 48
5.2.2 Puck Tracking Results . . . 48
5.3 Puck Motion Estimation and Planar Surface Experiments . . . 52
5.3.1 Planar Surface Stability Experiments . . . 52
5.3.2 Puck Motion Estimation Experiments . . . 53

6 Conclusion and Future Works **55**
6.1 Conclusion . . . 55
6.2 Future Work . . . 57

References **59**

APPENDICES **68**

A Supplementary Experiments **69**
A.1 Puck Detection Ablation Experiments . . . 69
A.2 Runtime Analysis . . . 72

B Additional Implementation Details **73**

B.1 Visual Object Tracking Training Parameters 73

B.2 Motion Estimation LSTM Model Architectures 75

List of Figures

2.1 Generic CNN architecture. 7
2.2 Residual block architecture used in Residual Networks (ResNets). 8
2.3 Feature Pyramid Network (FPN) architecture. 10
2.4 Small-object detection challenge in puck localization. 11
2.5 Overview of the SiamFC tracking architecture. 13
2.6 Long Short-Term Memory (LSTM) cell architecture. 14
2.7 Perspective distortion in image and rink coordinate spaces. 17

3.1 Context image generation pipeline. 22
3.2 PLUCC model architecture. 23
3.3 Temporally stable rink-plane reconstruction pipeline. 28
3.4 Rink plane estimation from monocular 3D reconstruction. 30

4.1 Challenging puck detection scenarios in the VIP-PuckDataset. 38
4.2 Puck detection in image and homographic coordinate spaces. 42

5.1 Qualitative PLUCC detection results under challenging broadcast conditions. 46
5.2 Tracker performance during puck occlusion. 50
5.3 Examples of visual tracking failures. 51
5.4 Forward LSTM trajectory predictions. 54

A.1 Context-only puck localization from player information. 70
A.2 Effect of target Gaussian heatmap standard deviation on model performance. 71

List of Tables

3.1 Recurrent motion models used in the pipeline. 32

4.1 VIP-PuckDataset statistics. 37
4.2 Challenge-type breakdown of labeled sequences. 37

5.1 Performance comparison of puck detection models in image coordinate space. 47
5.2 Comparison of puck detection models in homographic rink-space coordinates. 47
5.3 Tracking performance of image-space visual trackers. 49
5.4 Effect of homography stability on Kalman filter puck tracking performance. 53
5.5 Tracking performance of trajectory-based planar models. 53

A.1 Performance comparison of PLUCC and the context-only FCN-ResNet152. 70
A.2 Effect of context-driven dropout on PLUCC performance. 71
A.3 Runtime breakdown of the proposed pipeline. 72

B.1 Fine-tuning hyperparameters for the evaluated tracking models. 74
B.2 Forward LSTM architecture for planar puck tracking. 75
B.3 Dual-path LSTM architecture for planar puck tracking. 76
B.4 Bidirectional LSTM architecture for planar puck tracking. 77

Chapter 1

Introduction

1.1 Motivation

1.1.1 AI in Sports Analytics

Recent advances in computer vision and deep learning have significantly expanded the capabilities of automated sports analytics systems [3, 4, 36, 41, 43, 53, 59, 63, 65, 69]. Applications now extend beyond traditional player tracking and statistics generation to include tactical analysis, automated event detection, automated refereeing, broadcast augmentation, player performance evaluation, injury prevention, and large-scale gameplay understanding. These systems are increasingly being adopted across professional sports organizations, media companies, and sports analytics platforms due to the growing availability of broadcast video data and improvements in machine learning infrastructure [36, 51].

Automated analysis systems are particularly valuable in sports where large volumes of video are generated, but manual annotation remains expensive and time-consuming. As a result, there is increasing interest in developing low-cost vision-based systems capable of extracting meaningful game information directly from broadcast footage. Such systems have the potential to improve accessibility of advanced analytics for lower-level leagues and amateur organizations that lack access to expensive sensor-based tracking infrastructure. Beyond coaching and player development, applications also include automated match clipping, gameplay summarization, broadcast enhancement, and sports betting analytics [25, 51].

1.1.2 Object Detection and Tracking in Sports

A fundamental component underlying many sports analytics tasks is accurate object localization and tracking. The location and motion of the primary game object often serve as strong indicators of gameplay structure, possession, event development, and tactical decision-making. Consequently, object tracking systems have become central to automated analysis pipelines across numerous sports.

Ball detection and tracking have been extensively studied in sports such as soccer [1, 5, 16, 17, 20, 27, 47, 66], volleyball [11, 12, 14, 15], basketball [9, 10], and tennis [33, 58]. These systems enable downstream tasks, including trajectory forecasting, tactical analysis, automated commentary generation, and event segmentation. Many modern pipelines further integrate player tracking, pose estimation, and temporal reasoning to better understand interactions between players and the game object [3, 41, 43, 53, 65].

1.1.3 Applications of Puck Localization

In ice hockey analytics, accurate puck localization plays a particularly important role due to the puck's direct relationship with possession, offensive pressure, passing structure, and gameplay flow [22, 60, 73]. Reliable puck trajectories enable downstream systems for automated highlight generation, tactical analysis, gameplay understanding, and player evaluation. For example, automated clipping systems can identify important gameplay events by analyzing puck motion in conjunction with player positioning and rink location [51].

Despite its importance, reliable puck tracking remains considerably less explored than object tracking in many other sports [35, 42, 64, 76]. Compared to larger sports objects such as soccer balls or basketballs, the hockey puck occupies only a very small fraction of the image area and exhibits substantially different motion characteristics. These challenges make robust puck localization from broadcast video a uniquely difficult computer vision problem.

1.2 Challenges in Ice Hockey Puck Tracking

1.2.1 Visual Challenges

Puck detection in broadcast ice hockey video is challenging due to a combination of severe visual degradations and rapid gameplay dynamics. Unlike many sports objects, the puck

occupies approximately 0.005% of the total image area in standard broadcast footage [76]. Its small size significantly limits the amount of visual information available for reliable feature extraction.

Additional challenges arise from frequent player and stick occlusions, rapid puck motion causing motion blur, broadcast compression artifacts, and low contrast regions near rink boards and goal lines. As illustrated in Figure 4.1, the puck often becomes visually ambiguous due to deformation, partial visibility, or similarity to surrounding structures such as skate blades, rink markings, and broadcast overlays. These conditions significantly degrade the performance of conventional object detection and tracking systems.

1.2.2 Broadcast and Geometric Challenges

Beyond visual ambiguity, broadcast hockey video introduces substantial geometric instability that further complicates motion estimation and tracking. Broadcast cameras continuously pan, zoom, and change viewpoints in response to gameplay, resulting in highly dynamic image-space motion. Consequently, puck trajectories represented directly in image coordinates often contain artificial motion induced by camera movement rather than true puck dynamics.

Many existing approaches attempt to compensate for these effects using frame-wise homography estimation or rink registration techniques [23,32,54]. However, independently estimating geometric transformations for each frame introduces temporal inconsistencies that accumulate over time, producing unstable rink-space trajectories and noisy velocity estimates. These geometric instabilities become particularly problematic for downstream temporal models that rely on smooth and physically consistent motion representations.

1.2.3 Limitations of Existing Approaches

Existing approaches to puck localization have addressed the challenges of broadcast hockey using several different strategies, each with limitations that motivate further investigation.

Yang [76] addresses the limited visual information available from the puck by incorporating temporal features from a sequence of consecutive frames. This allows information from neighbouring frames to supplement the appearance of the puck in any individual frame. However, the method requires a sequence of nine frames to produce a single puck localization and remains dependent on the puck being visually observable within the temporal window. Consequently, its ability to localize the puck is limited when visual evidence is weak or unavailable due to occlusion.

Pidaparthy et al. [42] incorporate contextual information by combining estimated puck locations with player locations and optical flow to identify the region of play. While this demonstrates the value of player information for reasoning about puck location, the player-based context is used primarily to estimate a broader region of play rather than being integrated directly into the puck localization process. The spatial relationship between individual players and the puck therefore remains largely unexploited during detection.

Other approaches improve temporal consistency by applying filtering or trajectory refinement to sequences of puck detections [35, 64]. These methods can suppress isolated false detections and produce smoother trajectories when reliable puck observations are available. However, representing puck motion directly in image coordinates causes the resulting trajectory to contain both the physical motion of the puck and apparent motion introduced by broadcast camera movement. This makes image-space trajectories poorly suited to modelling the underlying motion of the puck.

Conventional visual object tracking provides another means of constructing trajectories by associating the puck across consecutive frames using its visual appearance. However, the puck frequently occupies only a few pixels and may be blurred, occluded, or visually similar to rink markings and other distractors. Under these conditions, appearance provides limited information for maintaining reliable long-term associations.

Together, these limitations leave two important aspects of broadcast puck tracking unresolved: effectively exploiting player-based context when direct visual evidence of the puck is insufficient, and representing puck motion in a form that is robust to both unreliable appearance and broadcast camera movement. These gaps motivate the research objectives presented in the following section.

1.2.4 Research Gap and Thesis Direction

The first objective of this work is to address the limited use of player-based context in existing puck localization methods. To this end, we propose PLUCC, a context-aware detection framework that incorporates player segmentation as an explicit spatial prior, allowing the detector to exploit the relationship between player positioning and puck location while maintaining single-frame operation.

The second objective is to determine whether modern appearance-based visual object trackers can reliably recover puck trajectories from broadcast video. Through a comparative evaluation of recent tracking methods, we investigate the extent to which the puck's small size, rapid motion, occlusion, and limited appearance information constrain conventional visual tracking.

Finally, to address the geometric instability of image-space puck trajectories, we propose a geometry-aware tracking framework in which puck motion is represented in a temporally stable planar coordinate system derived from monocular 3D reconstruction and geometric plane estimation. By separating puck motion from changes in broadcast camera viewpoint, this representation provides a more stable basis for temporal motion modelling.

Collectively, these contributions establish an end-to-end framework that progresses from context-aware puck detection, through an evaluation of the limitations of appearance-based visual tracking, to a geometry-aware representation for puck motion estimation. In doing so, the thesis investigates both the visual and geometric limitations of existing puck tracking approaches and provides a foundation for more robust long-term tracking in broadcast sports video.

1.3 Thesis Organization

The remainder of this thesis is organized as follows:

- Chapter 2 reviews the background and prior work in object detection, visual object tracking, geometric vision, and broadcast ice hockey puck tracking.
- Chapter 3 presents the proposed methodology, including the PLUCC detection framework, temporally stable rink-plane estimation, and planar puck motion estimation.
- Chapter 4 describes the dataset, experimental baselines, and evaluation metrics used throughout this thesis.
- Chapter 5 presents the experimental setup and evaluates the proposed methods through detection, tracking, and planar motion estimation experiments.
- Chapter 6 summarizes the contributions of this thesis, discusses its limitations, and outlines directions for future work.

Chapter 2

Background and Prior Works

2.1 Background

Puck localization in broadcast hockey video involves two closely related computer vision tasks: small, fast-moving object detection and small, fast-moving object tracking. The following sections provide the deep learning, computer vision, and geometric vision background necessary to understand the methods presented in this thesis.

2.1.1 Deep Learning Fundamentals

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep learning models designed to process image or grid-structured data. CNNs extract visual features through the convolution operation, in which a learnable kernel is translated across the input image to produce a feature map. These features are subsequently combined to form higher-order representations that capture information about the entire image [8]. When performing a convolution, at each spatial location, the kernel computes a weighted sum over a local neighborhood, allowing the network to identify specific visual patterns such as edges, textures, and corners. Mathematically, the output of a two-dimensional convolution is given by Equation 2.1.

$$Y(i,j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} X(i+m, j+n)\, K(m,n), \tag{2.1}$$

where $X \in \mathbb{R}^{H \times W}$ is the input feature map or image, $K \in \mathbb{R}^{(2k+1)\times(2k+1)}$ is the learnable kernel of size $(2k+1) \times (2k+1)$, and Y is the output feature map. Indices (i, j) specify the spatial location of the output feature map, and (m, n) index the elements of the convolution kernel.

After each convolution, a nonlinear activation function is applied to enhance the network's expressive capacity. CNNs predominantly utilize the Rectified Linear Unit (ReLU), defined in Equation 2.2.

$$\text{ReLU}(x) = \max(0, x), \tag{2.2}$$

The final feature maps are usually flattened and passed through one or more fully connected layers, which combine the learned feature embeddings to generate the network output. Figure 2.1 illustrates a generic CNN architecture.

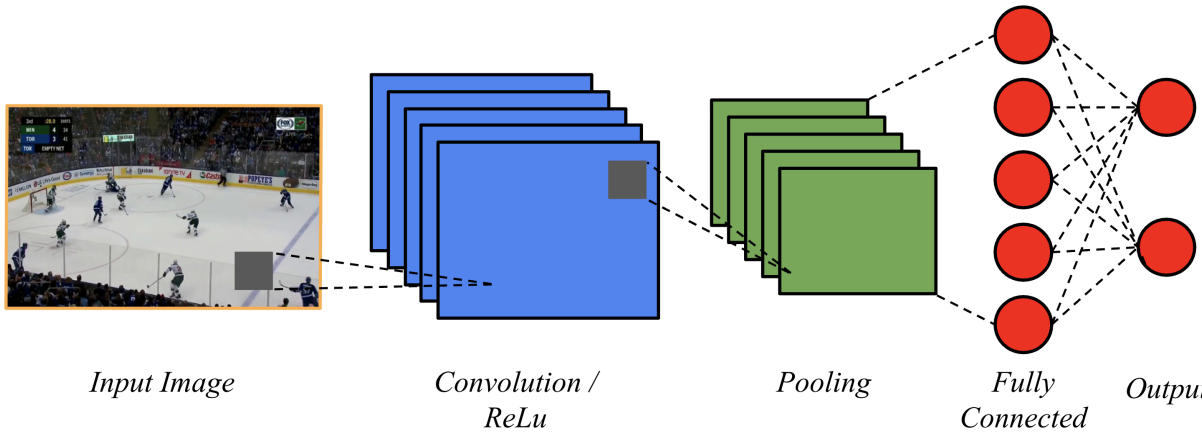


Figure 2.1: **Generic CNN architecture.** The input image is transformed through successive convolutional and nonlinear activation layers, producing structured feature representations. Pooling layers reduce the spatial dimensionality of feature maps whilst preserving discriminative information. The extracted features are subsequently processed by fully connected layers to generate the network output.

Residual Networks

Residual Networks (ResNets) address optimization challenges of deeper networks by reformulating the learning objective from directly learning a mapping to learning a residual

function [29]. Rather than requiring a sequence of convolutional layers to learn the desired mapping $H(x)$, a residual block instead learns the residual function

$$F(x) = H(x) - x, \tag{2.3}$$

such that the final output of the block becomes

$$Y = F(x) + x, \tag{2.4}$$

where x denotes the input feature map. This formulation allows the network to learn only the information that differs from the input, while an identity skip connection directly propagates the original features through the network.

Figure 2.2 illustrates a typical residual block.

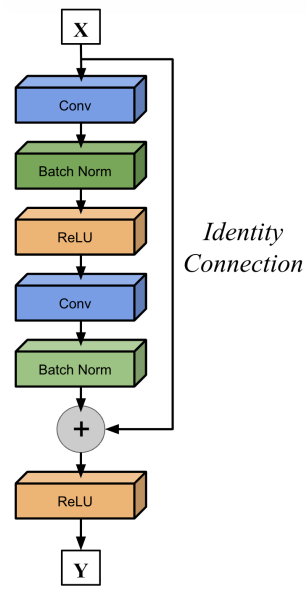


Figure 2.2: **Residual block architecture used in Residual Networks (ResNets).** The input **X** is transformed through convolution, batch normalization, and ReLU activation layers, while an identity connection bypasses these transformations. The transformed features are then combined with the original input through element-wise addition before the final activation, producing the output **Y**. The identity connection enables the block to learn a residual mapping relative to its input and facilitates the training of deeper neural networks.

Heatmap Regression

Heatmap regression formulates object localization as the prediction of a dense confidence map rather than directly regressing image coordinates or bounding boxes. During training, the ground truth location is represented as a two-dimensional Gaussian distribution centred on the target location,

$$H(x, y) = \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}\right), \tag{2.5}$$

where (x_0, y_0) denotes the ground truth object location and σ controls the spatial extent of the Gaussian. At inference, the object location is estimated from the peak response of the predicted heatmap. Compared with direct coordinate regression, heatmap regression preserves the spatial nature of the localization problem and provides smoother supervision during training, resulting in improved localization accuracy, particularly for small objects where precise coordinate regression is often unstable [21, 74].

2.1.2 Object Detection

Two-Stage Detectors

Deep learning object detectors are commonly categorized as either two-stage or single-stage architectures. Two-stage detectors, exemplified by Faster R-CNN [48], first generate a set of candidate object regions before performing object classification and bounding box refinement. By separating localization from classification, these methods typically achieve high detection accuracy, particularly for small or densely packed objects, but incur increased computational complexity and slower inference speeds.

Single-Stage Detectors

Single-stage detectors eliminate the explicit region proposal stage by directly predicting object classes and bounding boxes in a single forward pass. The YOLO family [45] is the most widely adopted example of this approach, achieving an effective balance between detection accuracy and computational efficiency. Although early single-stage detectors struggled with small object localization, modern architectures incorporate multi-scale feature fusion techniques to substantially improve performance while maintaining real-time inference.

Feature Pyramid Networks

Feature Pyramid Networks (FPNs) accurately localize objects across a wide range of scales by constructing a multi-scale feature hierarchy that combines high-resolution spatial information from shallow layers with semantically rich features from deeper layers [37]. As illustrated in Figure 2.3, an FPN employs a bottom-up encoder, a top-down decoder, and lateral skip connections to fuse features at multiple resolutions before prediction. FPNs significantly improve the detection of small objects and have become a standard component of many modern object detectors, including Faster R-CNN and recent YOLO architectures [45, 48].

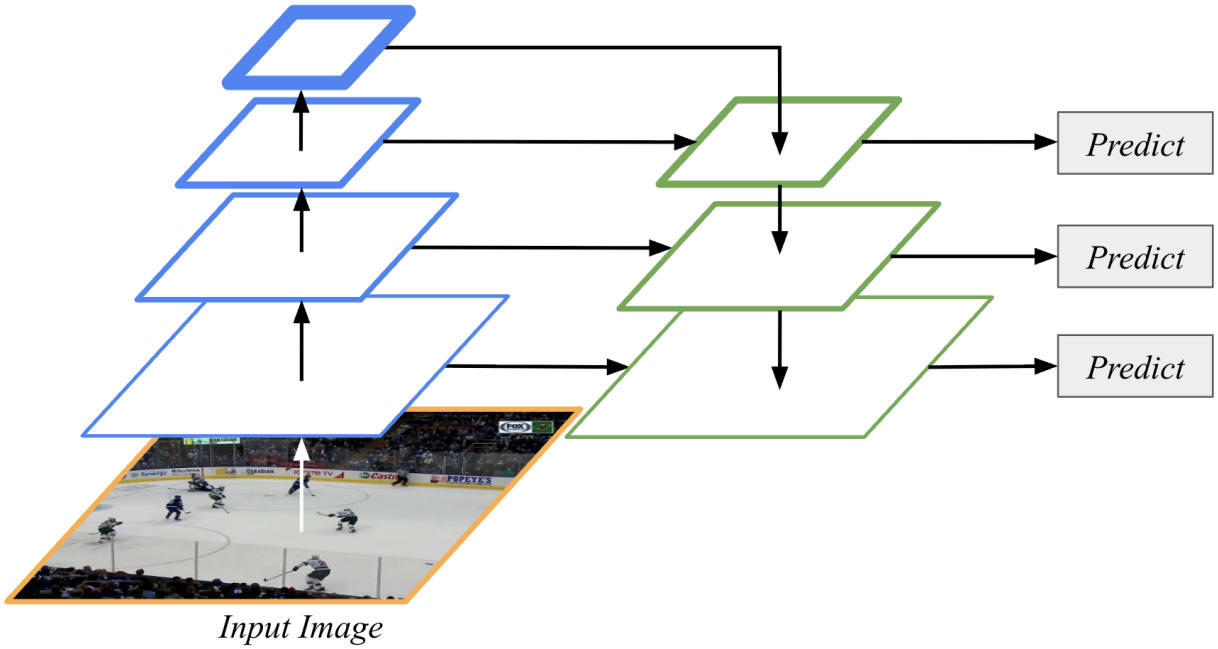


Figure 2.3: **Feature Pyramid Network (FPN) architecture.** The orange-bordered image represents the input image, which is processed by the encoder backbone (blue) to produce hierarchical feature representations at progressively lower spatial resolutions. The decoder pathway (green) combines higher-level semantic features with finer spatial information through lateral connections. Prediction heads (gray) attached at different pyramid levels produce outputs at multiple spatial resolutions.

Small Object Detection

Detecting an ice hockey puck is a challenging small object detection problem, with the puck occupying, on average, only 0.005% of the pixels in a broadcast frame [76]. As shown in Figure 2.4, the limited visual information available makes the puck highly susceptible to motion blur, low contrast, compression artifacts, and occlusion.

Small object detection is limited by the progressive reduction in spatial resolution within convolutional neural networks. As deeper layers learn increasingly semantic features, repeated downsampling removes the fine details required to localize tiny objects. Consequently, modern detectors commonly employ multi-scale feature aggregation and contextual reasoning, with recent methods also exploring Transformer architectures and image upscaling to improve the representation of small objects [70].

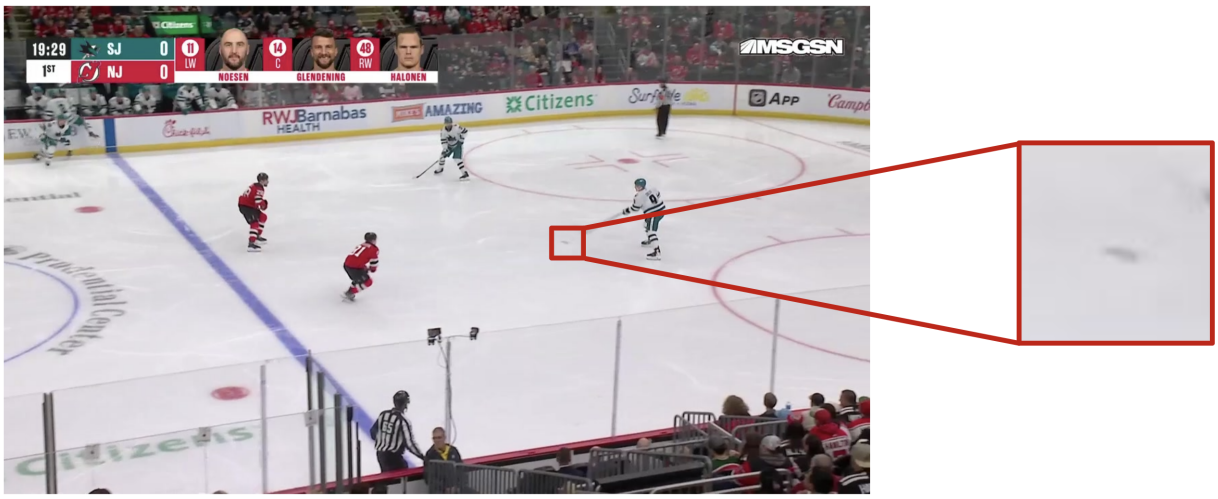


Figure 2.4: **Small-object detection challenge in puck localization.** A representative broadcast hockey frame is shown alongside a magnified view of the puck. The puck occupies only a few pixels in the full-resolution image, providing limited appearance information for detection.

2.1.3 Contextually Constraint Detection

Contextual constraints in detection algorithms are standard performance boosters in detection tasks where semantic correlations between spatial positions and object properties refine detection probabilities [13, 26, 55]. Shi et al. [55] enrich a feature pyramid network

for object detection using intermediary lateral modules that provide latent features spatial and high-frequency correlations, demonstrating improved performance for baseline models including Faster-RCNN [48].

2.1.4 Visual Object Tracking

Siamese Trackers

Visual object tracking has evolved through several architectural paradigms, each designed to enhance robustness against appearance variations, background clutter, and temporal uncertainty. Early deep learning–based trackers were largely characterized by Siamese network architectures, exemplified by SiamFC [6], which formulate tracking as learning a similarity function between an initial target template and a local search region. As illustrated in Figure 2.5, both the target template and search region are processed by a shared convolutional backbone to produce feature representations, which are subsequently compared through a cross-correlation operation to generate a similarity response map indicating the most likely target location. Although these approaches are computationally efficient, they rely on a fixed appearance representation and consequently exhibit substantial performance degradation under conditions such as occlusion, motion blur, and visually similar distractor objects.

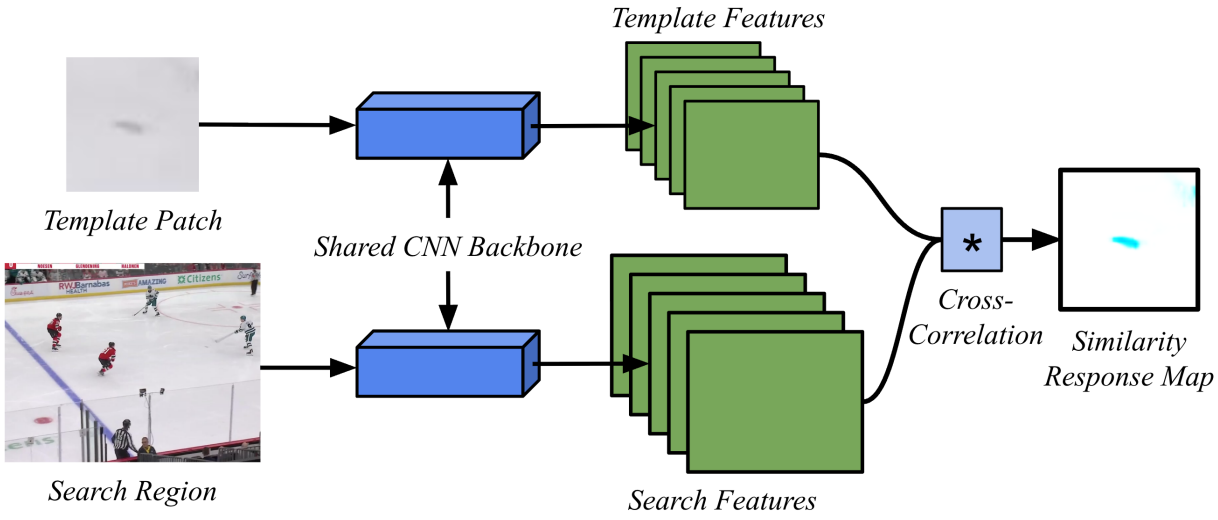


Figure 2.5: **Overview of the SiamFC tracking architecture.** A target template and search region are processed by a shared convolutional backbone to extract feature representations. Cross-correlation between the template and search features produces a similarity response map, whose peak identifies the predicted target location within the search region.

Online Optimization

Subsequent approaches introduced discriminative online model prediction to mitigate these limitations. DiMP [7] and SuperDiMP [18] adapt target-specific classifiers during tracking, improving resilience to background clutter but remaining vulnerable to corrupted updates when the target appearance is unreliable. PrDiMP [19] extends this framework by modeling uncertainty over target states, yet still depends on stable image-space observations.

Transformer Trackers

More recent Transformer-based trackers replace online optimization with global model prediction. ToMP [38] and TaMOs [39] leverage attention mechanisms to improve adaptation and re-detection. Despite these advances, all such trackers fundamentally assume that the target's appearance remains informative in image space—a premise that breaks down for tiny, fast-moving objects and a rapidly panning camera.

2.1.5 Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks are a form of recurrent neural network (RNN) designed to model temporal dependencies in sequential data [30]. Unlike feed-forward neural networks, recurrent networks maintain a hidden state that propagates information between successive time steps. However, conventional RNNs can struggle to learn long-term dependencies due to vanishing or exploding gradients. LSTMs address this limitation through a gated memory mechanism that regulates the retention, removal, and incorporation of information over time.

At each time step t, an LSTM receives the current input $\mathbf{x}_t$, the previous hidden state $\mathbf{h}_{t-1}$, and the previous cell state $\mathbf{c}_{t-1}$. Information is regulated through input, forget, and output gates, which determine what information is incorporated, retained, and propagated through the network. Figure 2.6 illustrates the information flow through an LSTM cell and the interaction between these gating mechanisms.

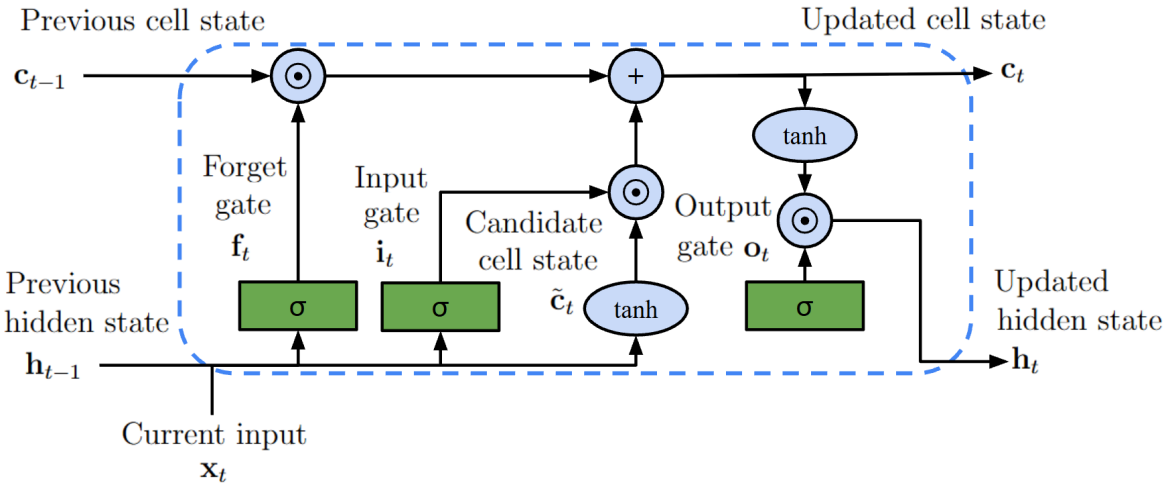


Figure 2.6: **Long Short-Term Memory (LSTM) cell architecture.** The current input $\mathbf{x}_t$, previous hidden state $\mathbf{h}_{t-1}$, and previous cell state $\mathbf{c}_{t-1}$ are processed through the forget, input, and output gates. These gates regulate the information retained and incorporated into the updated cell state $\mathbf{c}_t$ and the information propagated through the updated hidden state $\mathbf{h}_t$.

The central memory update can be expressed as

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{2.6}$$

where $\mathbf{f}_t$ and $\mathbf{i}_t$ denote the forget and input gates, respectively, $\tilde{\mathbf{c}}_t$ represents candidate information derived from the current input and previous hidden state, and $\odot$ denotes element-wise multiplication. The output gate subsequently controls how the updated cell state contributes to the hidden state $\mathbf{h}_t$. This gated memory mechanism allows LSTMs to selectively preserve information over extended temporal intervals, making them well suited for modelling object trajectories and other sequential motion patterns.

Forward LSTMs

A forward LSTM processes a sequence chronologically, such that the hidden state at time t contains information only from the preceding observations. Given a trajectory $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, the predicted state at time t is therefore conditioned on observations up to that time,

$$\hat{\mathbf{x}}_t = f(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}). \tag{2.7}$$

This causal structure makes forward LSTMs suitable for online tracking and motion prediction, where future observations are unavailable during inference. However, predictions during periods of missing observations must be inferred entirely from the trajectory preceding the gap.

Bidirectional LSTMs

Bidirectional LSTMs process a sequence in both temporal directions using separate forward and backward recurrent networks [52]. The forward network summarizes information preceding each time step, while the backward network incorporates information from subsequent observations. Their hidden representations are typically combined to produce an output conditioned on both past and future context,

$$\hat{\mathbf{x}}_t = f(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_T). \tag{2.8}$$

Access to observations on both sides of a temporal gap makes bidirectional models particularly suitable for offline trajectory reconstruction and interpolation. Unlike forward LSTMs, however, they are non-causal and therefore require future observations to be available during inference.

These properties make LSTMs well suited for temporal motion modelling. In this thesis, LSTM-based models are used to learn puck motion from sequential puck locations, as described in Section 3.2.2.

2.1.6 Geometric Vision

Homography

A homography is a projective transformation that maps points from one image plane to another while preserving the straight-line structure of the scene. Assuming all points lie on a common plane, a homography provides a complete description of the geometric relationship between two camera views and can therefore be used to perform perspective correction, image registration, and planar scene reconstruction.

A homography is represented by a 3×3 matrix H that relates corresponding points in homogeneous coordinates,

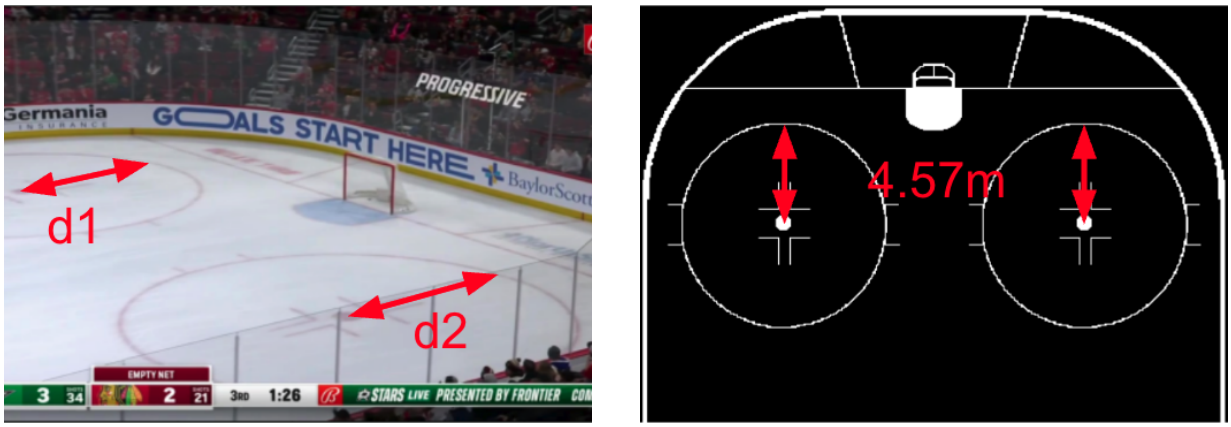
$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.9}$$

where (x, y) and (x', y') denote corresponding image coordinates, λ is an arbitrary scale factor arising from homogeneous coordinates, and $\mathbf{H}$ is the projective transformation matrix.

Given a sufficient number of point correspondences, the homography matrix can be estimated using the Direct Linear Transform (DLT), which formulates the estimation as a linear least-squares problem. Random Sample Consensus (RANSAC) is used to filter noise by iteratively estimating candidate homographies from randomly selected correspondences while rejecting geometric outliers. The resulting homography is subsequently refined using all inlier correspondences to obtain a robust estimate.

Planar homographies have become a fundamental component of sports analytics because many playing surfaces, including ice hockey rinks [23, 32, 54], can be approximated as planar environments. By estimating the homography between a broadcast image and a canonical rink template, player and puck detections can be projected into a common top-down coordinate system where metric distances and trajectories may be analyzed independently of the broadcast camera viewpoint. Figure 2.7 illustrates this transformation, showing how perspective distortion in the broadcast image causes equal real-world distances

to appear with different pixel lengths, while the corresponding rink-space representation preserves consistent metric distances across the playing surface.



(a) Broadcast-view image space.

(b) Top-down rink space.

Figure 2.7: **Perspective distortion in image and rink coordinate spaces.** In the broadcast view, the red arrows d_1 and d_2 mark equal physical offsets on the rink but appear with different pixel lengths due to perspective projection. After transforming the scene into rink space, the same offsets are represented from a top-down view, where equal real-world distances are mapped consistently across the playing surface.

Monocular 3D Reconstruction

Monocular 3D reconstruction aims to recover the three-dimensional geometry of a scene and the corresponding camera poses from monocular image sequences. Traditional approaches are based on Structure-from-Motion (SfM), which reconstructs sparse scene geometry by matching image features across multiple views while jointly estimating camera motion [61]. Recent learning-based methods instead infer dense scene geometry directly from images and simultaneously estimate camera poses, enabling dense point cloud reconstruction without explicit feature matching [67, 68]. In this thesis, monocular 3D reconstruction is used to recover a globally consistent representation of the ice surface, providing the geometric foundation for the temporally stable homography estimation described in Section 3.2.1.

2.2 Prior Works

2.2.1 Ice Hockey Puck Detection

There are four main methods for regressing the puck's location [42,51,76], often leveraging a combination of deep learning, temporal tracking, and/or classical detection techniques, and contextual cues.

Yang [76] highlights the shortcomings of the dated YOLOv3 [46] and Mask-RCNN [28] models on puck detection, notes their high false positive rates due to their model architecture being built around multiple object detection, and proposes a novel deep-learning architecture leveraging multi-headed learning and temporal features. The method requires a sequence of nine frames to regress a single detection, differentiating itself from the PLUCC model.

Pidaparthy et al.'s [42] system locates the location of play by regressing the puck location using a deep neural network. Their method leverages estimated player locations, optical flow, and the regressed puck location, enabling their system to move a camera to focus on a region of play. Pidaparthy et al. [42] note the need for more diverse and more extensive datasets and mention that player orientation and actions can be used to refine these systems further.

Most recently, Sarkhoosh et al. [51] propose an AI-based video cropping pipeline to tailor hockey video content for social media. Their method uses fine-tuned detection models such as Faster-RCNN [48] and YOLOv8 [62] to find regions of interest in the match by regressing the puck's location, noting that YOLOv8 X-Large outperformed all other models.

2.2.2 Ice Hockey Puck Tracking

In literature there are two methods [35,64] that deviate from raw puck detections and focus on spatio-temporal localization of the puck.

Vats et al.'s [64] PuckNet leverages a 3D convolutional neural network to regress a soft heatmap from short video clips representing the puck's general location. This network leverages temporal context to mitigate occlusion issues; however, it struggles with the puck's inherent small size and rapid motion, which can lead to mislocalizations. Furthermore, the model relies on the contextual cue that the puck is typically located in regions with high player density—an assumption that can result in localization errors when the

puck ventures into less congested areas. Furthermore, their model was only trained on the relative location of the puck and could not directly regress its precise location.

Li et al. [35] propose a tracking method that classifies puck motion into controlled and free-moving states. Their approach leverages image matching the puck for instances where it is visible and motion estimation during occlusions. However, Li et al. note the technique is prone to false positives because artifacts—such as the players' skates or noise in the background—can be misinterpreted as the puck.

2.2.3 Ice Rink Registration

Ice-rink registration in broadcast hockey video is commonly formulated as a per-frame homography estimation problem. Fani et al. [23] regress four control points on a canonical rink template from each frame using a ResNet-based model, recovering homographies via DLT. Because each frame is processed independently, temporal consistency is enforced only *post hoc* through smoothing of control-point trajectories, resulting in residual frame-to-frame jitter. Shang et al. [54] segment rink markings and iteratively refine homographies between broadcast frames and an overhead rink template, enabling generalization across rink geometries. Similarly, the HockeyRink dataset [32] introduces dense landmark annotations and keypoint-based homography estimation using YOLOv8, achieving high per-frame accuracy.

Across these approaches, homography estimation ultimately relies on single-frame visual evidence, with temporal stability either ignored or approximated via smoothing. These approaches introduce small but persistent coordinate inconsistencies that can significantly degrade downstream motion estimation. In contrast, our approach enforces temporal consistency by fitting a single planar surface to a 3D reconstruction across the entire sequence, yielding a stable mapping for puck-motion analysis.

Prior literature highlights three limitations that motivate the methodology developed in this thesis. First, existing puck detection methods remain constrained by the puck's limited visual appearance and make only limited use of contextual information. Second, puck tracking methods remain vulnerable to occlusion, rapid motion, and visually similar distractors when operating directly in image space. Finally, existing rink-registration approaches estimate geometry primarily on a per-frame basis, introducing temporal inconsistencies that complicate trajectory modelling. These limitations motivate the context-aware detection, temporally stable planar representation, and geometry-aware motion estimation methods introduced in the following chapter.

Chapter 3

System Overview and Methodology

This chapter presents the methodology developed for puck detection and motion estimation in broadcast hockey video. The proposed framework separates the problem into two complementary stages. First, context-aware puck detection estimates the puck location from individual broadcast frames by combining visual appearance with player-based contextual information. Second, puck detections are projected into a temporally stable planar representation of the rink, where recurrent models are used to estimate puck motion across time. Separating detection from motion estimation allows the detection model to exploit spatial context within individual frames while the temporal models reason about puck dynamics independently of broadcast camera motion.

3.1 Puck Detection Methodology

The first stage of the proposed framework addresses single-frame puck localization using the Player Localization Using Contextual Cues (PLUCC) model. PLUCC consists of three main components: (1) a feature pyramid encoder for extracting puck appearance features, (2) a context encoder for learning player-based contextual information, and (3) a gated decoder that combines these complementary representations to produce a puck localization heatmap. The following sections describe the contextual representation, model architecture, and objective function used to train the detector.

3.1.1 Context in Puck Detection

Relying solely on visual puck information is impractical for puck detection due to the puck's small size and the dynamic nature of the game, which often involves significant occlusion and motion blur, as shown in Figures 4.1c and 4.1a. Therefore, incorporating additional context is crucial to guide the model toward robust puck localization.

Prior research has revealed a significant correlation between puck location and players' positions on the rink [42, 64]. In Section A.1, we also highlight the relationship between player pose and puck location. However, directly extracting player pose, as explored in [2], is computationally expensive, and pose estimators often struggle under occlusion and motion blur.

To address these limitations, we use RGB player segmentation masks as an implicit contextual representation. Player bounding boxes are first extracted using a pretrained YOLOv5 [34] detector, after which the detected players are segmented using SAM2 [44]. The resulting segmentation preserves information about player location, shape, and orientation without requiring explicit pose estimation. Figure 3.1 illustrates this contextual input generation pipeline.

The contextual representation is not intended to identify the puck directly. Instead, it provides a spatial prior describing the configuration of players surrounding the play. While the puck's visual appearance may be weak or ambiguous because of its small size, motion blur, or visually similar rink artifacts, player positions and orientations remain informative about where the puck is likely to occur. The detector can therefore combine weak local appearance evidence from the broadcast image with broader information about the configuration of play. This complementary relationship motivates the separate appearance and context branches of the PLUCC architecture described in the following section.



Figure 3.1: **Context image (C_{RGB}) generation pipeline.** Given a broadcast image I_{RGB}, players are first localized using a pretrained YOLOv5 [34] detector. The resulting player detections are provided to SAM2 [44] to generate segmentation masks, which are combined to form the contextual input C_{RGB}. This representation preserves the spatial distribution, shape, and orientation of players while removing appearance information from the original broadcast frame.

3.1.2 PLUCC Model Architecture

The proposed PLUCC architecture, shown in Figure 3.2, is designed to combine two complementary sources of information. The feature pyramid encoder processes the full broadcast image and preserves fine-scale appearance information required to visually localize the puck. In parallel, the context encoder processes the player-only representation and learns features describing the spatial configuration of players surrounding the play. These representations are fused at multiple spatial resolutions by the gated decoder, allowing the final localization heatmap to depend jointly on local puck appearance and player-based contextual information.

Importantly, the context branch is not expected to independently determine the puck location. Rather, it provides contextual evidence that complements the appearance branch when visual puck features are weak or ambiguous. By maintaining the spatial structure of both representations during multi-scale fusion, the decoder can learn relationships between candidate puck features in I_{RGB} and the surrounding player configuration encoded by C_{RGB}.

Feature Pyramid Encoder

The Feature Pyramid Encoder extracts hierarchical appearance features from the broadcast image at multiple spatial resolutions. Following established approaches for small object detection [70], the encoder processes the full-resolution RGB image to preserve the fine

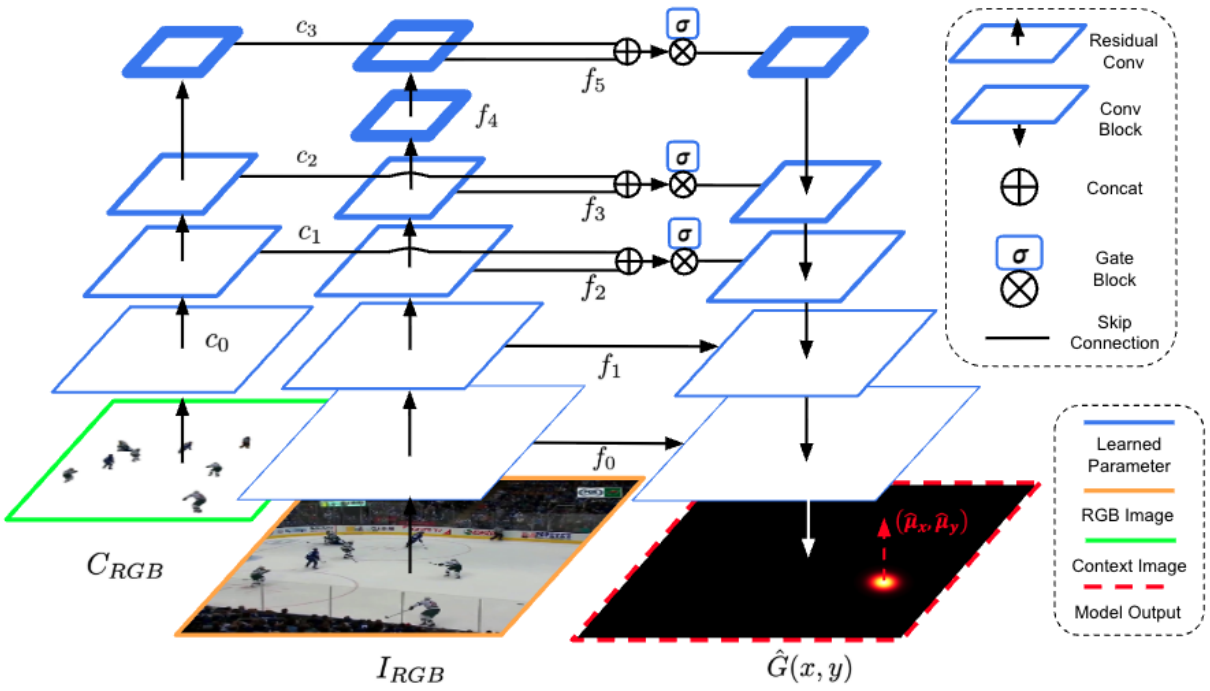


Figure 3.2: **PLUCC model architecture.** The full-resolution RGB frame I_{RGB} is processed by the feature pyramid encoder to extract visual puck features, while the half-resolution context image C_{RGB} is processed independently by the context encoder. Latent features from the feature pyramid and context encoders are denoted by f_n and c_n, respectively. The gated decoder combines these representations across multiple feature scales to produce the predicted Gaussian puck localization heatmap $\hat{G}(x, y)$.

spatial information required to localize the puck while incorporating higher-level semantic features from deeper layers of the network. This branch primarily learns visual features associated with the puck and its local surroundings, while the Context Encoder separately learns player-based contextual information.

The feature pyramid encoder is a modified ResNet-152 [29] backbone adapted from Wu et al. [72], selected based on its demonstrated performance for keypoint regression.

The input to the Feature Pyramid Encoder is the full-resolution RGB image, defined as $I_{RGB} \in \mathbb{R}^{B\times 3\times H\times W}$, where B denotes the batch size and H and W denote the image height and width, respectively. In this work, the input resolution is 720×1280 pixels.

The encoder produces a hierarchy of feature maps denoted by $\{f_0, f_1, f_2, f_3, f_4, f_5\}$. For the 720×1280 pixel input used in this work, the feature dimensions are

$$f_0 \in \mathbb{R}^{B\times 64\times 360\times 640}, \quad f_1 \in \mathbb{R}^{B\times 256\times 180\times 320},$$
$$f_2 \in \mathbb{R}^{B\times 512\times 90\times 160}, \quad f_3 \in \mathbb{R}^{B\times 1024\times 45\times 80},$$
$$f_4 \in \mathbb{R}^{B\times 2048\times 23\times 40}, \quad f_5 \in \mathbb{R}^{B\times 1024\times 23\times 40}.$$

Here, f_0 is obtained following the initial convolutional layer and retains the highest spatial resolution, while f_5 forms the bottleneck representation used to initialize the decoder. The intermediate features provide progressively lower-resolution and higher-dimensional representations of the input image.

Context Encoder

The context image, described in Section 3.1.1, is defined as $C_{RGB} \in \mathbb{R}^{B\times 3\times \frac{H}{2}\times \frac{W}{2}}$ and contains the RGB segmentations of all detected hockey players in I_{RGB}. For the input resolution used in this work, the context image has a spatial resolution of 360×640 pixels.

The Context Encoder processes C_{RGB} independently to produce three multi-scale feature maps, denoted by $\{c_1, c_2, c_3\}$. Their dimensions are

$$c_1 \in \mathbb{R}^{B\times 256\times 90\times 160},$$
$$c_2 \in \mathbb{R}^{B\times 512\times 45\times 80},$$
$$c_3 \in \mathbb{R}^{B\times 1024\times 23\times 40}.$$

The spatial resolutions of the context features are intentionally aligned with selected features from the Feature Pyramid Encoder. Specifically, c_1 matches the 90×160 spatial resolution of f_2, c_2 matches the 45×80 resolution of f_3, and c_3 matches the 23×40 resolution of the bottleneck feature f_5. Although the corresponding feature maps share the same spatial dimensions, their channel dimensions need not be identical because the features are concatenated before being processed by the gated decoder. This multi-scale alignment is important because the two branches serve different roles in localization. The feature pyramid features f_n retain visual evidence about candidate puck locations, whereas the context features c_n encode the spatial distribution and configuration of players. Aligning these representations spatially lets features describing a particular image region combine with contextual features describing the corresponding region of play. The decoder can therefore learn whether local appearance evidence is consistent with the surrounding player configuration rather than relying on puck appearance alone.

Gated Decoder

The gated decoder progressively reconstructs a high-resolution localization representation while combining appearance and contextual features at corresponding spatial scales. At decoding stages where both representations are available, the upsampled decoder features, feature-pyramid features, and context features are concatenated. This preserves the spatial information contained in each branch while allowing subsequent convolutional layers to learn joint representations of puck appearance and player configuration.

Because not every learned feature channel is equally informative for puck localization, a *GateBlock* is applied to the concatenated representation before convolution. The GateBlock learns a static per-channel weight (γ), allowing the network to control the relative contribution of the different learned feature channels during fusion. For an input tensor $F \in \mathbb{R}^{B\times C\times H\times W}$, the GateBlock computes a gating vector

$$g = \sigma(\gamma) \quad \text{with} \quad \gamma \in \mathbb{R}^{1\times C\times 1\times 1}, \tag{3.1}$$

where $\sigma(\cdot)$ is the sigmoid function. The gated output is then given by

$$\hat{F} = F \odot g, \tag{3.2}$$

with $\odot$ denoting element-wise multiplication.

The learned parameters γ remain fixed for a trained model during inference; therefore, the GateBlock is not an input-dependent attention mechanism in which the current player configuration dynamically switches individual RGB features on or off. Instead, spatial dependence is retained within the concatenated feature maps themselves, while the GateBlock learns the relative importance of their feature channels. The convolutional layers following each GateBlock then learn spatial relationships between the fused appearance and contextual representations. In this way, player context contributes spatial information about the configuration of play, while the gating operation controls how strongly the learned channels from the combined representation contribute to subsequent decoding.

When both the feature pyramid encoder feature f and the corresponding context encoder feature c are available, the fusion is defined as:

$$\tilde{F} = \mathrm{ConvBlock}\Big(\mathrm{GateBlock}\Big(\mathrm{Up}(\tilde{F}_{\mathrm{prev}}) \oplus f \oplus c\Big)\Big), \tag{3.3}$$

where $\mathrm{Up}(\cdot)$ upsamples the previous feature map $\tilde{F}_{\mathrm{prev}}$ to match the spatial dimensions of f and c using bilinear interpolation, $\oplus$ denotes concatenation, $\mathrm{GateBlock}(\cdot)$ applies

a sigmoid-based static gating operation, and ConvBlock$(\cdot)$ performs a 3×3 convolution followed by batch normalization and ReLU activation.

In stages without a corresponding context feature (e.g., for f_0 and f_1), the fusion is performed as:

$$\tilde{F} = \text{ConvBlock}\Big(\text{Up}(\tilde{F}_{\text{prev}}) \oplus f\Big). \tag{3.4}$$

The final heatmap logits are generated by upsampling the last refined feature map $\tilde{F}_{\text{final}}$ to the original input resolution and applying a 1×1 convolution:

$$\tilde{Z}(x, y) = \text{Conv}_{1\times1}\Big(\text{Up}(\tilde{F}_{\text{final}})\Big). \tag{3.5}$$

where $\tilde{Z}(x, y)$ represents the raw heatmap logits.

Lastly, the final predicted Gaussian heatmap $\hat{G}(x, y)$ is derived from normalizing and using a softmax operation over the logits spatial domain:

$$\hat{G}(x, y) = \frac{\exp(\tilde{Z}(x, y))}{\sum_{x',y'} \exp(\tilde{Z}(x', y'))}. \tag{3.6}$$

3.1.3 Objective Function

The PLUCC model is trained to minimize the distance between the predicted $\hat{G}(x, y)$ and ground truth $G(x, y)$ Gaussian heatmaps. To do so, we utilize Kullback-Leibler divergence loss ($\mathcal{L}_{KL}$) to penalize spatial displacement, encouraging the network to predict a peak precisely at the correct point.

To measure the similarity between the predicted and ground truth distributions, the KL divergence loss is computed as:

$$\mathcal{L}_{KL} = \frac{1}{B} \sum_{i=1}^{B} \sum_{x,y} G_i(x, y) \log \frac{G_i(x, y)}{\hat{G}_i(x, y)}, \tag{3.7}$$

where B is the batch size, and the loss is averaged across all samples in the batch to ensure stability.

Gaussian Heatmap Label. Gaussian heatmaps provide a soft training target with peak confidence at the object center. Inspiration for heatmap training stems from Tarashima et al.'s [57] baseline for sports ball detection, where they achieve enhanced accuracy by

employing position-aware model training with ground truth heatmaps, especially for small, fast-moving balls. This soft representation explicitly models spatial uncertainty and directs the model's attention precisely to the object center.

To derive a Gaussian heatmap label for training, the center of the ground truth bounding boxes (μ_x, μ_y) is the peak of the two-dimensional Gaussian. For each pixel (x, y) in the image, the Gaussian value is computed as:

$$G_{gt}(x, y) = \exp\left(-\frac{(x-\mu_x)^2 + (y-\mu_y)^2}{2\sigma^2}\right) \tag{3.8}$$

where σ^2 is the variance of the Gaussian label, a parameter that determines how soft or granular our target label is.

3.2 Puck Motion Estimation Methodology

This section outlines the puck motion estimation framework. It first covers the construction of a temporally stable planar representation of the rink using monocular 3D reconstruction and plane estimation. It then introduces the sequence-based models used to estimate puck trajectories within this planar coordinate system.

3.2.1 Temporally Stable Rink Plane Estimation

Rather than estimating homography independently for each frame, we leverage 3D reconstruction to recover a single, consistent rink plane across the entire sequence, thereby avoiding the frame-to-frame jitter inherent to single-frame 2D methods, which corrupts velocity and acceleration estimates in downstream tracking. This process consists of two stages: (1) 3D reconstruction with dynamic point filtering, and (2) planar surface localization, as illustrated in Figure 3.3.

3D Reconstruction and Dynamic Point Filtering.

Each broadcast frame is processed using the Pi3 monocular 3D reconstruction model [68], which produces: (1) a dense local point cloud $\mathbf{P}_{\text{local}} \in \mathbb{R}^{H \times W \times 3}$ in the camera coordinate

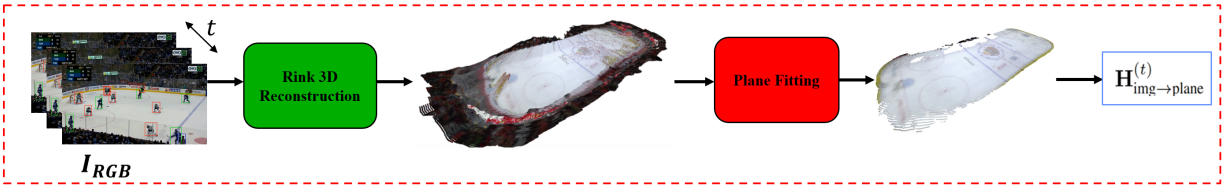


Figure 3.3: **Temporally stable rink-plane reconstruction pipeline.** The pipeline consists of two main stages. First, monocular 3D reconstruction recovers the scene geometry, after which points associated with dynamic objects are filtered to isolate the static rink environment. Second, a plane is robustly fitted to the cleaned 3D point cloud to recover a temporally stable representation of the playing surface.

frame, (2) its corresponding world-coordinate point cloud $\mathbf{P}_{\text{world}}$, and (3) the camera pose matrix $\mathbf{T} \in \mathbb{R}^{4\times4}$. Each local 3D point $\mathbf{p}_{\text{local}} \in \mathbb{R}^3$ is mapped into the global frame by

$$\mathbf{P}_{\text{world}} = \mathbf{R}\mathbf{P}_{\text{local}} + \mathbf{t}, \tag{3.9}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotation and translation components extracted from $\mathbf{T}$. Because Pi3 jointly reconstructs scene geometry and estimates camera poses across the sequence, the resulting world-coordinate reconstruction provides a common geometric reference from which we can estimate a single rink plane. This reconstruction aims not to recover metrically exact ice geometry, but to obtain a temporally consistent planar coordinate system for downstream puck-motion modelling.

Raw reconstructions include both the ice surface and dynamic objects (players, officials), which must be removed to isolate the rink plane. These dynamic points act as strong outliers, degrading downstream plane estimation. To address this, we use **PLUCC**'s context image binary mask $\mathbf{C_{RGB}}$ of player pixels. The mask is resized to match the point cloud resolution:

$$\mathbf{M}_{\text{local}} = \text{Resize}(\mathbf{C_{RGB}}, (H_{\text{local}}, W_{\text{local}}), \text{Nearest}). \tag{3.10}$$

where Nearest denotes nearest-neighbour interpolation, which preserves the binary nature of the segmentation mask during resizing. A value of $M_{\text{local}}[i,j] = 1$ denotes a pixel belonging to a segmented player.

Player-associated 3D points are then removed from the reconstruction:

$$\mathbf{P}_{\text{cleaned}}[i,j] = \begin{cases} \text{NaN}, & \mathbf{M}_{\text{local}}[i,j] = 1, \\ \mathbf{P}_{\text{local}}[i,j], & \text{otherwise.} \end{cases} \tag{3.11}$$

By filtering out these dynamic points, the resulting point cloud contains predominantly static rink geometry, yielding a substantially cleaner set of candidate points for the subsequent RANSAC-based plane fitting step. The filtered reconstruction is shown in Figure 3.4a.

This filtering specifically removes reconstructed geometry associated with segmented players; it does not assume that all remaining points belong to the ice surface. Static structures such as the boards, spectator regions, benches, and other reconstructed scene geometry may remain in the point cloud and are handled during the subsequent robust plane estimation stage.

Planar Surface Localization.

To recover the world–plane geometry, we fit a plane to the cleaned point cloud using RANSAC [24]. Although points associated with segmented players are removed during preprocessing, the reconstruction still contains substantial static geometry that does not belong to the ice surface, including the boards, stands, benches, and spectator regions. These structures can bias a least-squares plane estimate because they are not coplanar with the playing surface. RANSAC is therefore used to identify the dominant approximately planar structure while reducing the influence of reconstructed points that are geometrically inconsistent with that surface.

Each RANSAC iteration samples three non-collinear points ($\mathbf{p}_n$), estimates a candidate normal $\mathbf{n}_k$ and plane offset d_k,

$$\mathbf{n}_k = \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\|}, \qquad d_k = -\mathbf{n}_k \cdot \mathbf{p}_1, \tag{3.12}$$

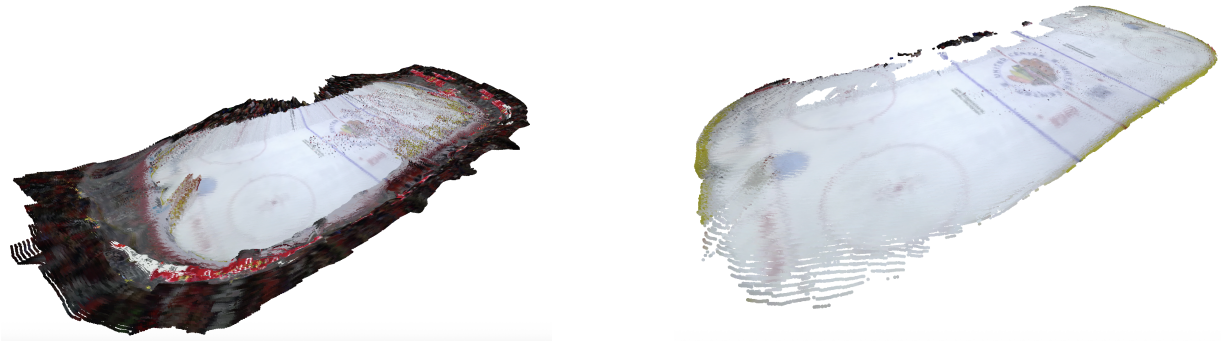
and evaluates its inliers. The best plane $(\mathbf{n}^*, d^*)$ is selected after N_{iter} iterations as the candidate supported by the largest set of geometrically consistent points, providing an estimate of the dominant approximately planar rink surface.

Once the rink plane has been estimated, we recover a mapping from the broadcast image to the plane by constructing a canonical 2D representation of the fitted surface. Let $\mathcal{P}_{\text{inlier}}$ denote the set of 3D points whose distance to the plane is less than $\epsilon = 0.001$. These points form a dense sampling of the reconstructed surface that is geometrically consistent with the estimated rink plane.

Flattening the Plane. We first construct an orthonormal basis $(\mathbf{u}, \mathbf{v})$ spanning the plane. All inlier points are projected onto this basis,

$$\mathbf{p}_{\text{plane}} = \begin{bmatrix} (\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{u} \\ (\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{v} \end{bmatrix}, \tag{3.13}$$

producing a 2D point cloud lying on the rink surface. The width of the plane image is chosen as the longest spatial extent of this flattened point set, while the height corresponds to the largest extent along the orthogonal direction. This produces a two-dimensional point cloud in the coordinate system of the fitted plane. Figure 3.4 illustrates the transition from the filtered monocular 3D reconstruction to the set of reconstructed points identified as consistent with the fitted rink plane. The flattened coordinates derived from these inliers form the 2D plane representation used for subsequent image-to-plane mapping.



(a) 3D ice hockey rink reconstruction after filtering dynamic player points.

(b) Planar surface localization showing points within $\epsilon = 0.001$ of the fitted plane.

Figure 3.4: **Rink plane estimation from monocular 3D reconstruction.** (a) Filtered 3D reconstruction containing predominantly static rink geometry after removing points associated with dynamic players. (b) RANSAC-based plane fitting identifies the dominant planar surface, with inlier points corresponding to reconstructed points within $\epsilon = 0.001$ of the estimated rink plane.

Constructing Image-to-Plane Correspondences.

For each inlier pixel (x_i, y_i) in the broadcast frame, Pi3 provides the associated 3D point $\mathbf{p}_i$. Since $\mathbf{p}_i$ is also mapped to a plane coordinate $\mathbf{p}_{\text{plane},i}$, we obtain direct correspondences

$$(x_i, y_i) \longleftrightarrow (u_i, v_i)$$

between image-space pixels and plane-image coordinates. Importantly, this step does not require camera intrinsics or knowledge of the camera matrix K; all correspondences are derived solely from the plane geometry and the Pi3-predicted 3D structure.

Estimating the Homography. Given these 2D–2D correspondences, the initial image-to-plane homography $\mathbf{H}_{\text{img}\to\text{plane}}$ is estimated using the Direct Linear Transform (DLT) with RANSAC to reject outliers:

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{H}_{\text{img}\to\text{plane}} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \tag{3.14}$$

The resulting planar representation prioritizes temporal consistency rather than metrically accurate reconstruction, absolute scale, or canonical orientation. Scale and rotation are intentionally unconstrained, and residual geometric error from the monocular reconstruction may remain in the estimated plane. However, because the same reconstructed plane provides the reference coordinate system throughout a sequence, downstream motion estimation can operate on relative puck displacements without requiring physical distance measurements.

Given temporally stable image-to-plane correspondences from the preceding section, each high-confidence PLUCC detection is mapped to normalized rink-plane coordinates. When detection is unreliable or absent, motion is recovered in this plane space and projected back to the broadcast image via the homography. This decouples short-term puck dynamics from camera motion and reduces the effect of frame-to-frame homography jitter on velocity estimates.

3.2.2 Motion Estimation LSTMs

To model the temporal dynamics of the puck, we employ Long Short-Term Memory (LSTM) networks [30], a recurrent architecture designed to capture temporal dependencies in sequential data. Rather than modelling motion directly in image space, the LSTMs operate on puck positions projected onto the temporally stable rink plane, reducing the influence of camera motion on the observed trajectory.

We model puck motion as a sequence of two-dimensional positions on the temporally stable rink plane. For each video sequence, let $\mathbf{p}_t = (u_t, v_t)^\top$ denote the puck center in normalized plane coordinates, obtained by transforming a PLUCC detection through

$\mathbf{H}_{\text{img}\to\text{plane}}$ and scaling by the sequence's plane-image extent. Sequences are processed in overlapping temporal windows, with each sequence maintaining its own coordinate frame consistent with the homography construction described above.

Three recurrent architectures share the same input representation and motion preprocessing but differ in how temporal context is aggregated: a unidirectional trajectory predictor for causal tracking, a dual-path model that fuses independent forward and backward temporal passes, and a bidirectional model for gap filling during occluded intervals. Table 3.1 summarizes the three architectures and their respective roles. Detailed layer-wise architectures and model dimensions are provided in Appendix B.2.

Table 3.1: **Recurrent motion models used in the pipeline.** All models operate on normalized plane coordinates $\mathbf{p}_t \in \mathbb{R}^2$.

Model	**Temporal context**	**Role in pipeline**
Forward LSTM	Causal (past $\to$ future)	Online tracking when PLUCC detection is missing; autoregressive prediction in plane space.
Dual-path LSTM	Independent forward and backward passes, fused per timestep	Offline trajectory refinement using full-sequence context without weight sharing across directions.
Bidirectional gap-filling LSTM	Bidirectional LSTM	Offline interpolation of contiguous detection gaps within a sequence.

Velocity-Augmented Input Representation

Raw plane positions alone under-specify fast puck motion. Before each recurrent module, we augment the position stream with finite-difference velocities. For a sequence $\{\mathbf{p}_t\}_{t=1}^{T}$, define

$$\mathbf{v}_t = \begin{cases} \mathbf{0}, & t = 1, \\ \mathbf{p}_t - \mathbf{p}_{t-1}, & t > 1. \end{cases} \tag{3.15}$$

Position and velocity are concatenated as $[\mathbf{p}_t; \mathbf{v}_t] \in \mathbb{R}^4$ and passed through a lightweight multilayer perceptron with ReLU activations. The resulting features are added residually to $\mathbf{p}_t$,

$$\tilde{\mathbf{p}}_t = \mathbf{p}_t + \mathrm{MLP}\big([\mathbf{p}_t; \mathbf{v}_t]\big), \tag{3.16}$$

yielding velocity-aware inputs $\tilde{\mathbf{p}}_t$ fed to the LSTM stack. This design injects local kinematic cues without altering the two-dimensional output space.

Forward LSTM Trajectory Predictor

The forward predictor encodes a causal history of puck motion. Given an input window $\{\tilde{\mathbf{p}}_t\}_{t=t-T+1}^{t}$, a stacked unidirectional LSTM produces hidden states $\mathbf{h}_t$, which are linearly projected to plane coordinates:

$$\hat{\mathbf{p}}_t = \mathbf{W}\mathbf{h}_t + \mathbf{b}. \tag{3.17}$$

During training, the model learns to forecast a future horizon autoregressively: the hidden state after the observation window is propagated forward, and each predicted position is fed back as the next input until the target horizon is reached. At inference, when PLUCC returns no detection (e.g., low-confidence heatmaps), the most recent T plane positions—detections and prior predictions—form the input window; a single-step autoregressive update yields $\hat{\mathbf{p}}_{t+1}$, which is mapped back to image space via $\mathbf{H}_{\text{img}\to\text{plane}}^{-1}$. Detected frames reset the history with ground-truth plane positions.

Dual-Path LSTM

Bidirectional LSTMs couple forward and backward passes through shared parameters. Our dual-path architecture instead runs two *independent* unidirectional LSTMs: one on $\{\tilde{\mathbf{p}}_t\}_{t=1}^{T}$ in chronological order and one on the time-reversed sequence. Let $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ denote their hidden states at time t (after reversing the backward path to align with frame indices). The fused representation uses learnable scalar weights α and β passed through sigmoid and normalized:

$$\mathbf{h}_t^{\text{fus}} = \frac{\sigma(\alpha)}{\sigma(\alpha)+\sigma(\beta)}\,\overrightarrow{\mathbf{h}}_t + \frac{\sigma(\beta)}{\sigma(\alpha)+\sigma(\beta)}\,\overleftarrow{\mathbf{h}}_t, \qquad \hat{\mathbf{p}}_t = \mathbf{W}\mathbf{h}_t^{\text{fus}} + \mathbf{b}. \tag{3.18}$$

Unlike a standard bidirectional LSTM, the two temporal directions do not share weights, allowing each path to specialize for causal prediction and retrospective smoothing, respectively. Training and autoregressive inference follow the same plane-coordinate protocol as the forward model; during open-loop prediction, the backward path retains its terminal hidden state while the forward path advances causally.

Bidirectional LSTM Gap Filling

Short occlusions and detection dropouts produce contiguous gaps in $\{\mathbf{p}_t\}$. The gap-filling model addresses these intervals using a single *bidirectional* LSTM over an entire sequence. Missing entries are first replaced with linear interpolation between the nearest observed

neighbors, producing an initial guess $\{\mathbf{p}'_t\}$ that is velocity-augmented as in Section 3.2.2. The bidirectional LSTM maps each timestep to a refined position $\hat{\mathbf{p}}_t$; observed detections are preserved, and only gap indices are overwritten:

$$\mathbf{p}_t^{\text{out}} = \begin{cases} \mathbf{p}_t, & t \notin \mathcal{G}, \\ \hat{\mathbf{p}}_t, & t \in \mathcal{G}, \end{cases} \tag{3.19}$$

where $\mathcal{G}$ is the set of gap frames. This offline, sequence-wise procedure exploits both past and future context within $\mathcal{G}$, which causal predictors cannot access at gap onset.

Gap filling is trained on trajectories with synthetically removed segments: random contiguous spans are masked, linearly initialized, and the network is supervised only on masked timesteps. This focuses learning on interpolation rather than replicating visible detections.

Training Objective and Supervision

All models are trained on puck trajectories derived from broadcast-style sequences with ground-truth bounding boxes. Detections are transformed into each sequence's rink-plane frame using the homography pipeline described above; positions are normalized to $[0, 1]^2$ per sequence. The forward and dual-path models minimize a masked mean-squared error between predicted and ground-truth future positions,

$$\mathcal{L}_{\text{traj}} = \frac{1}{|\mathcal{V}|} \sum_{(t,i)\in\mathcal{V}} \left\| \hat{\mathbf{p}}_t^{(i)} - \mathbf{p}_t^{(i)} \right\|_2^2, \tag{3.20}$$

where $\mathcal{V}$ indexes valid spatiotemporal entries in the batch. The gap-filling model uses the same metric but restricts the sum to synthetic gap indices $\mathcal{G}$. Optimization employs Adam with gradient clipping; trajectory models are trained with autoregressive rollouts (no teacher forcing) so inference and training dynamics align.

Integration with Detection and Plane Estimation

The full motion pipeline operates on broadcast sequences: (1) PLUCC produces a Gaussian heatmap for each frame; (2) the sequence homography maps detections to $\mathbf{p}_t$; and (3) the selected recurrent model supplies plane-space positions when detection fails or, for gap filling, refines an entire sequence offline before inverse mapping to image coordinates. Because the reconstructed planar coordinate system is not metrically calibrated, motion

supervision and inference rely on relative displacements in plane space rather than absolute physical distances or speeds. The resulting representation is therefore suitable for modelling temporally consistent puck motion, but does not directly provide measurements in real-world units.

The methods presented in this chapter define the complete framework for context-aware puck detection, temporally stable plane estimation, and geometry-aware motion modelling. Evaluating these components requires a dataset, representative baselines, and metrics that capture both localization accuracy and temporal tracking performance. The following chapter therefore establishes the experimental framework used to assess the proposed methods.

Chapter 4

Dataset, Baselines, and Metrics

This chapter defines the experimental framework used to evaluate the proposed puck detection and tracking methods. First, the VIP-PuckDataset and its training, validation, and test splits are described. Next, the detection and tracking methods used as experimental baselines are introduced. Finally, the evaluation metrics used to measure puck localization and tracking performance in both image and rink coordinate spaces are defined. Together, these components provide the basis for the experiments presented in Chapter 5.

4.1 Dataset

The VIP-PuckDataset contains 4,813 broadcast hockey sequences (∼150,000 frames) with varying frame rates, annotated with puck bounding boxes. Sequences are grouped by challenge type, including standard, blurry, yellow-line, and goal-line scenarios (Fig. 4.1). Typical sequences commonly contain only short intervals of visible-puck annotations before occlusion, and across all splits the puck is visible in only ∼68% of frames, reflecting the inherent difficulty of broadcast puck tracking.

Puck Detection and Tracking. 100,000 frames were used in training, and 30,000 for validation. The test set, comprised of 15,000 frames, and contains challenging scenarios, such as blurry (4.1a), goal-line (4.1e), and yellow-line (4.1f) pucks, to evaluate the model's performance under challenging conditions.

Table 4.1 summarizes VIP-PuckDataset composition. Challenge categories and their distribution within the dataset are highlighted in table 4.2 (standard broadcast, motion blur, yellow-line, and goal-line).

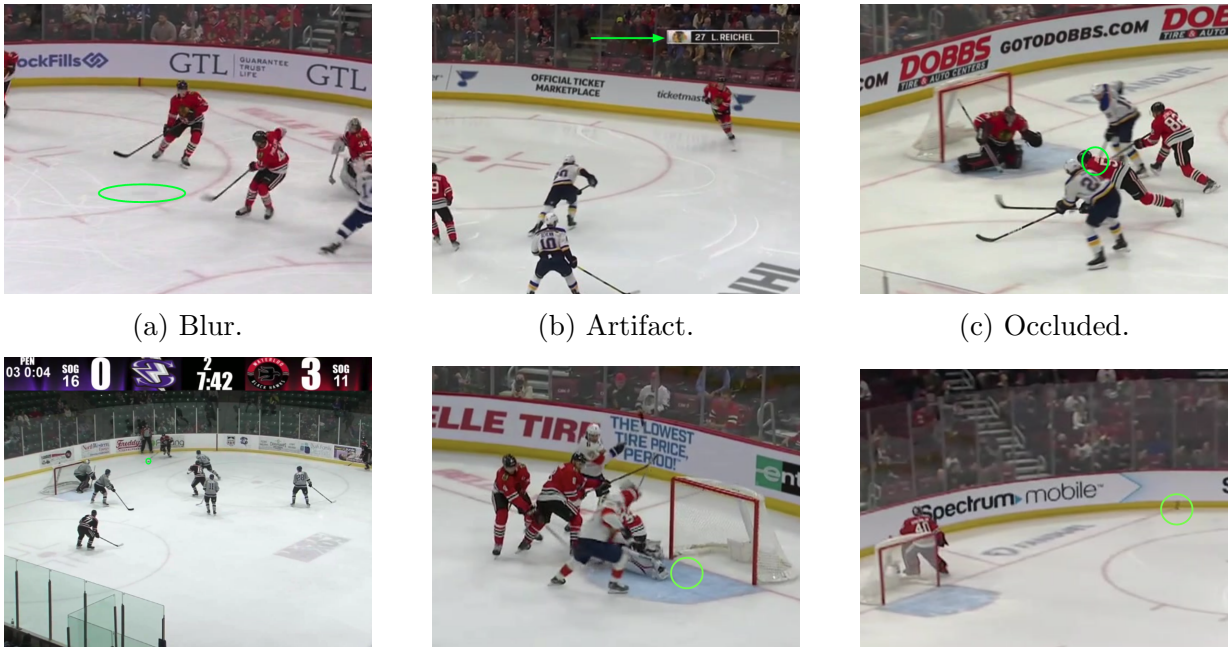
Table 4.1: **VIP-PuckDataset statistics.** Sequences contain per-frame puck bounding-box annotations. "Frames w/ puck" counts frames in which the puck is annotated as visible; remaining frames correspond to unlabeled or occluded intervals within each sequence.

Split	Seq.	Frames	w/ puck	Vis. (%)	Med. len.	Med. vis.	Mean len.
Train	2,541	101,623	71,492	70.4	16	13	40.0
Validation	726	30,040	20,369	67.8	16	13	41.4
Test	363	15,523	10,443	67.3	16	12	42.8
Labeled total	**3,630**	**147,186**	**102,304**	**69.5**	**16**	**13**	**40.5**
Other[†]	1,183	2,800	1,339	47.8	1	1	2.4
Full dataset	**4,813**	**149,986**	**103,643**	**69.1**	**12**	—	**31.2**

[†] Very short sequences; excluded from train/validation/test metrics above.

Table 4.2: **Challenge-type breakdown of labeled sequences.** Sequences are grouped according to their path/batch challenge tags. Values indicate the number of sequences belonging to each challenge type, with percentages representing the corresponding share of sequences within each dataset split.

Challenge	Train	Val.	Test	All labeled
Standard	1,679 (66.1%)	503 (69.3%)	233 (64.2%)	2,415 (66.5%)
Motion blur	666 (26.2%)	182 (25.1%)	97 (26.7%)	945 (26.0%)
Yellow-line	159 (6.3%)	31 (4.3%)	25 (6.9%)	215 (5.9%)
Goal-line	37 (1.5%)	10 (1.4%)	8 (2.2%)	55 (1.5%)



(a) Blur. (b) Artifact. (c) Occluded.

(d) Small. (e) Goal-line puck. (f) Yellow-line puck.

Figure 4.1: **Challenging puck detection scenarios in the VIP-PuckDataset.** Representative broadcast frames illustrating (a) motion blur, (b) broadcast artifact interference, (c) player occlusion, (d) the puck's small image footprint, (e) reduced visibility near the goal line, and (f) low contrast between the puck and yellow-line overlays.

Dataset limitations. VIP-PuckDataset clips are short: across labeled splits the median sequence length is only 16 frames (mean ≈40), and the median number of puck-visible frames per clip is 13 (75th percentile: 23; only 26% of clips contain 16–40 visible-puck frames). Coupled with heterogeneous broadcast frame rates in the source material (clips are commonly re-timed to 10–25 fps during preprocessing), these properties constrain what motion models can learn. Recurrent trackers and LSTMs in our pipeline use context windows of 10–20 frames—often comparable to or longer than an entire clip—so they primarily capture *short-horizon* interpolation and gap filling rather than sustained, long-term puck dynamics. Evaluation on the test split is stress-oriented (blur, yellow-line, and goal-line batches are over-represented relative to train), but it is not exclusively difficult footage: 64% of test sequences are tagged as standard broadcast. Reported tracking metrics should therefore be interpreted as performance on brief, occlusion-heavy snippets rather than full-game trajectory forecasting.

4.2 Baselines

4.2.1 Detection Baselines

We compare PLUCC against Faster R-CNN [48], YOLOv5 [45], the HockeyAI YOLOv8 model [31], and a heatmap-based FCN-ResNet152 [72]. Faster R-CNN, YOLOv5, FCN-ResNet152, and PLUCC were trained on the VIP-PuckDataset using the same training and evaluation splits. The FCN-ResNet152 additionally serves as an architectural ablation of PLUCC, isolating the contribution of contextual encoding and gated fusion. In contrast, the HockeyAI [31] model was not trained or fine-tuned on the VIP-PuckDataset and is evaluated directly as a publicly available, off-the-shelf detection baseline.

4.2.2 Tracking Baselines

We evaluate several state-of-the-art visual trackers with domain-specific fine-tuning: SiamFC [6], ARTrack [71], DiMP-50 [7], SuperDiMP-50 [18], ToMP-50 [38], PrDiMP-50 [19], and TaMOs-Swin Base [39]. Trackers are re-initialized after 10 consecutive missed frames, in accordance with standard evaluation practice.

4.3 Metrics

4.3.1 Puck Detection Metrics

Image Coordinate-Space Localization Error

To ensure a fair comparison across detection algorithms that produce heatmaps and those that produce bounding boxes, we calculate the pixel Euclidean distance ($\mathcal{D}_{pixel}$) between the predicted puck center $(\hat{\mu}_x, \hat{\mu}_y)$ and the ground truth center (μ_x, μ_y).

$$\mathcal{D}_{pixel} = \sqrt{(\hat{\mu}_x - \mu_x)^2 + (\hat{\mu}_y - \mu_y)^2} \tag{4.1}$$

The puck location is derived from a predicted heatmap ($\hat{G}(x, y)$) by taking the maximum point:

$$\hat{\mu}_x, \hat{\mu}_y = \arg\max_{(x,y)} \hat{G}(x, y). \tag{4.2}$$

Bounding box puck centers are taken from the center-most point of the bounding box such that:

$$\mu_x, \mu_y = \left(\frac{\hat{x}_{\min} + \hat{x}_{\max}}{2}, \frac{\hat{y}_{\min} + \hat{y}_{\max}}{2} \right). \tag{4.3}$$

Furthermore, we do not compute $\mathcal{D}_{pixel}$ for test samples that do not have a ground truth label. Models like YOLOv5 [45] and Faster-RCNN [48] can detect the object when it appears in the frame. The $\mathcal{D}_{pixel}$ metric is computed only for frames containing a ground-truth puck annotation. Frames without a ground-truth puck location are therefore excluded from the average localization error. This ensures a consistent comparison between bounding-box detectors, which explicitly predict object presence, and heatmap-based approaches, which directly regress puck location.

Average Precision is computed with a distance threshold $\tau \in \{5, 10, 25, 50\}$ pixels. A prediction is considered correct if the Euclidean distance $\mathcal{D}_{pixel}$ between the predicted puck center and the ground truth puck center is within the threshold τ. The mean Average Precision (mAP^{τ}) is computed as the mean of AP^{τ} values across all thresholds, providing a holistic measure of model performance across varying levels of localization accuracy.

Because the physical distance represented by a pixel displacement varies with camera viewpoint and image location due to perspective projection, $\mathcal{D}_{pixel}$ does not have a fixed conversion to a real-world distance. Consequently, image-space localization error is complemented by rink-space localization error, which measures prediction error after homographic projection into a common metric coordinate system.

Rink-Space Localization Error

The Rink-Space Localization Error (RSLE) compensates for the perspective distortion present in broadcast footage, where the same pixel error can correspond to vastly different real-world distances depending on the puck's location relative to the camera. Specifically, a pixel error ($\mathcal{D}_{pixel}$) near the camera, due to the viewing angle, may represent less of a physical distance on the rink compared to an error of $\mathcal{D}_{pixel}$ near the far boards. By transforming image coordinates into a standardized rink space, RSLE ensures that localization errors are measured in consistent physical units, removing the inherent bias of comparisons in image coordinate-space.

The RSLE metric for a single puck comparison can be formulated as:

$$RSLE = \sqrt{(\hat{x}_{rink} - x_{rink})^2 + (\hat{y}_{rink} - y_{rink})^2} \tag{4.4}$$

where $(\hat{x}_{rink}, \hat{y}_{rink})$ and (x_{rink}, y_{rink}) denote the predicted and ground-truth puck locations in rink coordinates, respectively.

Rink Space Localization Error Average Precision (AP^r) is the percentage of detections transformed to rink coordinates that lie within the puck radius, where $r = 3.81cm$ is the puck radius [40]. $AP^{r\times 2}$ is the percentage of detection within the puck diameter (twice the radius), and $AP^{r\times 4}$ is the percentage within twice the diameter.

Homography Estimation. We use Shang et al.'s [54] homography estimation process to map puck positions from the image plane to rink coordinates, for comparing detection model performance in metric units. Unlike our stable plane approach, this method computes temporally independent, metrically scaled and oriented correspondence between the 2D image coordinates and a predefined homographic space. We are only evaluating detection performance with this method, thus the temporal instability of this method is not a concern.

The transformation of the puck's image coordinates (μ_x, μ_y) into rink-space is performed in two steps. First, we warp the coordinates into a top-down view of the rink segmentation (with dimensions 720 by 1280), and then we scale these warped positions to match the rink dimensions.

First, the puck's position is represented in homogeneous coordinates:

$$\tilde{\mathbf{p}} = \begin{bmatrix} \mu_x \\ \mu_y \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{p}_x \\ \tilde{p}_y \\ \tilde{p}_z \end{bmatrix}. \tag{4.5}$$

Next, we apply the homography matrix to transform this point:

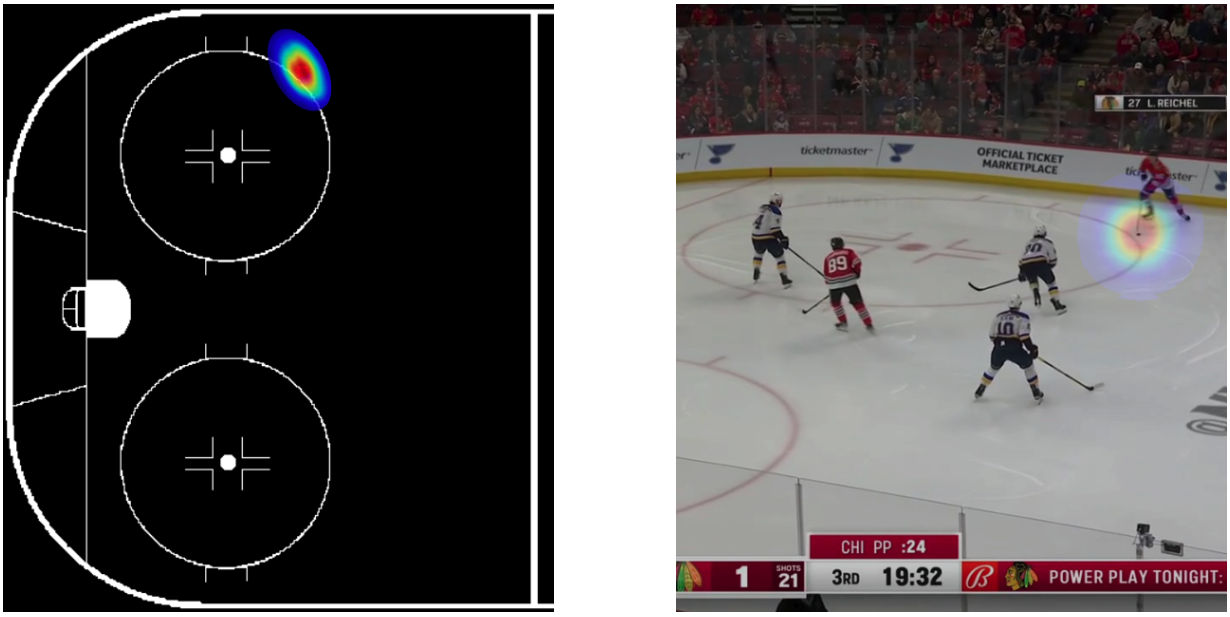
$$\tilde{\mathbf{p}}' = \mathbf{H}\,\tilde{\mathbf{p}}. \tag{4.6}$$

To convert the result back to inhomogeneous coordinates, we normalize by the third component:

$$(p_{\text{warp},x}, p_{\text{warp},y}) = \left(\frac{\tilde{p}'_x}{\tilde{p}'_z}, \frac{\tilde{p}'_y}{\tilde{p}'_z}\right). \tag{4.7}$$

Finally, using the dimensions of the homographic template (720 by 1280) and the standard NHL rink size (length $L = 61$ m and width $W = 25.9$ m) [56], we scale the warped coordinates to obtain the final rink-space coordinates:

$$x_{\text{rink}} = \frac{p_{\text{warp},x}}{1280} \times L, \quad y_{\text{rink}} = \frac{p_{\text{warp},y}}{720} \times W. \tag{4.8}$$



(a) Puck heatmap in homography.

(b) Puck heatmap in image coordinates.

Figure 4.2: **Puck detection in image and homographic coordinate spaces.** (a) The predicted puck heatmap transformed from image coordinates into the homographic rink representation. (b) The corresponding predicted puck heatmap overlaid on the broadcast frame in image coordinates.

4.3.2 Puck Tracking Metrics

All tracking metrics are computed in the *image coordinate space*, meaning that both predictions and ground truth annotations are evaluated directly in pixel coordinates without any projection to rink space. This ensures consistency with standard visual tracking benchmarks, but also implies that performance is affected by perspective distortion, scale variation, and camera motion inherent to broadcast footage.

AUC and mAP^{20} are used as the primary measures of tracking accuracy and are applied to both the visual object tracking and puck motion estimation experiments. Although the motion estimation models operate in plane space, their predictions are mapped back to image coordinates for evaluation. Using the same metrics for both experiments provides a common evaluation protocol, enabling direct comparison between conventional appearance-based visual tracking and the proposed geometry-aware motion estimation approach. Together, AUC and mAP^{20} capture complementary aspects of puck localization accuracy through bounding-box overlap and center-position error, respectively.

The Area Under the Curve (AUC) is computed from the success plot, which measures the Intersection over Union (IoU) between the predicted and ground truth bounding boxes across a range of thresholds. Specifically, for each IoU threshold $\tau \in [0, 1]$, the success rate is defined as the percentage of frames where the IoU exceeds τ. The success plot is obtained by sweeping τ over this interval, and the AUC corresponds to the integral of this curve, providing a threshold-independent measure of tracking accuracy. This metric rewards trackers that maintain consistently high overlap with the ground truth across varying levels of strictness.

In addition to overlap-based evaluation, we report mAP^{20}, which measures center-based localization accuracy. For each frame, the Euclidean distance between the predicted puck center and the ground truth center is computed in pixel space. A prediction is considered correct if this distance is within a fixed threshold of 20 pixels. The mAP^{20} score is then defined as the percentage of frames that satisfy this criterion over the full sequence. Unlike IoU-based metrics, mAP^{20} is less sensitive to bounding box size and instead focuses on the precision of the predicted puck location, making it particularly suitable for small-object tracking scenarios such as puck tracking.

Finally, we evaluate tracker robustness through the average number of restarts per sequence. A restart is triggered when the tracker loses the target, defined as the predicted bounding box having an IoU below 0.1 with the ground truth for a sustained period of 10 consecutive frames. Once this condition is met, the tracker is re-initialized using the ground truth annotation. The reported restart metric is averaged across all sequences and reflects the tracker's ability to maintain lock on the puck over time. Lower restart counts indicate greater temporal stability and robustness to occlusions, motion blur, and distractors.

Chapter 5

Experiments and Results

This chapter evaluates the proposed methods through three sets of experiments. First, the PLUCC model is evaluated against existing puck detection approaches to determine the effectiveness of incorporating player-based contextual information. Second, modern visual object trackers are evaluated on broadcast hockey sequences to assess their ability to maintain puck trajectories under challenging visual conditions. Finally, the proposed temporally stable planar representation and recurrent motion models are evaluated to determine whether modelling puck motion in plane space provides a more stable alternative to conventional image-space tracking. Together, these experiments evaluate the detection, tracking, and motion estimation components of the proposed framework.

5.1 Detection Experimental Setup and Results

5.1.1 Puck Detection Experiment Implementation Details

The PLUCC model consists of 114 million trainable parameters and was trained on an NVIDIA H100 over 25 epochs and with a batch size of 10. At inference, PLUCC and context image generation operate together at 6.03 frames per second. The feature pyramid encoder input (I_{RGB}) size was 720 by 1280 pixel (height by width) images, and the context encoder was fed 360 by 640 context images (C_{RGB}). The initial learning rate was 0.001 and would be reduced by a factor of ten if no improvement over the validation set occurred for five epochs. Lastly, a Gaussian standard deviation (σ) of 5.0 was used, with further reasoning highlighted in Section A.1. Training data augmentation included random flipping, Gaussian blurring, added noise, and normalization.

Context-driven Dropout

To encourage the PLUCC model to exploit contextual information rather than relying primarily on appearance features, I_{RGB} is withheld during training with a probability of $p_{drop} = 1\%$. During these iterations, the network must regress the puck location solely from contextual cues encoded by the positions and orientations of the players. This encourages the context encoder to learn features that remain informative when direct visual evidence of the puck is weak or unavailable. The effectiveness of context-driven dropout is evaluated through an ablation study in Section A.1.

5.1.2 Detection Results

The results in Table 5.1 compare our PLUCC method with other commonly used single frame puck detection baseline models [45, 48], and the FCN-ResNet152 [72] in image coordinates on the VIP-PuckDataset test set. Our model outperforms baseline object detection models across all values of τ with a 12.3% increase in mAP^{τ} from Faster-RCNN, the best-performing object detection model. Furthermore, the 3.9% increase in mAP^{τ} between the FCN-ResNet152 and PLUCC highlights the enhancement gained from incorporating the context encoder and gated decoder in the model architecture. However, this enhanced detection performance is counterbalanced by the fact that the inference speed of our model is heavily limited by the preprocessing required to generate the C_{IRG}. Figure 5.1 shows the model's Gaussian heatmap outputs, demonstrating robust detections under challenging circumstances.



(a) Blur: full frame



(b) Blur: zoomed



(c) Artifact: full frame



(d) Artifact: zoomed



(e) Occlusion: full frame



(f) Occlusion: zoomed



(g) Yellow-line: full frame



(h) Yellow-line: zoomed

Figure 5.1: **Qualitative PLUCC detection results under challenging broadcast conditions.** Each row shows a full broadcast frame (left) and a magnified view of the corresponding puck localization (right).

Table 5.1: **Performance comparison of puck detection models in image coordinate space.** The table metrics include mean Average Precision (mAP^{τ}) for all ($\tau \in \{5, 10, 25, 50\}$), Average Precision for each ($AP^{\tau \in \{5,10,25,50\}}$), the average Euclidean distance error ($\mathcal{D}_{pixel}$) between predicted and ground truth puck locations, and average inference speed in frames per second (FPS).

Method	mAP^{τ}	AP^{5}	AP^{10}	AP^{25}	AP^{50}	$\mathcal{D}_{pixel}$	FPS
YOLOv5-Large [45]	56.3	51.5	57.5	57.9	58.4	55.2	-
Faster-RCNN [48]	71.3	69.6	70.9	72.0	73.0	71.4	32.3
HockeyAI (YOLOv8)* [31]	41.5	41.3	41.4	41.5	41.7	81.3	60.1
FCN-ResNet152($\sigma = 5$) [72]	79.6	74.8	79.9	81.3	82.4	52.11	45.8
PLUCC ($\sigma = 15$)	81.6	76.2	82.0	83.7	84.7	48.8	6.0
PLUCC ($\sigma = 5$)	**83.5**	**82.2**	**83.6**	**84.2**	**85.0**	**47.0**	6.0

*Off-the-shelf model evaluated without training or fine-tuning on the VIP-PuckDataset.

Table 5.2: **Comparison of puck detection models in homographic rink-space coordinates.** AP^{r} represents the average precision when the predicted puck location falls within the puck radius ($r = 3.81$ cm), $AP^{r \times 2}$ measures average precision when the prediction is within twice the puck radius (puck diameter), and $AP^{r \times 4}$ measures average precision of predictions within four times the radius (twice the diameter). The average rink-space localization error is the $RSLE_{avg}$ metric in meters.

Method	AP^{r}	$AP^{r \times 2}$	$AP^{r \times 4}$	$RSLE_{avg}(m)$
YOLOv5-Large [45]	0.17	2.69	43.11	3.14
Faster-RCNN [48]	18.58	19.14	20.06	3.76
HockeyAI (YOLOv8)* [31]	18.87	32.71	40.9	1.58
FCN-ResNet152($\sigma = 5$) [72]	41.85	54.23	62.74	3.89
PLUCC ($\sigma = 15$)	20.69	40.35	67.68	1.10
PLUCC ($\sigma = 5$)	**43.59**	**62.94**	**81.23**	**1.05**

*Off-the-shelf model evaluated without training or fine-tuning on the VIP-PuckDataset.

Results shown in Table 5.2 highlight the PLUCC method's superior detection accuracy in the rink-space coordinates, surpassing the strongest baseline object detector, Faster-RCNN [48] by 25.01% in mAP^{r}, 43.8% in $mAP^{r \times 2}$, and is on average 2.71 meters closer to the puck. Furthermore, the model outperforms the FCN-ResNet152 [72] by 1.74% in

mAP^r, demonstrating the strength of the standalone feature pyramid network in localization and the refinement the context encoding provides.

5.2 Puck Tracking Experimental Setup and Results

5.2.1 Puck Tracking Experiment Implementation Details

All tracking experiments use 1280×720 pixel broadcast footage. Model fine-tuning took place on 2 x A6000 GPUs, and the selected backbone for each network was the most extensive pre-trained backbone available. We also froze most model backbones during fine-tuning to prevent overfitting and stabilize training on our small dataset, while allowing later layers and heads to specialize. Lastly, all models were fine-tuned for 30 epochs with a stepwise learning rate schedule that decays the learning rate at 15 and 25 epochs, with a decay factor $\gamma = 0.2$. Specific per-model training parameters are outlined in Table B.1.

Artificial occlusion. To evaluate tracker robustness under occlusion, we follow the synthetic occlusion protocol of Xu et al. [75]. For each frame, the ground-truth puck bounding box region is masked by replacing pixel values with the mean RGB computed from a surrounding ring of radius $r = 5$ pixels. This removes direct visual evidence of the puck while preserving local background statistics, simulating realistic occlusion scenarios without introducing unnatural artifacts. The first three frames of each sequence remain unoccluded to allow tracker initialization.

5.2.2 Puck Tracking Results

Table 5.3 presents a comparison of image-space visual trackers under both standard and artificial occlusion settings. Under standard evaluation, PrDiMP achieves the highest AUC (31.25%), while TaMOs attains the best localization precision (mAP^{20}) and the fewest tracker restarts. However, under artificial occlusion, all visual trackers exhibit a substantial drop in performance, with increased restart frequency and degraded localization accuracy, highlighting their reliance on visual appearance.

Global vs. Local Search Behavior. A key distinction between these models lies in how they process spatial context. TaMOs performs *global frame-level reasoning*: the

Table 5.3: **Tracking performance of image-space visual trackers.** *Standard test:* unmodified GT. *Artificial occlusion:* GT puck region masked using ring-based mean RGB (r=5 px).

	Standard			Artificial Occlusion		
Method	**AUC(%)**	**mAP^{20}(%)**	**Restarts**	**AUC(%)**	**mAP^{20}(%)**	**Restarts**
Siamese-FC	22.09	34.46	1.33	-	-	-
ToMP-50	25.38	46.11	1.12	6.59	11.64	2.07
ARTrack	20.0	52.2	1.35	**9.19**	24.3	**1.02**
DiMP-50	27.05	40.61	1.29	5.12	8.20	2.21
Super DiMP-50	29.85	44.75	1.20	6.74	11.14	2.11
PrDiMP-50	**31.25**	47.75	1.12	6.68	11.17	2.09
TaMOs-Swin Base	14.66	**55.65**	**0.64**	6.43	**26.96**	1.13

Transformer backbone encodes the entire image at once and predicts object-specific filters in parallel. This global context allows TaMOs to re-detect the puck after tracking failures without requiring the target to remain within a localized *search region*, contributing to its lower restart frequency relative to the other evaluated trackers. Because Siamese-FC, DiMP-family trackers, ToMP, and ARTrack restrict inference to a cropped neighborhood around the last known location, any tracking failure that pushes the true puck position outside this search space results in irreversible drift. This explains why TaMOs exhibits the fewest restarts (0.64 per sequence), while template-based trackers fail to recover after losing the puck.

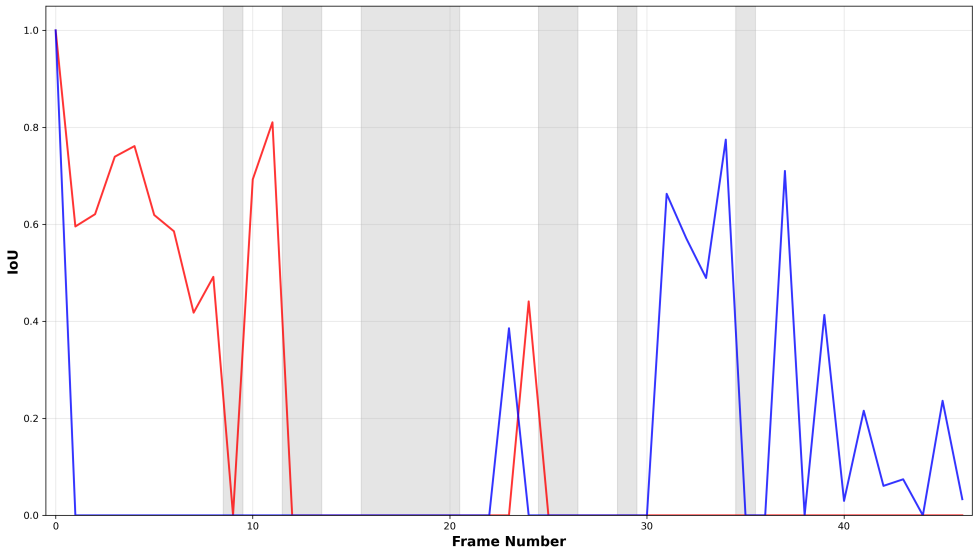


Figure 5.2: **Tracker performance during puck occlusion.** Intersection over Union (IoU) across frames for **PrDiMP** (red) and **TaMOs** (blue) over a representative sequence. Grey shaded regions denote intervals in which the puck is occluded and ground-truth annotations are unavailable; IoU values within these intervals therefore do not represent valid localization measurements.

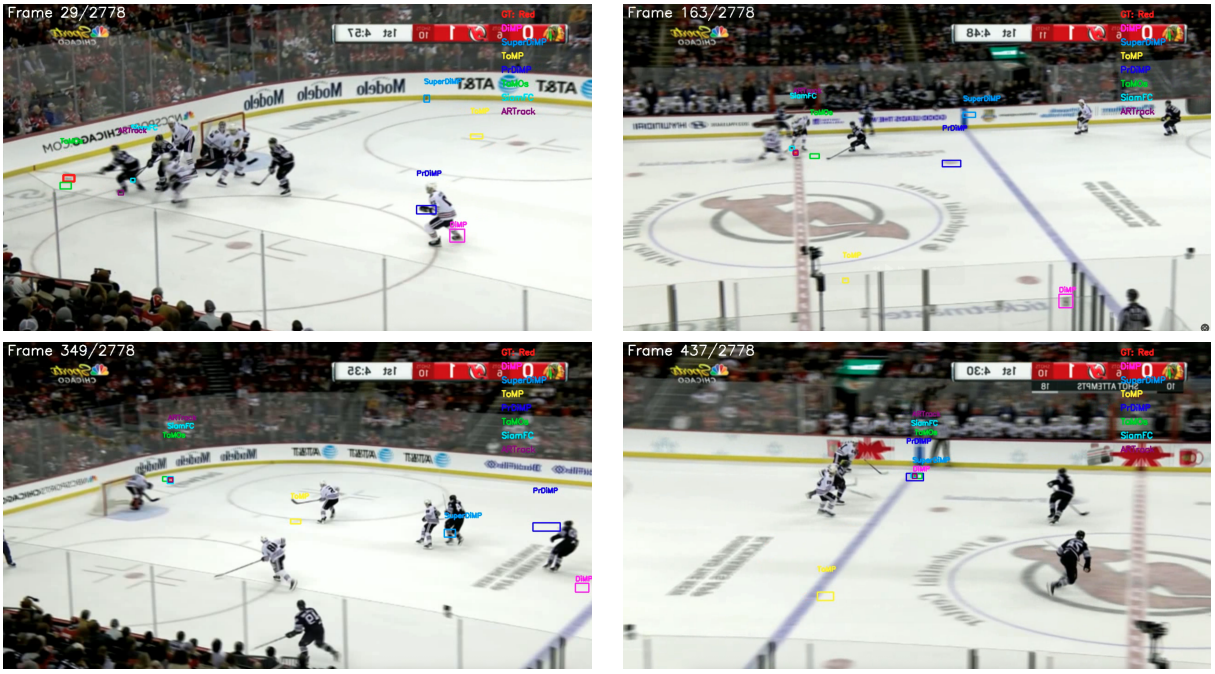


Figure 5.3: **Examples of visual tracking failures.** Predictions from multiple visual trackers are overlaid with colored bounding boxes: **red** = ground truth, **magenta** = DiMP, **orange** = SuperDiMP, **cyan** = ToMP, **blue** = PrDiMP, **green** = TaMOs, **yellow** = SiamFC, and **purple** = ARTrack.

Occlusions and Recovery Dynamics. Figure 5.2 illustrates the different recovery behaviours of PrDiMP and TaMOs over a sequence. PrDiMP maintains relatively high overlap with the ground truth and recovers successfully after the first occluded interval. However, after losing the puck around frame 12, its IoU remains at or near zero for the remainder of the sequence, aside from a brief localization near frame 24. The PrDiMP curve overlaps the horizontal axis from approximately frames 30–45, indicating that the tracker has lost the puck and fails to re-acquire it despite subsequent observations.

TaMOs exhibits the opposite behaviour. Although it loses the puck early in the sequence, it repeatedly re-acquires the target later, most notably after frame 30, where several periods of substantial overlap are observed. This ability to recover after tracking failure is consistent with its global frame-level search strategy, which does not require the puck to remain within a local search region centred on the previous prediction. In contrast, once PrDiMP drifts sufficiently far from the puck, the target may fall outside its local search region, preventing subsequent recovery. These recovery dynamics help explain the lower average restart frequency observed for TaMOs in Table 5.3.

Failure Modes: Drift, Stick-Locking, and Small-Object Ambiguity. Qualitative examples in Figure 5.3 highlight failure modes common across the evaluated trackers. The most severe issues occur in models like Siamese-FC, whose fixed-template matching causes them to "lock onto" visually similar distractors such as hockey sticks, skates, or patchy ice textures. When the puck passes behind a stick or merges with player clutter, these trackers frequently commit to the wrong object and never recover.

Additionally, many trackers struggle under *extreme motion blur*, where the puck becomes both aliased and elongated. Architecturally, this weakness is linked to the use of early convolutional layers with a stride of two, which discards fine-grained spatial detail that is critical for detecting a 3–5 pixel object in broadcast footage. Even high-performing modern architectures exhibit degraded confidence under these conditions, leading to sudden drift or delayed recovery.

5.3 Puck Motion Estimation and Planar Surface Experiments

5.3.1 Planar Surface Stability Experiments

To isolate the role of geometric stability, we compare two Kalman filter–based tracking pipelines that differ only in how planar coordinates are constructed. The first uses per-frame homographies estimated independently for each frame, while the second uses a single temporally stable planar representation derived from 3D reconstruction. As shown in Table 5.4, the stable planar representation significantly improves all tracking metrics, reducing localization error and tracker restarts. This experiment demonstrates that homography stability is critical for reliable motion estimation.

Table 5.4: **Effect of homography stability on Kalman filter puck tracking performance.** Tracking accuracy and stability are reported for per-frame homography estimation and the proposed temporally stable planar surface representation.

Metric	Per-Frame Homography [54]	Planar Surface
$\mathcal{D}_{pixel}$ (px)	183.20	**42.96**
mAP^{20}(%)	33.88	**51.34**
AUC (%)	5.17	**7.39**
Avg. Restarts / Seq	2.10	**1.54**
Total Restarts	762	**560**

5.3.2 Puck Motion Estimation Experiments

Table 5.5 summarizes the performance of the evaluated recurrent models. Among the trajectory-only methods, the Bidirectional LSTM achieves the highest overall tracking accuracy, obtaining an AUC of 30.9%, while the Forward LSTM achieves the highest localization precision with an mAP^{20} of 48.8%. The Dual-Path LSTM provides comparable precision but does not improve overall trajectory quality. These results indicate that the proposed planar representation captures sufficient temporal structure for sequence models to recover meaningful puck motion despite relying exclusively on projected coordinates.

Table 5.5: **Tracking performance of trajectory-based planar models.** Models operate solely on projected puck coordinates and are evaluated under standard conditions without image-space occlusion.

Method	**AUC(%)**	**mAP^{20}(%)**	Restarts
Forward LSTM	24.4	**48.8**	-
Bidirectional LSTM	**30.9**	46.0	-
Dual-Path LSTM	22.2	46.2	-
Forward LSTM (with restarts)	**46.0**	**74.1**	0.85

To provide a direct comparison with the visual object trackers, we additionally evaluate the Forward LSTM using the same restart protocol defined in Section 4.3.2. Under this protocol, a restart is triggered when the predicted puck location fails to maintain sufficient

overlap with the ground truth for 10 consecutive frames, after which the model is re-initialized using the ground-truth puck location. Applying this restart protocol increases the AUC from 24.4% to 46.0% and the mAP^{20} from 48.8% to 74.1%, while requiring an average of 0.85 restarts per sequence. Using the same restart criterion ensures that the Forward LSTM and image-space visual trackers are evaluated under a consistent recovery protocol.

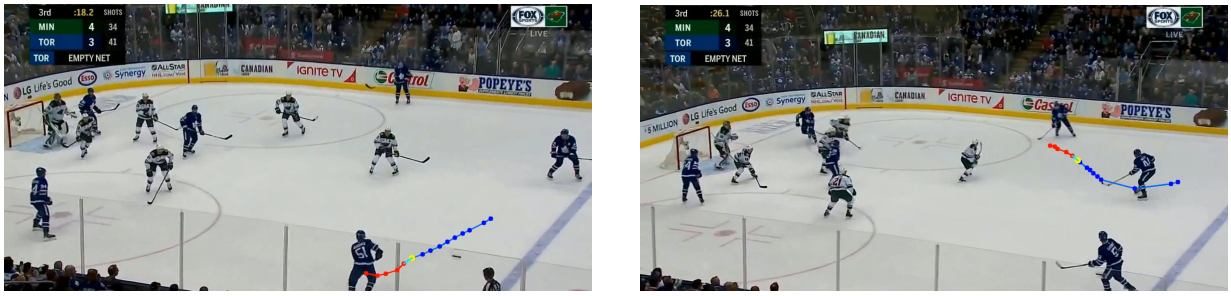


Figure 5.4: **Forward LSTM trajectory predictions.** The yellow marker denotes the current puck location, **blue** points show past puck positions, and **red** points represent the future puck trajectory predicted by the forward LSTM.

Qualitative examples of the predicted trajectories are shown in Figure 5.4. The forward LSTM produces smooth future trajectory estimates that generally follow the underlying puck motion. Although prediction uncertainty naturally increases as the forecasting horizon grows, the learned trajectories remain physically plausible within the normalized rink coordinate system. This behaviour highlights an advantage of the proposed geometric representation: by removing camera-induced motion and perspective distortion, the recurrent model learns the puck's dynamics rather than compensating for changes in the broadcast viewpoint.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Accurate localization and tracking of small, fast-moving objects remain among the most challenging problems in sports computer vision. Although recent advances in deep learning have improved object detection and visual tracking, broadcast ice hockey continues to pose distinct difficulties because of the puck's extremely small size, rapid motion, frequent occlusions, motion blur, and continuously changing camera viewpoints. These problems limit the effectiveness of conventional appearance-based methods and motivate tracking systems that exploit additional contextual and geometric information.

This thesis presented an end-to-end framework for broadcast ice hockey puck localization and tracking that progressively addresses these limitations through three complementary contributions. First, we introduced a context-aware puck detector, Player Localization Using Contextual Cues (PLUCC), to improve single-frame puck localization by incorporating player segmentation as a contextual prior. By combining appearance features with semantic context, the proposed architecture demonstrated improved localization effectiveness over appearance-only detectors while continuing efficient inference. These results show the importance of contextual reasoning for small object detection, particularly under challenging conditions such as occlusion, motion blur, and low contrast.

The second contribution investigated whether recent appearance-based visual object trackers can maintain reliable puck trajectories in broadcast hockey. The assessment demonstrated that, despite considerable progress in generic visual object tracking, modern trackers remain limited when applied to hockey puck tracking. All evaluated methods

showed frequent tracking failures, drift, and re-initialization, showing that improvements in appearance modelling alone are insufficient for reliable long-term puck tracking in broadcast video.

The final contribution proposed a geometry-aware tracking method based on temporally stable planar motion estimation. Rather than representing puck motion in image coordinates, we used monocular 3D reconstruction to recover a consistent rink-surface representation, from which we estimated a single stable planar coordinate system. By projecting detections into this coordinate system, we substantially reduced camera-induced motion and perspective distortion, producing trajectories that more closely reflect the puck's true physical motion. Experiments showed that this representation improved the stability of downstream motion estimation and enabled trajectory prediction using lightweight recurrent neural networks that operate solely on projected puck coordinates.

Collectively, this thesis shows that robust puck tracking requires more than increasingly sophisticated appearance models. Instead, incorporating contextual information for detection and representing motion within an appropriate geometric coordinate system provides a more reliable basis for tracking. Although developed for broadcast ice hockey, the underlying planar surface localization method is largely sport-agnostic and can be extended to other broadcast sports involving small balls or pucks, including soccer, tennis, basketball, and baseball. More broadly, this work shows the importance of combining semantic reasoning with geometric scene understanding when designing computer vision systems for dynamic broadcast environments.

Despite these contributions, several limitations remain. The proposed geometric method relies on the quality of the monocular 3D reconstruction, which may degrade under extreme close-up shots or visually ambiguous scenes. Furthermore, the proposed motion estimation framework accounts only for planar puck motion and does not explicitly model airborne trajectories or puck interactions with players and the boards. Finally, the recurrent trajectory models are intentionally lightweight and do not exploit richer temporal representations that may further improve long-term prediction, a limitation further exacerbated by the short, variable-frame-rate video sequences the models were trained on.

To summarize, this thesis advances broadcast hockey analytics by introducing a context-aware detection framework, demonstrating the shortcomings of current appearance-based tracking algorithms, and proposing a geometry-aware motion representation for stable puck tracking. Together, these contributions lay the groundwork for future research on geometrically informed sports analytics systems and highlight the critical role of scene geometry in enabling robust long-term object tracking.

6.2 Future Work

The findings of this thesis suggest several directions for extending the proposed framework. In particular, future work may expand the generality of the planar tracking framework, incorporate richer contextual information into both detection and motion estimation, and improve long-term trajectory modelling through more comprehensive data and temporal architectures.

1. **Extension to other planar sports.** The geometry-aware tracking framework developed in this thesis is largely independent of ice hockey-specific scene geometry. The planar representation is recovered directly from monocular 3D reconstruction and does not require a predefined rink template or sport-specific geometric calibration. Consequently, the same methodology could be applied to other sports in which the object of interest moves predominantly over a planar playing surface, such as soccer, basketball, tennis, or lacrosse. Future work should evaluate the framework across multiple sports and broadcast environments to determine how well the proposed geometric representation generalizes without domain-specific fine-tuning.

2. **Explicit contextual representations for puck detection.** PLUCC demonstrates that player segmentation provides useful contextual information for puck localization; however, the segmentation masks encode player pose and orientation only implicitly. Future work could incorporate more explicit representations of player behaviour, such as body orientation, stick position, or estimated gaze direction derived from 3D human pose. These cues could provide stronger information about where a player is directing their attention or interacting with the puck, particularly when the puck itself is occluded or visually indistinguishable. An adaptive mechanism could additionally determine when direct puck appearance is reliable and when greater emphasis should be placed on contextual information.

3. **Context-aware motion estimation.** The motion estimation models presented in this thesis operate solely on the puck's projected planar coordinates and therefore model trajectory dynamics independently of the players surrounding the puck. However, puck motion is strongly influenced by player interactions. Player position and orientation, stick position, and their motion relative to the puck could therefore provide valuable information when predicting puck trajectories through periods of missing visual evidence. Extending the planar representation to include these contextual features would allow the temporal model to reason jointly about puck and player interactions, potentially improving trajectory estimation during passes, shots, deflections, and occlusions.

4. **Long-term trajectory and occlusion modelling.** A primary limitation of the current dataset is the short duration of its annotated puck sequences and the absence of puck annotations during complete occlusions. Consequently, the recurrent models developed in this thesis are primarily evaluated on short-term motion estimation and cannot be trained to explicitly recover long-term trajectories through occlusion. Future datasets should contain longer continuous sequences and provide puck position annotations through occluded frames where the puck location can be reliably established. Such data would also enable the investigation of more recent sequence modelling architectures, including Transformers, which are capable of modelling longer-range temporal dependencies than the lightweight recurrent models considered in this work. Combined with richer player-based contextual information, these models could learn longer-term puck dynamics and recover trajectories through extended periods in which direct visual observations are unavailable.

References

[1] Yasuo Ariki, Tetsuya Takiguchi, and Kazuki Yano. Digital camera work for soccer video production with event recognition and accurate ball tracking by switching search method. In *2008 IEEE International Conference on Multimedia and Expo*, pages 889–892, 2008.

[2] Bavesh Balaji. Language guided out-of-bounding box pose estimation for robust ice hockey analysis. Master of applied science thesis, University of Waterloo, Waterloo, Ontario, Canada, 2024.

[3] Bavesh Balaji, Jerrin Bright, Yuhao Chen, Sirisha Rambhatla, John Zelek, and David Clausi. Seeing beyond the crop: Using language priors for out-of-bounding box keypoint prediction. *Advances in Neural Information Processing Systems*, 37:102897–102918, 2024.

[4] Bavesh Balaji, Jerrin Bright, Sirisha Rambhatla, Yuhao Chen, Alexander Wong, John S. Zelek, and David A. Clausi. Domain-guided masked autoencoders for unique player identification. In *Proceedings of the 21st Conference on Robots and Vision (CRV)*, May 2024.

[5] Michael Beetz, Nicolai von Hoyningen-Huene, Bernhard Kirchlechner, Suat Gedikli, Francisco Siles, Murat Durus, and Martin Lames. ASPOGAMO: Automated sports games analysis models. *International Journal of Computer Science in Sport*, 8(1):1–21, 2009.

[6] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision – ECCV 2016 Workshops*, volume 9914 of *Lecture Notes in Computer Science*, pages 850–865. Springer, 2016.

[7] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6182–6191, October 2019.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, 1 edition, 2006.

[9] Bodhisattwa Chakraborty and Sukadev Meher. 2D trajectory-based position estimation and tracking of the ball in a basketball video. In *Second International Conference on Trends in Optics and Photonics*, Kolkata, India, December 2011. Department of Applied Optics and Photonics, University of Calcutta. December 7–9, 2011.

[10] Bodhisattwa Chakraborty and Sukadev Meher. Real-time position estimation and tracking of a basketball. *2012 IEEE International Conference on Signal Processing, Computing and Control*, pages 1–6, 2012.

[11] Hua-Tsung Chen, Hsuan-Shen Chen, and Suh-Yin Lee. Physics-based ball tracking in volleyball videos with its applications to set type recognition and action detection. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 1, pages I–1097–I–1100, 2007.

[12] Hua-Tsung Chen, Wen-Jiin Tsai, Suh-Yin Lee, and Jen-Yu Yu. Ball tracking and 3d trajectory approximation with applications to tactics analysis from single-camera volleyball sequences. *Multimedia Tools Appl.*, 60(3):641–667, October 2012.

[13] Zhe Chen, Jing Zhang, Yufei Xu, and Dacheng Tao. Transformer-based context condensation for boosting feature pyramids in object detection. *International Journal of Computer Vision*, 131(10):2738–2756, June 2023.

[14] Xina Cheng, Masaaki Honda, Norikazu Ikoma, and Takeshi Ikenaga. Anti-occlusion observation model and automatic recovery for multi-view ball tracking in sports analysis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1501–1505, 2016.

[15] Xina Cheng, Xizhou Zhuang, Yuan Wang, Masaaki Honda, and Takeshi Ikenaga. Particle filter with ball size adaptive tracking window and ball feature likelihood model for ball's 3d position tracking in volleyball analysis. In *Advances in Multimedia Information Processing – PCM 2015*, volume 9314 of *Lecture Notes in Computer Science*, pages 203–211. Springer, 2015.

[16] Kyuhyoung Choi and Yongdeuk Seo. Probabilistic tracking of the soccer ball. In *Statistical Methods in Video Processing*, volume 3247 of *Lecture Notes in Computer Science*, pages 50–60. Springer, 2004.

[17] Kyuhyoung Choi and Yongdeuk Seo. Tracking soccer ball in tv broadcast video. In *Image Analysis and Processing – ICIAP 2005*, volume 3617 of *Lecture Notes in Computer Science*, pages 661–668. Springer, 2005.

[18] Seokeon Choi, Junhyun Lee, Yunsung Lee, and Alexander Hauptmann. Robust long-term object tracking via improved discriminative model prediction. In *Computer Vision – ECCV 2020 Workshops*, volume 12539 of *Lecture Notes in Computer Science*, pages 602–617. Springer, 2020.

[19] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7183–7192, June 2020.

[20] Tiziana D'Orazio, Marco Leo, Nicola Mosca, Paolo Spagnolo, and Pier Luigi Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 559–564. IEEE, September 2009.

[21] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

[22] Xin Duan. *Automatic determination of puck possession and location in broadcast hockey video*. PhD thesis, University of British Columbia, 2011.

[23] Mehrnaz Fani, Pascale Brunelle Walters, David A. Clausi, John Zelek, and Alexander Wong. Localization of ice-rink for broadcast hockey videos. arXiv preprint arXiv:2104.10847, 2021.

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

[25] René Manassé Galekwa, Jean Marie Tshimula, Etienne Gael Tajeuna, and Kyamakya Kyandoghere. A systematic review of machine learning in sports betting: Techniques, challenges, and future directions. arXiv preprint arXiv:2410.21484, 2024.

[26] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1134–1142, December 2015.

[27] Xiao Han, Qi Wang, and Yongbin Wang. Ball tracking based on multiscale feature enhancement and cooperative trajectory matching. *Applied Sciences*, 14(4):1376, 2024.

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[31] Mehdi Houshmand Sarkhoosh, Sushant Gautam, Cise Midoglu, Saeed Shafiee Sabet, Tomas Kupka, and Pål Halvorsen. HockeyAI: A multi-class ice hockey dataset for object detection. In *Proceedings of the 16th ACM Multimedia Systems Conference*, MMSys '25, pages 228–234, New York, NY, USA, 2025. Association for Computing Machinery.

[32] Mehdi Houshmand Sarkhoosh, Sushant Gautam, Cise Midoglu, Saeed Shafiee Sabet, Tomas Kupka, and Pål Halvorsen. HockeyRink: A dataset for precise ice hockey rink keypoint mapping and analytics. In *Proceedings of the 16th ACM Multimedia Systems Conference*, MMSys '25, pages 249—-255, New York, NY, USA, 2025. Association for Computing Machinery.

[33] Yuto Kase, Kai Ishibe, Ryoma Yasuda, Yudai Washida, and Sakiko Hashimoto. Locating tennis ball impact on the racket in real time using an event camera. In *Sports Analytics*, volume 15925 of *Lecture Notes in Computer Science*, pages 99–115, Cham, 2026. Springer Nature Switzerland. ISACE 2025.

[34] Rahima Khanam and Muhammad Hussain. What is YOLOv5: A deep look into the internal features of the popular object detector. arXiv preprint arXiv:2407.20892, 2024.

[35] Muyu Li, Henan Hu, and Hong Yan. Ice hockey puck tracking through broadcast video. *Neurocomputing*, 551:126484, 2023.

[36] Wenxi Li, Moran Liu, Jiaxin Liu, Baosheng Zhang, Tao Yu, Yuchen Guo, and Qionghai Dai. A review of artificial intelligence for sports: Technologies and applications. *Intelligent Sports and Health*, 1(3):113–126, 2025.

[37] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[38] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8721–8730, 2022.

[39] Christoph Mayer, Martin Danelljan, Ming-Hsuan Yang, Vittorio Ferrari, Luc Van Gool, and Alina Kuznetsova. Beyond sot: Tracking multiple generic objects at once. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6826–6836, January 2024.

[40] Brian Nemeth. Hockey puck math: For radius, diameter, area, circumference, and volume. https://bleacherreport.com/articles/412092-hockey-puck-math-for-radius-diameter-area-circumference-and-volume, 2018. Accessed: 2025-03-10.

[41] Ken Nsiempba, Amir Nazemi, David Clausi, and John Zelek. Leveraging player tracking for event detection in ice hockey. *Journal of Computational Vision and Imaging Systems*, 10(1):69–74, 2024.

[42] Hemanth Pidaparthy and James H Elder. Keep your eye on the puck: Automatic hockey videography. In *Winter Conference on Applications in Computer Vision (WACV)*, 2019.

[43] Harish Prakash, Jia Cheng Shang, Ken M. Nsiempba, Yuhao Chen, David A. Clausi, and John S. Zelek. Multi player tracking in ice hockey with homographic projections. In *Proceedings of the 21st Conference on Robots and Vision (CRV)*, May 2024.

[44] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in

images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

[45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[46] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018. Technical report.

[47] Jinchang Ren, James Orwell, Graeme A. Jones, and Ming Xu. Tracking the soccer ball using multiple fixed cameras. *Computer Vision and Image Understanding*, 113(5):633–642, 2009. Computer Vision Based Analysis in Sport Environments.

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[49] Liam Salass, Jerrin Bright, Amir Nazemi, Yuhao Chen, John Zelek, and David Clausi. Ice hockey puck localization using contextual cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 6106–6115, June 2025.

[50] Liam Salass, John Zelek, and David Clausi. Geometrically aware ice hockey puck tracking vianbsp;stable planar representations. In *Sports Analytics: Third International Conference, ISACE 2026, Vancouver, BC, Canada, June 1–2, 2026, Proceedings*, page 215–232, Berlin, Heidelberg, 2026. Springer-Verlag.

[51] Mehdi Houshmand Sarkhoosh, Sayed Mohammad Majidi Dorcheh, Cise Midoglu, Saeed Shafiee Sabet, Tomas Kupka, Dag Johansen, Michael A. Riegler, and Pål Halvorsen. AI-Based Cropping of Ice Hockey Videos for Different Social Media Representations. *IEEE Access*, 12:118227–118249, 2024.

[52] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[53] Marjan Shahi, David Clausi, and Alexander Wong. Goalienet: A multi-stage network for joint goalie, equipment, and net pose estimation in ice hockey. *arXiv preprint arXiv:2306.15853*, 2023.

[54] Jia Cheng Shang, Yuhao Chen, Mohammad Javad Shafiee, and David A. Clausi. Rink-agnostic hockey rink registration. In *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports*, MMSports '23, pages 73—81, New York, NY, USA, 2023. Association for Computing Machinery.

[55] Zican Shi, Jing Hu, Jie Ren, Hengkang Ye, Xuyang Yuan, Yan Ouyang, Jia He, Bo Ji, and Junyu Guo. HS-FPN: High frequency and spatial perception FPN for tiny object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6896–6904, 2025.

[56] Athletica Sport Systems. Size of hockey rinks: Why the us rink is smaller than the eu rink. https://www.athletica.com/size-of-hockey-rinks/. Accessed: 2025-03-11.

[57] Shuhei Tarashima, Muhammad Abdul Haq, Yushan Wang, and Norio Tagawa. Widely applicable strong baseline for sports ball detection and tracking. In *34th British Machine Vision Conference (BMVC)*. BMVA, 2023.

[58] Binren Tian, Debing Zhang, and Chun Zhang. High-speed tiny tennis ball detection based on deep convolutional neural networks. In *2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 30–33, 2020.

[59] Sijia Tian. Group event recognition in ice hockey. Master of science thesis, University of British Columbia, 2018.

[60] Moumita Roy Tora, Jianhui Chen, and James J Little. Classification of puck possession events in ice hockey. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 147–154. IEEE, 2017.

[61] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 203(1153):405–426, 1979.

[62] Rejin Varghese and Sambath M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6, 2024.

[63] Kanav Vats, Mehrnaz Fani, David A Clausi, and John Zelek. Multi-task learning for jersey number recognition in ice hockey. In *Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports*, pages 11–15, 2021.

[64] Kanav Vats, Mehrnaz Fani, David A. Clausi, and John Zelek. Puck localization and multi-task event recognition in broadcast hockey videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4562–4570, 2021.

[65] Kanav Vats, Pascale Walters, Mehrnaz Fani, David A Clausi, and John S Zelek. Player tracking and identification in ice hockey. *Expert systems with applications*, 213:119250, 2023.

[66] Jorge Armando Vicente-Martínez, Moisés-Vicente Márquez-Olivera, Abraham García-Aliaga, and Viridiana Hernández-Herrera. Adaptation of YOLOv7 and YOLOv7_tiny for soccer-ball multi-detection with DeepSORT for tracking by semi-supervised system. *Sensors*, 23(21):8693, 2023.

[67] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5294–5306, June 2025.

[68] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations (ICLR)*, 2026.

[69] Saef Wbaid, Azizov Marufkhuja Ahmatkhonovich, Hilola Pulatova, and Md Afzal. Artificial intelligence-powered sports analytics: Enhancing performance through data science. In *2025 International Conference on Computational Innovations and Engineering Sustainability (ICCIES)*, pages 1–6, 2025.

[70] Wei Wei, Yu Cheng, Jiafeng He, and Xiyue Zhu. A review of small object detection based on deep learning. *Neural Computing and Applications*, 36(12):6283–6303, April 2024.

[71] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9697–9706, June 2023.

[72] Yangzheng Wu, Mohsen Zand, Ali Etemad, and Michael Greenspan. Vote from the center: 6 DoF pose estimation in RGB-D images by radial keypoint voting. In *Computer Vision – ECCV 2022*, volume 13670 of *Lecture Notes in Computer Science*, pages 335–352, Cham, 2022. Springer.

[73] Greg Wyshynski. NHL brings advanced puck tracking stats to public. ESPN, October 2023. Published October 23, 2023; accessed August 14, 2026.

[74] Chen Xin, Thomas Motz, Andreas Hartel, and Enkelejda Kasneci. Ocdet: Object center detection via bounding box-aware heatmap prediction on edge devices with npus. *arXiv preprint arXiv:2411.15653*, 2024.

[75] Hao Xu, Arbind Agrahari Baniya, Sam Wells, Mohamed Reda Bouadjenek, Richard Dazeley, and Sunil Aryal. TOTNet: Occlusion-aware temporal tracking for robust ball detection in sports videos. *Computer Vision and Image Understanding*, 264:104657, 2026.

[76] Xionghao Yang. Where is the puck? tiny and fast-moving object detection in videos. Master of engineering thesis, McGill University, Montreal, Quebec, Canada, 2021.

APPENDICES

Appendix A

Supplementary Experiments

A.1 Puck Detection Ablation Experiments

Our ablation studies highlight the performance of an independent context encoder, the selection of the optimal Gaussian standard deviation (σ) parameter, and the effects of including context-driven dropout in our training.

Independent Context Encoder

Training the FCN-ResNet152 [72] independently on context images C_{RGB} (no raw RGB images including puck features) resulted in poor detection accuracies, highlighted in Table A.1. However, this ablation experiment produced heat maps that peak in regions corresponding to the direction of the player's gaze, shown in Figure A.1. This implies that the context encoder learns to look where players focus, an important cue in hockey where a player's line of sight often correlates with puck location.

Sigma Parameter Selection

To select the optimal Gaussian heatmap standard deviation, we conducted an ablation study by training models with different σ values: 5, 15, 25, and 50. Figure A.2 highlights the varying model's performances over average precision threshold $\tau \in 5, 10, 25, 50$ on the test set, where the model trained with $\sigma = 5$ achieved the best performance across all evaluation thresholds τ. The model performs best at lower thresholds (e.g., $\tau = 5$ and

Table A.1: **Performance comparison of PLUCC and the context-only FCN-ResNet152.** The FCN-ResNet152 is trained independently using only context images, providing a comparison with PLUCC's combined appearance and contextual representations.

Method	mAP^{τ}	AP^{5}	AP^{10}	AP^{25}	AP^{50}	$\mathcal{D}_{pixel}$
FCN-ResNet152 (Only Context Image)	6.0	0.2	1.0	5.7	18.0	178.5
PLUCC ($\sigma = 5$)	**83.5**	**82.2**	**83.6**	**84.2**	**85.0**	**47.0**



Figure A.1: **Context-only puck localization from player information.** Heatmap generated by the independently trained context encoder when processing a single segmented player. The network highlights regions along the player's line of sight, implicitly predicting the location of the occluded puck behind the boards.

$\tau = 10$), indicating that a smaller σ results in a more focused heatmap. Furthermore, at higher thresholds (e.g., $\tau = 25$ and $\tau = 50$), the performance of all models converges, suggesting that the benefit of a more granular heatmap is most significant when exact localization accuracy is required.

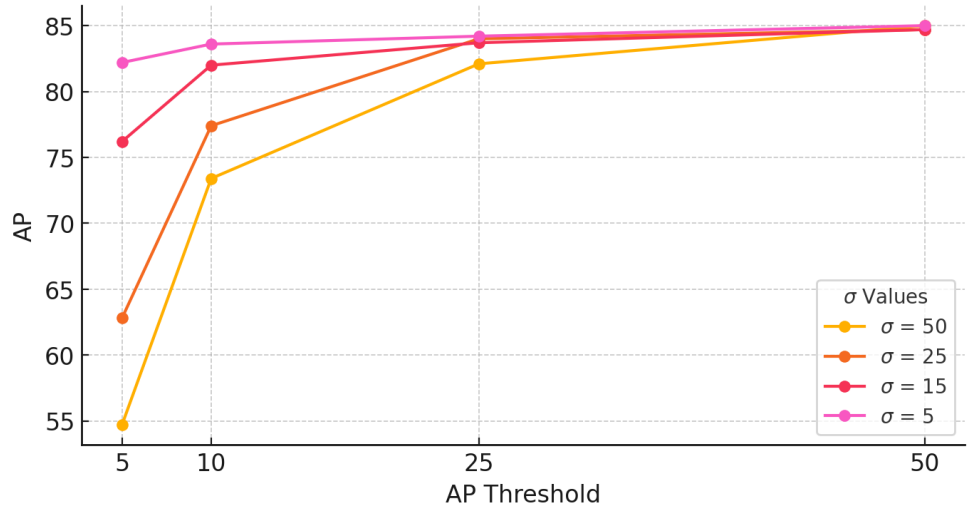


Figure A.2: **Effect of target Gaussian heatmap standard deviation on model performance.** Comparison of models trained using different target Gaussian heatmap standard deviations (σ).

Context-driven Dropout

Table A.2 highlights the performance boost of including a dropout of the I_{RGB} input image into the feature pyramid encoder with a probability of $p_{drop} = 1\%$, the network is compelled to leverage information from the contextual image (C_{RGB}) — such as player positions and orientations — which are indicative of the puck's location. The increase in accuracy highlights the necessity of forcing the model to rely more on contextual cues when the direct visual features are unreliable.

Table A.2: **Effect of context-driven dropout on PLUCC performance.** Comparison between PLUCC trained with a context-driven dropout probability of 1% and PLUCC trained without context-driven dropout.

Method	mAP^5	$\mathcal{D}_{pixel}$	AP^r	$RSLE_{avg}(m)$
PLUCC ($p_{drop} = 0\%$)	78.7	61.15	39.9	1.36
PLUCC ($p_{drop} = 1\%$)	**82.2**	**47.0**	**43.6**	**1.05**

A.2 Runtime Analysis

Table A.3 presents the runtime of the proposed pipeline. The majority of the computational cost is associated with monocular 3D reconstruction using Pi3 and the one-time plane estimation stage. In contrast, homography projection and recurrent trajectory prediction contribute negligible computational overhead, with the LSTM requiring only 0.28 ms for a 16-frame sequence. Since plane estimation is performed once per sequence and subsequently reused, the proposed motion estimation framework introduces minimal additional runtime beyond the initial geometric reconstruction.

Table A.3: **Runtime breakdown of the proposed pipeline.** Runtime is reported in milliseconds (ms) for a 16-frame sequence using an NVIDIA RTX 6000 Ada GPU.

Component	ms
Context image generation	144.84
PLUCC	21.83
Pi3 preprocess	6.37
Pi3 DNN forward (amortized)	213.84
Plane RANSAC (once)	396.06
Plane RANSAC (per frame, amortized)	24.75
Homography projection (each direction)	< 0.02
LSTM (causal, T=10)	0.28
Sum (illustrative)	**411.91**

Appendix B

Additional Implementation Details

B.1 Visual Object Tracking Training Parameters

To ensure a consistent evaluation of visual object tracking methods, each tracker was trained or fine-tuned on the VIP-PuckDataset using the configurations summarized in Table B.1. Where applicable, pretrained model weights were retained and selected network components were fine-tuned, while SiamFC was trained from scratch on the puck sequences. The learning rates, optimization strategies, and frozen backbone components were selected according to the requirements of each tracking architecture.

Table B.1: **Fine-tuning hyperparameters for the evaluated tracking models.** Learning rates are reported either globally or for individual network components. Abbreviations: `clf` = classifier, `bb reg` = bounding box regressor, and `feat` = feature extractor.

Model	**Optimizer**	**Learning Rates**	**Backbone Strategy**
KeepTrack	Adam	All parameters: 1×10^{-5}; milestones at epochs 10 and 20.	Feature extractor frozen.
SuperDiMP	Adam	Global: 2×10^{-4}; classifier init: 5×10^{-5}; classifier opt.: 5×10^{-4}; bounding box regressor: 10^{-3}; layer3: 2×10^{-5}.	`conv1`–`layer2` frozen.
DiMP50	Adam	Global: 2×10^{-4}; classifier init: 5×10^{-5}; classifier opt.: 5×10^{-4}; classifier features: 5×10^{-5}; feature extractor: 2×10^{-5}.	Early backbone layers effectively unchanged.
PrDiMP50	Adam	Global: 2×10^{-4}; classifier: 10^{-3}; bounding box regressor: 10^{-3}; feature extractor: 2×10^{-5}.	Backbone lightly updated.
ToMP50	AdamW	Global: 2×10^{-4}; head: 10^{-4}; layer3: 2×10^{-5}.	`conv1`, `bn1`, `layer1`, and `layer2` frozen; BatchNorm layers frozen.
TaMOs (Swin-B)	AdamW	Global: 2×10^{-4}; head: 10^{-4}; backbone stage 2: 2×10^{-5}.	Backbone stages 0 and 1 frozen; BatchNorm frozen; gradient clipping = 0.1.
ATOM	Adam	Bounding box regressor only: 10^{-4}; milestones at epochs 10 and 20.	Classifier and backbone frozen.
ARTrack (ViT-L)	AdamW	Global: 4×10^{-6}; backbone: $0.1\times$ global learning rate; step scheduler with no decay during 30 epochs.	ViT-L backbone lightly updated from pretrained checkpoint; gradient clipping = 0.1.
SiamFC	Adam / SGD	Fixed learning rate: 10^{-3}; no decay during 30 epochs.	Convolutional embedding trained from scratch using puck sequences.

B.2 Motion Estimation LSTM Model Architectures

motion models introduced in Section 3.2.2 share a common input representation but differ in how temporal information is processed. Each model operates on normalized two-dimensional puck coordinates in the temporally stable plane representation and incorporates finite-difference velocity information prior to recurrent processing. The detailed architectures of the forward, dual-path, and bidirectional gap-filling LSTM models are provided in Tables B.2–B.4.

Table B.2: **Forward LSTM architecture for planar puck tracking.** Layer-wise architecture of the forward LSTM model. Abbreviations: B = batch size, T = sequence length, MLP = multilayer perceptron, and ReLU = rectified linear unit. The model uses a hidden size of 128, two LSTM layers, and a dropout probability of 0.1.

Stage	Description	Dims
Input	Normalized rink-plane puck coordinates $\mathbf{p}_t = (u_t, v_t)^\top$ at each time step	$(B, T, 2)$
Velocity features	Finite differences $\Delta\mathbf{p}_t = \mathbf{p}_t - \mathbf{p}_{t-1}$ ($\Delta\mathbf{p}_0 = \mathbf{0}$); concatenate $[\mathbf{p}_t; \Delta\mathbf{p}_t]$	$(B, T, 4)$
Velocity MLP	Linear(4→64), ReLU, Linear(64→2); residual addition: $\tilde{\mathbf{p}}_t = \mathbf{p}_t + \mathrm{MLP}([\mathbf{p}_t; \Delta\mathbf{p}_t])$	$(B, T, 2)$
Recurrent core	Unidirectional LSTM, 2 layers, hidden size 128, inter-layer dropout 0.1	$(B, T, 128)$
Output head	Linear projection to plane coordinates	$(B, T, 2)$
Role	Forward trajectory prediction from past plane positions.	

Table B.3: **Dual-path LSTM architecture for planar puck tracking.** Layer-wise architecture of the dual-path LSTM model. Abbreviations: B = batch size, T = sequence length, MLP = multilayer perceptron, ReLU = rectified linear unit, fwd = forward path, and bwd = backward path. Each path uses a hidden size of 128, two LSTM layers, and a dropout probability of 0.1, followed by weighted fusion.

Stage	Description	Dims
Input	Normalized rink-plane puck coordinates $\mathbf{p}_t = (u_t, v_t)^\top$ at each time step	$(B, T, 2)$
Velocity features	Finite differences $\Delta\mathbf{p}_t = \mathbf{p}_t - \mathbf{p}_{t-1}$ ($\Delta\mathbf{p}_0 = \mathbf{0}$); concatenate $[\mathbf{p}_t; \Delta\mathbf{p}_t]$	$(B, T, 4)$
Velocity MLP	Linear(4→64), ReLU, Linear(64→2); residual addition: $\tilde{\mathbf{p}}_t = \mathbf{p}_t + \mathrm{MLP}([\mathbf{p}_t; \Delta\mathbf{p}_t])$	$(B, T, 2)$
Forward path	Independent unidirectional LSTM on $\tilde{\mathbf{p}}$ (2 layers, hidden size 128, dropout 0.1)	$(B, T, 128)$
Backward path	Independent unidirectional LSTM on the time-reversed $\tilde{\mathbf{p}}$ sequence; outputs are reversed back to the original temporal order (2 layers, hidden size 128, dropout 0.1)	$(B, T, 128)$
Fusion	Learnable weights α_f, α_b; $w_f = \sigma(\alpha_f)$ and $w_b = \sigma(\alpha_b)$, normalized to sum to 1; $\mathbf{h}_t = w_f \mathbf{h}_{f,t} + w_b \mathbf{h}_{b,t}$	$(B, T, 128)$
Output head	Linear projection to plane coordinates	$(B, T, 2)$
Role	Combines independent forward and backward temporal context without sharing weights between directions.	

Table B.4: **Bidirectional LSTM architecture for planar puck tracking.** Layer-wise architecture of the bidirectional gap-filling LSTM model. Abbreviations: B = batch size, T = sequence length, MLP = multilayer perceptron, ReLU = rectified linear unit, and BiLSTM = bidirectional LSTM. The model uses a hidden size of 128 per direction, two LSTM layers, and a dropout probability of 0.1.

Stage	Description	Dims
Input	Normalized rink-plane puck coordinates $\mathbf{p}_t = (u_t, v_t)^\top$; missing or occluded positions are zeroed or interpolated	$(B, T, 2)$
Velocity features	Finite differences $\Delta\mathbf{p}_t = \mathbf{p}_t - \mathbf{p}_{t-1}$ ($\Delta\mathbf{p}_0 = \mathbf{0}$); concatenate $[\mathbf{p}_t; \Delta\mathbf{p}_t]$	$(B, T, 4)$
Velocity MLP	Linear(4→64), ReLU, Linear(64→2); residual addition: $\tilde{\mathbf{p}}_t = \mathbf{p}_t + \mathrm{MLP}([\mathbf{p}_t; \Delta\mathbf{p}_t])$	$(B, T, 2)$
Recurrent core	Bidirectional LSTM, 2 layers, hidden size 128 per direction, inter-layer dropout 0.1; forward and backward hidden states are concatenated	$(B, T, 256)$
Output head	Linear projection to plane coordinates	$(B, T, 2)$
Gap masking	Optional valid-position mask preserves observed positions and replaces predictions only at missing or occluded frames	$(B, T, 2)$
Role	Fills trajectory gaps using both past and future temporal context.	