# Deep Reinforcement Learning Project Proposal

In multi-view 3D sports reconstruction, using all available camera views simultaneously is computationally expensive and often redundant. Our project aims to develop an RL agent that dynamically selects the optimal subset of camera views to maximize reconstruction quality while minimizing computational cost.

## Plan to Do and How

### Sequential Nature of the Problem

The problem is sequential because:

- The agent selects camera views one at a time
- Each selection affects the information available for the next selection
- The quality of the final reconstruction depends on the sequence of views selected

### State Space

The state space consists of:

- Current reconstruction quality metrics (e.g., reprojection error, point cloud density)
- Features extracted from each available camera view (e.g., view angle, coverage of the scene)
- Athletes' positions and movements extracted from each view
- Occlusion information from current views
- Previously selected camera views

### Action Space

The action space is discrete, representing the selection of one of N available camera views at each step. For example, with 8 cameras, the action space would be {0,1,2,3,4,5,6,7}.

### Transition Function

The transition function maps the current state and chosen action (camera view) to the next state:

- Adding the selected view to the reconstruction
- Updating the reconstruction quality metrics
- Updating features based on the new combined set of views
- Updating occlusion information

### Reward Function

The reward function will be designed to balance reconstruction quality and computational efficiency:

- Positive reward for improvement in reconstruction quality (measured by reduction in reprojection error, increase in point cloud density, reduction in occlusions)

- Negative reward proportional to computational cost of adding the view
- Terminal reward based on final reconstruction quality compared to using all views

### RL Algorithm

I plan to use Deep Q-Network (DQN) as our primary RL algorithm, I will look into different algorithms once done more research.

## Baseline Comparison

We will compare our approach against:

- Using all available camera views (maximum quality but highest computational cost)
- Random selection of camera views

## Implementation Plan

- **Weeks 1-2**: Set up the reconstruction pipeline using COLMAP and prepare the dataset
- **Weeks 3-4**: Implement the RL environment with the defined state space, action space, and reward function
- **Week 5**: Train the RL agent using PPO and DQN algorithms
- **Week 6**: Evaluate against baseline methods and optimize hyperparameters

## Stretch Goals

If time permits, I aim to:

- Implement a more sophisticated reward function that considers temporal consistency in sports videos
- Extend the approach to handle dynamic camera selection in real-time streaming scenarios