Examples of Reinforcement Learning Applications

Games, Robotics, and Recommendation Systems

Sarwan Ali

Department of Computer Science
Georgia State University

Real-World RL Applications

Today's Learning Journey

Why Study RL Applications?

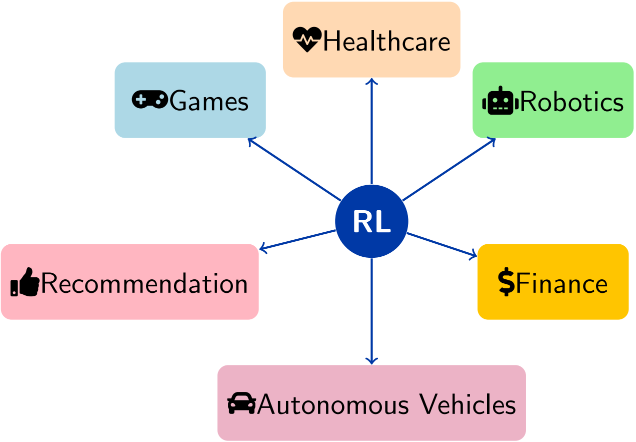
Understanding real-world applications helps us:

- **Connect Theory to Practice** - See how abstract concepts solve real problems
- **Identify Problem Patterns** - Learn to recognize when RL is appropriate
- **Understand Design Choices** - Why certain algorithms work better for specific domains
- **Appreciate Impact** - See how RL transforms industries and society

Key Insight

Each application domain presents unique challenges that drive RL research and innovation.

RL Application Landscape



Focus Areas: Games, Robotics, Recommendation Systems

Games: Why Perfect for RL?

Why games are ideal RL testbeds:

- **Clear Rules** - Well-defined state spaces and actions
- **Objective Rewards** - Win/lose, points, scores
- **Controlled Environment** - Reproducible, safe testing
- **Scalable Complexity** - From simple to incredibly complex
- **Human Benchmarks** - Clear performance comparison

Game Environment

State $\rightarrow$ Action
Action $\rightarrow$ Reward
Learn $\rightarrow$ Improve

Evolution of Game AI

Checkers (1950s) $\rightarrow$ Chess (1997) $\rightarrow$ Go (2016) $\rightarrow$ StarCraft II (2019) $\rightarrow$ Multi-agent games

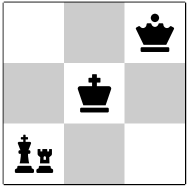
Chess: Deep Blue to AlphaZero

Traditional Approach (Deep Blue):

- Hand-crafted evaluation functions
- Minimax with alpha-beta pruning
- Massive computational power
- Domain-specific optimizations

RL Approach (AlphaZero):

- Self-play learning
- Monte Carlo Tree Search + Neural Networks
- No human chess knowledge
- Generalizable architecture



RL Learning:
Play → Learn → Improve

Key Achievement

AlphaZero learned to play chess at superhuman level in 4 hours, discovering novel strategies never seen before.

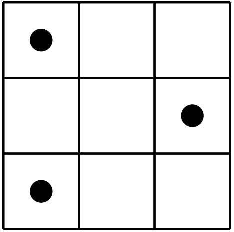
Go: The Ultimate Board Game Challenge

Why Go was considered impossible:

- State space: 10^{170} possible positions
- Branching factor: 250 moves per turn
- No good evaluation function
- Requires intuition and pattern recognition

AlphaGo's Revolutionary Approach:

1. **Supervised Learning** - Learn from human expert games
2. **Reinforcement Learning** - Self-play improvement
3. **Monte Carlo Tree Search** - Guided exploration
4. **Deep Neural Networks** - Position evaluation



$19 \times 19 = 361$ positions

Historic Milestone

AlphaGo defeated Lee Sedol (9-dan professional) 4-1 in March 2016, marking a breakthrough in AI capabilities.

Video Games: Complex Multi-Agent Environments

Atari Games (DQN):

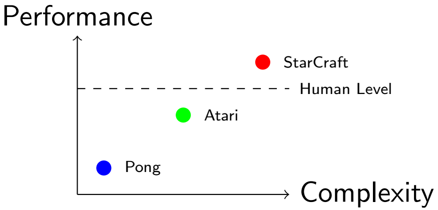
- Raw pixel input
- Simple action spaces
- Single-agent environments
- Human-level performance on many games

StarCraft II (AlphaStar):

- Real-time strategy
- Partial observability
- Long-term planning
- Multi-agent coordination

Dota 2 (OpenAI Five):

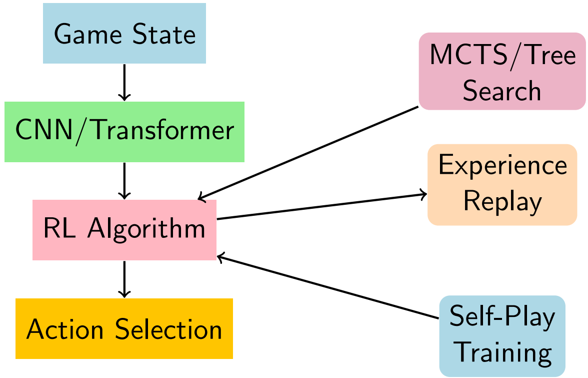
- 5v5 team coordination
- Complex item builds
- 45+ minute games
- Defeated professional teams



Key Insights

Video games provide increasingly complex environments that push the boundaries of multi-agent RL and long-term planning.

Game RL: Technical Components



Common Challenges: Credit assignment, exploration vs exploitation, multi-agent coordination

Robotics: From Simulation to Reality

Why robotics needs RL:

- **Uncertain Environments** - Real world is unpredictable
- **Continuous Control** - Fine motor skills and smooth movements
- **Adaptation** - Must handle new situations and objects
- **Multi-modal Sensing** - Vision, touch, proprioception
- **Safety Constraints** - Cannot damage robot or environment

Unique Challenges:

- Sample efficiency (real-world data is expensive)
- Sim-to-real transfer
- Hardware limitations

Robot Learning

Sensors $\rightarrow$ State
Policy $\rightarrow$ Actions
Environment $\rightarrow$ Feedback

Continuous adaptation

Key Insight

Robotics RL must balance exploration with safety, making it one of the most challenging application domains.

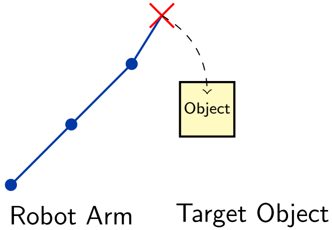
Manipulation: Learning to Grasp and Move

Grasping Challenges:

- Object shape and material variations
- Force control and tactile feedback
- 6-DOF pose estimation
- Collision avoidance

RL Solutions:

- **Sim-to-Real** - Train in simulation, transfer to reality
- **Curriculum Learning** - Start simple, increase complexity
- **Multi-task Learning** - Share knowledge across objects
- **Demonstration + RL** - Bootstrap with human demos



Success Stories

OpenAI's robotic hand solving Rubik's cube, Google's everyday object manipulation

Locomotion: Learning to Walk, Run, and Navigate

Locomotion Control:

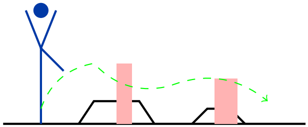
- Dynamic balance and stability
- Terrain adaptation
- Energy efficiency
- Robust to disturbances

Navigation Tasks:

- Path planning in unknown environments
- Obstacle avoidance
- SLAM (Simultaneous Localization and Mapping)
- Social navigation (moving around humans)

RL Approaches:

- Model-free policy learning (PPO, SAC)
- Hierarchical RL (high-level goals, low-level control)
- Multi-agent coordination for swarm robotics



Terrain Navigation

Examples

Boston Dynamics' Atlas, ANYmal quadruped, autonomous drones

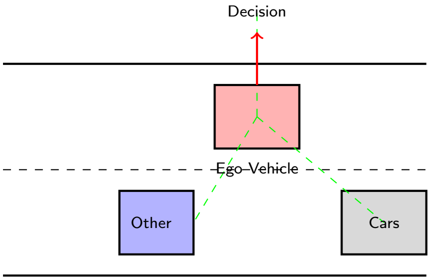
Autonomous Vehicles: RL on the Road

Driving Challenges:

- Multi-agent environment (other vehicles, pedestrians)
- Safety-critical decisions
- Uncertain human behavior
- Weather and lighting conditions
- Regulatory compliance

RL Applications:

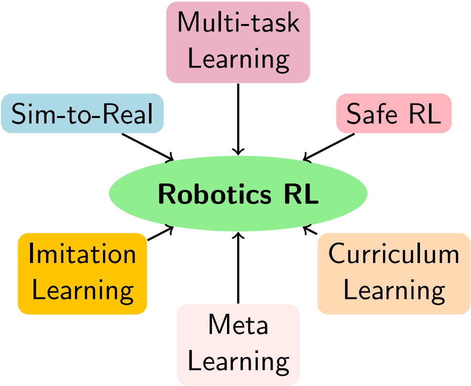
- Lane changing and merging
- Intersection navigation
- Parking and low-speed maneuvers
- Energy optimization
- Route planning



Safety First

RL must be combined with formal verification and safety constraints

Robotics RL: Key Techniques



Integration is Key: Most successful robotic systems combine multiple techniques to achieve robust performance.

Recommendation Systems: The Personal Touch

Why RL for Recommendations:

- **Sequential Decision Making** - What to show next matters
- **Long-term User Satisfaction** - Beyond immediate clicks
- **Exploration vs Exploitation** - Try new content vs safe bets
- **Dynamic Preferences** - User interests evolve over time
- **Multi-objective Optimization** - Engagement, diversity, fairness

Traditional vs RL Approach:

- Traditional: Predict rating/click probability
- RL: Maximize long-term user engagement

RecSys RL

User $\rightarrow$ State
Recommend $\rightarrow$ Action
Feedback $\rightarrow$ Reward

Personalized experience

Key Insight

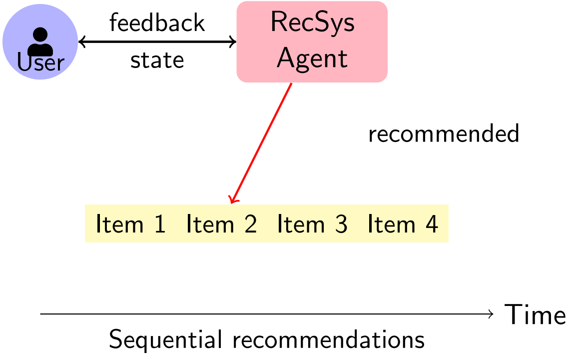
RL treats recommendation as a sequential decision problem, optimizing for long-term user satisfaction rather than immediate clicks.

RL Components in RecSys:

- **State** - User profile, context, history
- **Action** - Item/content to recommend
- **Reward** - Click, rating, time spent, purchase
- **Policy** - Recommendation strategy
- **Environment** - User behavior model

Challenges:

- Large action spaces (millions of items)
- Sparse and delayed rewards
- Cold start problem
- Filter bubbles and diversity



Real-World RecSys Applications

YouTube (Google):

- Billion+ hours watched daily
- Multi-objective: watch time, clicks, satisfaction
- Deep RL for video sequencing
- Handles fresh content and trending topics

Netflix:

- Personalized homepage layouts
- Contextual bandits for A/B testing
- Long-term user retention focus
- Content production guidance

Amazon:

- Product recommendations
- Cross-selling and upselling
- Inventory and pricing optimization
- Supply chain integration

Spotify:

- Playlist generation (Discover Weekly)
- Music flow and transitions
- Artist promotion balance
- Audio content recommendations

Business Impact

RL-based recommendation systems drive significant revenue: 35% of Amazon sales, 80% of Netflix views, 70% of YouTube watch time.

RecSys RL: Technical Approaches

Contextual Bandits

- Single-step decisions
- Fast online learning
- Exploration-exploitation
- Industry standard

Deep Q-Networks (DQN)

- Multi-step planning
- Value-based learning
- Experience replay
- Large action spaces

Actor-Critic Methods

- Policy optimization
- Continuous actions
- Multi-objective
- Long-term rewards

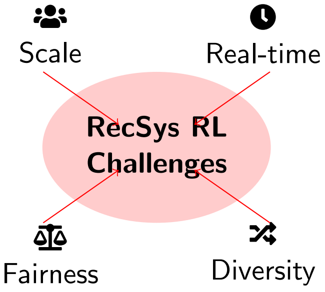
Hybrid Approaches: Many systems combine multiple techniques, e.g., bandits for exploration + deep RL for long-term optimization.

Technical Challenges:

- **Cold Start** - New users/items with no history
- **Scalability** - Millions of users and items
- **Real-time Inference** - Sub-100ms response times
- **Offline Evaluation** - Hard to simulate user behavior

Business Challenges:

- **Filter Bubbles** - Over-personalization
- **Fairness** - Equal exposure for content creators
- **Diversity** - Avoiding repetitive recommendations
- **Explainability** - Why this recommendation?



Solution Approaches

Multi-objective RL, constrained optimization, ensemble methods, and hybrid human-AI systems.

Common Patterns Across Domains

Games

Perfect info
Clear rewards
Simulated env

Common Challenges

- Exploration
- Sample efficiency
- Generalization
- Multi-objective

Robotics

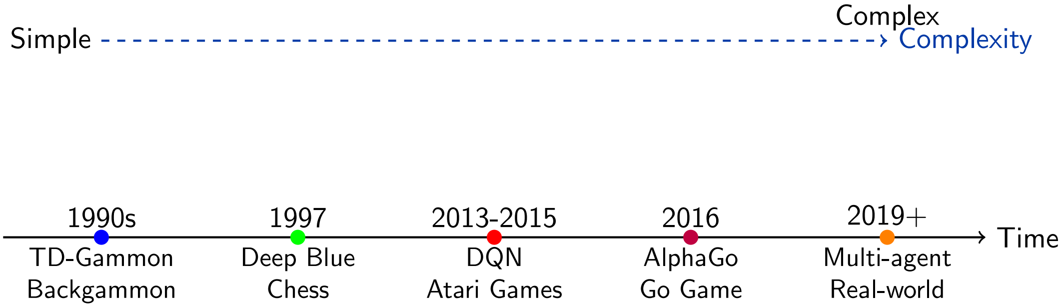
Physical world
Safety critical
Continuous control

RecSys

Human behavior
Large scale
Business metrics

Key Insight: Despite different domains, core RL principles and challenges remain consistent.

Evolution of RL Applications



Trend: From simple, controlled environments to complex, multi-agent, real-world applications.

Current Frontiers and Challenges

Technical Frontiers:

- **Sample Efficiency** - Learning from fewer interactions
- **Transfer Learning** - Knowledge across domains
- **Multi-agent Systems** - Coordination and competition
- **Safe RL** - Constraints and guarantees
- **Offline RL** - Learning from logged data

Emerging Applications:

- Drug discovery and healthcare
- Climate modeling and sustainability
- Scientific discovery automation
- Creative content generation

Societal Challenges:

- **Fairness and Bias** - Ensuring equitable outcomes
- **Transparency** - Explainable AI decisions
- **Privacy** - Learning without exposing data
- **Robustness** - Handling adversarial inputs
- **Alignment** - Ensuring AI goals match human values

Future Vision

RL systems that are sample-efficient, safe, interpretable, and aligned with human values.

Key Takeaways

What we learned about RL applications:

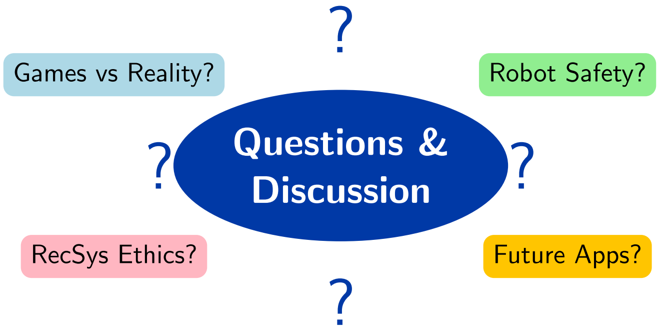
1. **Domain Diversity** - RL successfully tackles vastly different problem types
2. **Common Principles** - Despite domain differences, core RL concepts remain universal
3. **Progressive Complexity** - Applications have evolved from simple to incredibly sophisticated
4. **Real-World Impact** - RL drives significant technological and business advances
5. **Ongoing Challenges** - Technical and societal issues continue to drive research

Looking Forward

The next decade will see RL applications become more sample-efficient, safer, and more closely integrated with human decision-making processes.

RL: From Games to Real-World Impact

Questions and Discussion



Discussion prompts:

- Which application domain do you find most interesting and why?
- What ethical considerations should guide RL application development?
- How might RL applications evolve in the next 5-10 years?