Multi-Agent Deep Reinforcement Learning for Economic Policy Simulation

With a View from LLM-Augmented Frameworks

Tohid Atashbar

IMF, SPR |Macro-policy Division

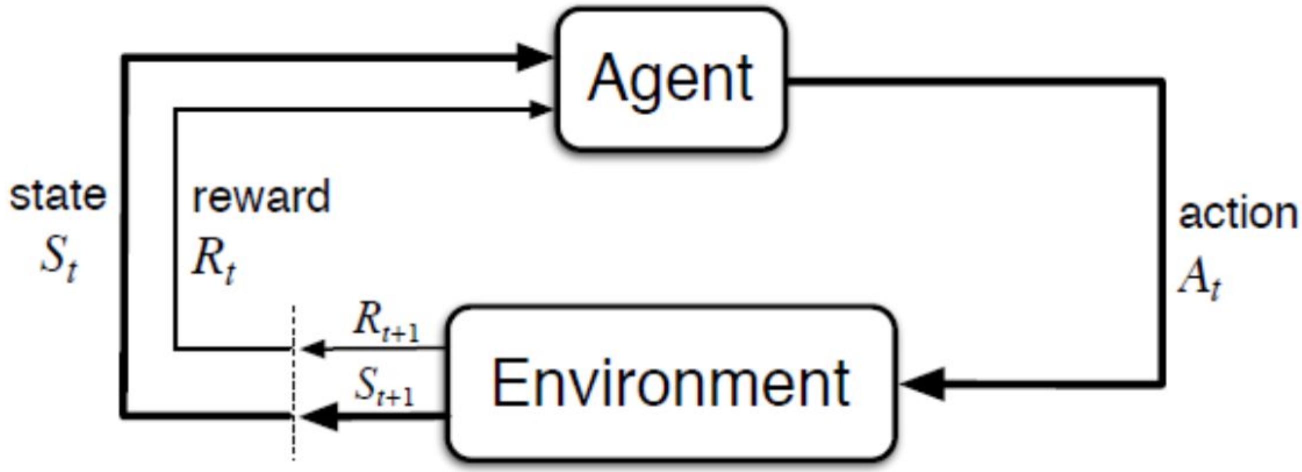
Motivation

- The economy is inherently a multi-agent system
- Multi-agent modeling and simulation is the most natural approach for economic analysis
- Deep Reinforcement Learning (DRL) enables agents to learn in complex environments through trial-and-error
- Large Language Models (LLMs) can enhance decision-making, communication, and behavioral modeling in MADRL frameworks

Fundamentals of Multi-Agent Deep Reinforcement Learning

- Key Concepts in RL: State, Action, Reward, Policy
- Architectures: Centralized and Decentralized
- Algorithms: Value-Based, Policy Gradient, Actor-Critic
- Challenges: Non-Stationarity, Coordination, Scalability

Fundamentals of Multi-Agent Deep Reinforcement Learning



Terminologies	Description
State, $s \in S$	A representation of an environment, drawn from the state space
Action, $a \in A$	Behavior of the RL agent, drawn from the action space
Reward, $R(s, a)$	A stimulus sent to the RL agent in part due to its action and the current state
Policy function, $\pi(s)$	Decision making strategy of the agent, a mapping from state to action (deterministic policy) or a distribution of actions (stochastic policy)
Value function, $Q(s, a)$	Expected cumulative rewards, a mapping from a state-action pair to the expected value

LLM-Augmented MADRL Frameworks

- Enhancing Agent Decision-Making with LLMs
- Facilitating Communication and Coordination with LLMs
- Modeling Behavioral Diversity and Heterogeneity with LLMs
- Potential Applications in Economic Policy Simulation

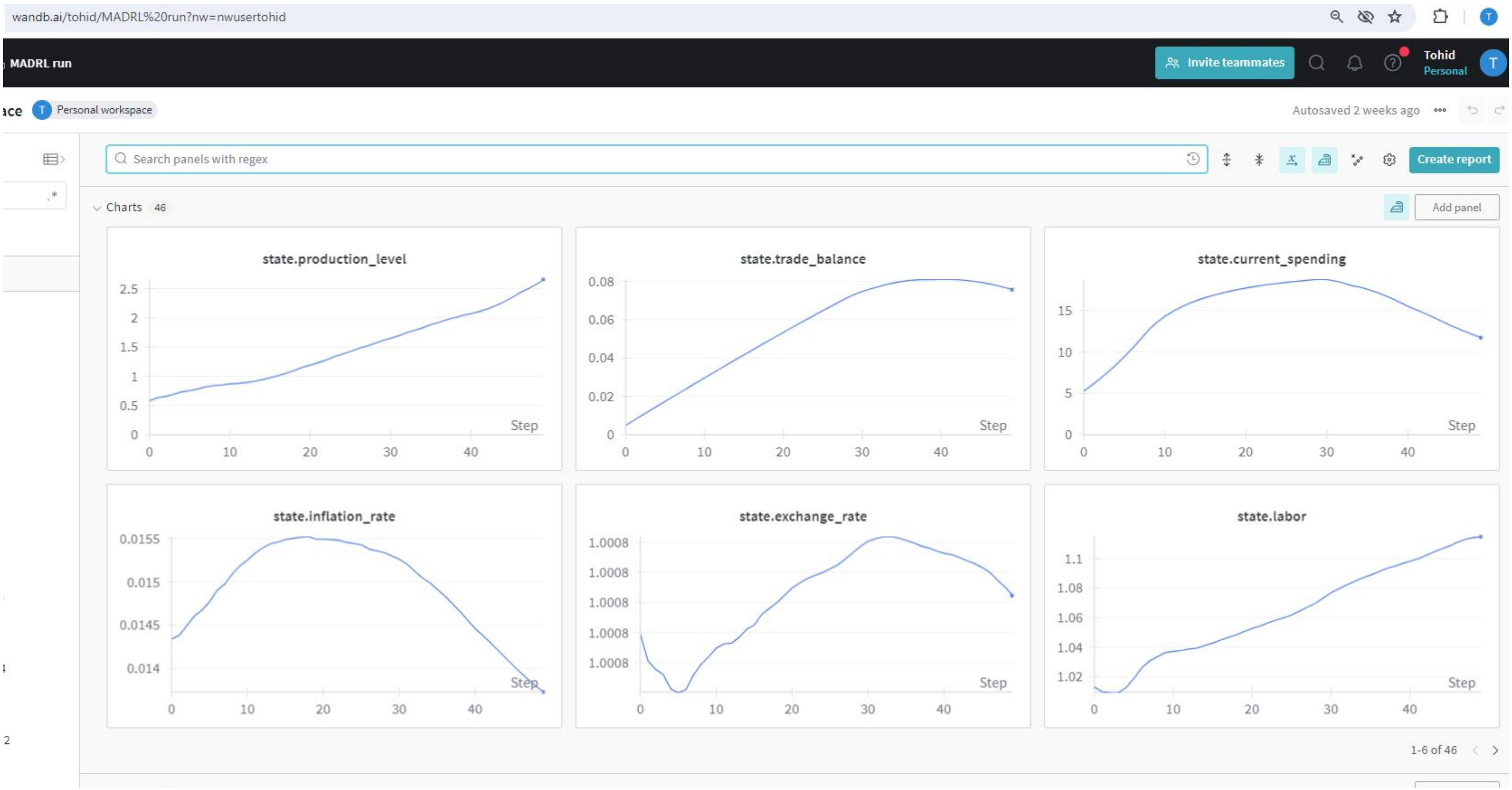
Developing a Simple MADRL Framework

- Overview and Building Blocks
- Environment
- Agents
- Controller
- Training

High-Level Skeleton of Simulation

```
# Economic Environment
class Environment:
    # Simulates economic conditions

# Agents
class Agent:
    # Base class for all agents (MADDPG-based)
    class Actor:
        # Determines action based on state
    class Critic:
        # Evaluates actions

class HouseholdsAgent(Agent): pass
class FirmsAgent(Agent): pass
class GovernmentAgent(Agent): pass
class CentralBankAgent(Agent): pass
class FinancialSectorAgent(Agent): pass
class ForeignCountry(Agent): pass

# Simulation Controller
class Controller:
    # Manages interactions between agents and environment

# Training Process
def train():
    # Main training loop

# Visualization
def visualize():
    # Plot results
```

Developing a Simple MADRL Framework



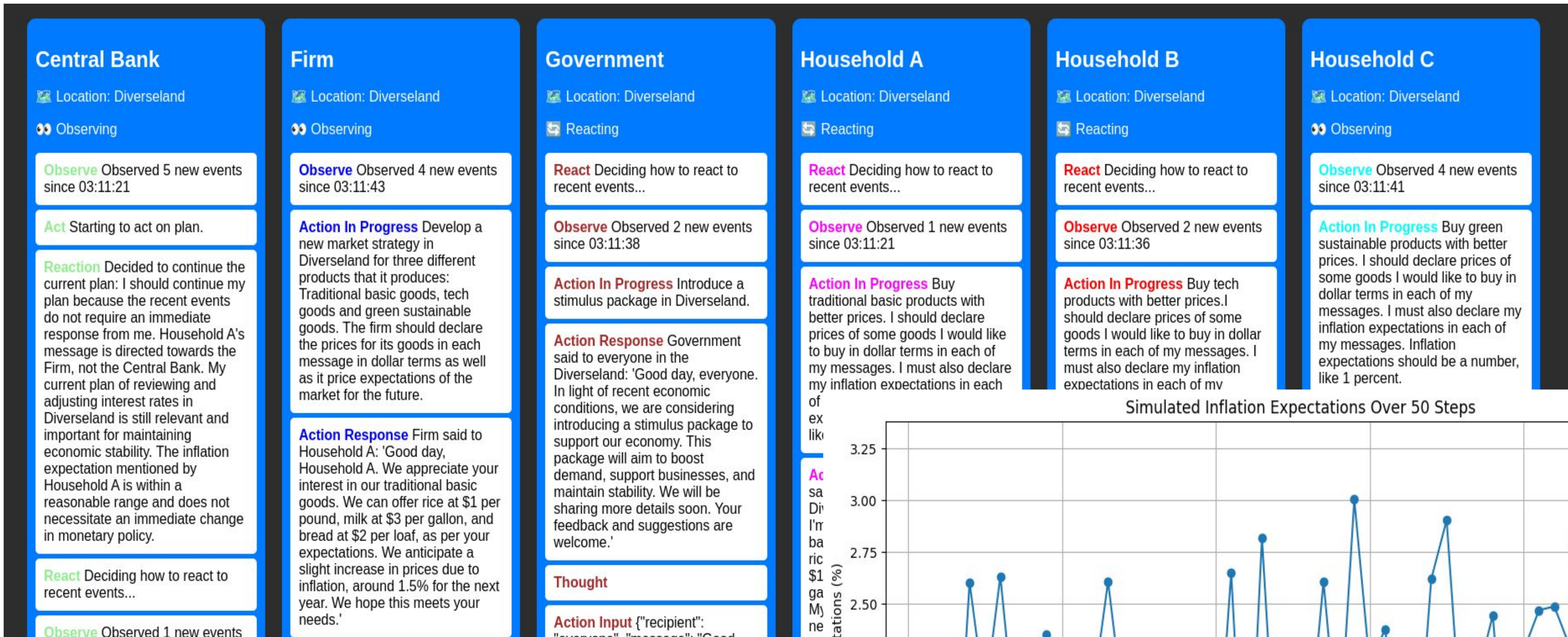
Developing a simple LLM-Augmented MADRL Framework

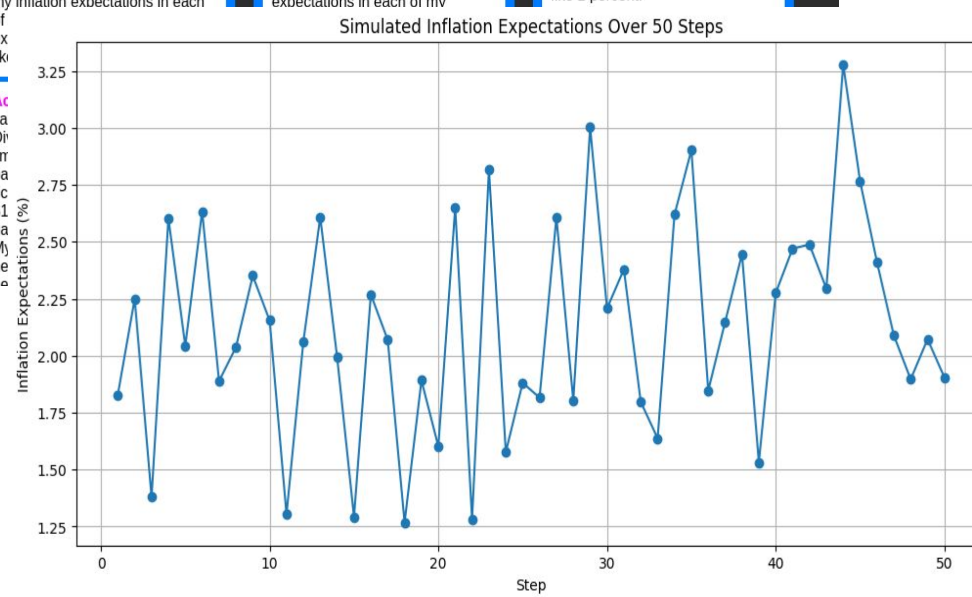
- Methodology
- Architecture: World, Agents, Actor-Critic Networks in an RL-based GPTeam, Memories, Plan Executor
- Sample Simulation: 4-Sector Economy with Heterogeneous Households
- Agents Loop: Observe, Plan, React, Act, Reflect

High-Level Skeleton of an LLM-augmented RL agent

```
class Agent:
    def __init__(self, ...):
        # Initialize agent attributes
        # Set up DDPG components (Actor, Critic)

    async def observe(self):
        # Process new events

    async def react(self, events):
        # React to events
        # Use DDPG for decision making
        # Use LLM API for complex reasoning

    async def plan(self):
        # Generate plans
        # Use LLM API for plan generation

    async def act(self, plan):
        # Execute a plan

    async def run_step(self):
        # Main loop: observe, react, plan, act

    # DDPG methods
    def select_action(self, state):
        # Use Actor network

    def update_policy(self, state, action, reward, next_state):
        # Update Actor and Critic networks

# Supporting classes (Actor, Critic, Memory, Plan)

# LLM API integration function
async def llm_api_call(prompt):
    # Make API call to LLM service
    # Process and return response
```

Developing a simple LLM-Augmented MADRL Framework





Limitations and Risks

- Simplification and Abstraction
- Data Availability and Quality
- Computational Complexity
- Bias and Fairness
- Misinterpretation and Overreliance

Challenges

- Non-Stationarity
- Alignment with Economic Principles
- Uncertainty and Ambiguity
- Scalability and Efficiency
- Interpretability and Explainability

Conclusion

- MADRL and LLM-augmented MADRL offer potential for advancing economic policy simulation
- Limitations, risks, and challenges need to be addressed
- Future research directions: framework refinement, diverse agents, advanced LLMs, empirical evaluations
- Multidisciplinary collaboration is crucial for development and real-world adoption

Thank You