REINFORCEMENT LEARNING

Lecture 5 : DRL Trends

Ibrahim Sammour

January | 2024

Sample Efficiency

- **Problem**
  - High sample complexity, requiring vast amounts of data for effective learning.
  - Develop DRL algorithms that can achieve optimal performance with fewer training samples.
- Reduced data requirements enable faster training and deployment of RL models.
- Crucial for real-world applications where collecting extensive data can be time-consuming or impractical.

- **Techniques**
  - Experience Replay
  - Prioritized Experience Replay

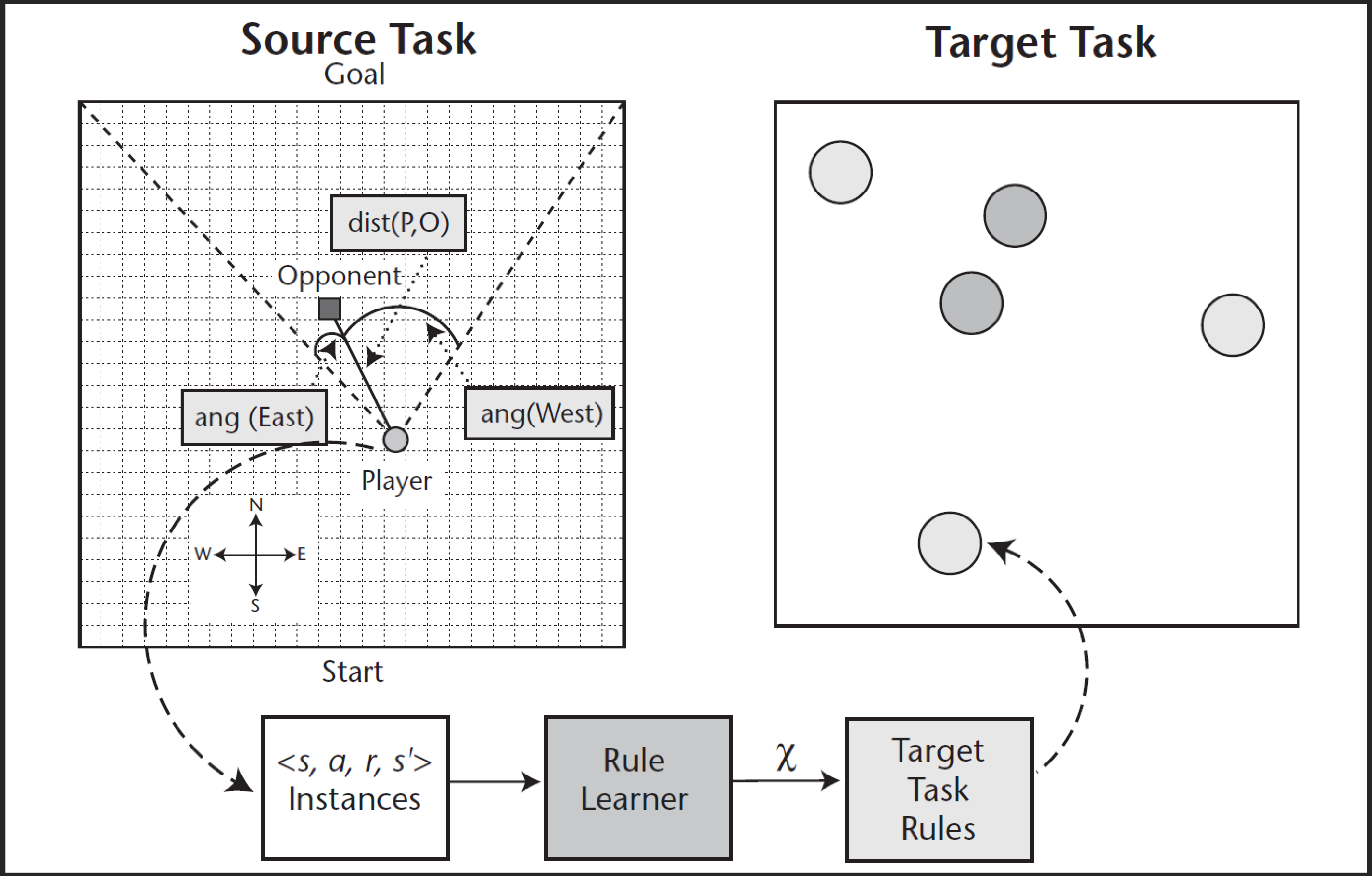
Sample Efficiency

- **Experience Replay**
  - **Past experiences are stored and randomly sampled during training.**
  - **Breaks temporal correlations in the data.**
  - **Enhances learning stability and convergence.**

- **Prioritized Experience Replay**
  - **Prioritizes experiences based on the temporal difference between predicted and actual rewards.**
  - **Allows the model to prioritize and learn from more informative samples.(large TD difference i.e.)**
  - $P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$
  - $P(i)$ **is the probability of sampling the i-th experience.**
  - p_i **is the priority of the i-th experience, proportional to the absolute TD error.**
  - α **is a hyperparameter that controls the degree of prioritization.**

Transfer Learning

- **Process of applying knowledge acquired from one domain (source task) to enhance learning in a different but related domain (target task).**



Transfer Learning

- Enhance generalization and efficiency by using pre-existing knowledge.
- Minimize the need for extensive training on each specific task.
- Reduces the computational cost and training time for new tasks.

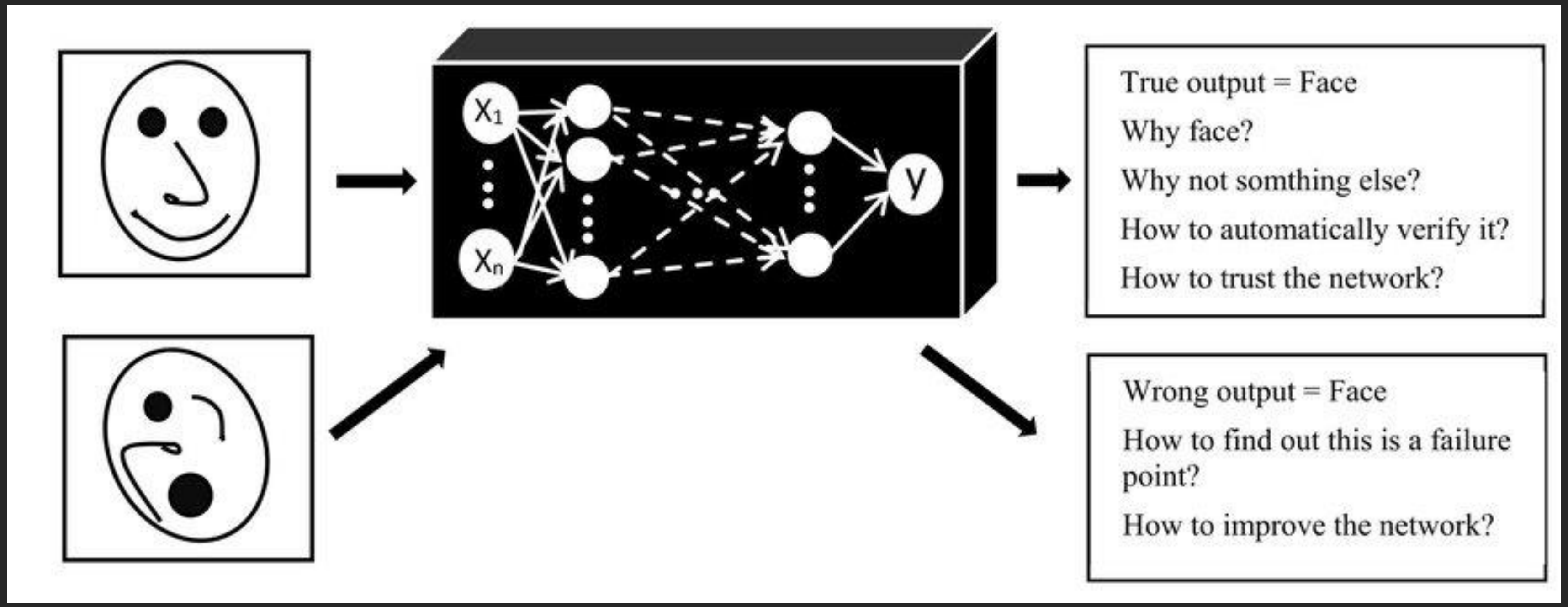
- Homogeneous transfer involves transferring knowledge between tasks with the same state space. (Train on a maze and transfer to a different maze- same environment)

- Heterogeneous transfer involves transferring knowledge between tasks with different state spaces. (Train on a maze and transfer to an environment where its supposed to pick objects)

Explainability

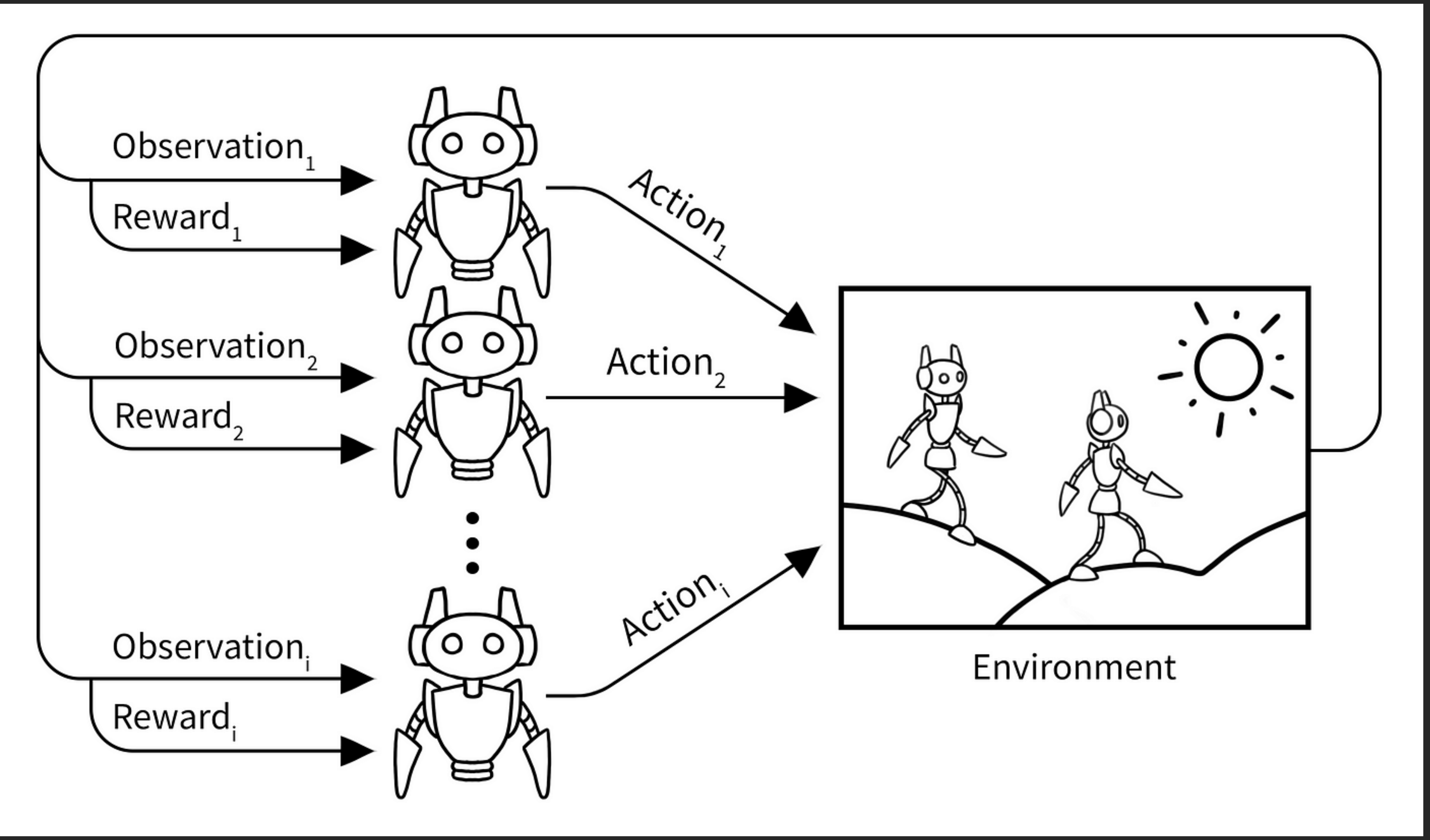
- **DRL models involve complex and deep neural networks, making it challenging to understand the decision-making process.**

- **Explainability:**
  - **Examining the importance of different states in decision-making (play around with the inputs).**
  - **Visualize the states and actions taken during training to understand it better.**
  - **Identify critical learning points and potential areas for improvement.**



Multi-Agent DRL

- **Multiple agents interacting in a shared environment.**
- **Aims to improve cooperation, competition, and communication among agents for more sophisticated decision-making.**



Multi-Agent DRL

- Cooperative Agents
  - Agents work together to achieve a common goal.
  - Challenge: Balancing individual goals with the common objective.

- Individual Agents
  - Agents act independently with no direct collaboration.
  - Challenge: Coordinating actions for mutual benefit is sacrificed for individual optimization. (Competetive)

Multi-Agent DRL

- Centralized Training - Deployment
  - A central entity makes decisions for all agents. (single DRL model)
  - Example: Controlling a swarm of drones.
  - Global coordination - potential scalability issues.

- Decentralized Training - Deployment
  - Each agent independently makes decisions. (Multiple DRL models)
  - Example: Autonomous vehicles making decisions based on local observations.
  - Scalable - not easy to achieve global coordination.

Multi-Agent DRL

- **Challenges in Multi-Agent RL**
  - **Credit Assignment**
    - **Which agent contributed more to the result**
  - **Exploration**
    - **Balancing exploration and exploitation in a shared environment**
  - **Non-Stationary Environment**
    - **The environment is dynamic and influenced by the actions of other agents**

Multi-Task Reinforcement Learning

- Splitting a task into multiple smaller ones.
- The aim is to:
  - Isolate problems.
  - Easily adapt to new tasks (reduced training time).
  - Improves explainability.

Before

DRL Model
The agent opens a door and cleans a room

After

DRL Model 1
Open a door

DRL Model 2
Clean a room

Adversarial Training

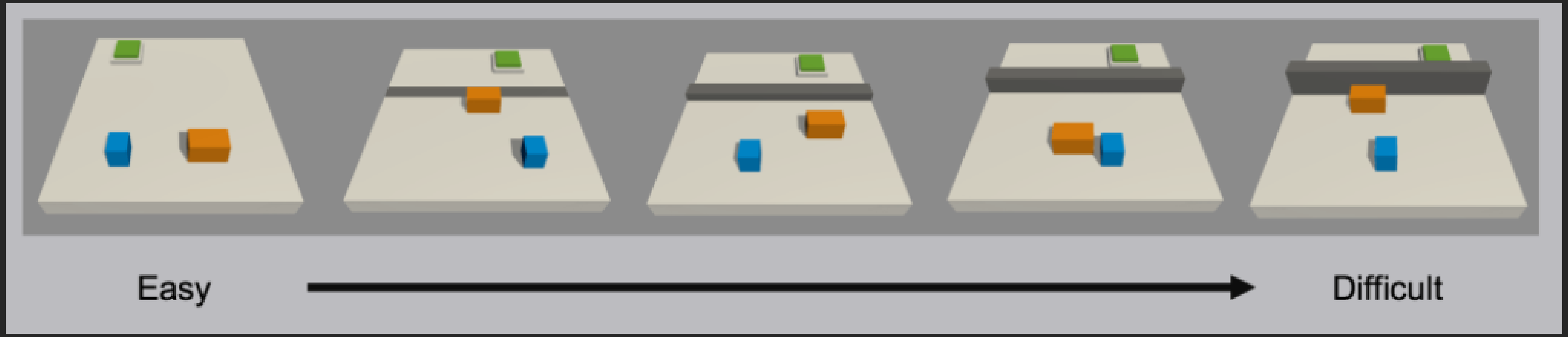
- Exposing the model to adversarial scenarios during training.
- Strengthen the DRL model's ability to perform well on unseen or altered inputs (states).
- The aim is to challenge the model to adapt and generalize more effectively.

- Example:
  - Training a drone to navigate in an environment
    - Introduce unpredictable wind patterns.
    - Simulate temporary sensor failures.

Curriculum Learning

- Start with simple tasks and progressively increase the complexity over time.
- The aim is to
  - Facilitates a smoother learning trajectory, preventing the agent from becoming overwhelmed by challenging tasks initially.
  - Reduce training time.
  - Improve generalization.

- Complexity adjustment is based on the learning progress of the agent.



Ensemble Learning

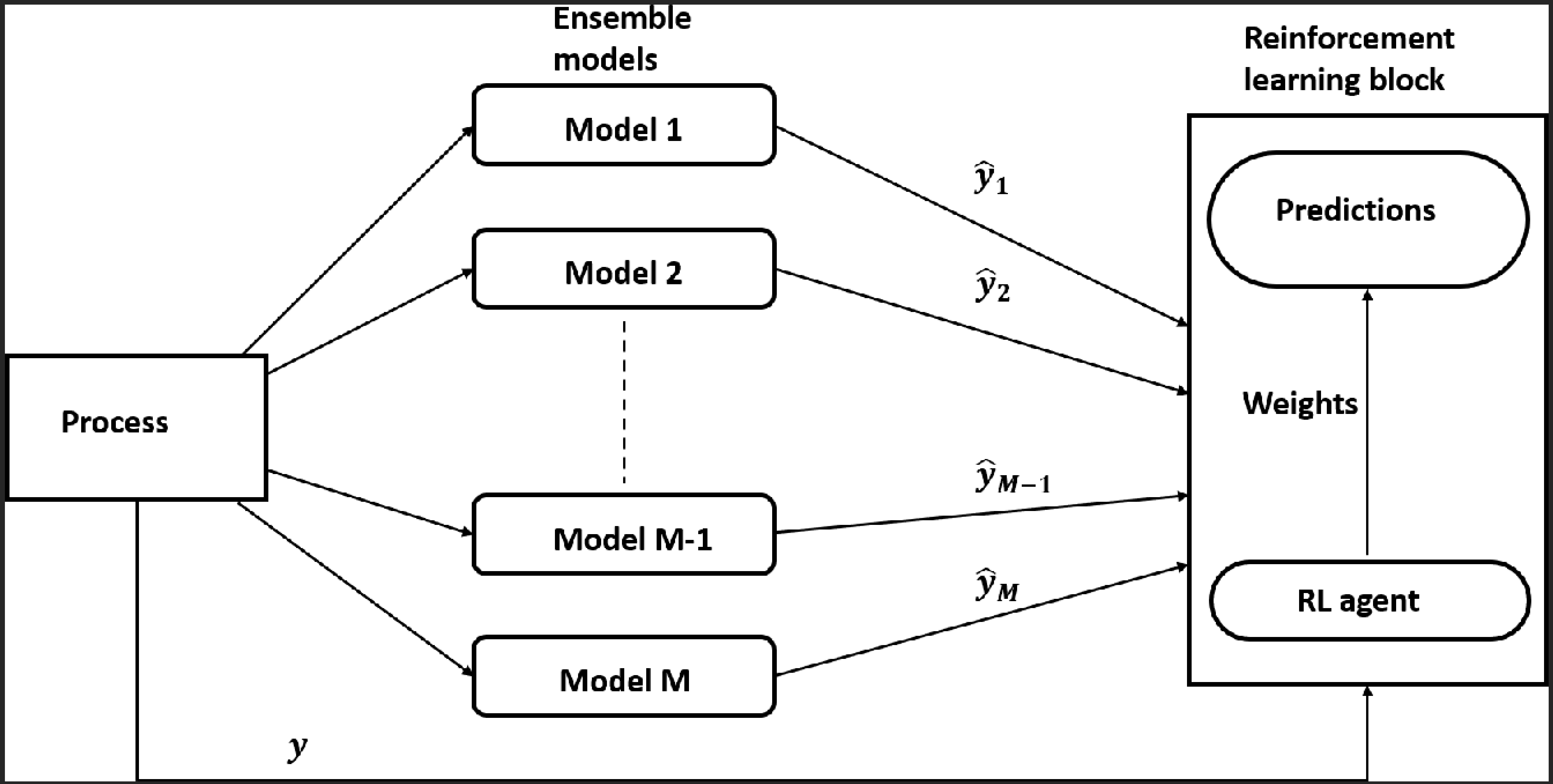
- Combine predictions from multiple DRL models to enhance performance and robustness.
- Use predictions from multiple DRL models.

- Majority Voting
    - Each model "votes" on the decision, and the most frequent choice is selected.

- Soft Voting
    - DRL Models provide probability estimates (actions), and the final decision is based on weighted averages.

Ensemble Learning

Soft Voting



Meta Learning

- Learning to Learn
- An agent is trained to rapidly adapt to new tasks with minimal data.

- Initial Task Set
  - Train the robot on a set of initial tasks involving holding objects of different shapes and weights.

- Rapid Adaptation
  - Introduce a new task with a new object (shape or weight) not seen during initial training.
  - Measure the robot's ability to adapt quickly based on its meta-learned capabilities.