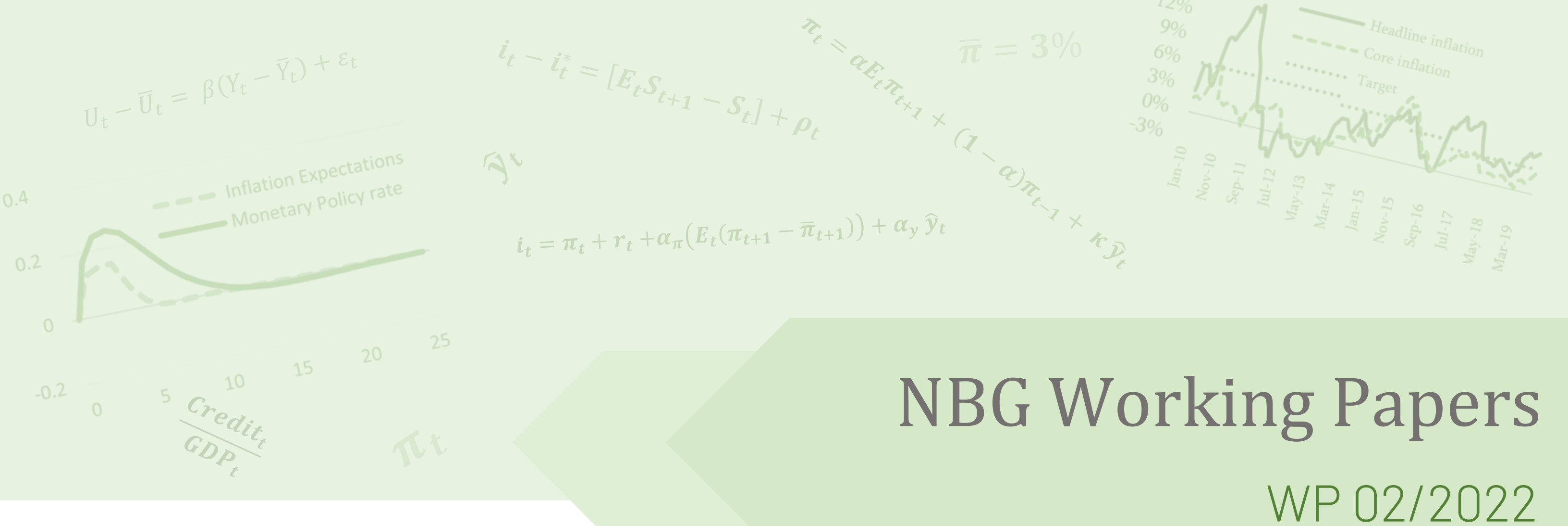


NBG Working Papers

WP 02/2022

Application of Artificial Intelligence for Monetary Policy-Making

by Mariam Dundua and Otar Gorgodze

***The National Bank of Georgia's (NBG) Working Papers are published to elicit comments and encourage debate on ongoing research**. Working Paper Series aim to present original research contributions relevant to central banks. The views expressed here are those of the author(s) and do not necessarily represent the views of the NBG. No responsibility for them should be attributed to the NBG.*

საქართველოს ეროვნული ბანკი
National Bank of Georgia

Application of Artificial Intelligence for Monetary Policy-Making[*]

Mariam Dundua[†] and Otar Gorgodze[‡§]

November 2022

Abstract

The recent advances in Artificial Intelligence (AI), in particular, the development of reinforcement learning (RL) methods, are specifically suited for application to complex economic problems. We formulate a new approach looking for optimal monetary policy rules using RL. Analysis of AI generated monetary policy rules indicates that optimal policy rules exhibit significant nonlinearities. This could explain why simple monetary rules based on traditional linear modeling toolkits lack the robustness needed for practical application. The generated transition equations analysis allows us to estimate the neutral policy rate, which came out to be 6.5 percent. We discuss the potential combination of the method with state-of-the-art FinTech developments in digital finance like DeFi and CBDC and the feasibility of MonetaryTech approach to monetary policy.

JEL Codes: C60, C61, C63, E17, C45, E52
Keywords: Artificial Intelligence; Reinforcement Learning; Monetary policy

The National Bank of Georgia's (NBG) Working Papers are published to elicit comments and encourage debate on ongoing research*. Working Paper Series aim to present original research contributions relevant to central banks. The views expressed here are those of the author(s) and do not necessarily represent the views of the NBG. No responsibility for them should be attributed to the NBG. The working papers have not been peer-reviewed.*

[*] We would like to thank Shalva Mkhatrishvili (NBG), Papuna Lezhava (NBG) and Mikheil Gavasheli (NBG) for useful comments and discussion. We would also like to thank Financial and Supervisory Technology Development, Macroeconomics and Statistics, and Financial Stability Departments at the NBG for their feedback. All errors and omissions remain with the authors.

[†] Corresponding author: Mariam Dundua, National Bank of Georgia, Leading Specialist at Financial and Supervisory Technology Development Department (e-mail: mariam.dundua@iset.ge)

[‡] Contributed to the research when working as the Head of Financial and Supervisory Technologies Department at National Bank of Georgia. Currently, working as a Resident Advisor on Banking Supervision Senior at the IMF (e-mail: ogorgodze@imf.org)

[§] Disclaimer: The views expressed here are of the authors and should not necessarily coincide with opinions held at NBG or IMF.

Contents

INTRODUCTION 3
LITERATURE REVIEW 5
Review of applications of RL in economics 5
Review of simple rules in monetary policy 6
METHODOLOGY 9
a. Reinforcement Learning Fundamentals 9
b. Policy-based model-free methods 11
Actor-Critic methods 12
Proximal Policy Optimization 12
c. Recurrent Neural Networks (GRU) 13
PROBLEM DEFINITION 15
MODELLING STEPS 16
1st stage: Generate transition equations for Gym environment 16
Macroeconomic Modelling: GRU 17
Interpret GRU model 19
2nd stage: Training Macro Agent 22
3rd Stage: Making sense of training results 25
Explainibility Technique of RL agent behavior 25
Testing the agent against the shocks 27
Nonlinearity of RL Agent Policy Reaction Function 33
Counterfactual analysis 36
Conclusion 40
References 41
Appendix 44
Appendix A: GRU model Structure 44
Appendix B: Hyperparameters of GRU model 44
Appendix D: Variable Importance For GRU Model based on SHAP Value 48
Appendix E: GRU model output explainability by SHAP Value 49
Appendix F: Testing Result against shocks 52
Appendix G: Counterfactual analyses multi-level decision 55

INTRODUCTION

Rules vs. discretion is a central debate for monetary policy. The inflation targeting (IT) regimes that spread from the early 1990s was a step towards more rules-based policy and explicitly defined objective function of the central bank. However, any attempts to tie operational procedures of a central bank to an objective function remains elusive. So called simple rules for monetary policy remain only indicative and cannot be used in a literal sense in practice. The problem of making monetary policy rules-based is most strikingly apparent for crypto assets where monetary policy is programmed and, indeed, rules-based by default, but these assets exhibit wild price instability.

We propose to use AI to guide design of practical monetary policy rules. For the empirical application of the rules design we use Georgian economy where the Central Bank has explicit IT regime. (Monetary Policy Operations Manual, 2021). The monetary policy committee, when setting monetary policy rates, makes decisions based on the inflation forecast, the current macroeconomic environment, and the state of the financial markets. The decision-making process is sequential and can be expressed as a Reinforcement Learning framework. It gives us an opportunity to describe everything as an environment for RL agent and changes to monetary policy rate as an agent action. RL approach can be considered as at least complement to the Forecasting and Policy Analysis System, which is one of the key components for monetary policy conduct to ensure the monetary policy rate is forward-looking. This paper will discuss the RL framework possibility during the monetary policy decision-making process.

Models developed in macroeconomic theory are usually limited to simplistic behavioral models, which have restricted opportunities to conduct policy experimentation. More complex models have another problem. The common approach for modeling macroeconomics is a Dynamic Stochastic General Equilibrium (DSGE) model. However, these models, while useful in some applications, have their limitations. To solve DSGE nonlinearly is very challenging, and linearized models may have less robustness in the face of large shocks than nonlinear DSGE models. The curse of dimensionality is another problem of this approach.

In response to this challenge, deep learning-based economic simulation can be a powerful framework to eliminate these limitations. Economic simulation allows training the reinforcement learning macro agent, which will learn nonlinear decision-making rules. Deep learning models can learn a nonlinear pattern in high dimension. The RL models have superhuman performance in many sequential highdimensional tasks (Silver, et al., 2017) (Oriol Vinyals, 2019) (Kober, Bagnell, & Peters, 2013). Such process of learning most closely mimics human decisions. The agent learns policy in such a way to maximize long-term reward. In this manner, it is similar to cumulated discounted future utilities in economics. The agent learns policy through interacting with the environment. In this paper, the environment is an empirical model trained on the historical macroeconomic data of Georgia. This empirical model describes a relationship between macroeconomic variables and predicts future values. The paper uses deep learning-based recurrent neural networks, specifically Gated Recurrent Unit (GRU), to train the empirical model. The GRU model is successfully used for sequential tasks (Chung, Gulcehre, Cho, & Bengio, 2014). The macro agent obtains observations about macroeconomic variables from the GRU model and decides the policy rate. Based on the reward, the agent obtains signals how good his decision is. The paper uses advanced RL technics to train RL macro agent. Agents' reward function is defined as a marginal change in inflation deviation from its target with negative sign. Inflation deviation from its target is represented with a custom loss function. Testing the RL model results shows that RL macro agent can attain the national bank of Georgia's mandate and stabilize the inflation rate close to a target value.

In this paper, we developed RL based macro agent that can decide on policy rate changes. The empirical results are presented on the example of the Georgian economy, using the advanced RL models and frameworks. The rest of the paper is structured as follows: First part is a preview of the application of RL methods in economics and simple rules in monetary policy. The next part describes the methodology. After defining the problem, three-stage approach central to this paper is explained. In the first stage, the GRU model is trained for environment modeling; in the next step RL model is trained, and the last step interprets the model results. Concluding remarks summarize the paper's main findings and highlight the possible expansion of the results.

LITERATURE REVIEW

Review of applications of RL in economics

It can be said that reinforcement learning application in economics is still in its infancy, however it has many applications in agent-based, state-based models or areas of repetitive interaction, such as game theory. There exist several applications where reinforcement learning methods can be used. For almost every problem which is classical optimal control, the RL framework can be used, since economics is a sequential imperfect observation game.

Recently Reinforcement Learning is widely used for economic simulation. RL is seen as a powerful tool for analyzing possible policies, presenting different scenarios and analyzing the results. This is especially important for policy makers. There are several trials to solve DSGE models using RL. In a DSGE model, it is challenging to find an optimal policy, because of its nonlinearity. The curse of dimensionality is another problem for DSGE. The latest awesome application of RL agents in economics have been attempted from Salesforce, Harvard University, and You.com (2021) and (2020). The paper has designed an optimal taxation economic policy via two-level deep reinforcement learning. The AI Economist recommends how to set taxes, subsidize and redistribute the wealth, while AI agents pay taxes. The paper conducts millions of economic simulations. In this simulation AI agent and AI Economist continuously learn to adapt to each other. The AI economist improves the equality productivity goal, set by the government. The AI Economist shows that economic policy using RL is sound, viable, flexible and effective. The work from the Bank of England solved heterogeneous general equilibrium economic models with Deep Reinforcement Learning (Hill, Bardoscia, & Turrell, 2021). The COVID-19 pandemic emphasized the importance of heterogeneous agents and RL allows heterogeneity to be easily built into the model. The paper shows that it is possible to analyses different potential scenarios in order to make various policy decisions and using reinforcement learning, it is possible to solve rational expectation models, by agent learning. One of the latest works using deep reinforcement learning was done by (Chen, et al., 2021). They trained forward-looking household RL agents and used the classical model which is an interaction of monetary and fiscal policy. The agents interact with the environment, it has no prior knowledge about the environment, only knows its utility function, observes economic variables, and does not know economic structure. Agent learns nonlinear decision rules, by taking action which is a decision about consumption at this particular period. The paper found out that when both monetary and fiscal policy is active or passive, AI agents can learn these equilibria, whereas an adaptively Learning Agent cannot. Curry, Trott, Phade (2022) used deep RL for finding general equilibria in many-agent economic simulation. This paper also uses hundreds of heterogonous RL agents, a scale that has not been studied before with strategic agents. Their modelling methods are flexible and can find a stable solution in a variation of real-business-cycle DSGE models. They find equilibria in open and closed economies. A little bit earlier, Sinitskaya and Tesfatsion (2015), did a similar job. They used tabular Q-learning to train a small number of consumers and firms. Their consumer and workers were non-heterogeneous. The latest paper by Deutsche Bundesbank (Hinterlang & Tänzer, 2021) tries to compute optimal monetary reaction function by RL. This paper is the most similar what we are trying to do in our work as both of this paper aim is to find optimal monetary policy. In the first step the paper uses ANN specification to capture nonlinearities presented in the U.S quarterly data and estimate model transition equations, while our paper uses GRU network in order to take into consideration sequential information which is typical for time series data. In the second step they applied RL to compute optimal monetary policy reaction function. The papers take into consideration zero low bound constrain for policy rate, convex Phillips/IS curves and asymmetric preferences. All the result shows that resulting policy reaction functions outperforms other common rules as well as the actual federal funds rate.

Our work uses data-driven estimation for dynamic economic simulation, instead of theory. The modern RL agent tries to learn optimal monetary policy and make the best decision about key policy rates, to achieve the policy objective. The use of this kind of approach implies at least partial automation of monetary policy conduction.

Review of simple rules in monetary policy

Search for optimal rules for monetary policy conduct is one of the fundamental questions of monetary policy research and has profound practical implications for central banks. It is interesting that this question over the last years acquired another dimension due to development of decentralized crypto assets and stable coins. Monetary policy rules embedded in a decentralized code is a perfect example of rules-based monetary policy. However, currently it is obvious that they cannot compete with central banks monetary policy committees in policy-making, who use simple monetary rules only to support and guide their decisions. All these point to the fact that it is not easy to choose one policy rule, which will be a precondition for the sustainable development of the economy, because the macroeconomic environment is characterized by high uncertainty.

To review history, importance of policy rules was always emphasized by economists as early as Adam Smith, Henry Thornton, David Ricardo, Irving Fisher and Knut Wicksell who have argued importance of the well-regulated and rule-guided monetary policy (Taylor & Williams, 2011).

The simplest and one of the oldest monetary policy rules were proposed by Freidman (1960), (1968) known as Friedmans' k-Percent Rule. Following research of the Great Depression by Freidman and Schwartz (1963) and subsequent criticism of the monetary policy mistakes made in response to the crisis, Friedman suggested that the money supply be increased at constant growth rate. One of the main challenges Friedmans' k-Percent Rule facing is that its short-run stability depends on the stability of (i) real GDP growth and (ii) velocity of money growth. Furthermore, the central bank can directly influence only high-powered money which is not linked to aggregated demand making this rule unpractical.

Intersect rate is most common instrument used to conduct monetary policy. Therefore interest rate rules are more popular. Most popular interest rate is Taylor Rule and its modifications. According to the Taylor rule (1993) changes in short-term interest rate depend on equilibrium real interest rate, inflation deviation from its target, and the real output gap. Taylor rule says that the central bank will react by changing the interest rate in response to both inflation and output gap. But there are lags in both the measurement of inflation and in the transmission of monetary policy decisions. These considerations motivate another extension of the Taylor rule, which assumes that policy responds not to current inflation but to expected one. This is called the forward-looking Taylor rule. Taylor rule reacts to economic conditions, by adjusting short-term interest rate, hence it is more flexible, than the Friedman rule. The main problem this rule is facing is to find the suitable coefficient for output gap and inflation. If there is no right balance between coefficients, central banks make the decision based on incorrect information, which leads to resource misallocation.

Use of new Keynesian DSGE models created opportunity to have analytical framework to analyze performance of optimal policy rules. Taylor and Williams (2011) summarizes large literature devoted to this approach and identify three general characteristics of simple optimal policy rules:

(1) an interest rate instrument performs better than a money supply instrument;

(2) interest rate rules reacting to both inflation and real output worked better than rules that focused on either one;

(3) interest rate rules that reacted to the exchange rate are inferior.

However, it is hard to identify which simple rule is preferable and results in the literature are inconclusive and contradictory. Knotek at.al. (2016) summarizes 7 simple rules that is used by the Federal Reserve to inform monetary policy. The performance of these rules are updated and published quarterly (Cleveland, n.d.). The results indicate large within-rule variation as well as large deviations from the actual monetary policy.

The fact that there are many versions of the simple monetary rules and they are not used more actively for the actual monetary policy decisions could be result of their simplicity. Actual policy rules should take into account more variables than is indicated by the simple monetary rules. Research on optimal policy rules aims to address this shortcoming by formulating optimal policy as optimal control problem and taking into account all relevant information for monetary policy (Giannoni & Woodford (2005); Svensson (2010); Woodford (2010)). However, research in optimal policy rules has even less practical relevance and they are rarely used by the policy makers. The drawback of optimal rules are that they are highly sensitive to model specification as well as suffer from complexity and are difficult to analyze and explain.

It should be noted that a search for optimal policy using micro-founded or structural models has an inherent problem that stems from the limitations of traditional linear modeling toolkits that economists use. The traditional models by economists are analytical simplification of the reality and cannot capture all relevant information needed for the conduct of monetary policy. if you add more equations and variables, analytical models become difficult to solve and interpret. An optimal monetary policy rule is derived from the incomplete model of a complex macroeconomic environment. As a result there is no consensus between economists about one particular policy rule.

However, more complex rules are difficult to construct using with traditional economic tools and often exhibit instability.

Finding practical optimal policy rule will require more complex modeling approach one that treats blurred distinction between model and data. Modern AI toolkits could hold promise to solve address this problem.

METHODOLOGY

a. Reinforcement Learning Fundamentals

Reinforcement learning stands out as a potential path to general intelligence. Researchers have trained AI agent to achieve impressive control across various robotic systems. Reinforcement learning has been leveraged in multiple real-world use cases, including application in self-driving cars, in industry reinforcement learning-based robots are used to perform various tasks, it has application for automated stock trading in finance, also in healthcare, where patients can receive treatment from policies learned from RL systems, and there are many other real-life RL applications.

Every reinforcement learning problem is the interaction between agent and environment. Agent observes the environment, makes the decision, and, based on the decision, takes some action $a \in A$ and transmits from one state $s \in S$ to another. At which state the agent will end up is defined by transition probabilities (P). The environment changes when the agent acts and might be changed by itself. The change in the environment following an agent's action is determined by a model that we may or may not know.

After the agent takes the action, it receives the Reward $r \in R$. The Reward is the signal for agents how good this state is. Agent's goal is to maximize cumulative expected reward, this is called return. At time t, return is:

$$max_{\pi} \, E_t[G_t] \quad G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \qquad (1)$$

Where γ is the discount factor, which shows how much the agent values the future.

The model defines transition probabilities and reward functions. The transition is modelled as a Markov decision process (MDP):

$$T: Pr\,(S_{t+1}|S_t, a_t) \; (2)$$

There are two possibilities about the model:

- Model is known: When there is everything known about the environment, Dynamic programming can be used to find optimal solutions or do model-based Reinforcement learning.
- Model is unknown: do the model-free Reinforcement learning.

Policy determines what action to take in a particular state s. The policy may be deterministic or stochastic:

$$\text{Deterministic: } a_t = \pi(s) \quad (3)$$

$$\text{Stochastic: } \pi(a|s_t) = P_\pi[A = a|S = s] \quad (4)$$

Each time the agent finds itself in a particular situation, it will decide about a certain action and face the consequences of that action in the future. It implies that the state has value (value function $V(s)$), and moreover, state and action pairs (action-value function) $q(s, a)$ have value as well. The agent uses future values, to evaluate the state s. It is an explicit measure of how good the state is. RL tries to learn policy and value function. Mathematically, the value function is defined:

$$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|S_t = s] for\ all\ s \in S \quad (5)$$

The same way is defined action-value function, which shows how good this particular state (s) is if agent takes action a and follows the policy π. In other words, it is expected return at state s selecting an action a and following the policy π.

$$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|S_t = s, A_t = a] for\ all\ s \in S \quad (6)$$

The difference between action-value function and state-value function is called “advantage”

$$A_\pi(s, a) = v_\pi(s) - q_\pi(s, a) \quad (7)$$

On the other hand, if action is taken to maximize the action-value function, it is called the optimal action-value function, and in the mathematical form it is the following:

$$q^*(s, a) = max_{\ \pi}\, q_\pi(s, a) \quad (8)$$

And the same for optimal value function:

$$V^*(s) = \max_\pi q_\pi(s, a) \quad (9)$$

And the policy that gives the optimal value function is called optimal policy π^*:

$$\pi^* = \underset{\pi}{arg\,max}\, V_\pi(s) \quad (10)$$

In a more formal way, almost all reinforcement model problem can be written as Markov decision Process (MDP). As MDP suggests each state depends only on previous state and agents' action, called "Markov" property. Any MDP contains 5 tuple $< S, A, R, P, S_0 >$

Value function $V(s)$ can be decomposed as immediate reward plus discounted future values.

$$V(s) = E[R_{t+1} + \gamma V(S_{t+1}) | S_t = s] \quad (11)$$

And similarly for action-value function:

$$q(s,a) = E[R_{t+1} + \gamma E_{a \sim \pi} q(S_{t+1}, a) | S_t = s, A_t = a] \quad (12)$$

These kinds of equations (11) and (12) are called Bellman equations and to solve these equations Dynamic programming (DP) can be used. In case of DP, the model is known and it iteratively evaluates value functions and improves policy.

In some cases, the probabilities of getting from one state to another are unknown. An agent interacts with the environment, or in other words, sample from the environment and, this way, learn the distribution of the environment. The more the agent interacts with the environment, which means sampling from each state and action, the closer the average reward is to the actual reward. It is true because of the law of large numbers. In some cases, the environment contains large state space, and this approach is not practical. In this case, we can parametrize the action-value function and use a deep neural network in order to approximate the action-value function:

$$q^*(s,a) \approx q(s,a,\theta) \quad (13)$$

The update rule for θ derives from various reinforcement learning algorithms; one example is deep Q-learning. It is a value-based model-free reinforcement learning problem.

b. Policy-based model-free methods

It is possible to directly approximate policy function $\pi(a|s,\theta)$ instead of the value function and update θ by some learning algorithms. Policy-based approaches are efficient when the environment is in continuous space. Since, in our setup, there are an infinity of many actions and states, value-based approaches are computationally expensive and ineffective. Our monetary policy agent faces continuous space, so policy-based methods are more relevant to use in this case.

In the policy gradient method, the goal is to approximate policy function and the policy function itself is a stochastic function. The algorithm is trained to maximize the expected return. Gradient ascent can be used to find the best θ, which provides the maximum expected returns.

The agent learns the environment by using policy and uses the same policy in order to explore and update the model, that is why policy gradient and actor-critic model is on policy model-free learning.

Actor-Critic methods

If both, the value function and policy are learned by the agent it is the Actor-Critic Algorithm. In this kind of algorithm, there are simultaneously two functions, the goal for both of them is to improve each other. One of them called actor and the second critic:

- Critic aims to approximate value function $V(s, w)$ or action-value function $q(s, a, w)$, depending on the algorithms, and it has its own learnable parameters w;
- Actors who try to approximate policy function $\pi(a|s, \theta)$ also has their learnable parameter θ, and this parameter is updated by critic suggestion.

This paper uses Proximal Policy Optimization (PPO) actor-critic algorithms to train monetary policy agent.

Proximal Policy Optimization

PPO proximal policy optimization (2017) is an actor-critic algorithm, which is the simplification of Trust Region Policy Optimization (TRPO) (2017) algorithms, but the performance is the same. TRPO objective functions often lead to extremely large parameter updates. PPO introduce clipping term to the objective function and discourages large policy change.

Let's denote $r(\theta)$as a probability ratio between old and new policies:

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} \quad (14)$$

PPO tries to constraint $r(\theta)$ to stay in a small interval $[1-\varepsilon, 1+\varepsilon]$, where ε is a hyperparameter. Clip objective function for PPO is:

$$J^{Clip}(\theta) = E[min\left(r(\theta), \hat{A}_{\theta_{old}}(s,a), clip(r(\theta), 1-\varepsilon, 1+\varepsilon\ )\hat{A}_{\theta_{old}}(s,a)\right)] \quad (15)$$

The function $clip(r(\theta), 1-\varepsilon, 1+\varepsilon\ )$ clips the ratio to between $1-\varepsilon$ and $1+\varepsilon$. The objective function takes a minimum between the old value and the clipped one. It forces not to update parameters extremely. PPO guarantees more stable policy networks, it is also easier, thus it is chosen to train macro agent.

c. Recurrent Neural Networks (GRU)

Gated Recurrent Units (GRU) (2014) is version of recurrent neural networks (RNN). RNN are used for sequential data. GRU is slightly simplified version of Long Short-Term Memory Units (LSTMs) (Hochreiter, 1997), which is the most widely used version of RNN. General RNN architecture suffers from vanishing and exploding gradients problem (Bengio, Simard, & Frasconi, 1994). Both of this version of RNN, are effective solution for this problem. The main reason why GRU was adopted in this paper is that it has fewer training parameters than LSTM, thus requiring less data for generalization.

Let's begin classical Multilayer perceptron (MLP), with single layer. Activation function of hidden layer denote by φ. The hidden layer is calculated as:

$$h = \varphi(x_t * W^{xh} + b_h) \quad (16)$$

In equation (16) $h = \varphi(x_t * W^{xh} + b_h)$ (*16*there is a weight parameter W^{xh} and a bias parameter b_h. h is used as the input of the last layer, so output layer will be:

$$o = h * W + b \qquad (17)$$

Where o is output variable, W and b is the weight and bias parameters of the last layer. This is classical MLP with one hidden layer. In case of sequential data, must take advantage of the fact that it is possible to determine the future based on the past. In order to carry out past information, we can save the hidden layer computed in the previous time step h_{t-1} and introduced new weight parameters W^{hh} to describe how to use the hidden variable of the previous time step in the current time step. More precisely, to calculate current time step hidden layer, we need previous time step hidden variable and input of the current time step.

$$h_t = \varphi(x_t * W^{xh} + h_{t-1}W^{hh} + b_h) \quad (18)$$

So, relationship between h_t and h_{t-1}captures sequential information up to current time step. This kind layer called recurrent layer. This is simple Recurrent Neural Network.

As mentioned above, GRU is one of the RNN. It has update and reset gate. It means, we have tool which determines when a hidden state should update and when it should be reset. This tool is learnable.

“Update gate” controls how much of the past information passed to the future. If the first observation is very important to determine future, the network does not update hidden state after this important observation. “Update gate” is defined by the following formula:

$$z_t = \sigma(W^z x_t + U^z h_{t-1} + b_z) \quad (19)$$

Where x_t is the network unit and W^z is its weight. The same is true for h_{t-1} and b_z is bias parameter. On their sum is applied sigmoid activation function.

“Reset gate” controls how much of the past information to forget. If some past observation is not important to determine future values, the network will skip this observation. The “Reset gate” is defined by:

$$r_t = \sigma(W^z x_t + U^z h_{t-1} + b_r) \quad (20)$$

Where parameters has the same meaning as in the “update gate”.

To integrate reset gate in our state updating process. It leads candidate hidden state. It is computed:

$$\widetilde{h_t} = tanh(W x_t + U(r_t h_t) + b) \quad (21)$$

The $\widetilde{h_t}$ is candidate hidden state, because there is room to take into consideration update gate. When reset gate is close to 1 this network is very like to (18), which is general RNN. When reset gate is close to 0, the result is that candidate hidden state is simple MLP with just $x_t$input.

The hidden state h_t at time t is weighted combination of candidate activation $\widetilde{\mathrm{h_t}}$ and previous activation $\mathrm{h_t}$. As weight are used “update gates” $\mathrm{z_t}$.

$$h_t^j = (1 - z_t) * h_{t-1}^j + z_t * \tilde{h}_t^j \quad (22)$$

The update gate determines how much the new candidate state can be used. When update gate is close to 1, the network uses the old hidden state, and ignores information from x_t, while update gate is close to 0, the new hidden state h_t^j is close to old hidden state h_{t-1}^j . This kind architecture helps to avoid vanishing and exploding gradients problem and efficiently pass sequential information.

PROBLEM DEFINITION

We model the macroeconomic environment for the macro agent as Markov Decision Process. As in MDP, here are State-space, on which agent interacts, Action space - agent decision about monetary policy after the interacting with environment, Reward – agent reward receiving after taking a specific action, Policy – agent decision-making strategy at a particular state, which represents probability distribution over the state space and expected reward after taking action and follow the optimal policy.

- State Space $S \in R^6$, where Agent at time t observes current environment, real effective exchange rate, annual inflation, economic growth, and current loss defined by (24) equation
- Action Space is in $a \in R^1$. An agent takes action from$[-0.5,0.5]$ interval, which means agent can maximum increase or decrease policy rate by 0.5 percentage points at every period.
- Reward $r(s, a, s')$ agent has at state s after taking action a and arriving a new state s'. Reward is defined by following equation:

$$r(t) = -1 * (loss_t - loss_{t-1}) \quad (23)$$

$$loss_t = \begin{cases} 0.98 * loss_{t-1} + 0.02 * abs(inflation_t - 3) \; if \; 0 < Policy\ Rate_t < 13^{**} \\ 0.98 * loss_{t-1} + 0.02 * abs(inflation_t - 3) + 1.5 \; else \end{cases}$$

(24)

Agent needs to know that, the policy rate should neither be too high nor negative. Thus, when the policy rate is negative or greater than 13 percent, the lost at time t increases by 1.5. It is penalty term for agent, when the policy rate is not in the desired range. Policy $\pi(s)$ agent decision making strategy at state s, which represent probability distribution over the state;

- Expected reward $Q_\pi(s, a)$ after taking action a, at state s and after following the optimal policy.

The Agents' environment is written in Gym [††] environment of Open AI.

[**] The choice of the upper bound does not change the results of RL Model, if the upper bound is less than 20

[††] https://gym.openai.com/

Modelling Steps

There are three stages used in the modeling:

1. Generate transition equations for the Gym environment. It is essential to have a environment to train the RL model. We use neural networks to estimate economic environment were dynamic optimization will take place.
2. Train AI policy model conditional on the generated transition equations. RL algorithm learns optimal policy constrained by the environment from the first stage.
3. Interpret monetary policy generated by RL. This step is necessary due to black box approach nature of AI.

1st stage: Generate transition equations for Gym environment

Generally, an agent does not know precise environment dynamics. It interacts with the environment to form an idea of the dynamics of the environment. Due to the macroeconomic environment, the macro agent cannot interact directly with macro variables. However, when interacting with estimated equations, a macro agent can form an idea of macro dynamics. This work proposes to use Recurrent Neural Networks (RNN), more specifically Gated Recurrent Units (GRU) (2014), to estimate the macroeconomic environment. The RNN is enabled to make the use of sequential information. It has a memory that captures information that has been calculated so far. That is why RNN is convenient for sequential type information, in particular for time-series data. To train the GRU model, we are using the TensorFlow framework.

To learn macroeconomic behavior in Georgia, RL macro agent observes different macro variables; these variables are part of the RL agents' environment. In order to enable the agent to discover how to adjust policy rate in response to the current macroeconomic situation, the agent needs to know the current inflation rate as it is a key indicator for price stability. More importantly, the macro agent tries to minimize future inflation deviation from the target. In addition, the agent observes real growth in Gross Domestic Income as part of the environment. To derive policy rate path, agent needs to know the value of local currency relative to trading partners in real terms. The annual increase in Real Effective Exchange Rate (REER) is another variable included in the macro agent environment. The annual growth rate of the M2 money aggregate is the last variable in given environment. As annual growth rates are taken for all the variables, they is no need for seasonal adjustment or differencing.

The necessary data to conduct this research are retrieved from the database of the National Bank of Georgia[‡‡]. Gross Domestic Product (GDP) is available quarterly. The paper uses GRU networks to describe the dynamics of the environment. Therefore, it needs observation as much as possible to train the network. A preliminary assessment of economic growth provided by the National Statistics Office of Georgia[§§] is used as a proxy for real GDP growth. All other variables used by the paper are available monthly. Available data starts from January 2010, and the last observation is in March 2022.

Macroeconomic Modelling: GRU

Macroeconomic indicators are closely related. Hence, it is better to evaluate all the equations together in one network instead of each equation having its network. The paper uses a GRU network architecture with several nodes in the last layer. The number of nodes in the last dense layers matches the number of variables the model tries to predict. The GRU predicts the next month's values. These variables are inflation rate, M2 annual growth rate, real effective exchange rate, and real economic growth. GRU network has five input nodes. The nodes/variables in the last layer are added policy rate. The model simultaneously tries to predict the inflation rate, M2 annual growth rate, real effective exchange rate, real economic growth, and input variables are annual inflation rate, M2 annual growth rate, real effective exchange rate, real economic growth, and policy rate. The lag values in the GRU model were selected based on forecast accuracy and economic sense. Mathematically it is represented by the following formula:

$$y_{t+1} = f(x_t) \quad (25)$$

Where f represents some function learnable by the GRU network

$$x_t = \left(policy_{t-4}, M2_{g_t}, REER_t, Inflation_t, Economic\_g_t\right) \quad (26)$$

[‡‡] https://www.nbg.gov.ge/index.php?m=2

[§§] https://www.geostat.ge/ka

$$y_t = \left(M2_{g_t}, REER_t, Inflation_t, Economic\_g_t\right) \quad (27)$$

The paper uses a fixed partition to train the model. 10% of the data was used for testing, and the rest were split into training and test data, 80% and 20%, respectively. The splitting was conducted to ensure that the test set contains a minimum of the whole year. The testing period starts in February 2021 and ends in March 2022. After the model was trained and evaluated during the validation period, where the model had desired performance, the model was retrained using both the training and validation data. Then evaluate the model during the test period to see if the model performs just as well. The time series model must be trained on the latest available data, which is why the validation set is also included in the training set, but we keep the test as an evaluation set. This process is shown in Figure 1.

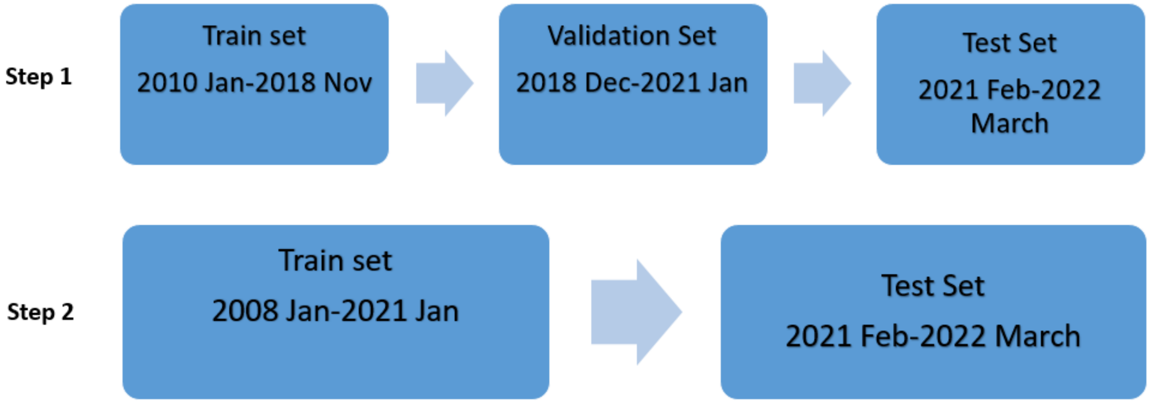


Figure 1 The Training Process of GRU model

In a neural network, it is required to normalize data to avoid the influence of some irrelevant variables on the prediction caused by different scales. Normalization also helps to prevent the gradient explosion. The study uses a min-max normalization: subtracting the minimum value from each observation and dividing by the difference between the maximum and minimum values. L2 regularization is used with parameter 0.01 to avoid the overfitting of models. The model uses Mean Square Error (MSE) as a loss function. Mean absolute errors (MAE) are applied to evaluate the model's performance. The model uses the "Adam" optimization algorithm. More information on hyperparameters can be found in and the general structure of the GRU model is presented in Appendix A.

The model result evaluated on test data can be found in Table 1.

Variable	MAE
Inflation	1.51
REER	3.99
M2 Growth	1.09
Economic Growth	6.96

Table 1 MAE for GRU model

Interpret GRU model

Recurrent Neural Networks are part of the "Black Box" models, which means they are quite complex and not easily interpretable. For our goals, the economic sense is essential. SHAP (Shapley Additive Explanations) (2017) values are used to check the economic soundness of the GRU model. It is one of the leading machine learning explainability technics. SHAP value shows the impact of the feature on output relative to an average value. The important variables for each model output variable are calculated based on the mean absolute value of SHAP. The idea is that the feature with large mean absolute Shap values is important. The feature importance of the GRU model can be seen in Appendix D. The variable importance tells us nothing about how the explanatory variable affects the forecast. Instead, a SHAP dependence plot is used to understand the effect of features on the forecast. For each observation, the feature value is plotted on the x-axis, and the corresponding Shapley value is on the y-axis. When the value in the scatter plot is positive (negative), it means that at this point, feature value has a positive (negative) impact on prediction. It should be noted that the analysis of the SHAP value is done on the entire historical data set. From Figure 2, it can be seen that the lagged value of the policy rate decreases inflation when it is more than 6.5 percentage point, and below this value impact on inflation prediction is the upward direction. Therefore, it can be said that 6.5 percentage point is a neutral policy rate for inflation and other variables (See. Figure 41, Figure 46,Figure 50). With SHAP dependence plot, we can find the steady state of other variables. The steady state of inflation is about 4% (See. Figure 40, Figure 44,Figure 48). This helps us to calculate the steady state of real rate, which is about 2.5%. From Figure 3, it seems that based on the GRU model, real appreciation of the Georgian Lari leads to a decrease in the inflation forecast. In contrast, real depreciation increases the inflation forecast. Likewise, when M2 growth (Figure 4) is above about 15%, it increases the inflation, and below this point decreases. The M2 Growth rate is the least important variable for inflation forecasting (Figure 37). Economic Growth (Figure 5) increases the inflation forecast when there is high and positive economic Growth. In contrast, negative or small positive economic Growth leads to a reduction in inflation. The SHAP dependence plot for other target variables can be found in Appendix E. We can say that the interpretation of the GRU model results is very close to the economic sense.

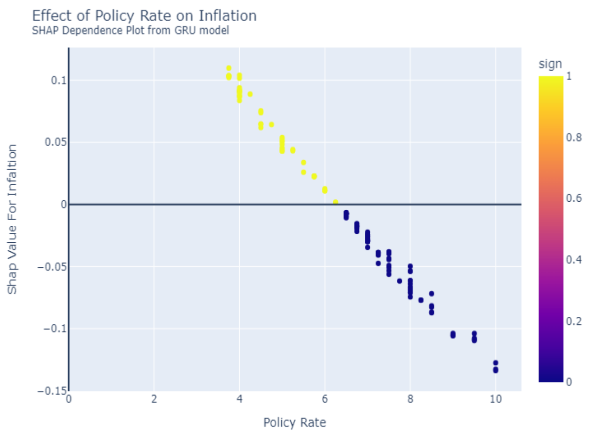


Figure 2 Effect of policy rate on Inflation Forecast

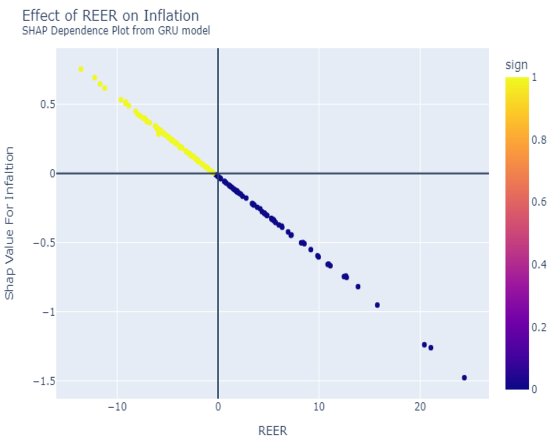


Figure 3 Effect of REER on Inflation Forecast

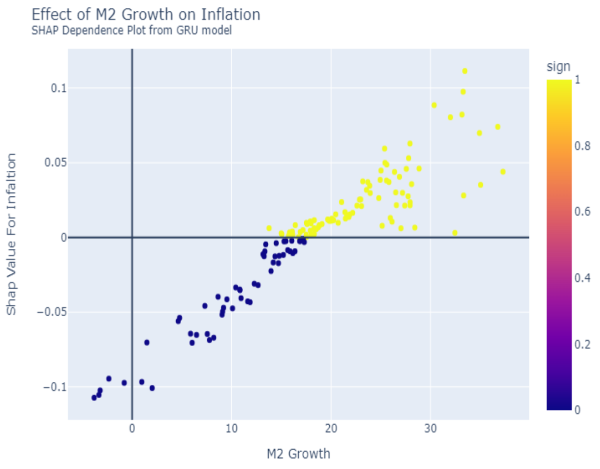


Figure 4 Effect of M2 growth on Inflation Forecast

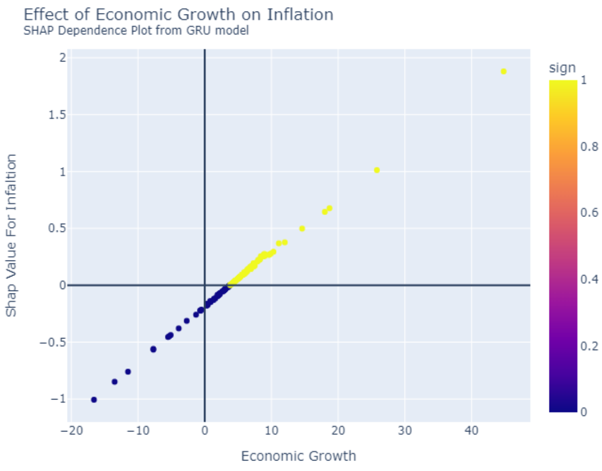


Figure 5 Effect of Economic Growth on Inflation Forecast

The performance of the GRU model affects how the macro agent perceives the environment. Of course, the accuracy of the GRU model is important. However, suppose the model adheres to the general vector of the prediction when forecasting. In that case, it may be sufficient for the macro agent to learn the optimal monetary policy. We can say that the macro agent is looking not at the absolute values of the forecast but the relative ones. In Appendix C, we can see the GRU model at least keeps the forecasting vector, where the real and forecasting value of the model are plotted.

2nd stage: Training Macro Agent

The output of GRU model is part of the macro agents' environment. Agent observes inflation rate, M2 annual growth rate, real effective exchange rate, economic growth rate, and current loss and then makes decisions about policy rate path. The initial state when the agent starts learning is March 2022, which is the latest available data point. From April 2022, the macro agent decides to change the policy rate and, from this date, starts macroeconomic simulation for the next 8192 months. More specifically, each month macro agent changes the policy rate. Modified policy rates affect the environment/macro variables, and all variables interact with each other. The size of the simulation period is 8192 months, which means the GRU network creates 8192 observations for the macro agent to learn optimal policy. Figure 6 shows a short overview of Reinforcement Learning based macro agent.

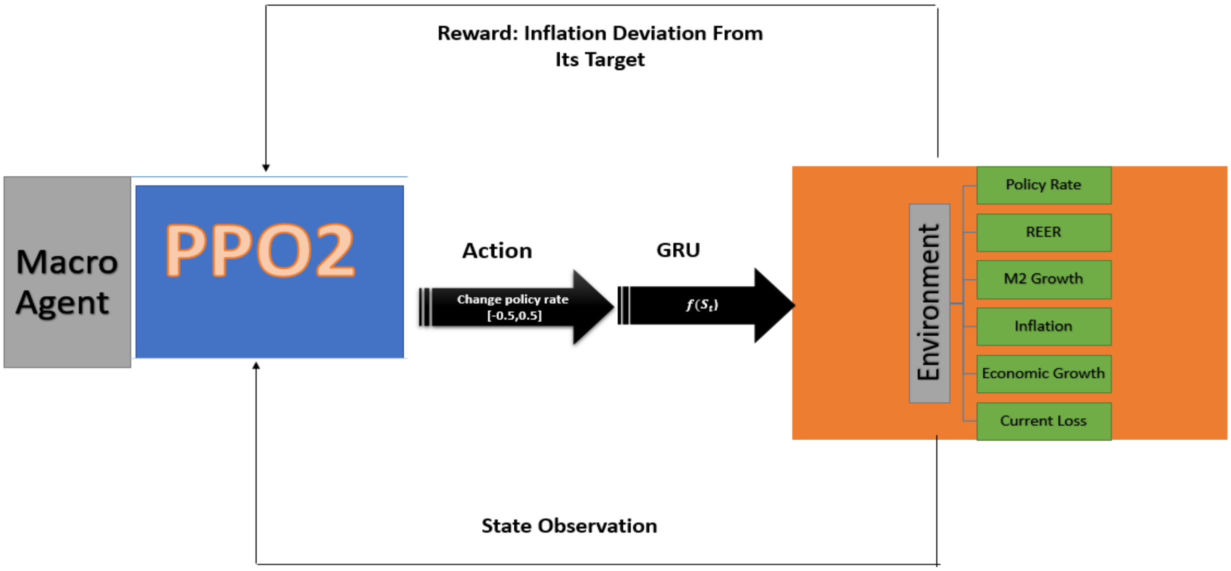


Figure 6 Overview of reinforcement learning based monetary policy decision

To train macro agent, the paper uses modern RL algorithms PPO2. Stable Baselines[***] libraries are used for model implementations. RL agent are trained with 98304 total timestamps. From 2018, the inflation target is set at 3 percent. Thus, the objective of the macro agent is to learn the policy rate so that inflation would maintain close to 3 percent. Figure 7 and Figure 8 show the learning process of Macro agent. For the sake of better visualization, Figure 8 focuses only on approximately last 300 observations from the simulated data (Figure 7). There is a clear negative relationship between inflation and policy rate. The key policy rate tightens when inflation is high, and vice versa. Both figures depict that each time macro agent chooses the monetary policies which minimize the loss function as the curve is monotonically decreasing. In turn, the inflation rate is stable and very close to the target value. Therefore, we can say that the agent has learned the optimal policy to achieve the policy objective. Fluctuations characterize the policy rate but mostly vary between 4 to 6 percent, which are mainly due to stochastic policy during the training process.

[***] https://stable-baselines.readthedocs.io/en/master/

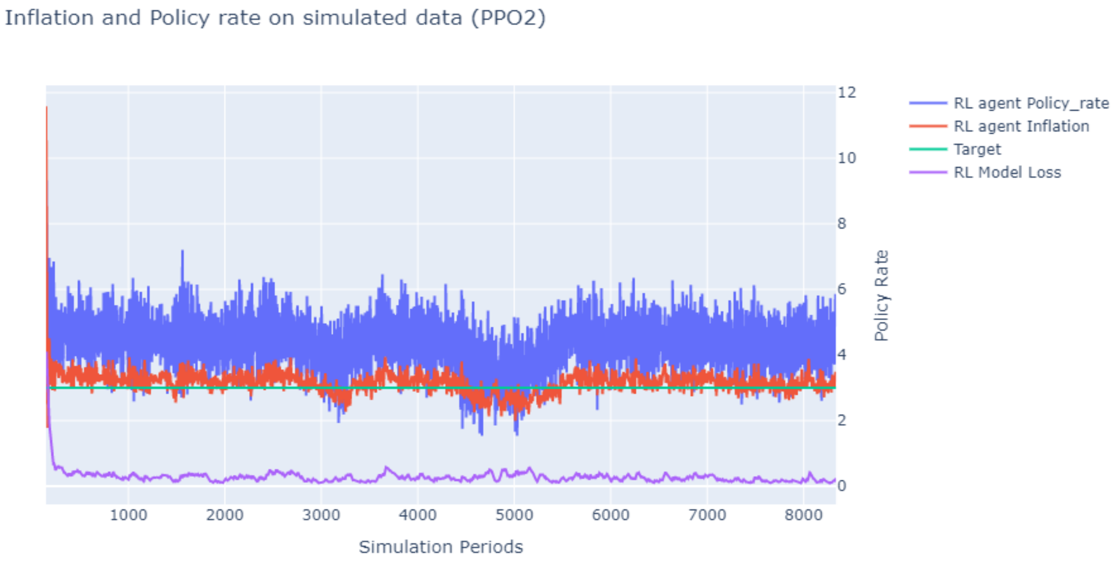


Figure 7 Inflation and Policy Rate during the training

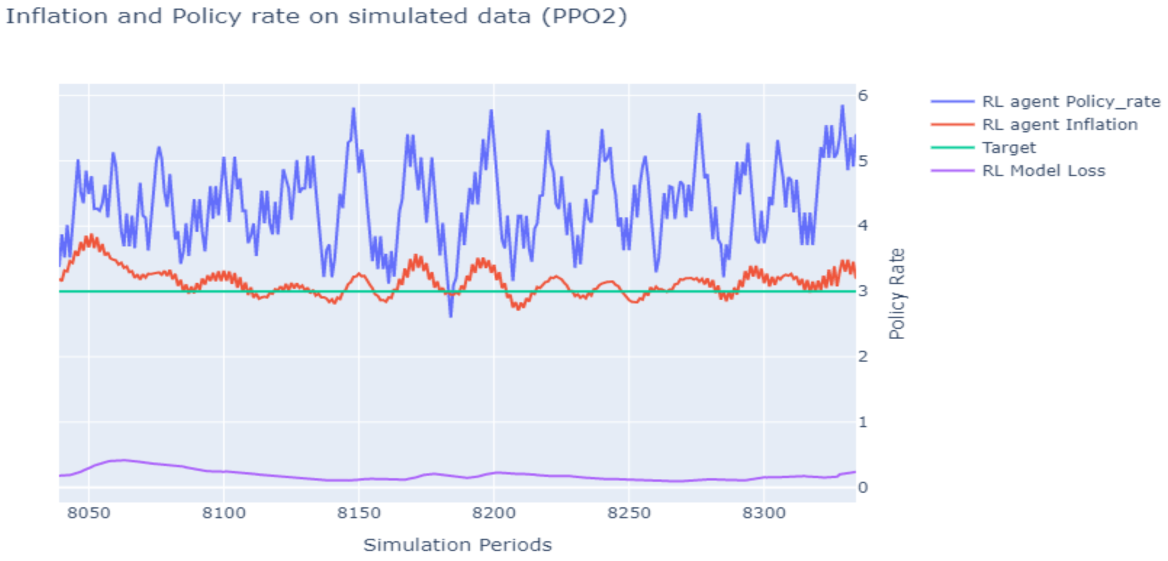


Figure 8 Inflation and Policy Rate during the training last 300 observation

3rd Stage: Making sense of training results

Explainibility Technique of RL agent behavior

The next important step is to explain the behavior of the Reinforcement learning agent. This step is particularly vital as the logic of AI represents a “black box” and we need additional tools to check if it is consistent with basic economic intuition. This step also serves as the mitigant of the overfitting and data mining risks.

We generated 30000 observations on which were trained the GRU model. In addition, during the simulation period, we added random noise to the environment so that the model would not make a decision constantly in a steady state and see other states as well. Random noise is generated from a normal distribution: $N(0{,}0.3 * std(X_{i_t}))$ where$X_{i_t} \in \{M2_{g_t}, REER_t, Inflation_t, Economic_g_t\}$ and std is standard deviation of appropriate variable is calculated based on historical data.

The GRU model is trained on the generated observations, where the target variable is action from the RL model, and feature vectors are policy rate, real effective exchange rate, annual inflation, M2 growth, and economic growth.

A trained GRU model tries to learn how an RL agent makes a decision. If we explain the GRU model, ultimately, it means that we can explain the behavior of the RL agent. To explain the GRU model again, we will use the Shap value. Figure 9 shows variables that are most important when the agent makes a decision. This graph can be interpreted as a ranking of the importance of the variables in the monetary reaction function. Economic growth and REER are the most important variables. The inflation rate and the policy rate are also very important. The macro agent gives the least attention to the M2 growth rate. Figure 10, Figure 11, Figure 12, Figure 13, Figure 14 represents SHAP dependence plots for RL agents’ action for different variables from environment. The results are in sync with the economic intuition and characteristics of the modeled economy. Figure 12 indicates that the agent tightens policy rate when inflation is approximately above 3%.

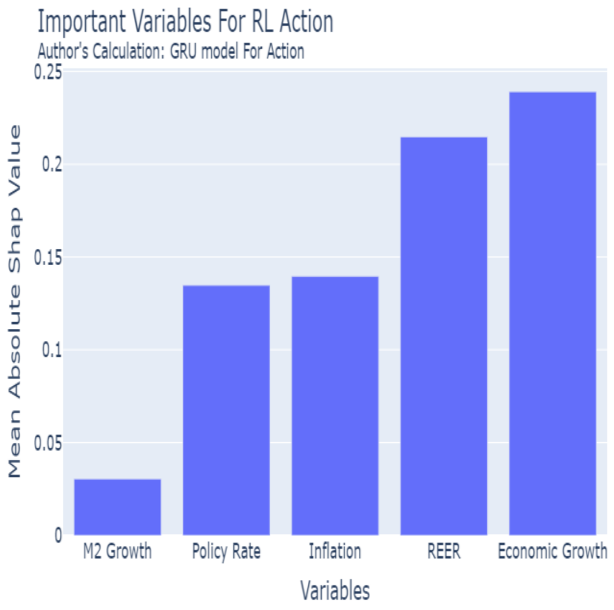


Figure 9 Variable Importance for RL Agent

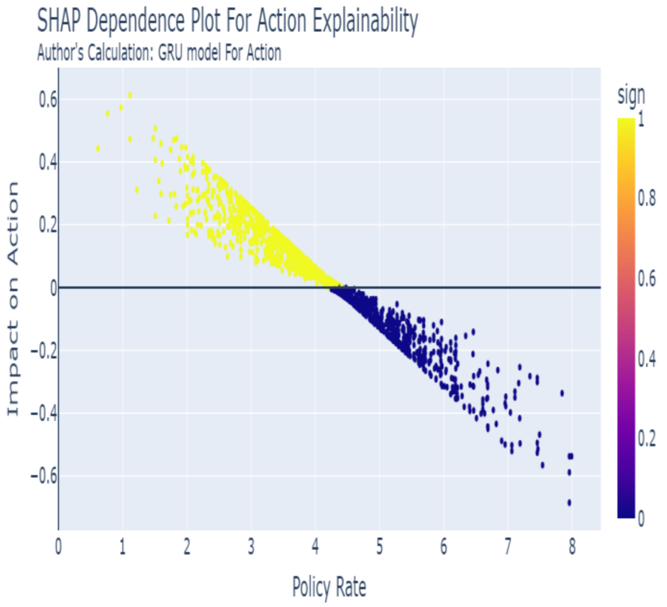


Figure 10 Effect of Policy Rate on RL Agent Action Forecast

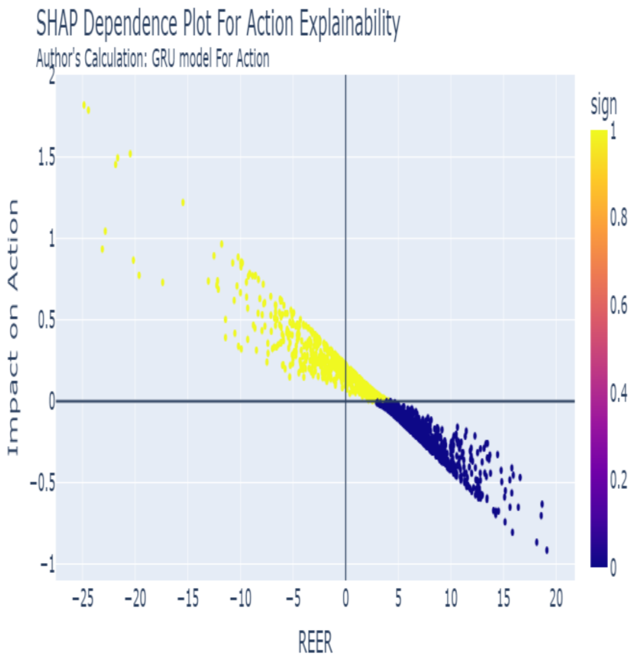


Figure 11 Effect of REER on RL Agent Action Forecast

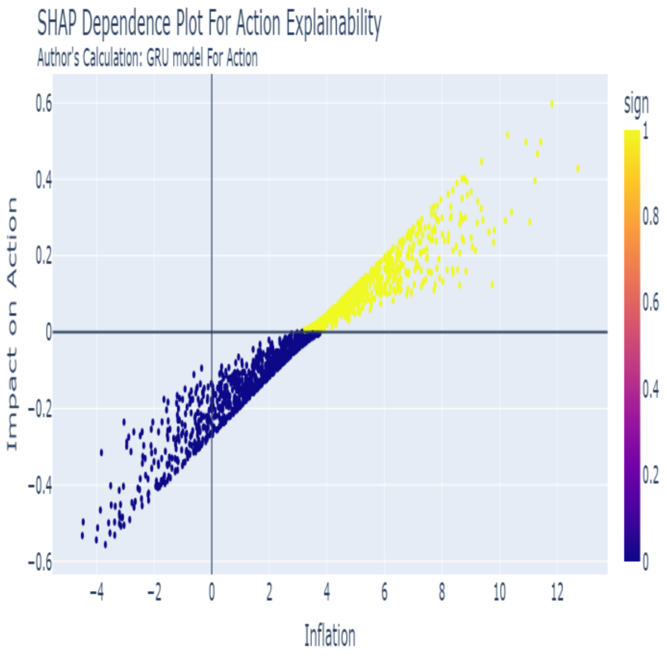


Figure 12 Effect of Inflation on RL Agent Action Forecast

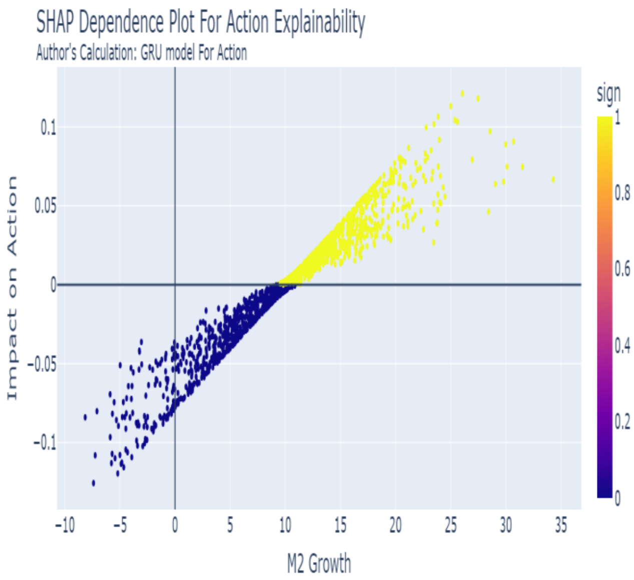


Figure 13 Effect of Policy M2 Growth Agent Action Forecast

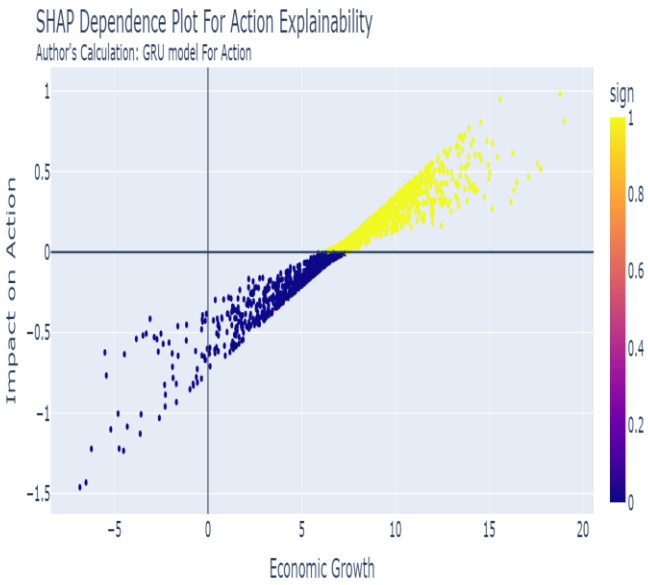


Figure 14 Effect of Economic Growth on RL Agent Action Forecast

Testing the agent against the shocks

The next step for testing the macro agent is to simulate the behavior of the macro agent and macroeconomic variables in response to the structural shocks. This type of analysis will help us better understand the behavior of RL agent and contrast it with economic intuition. Shock is added for each macroeconomic variable separately, which is positive or negative. The size of the shock is two standard deviations of actual data. The shock is persistent, which lasts six months, and the duration of simulation periods is 300 months. Recall that we start the shock simulation from the steady state.

Figure 15, Figure 16, Figure 17, Figure 18 represent deflationary shock results, with about 4% deflation. To respond the shock, the macro agent lowers the policy rate below the projected steady-state to stimulate demand, make inflation higher, and approaches its target value (see Figure 15). We can see that the macro agent managed to cope with the shock and brought inflation back to the target rate. In Figure 16 we see that a reduced policy rate inevitably increases inflation. From this figure, it can be noticed that the real effective exchange rate contributes the largest share to the increase in inflation. In turn, a declined policy rate reduces the real effective exchange rate (see. Figure 17). Thus, the policy rate affects inflation through REER. This result is consistent with Figure 9, where the real effective exchange rate is one of the main variables for the macro agent in the decision-making process. REER carries the strongest transmission mechanism of monetary policy. When the macro agent decreases the policy rate, demand for local currency assets decreases, the exchange rate depreciates in the first few months, which leads to increase in inflation. It should also be noted that economic growth is an important variable in inflation growth and together with the exchange rate, it plays a significant role bringing inflation back to the target (Figure 16). From Figure 18, we can see that the main reason for the decrease in policy rate is deflation. This confirms that agent made a right decision. The GRU model from the previous paragraph is used to obtain the above-mentioned decomposition, which tries to explain the action of the macro agent.

The same conclusion in the opposite direction can be drawn when there is a sharp increase in inflation that is not a result of pressures in the economy. It is called cost-push shock. When there is a cost-push shock, the macro agent tightens the policy rate. Tighter monetary policy curbs demand, thus inflation decreases. After certain periods, inflation is near to its target rates (See. Figure 19). The policy rate reduces inflation through exchange rate and economic growth effects (See. Figure 20 and Figure 21). The macro agents’ response to inflationary shocks makes economic sense. Most importantly, the agent can return inflation to the target rate after the shock. In Appendix F, there can be seen RL agent response to other structural shocks.

Deflationary Shock

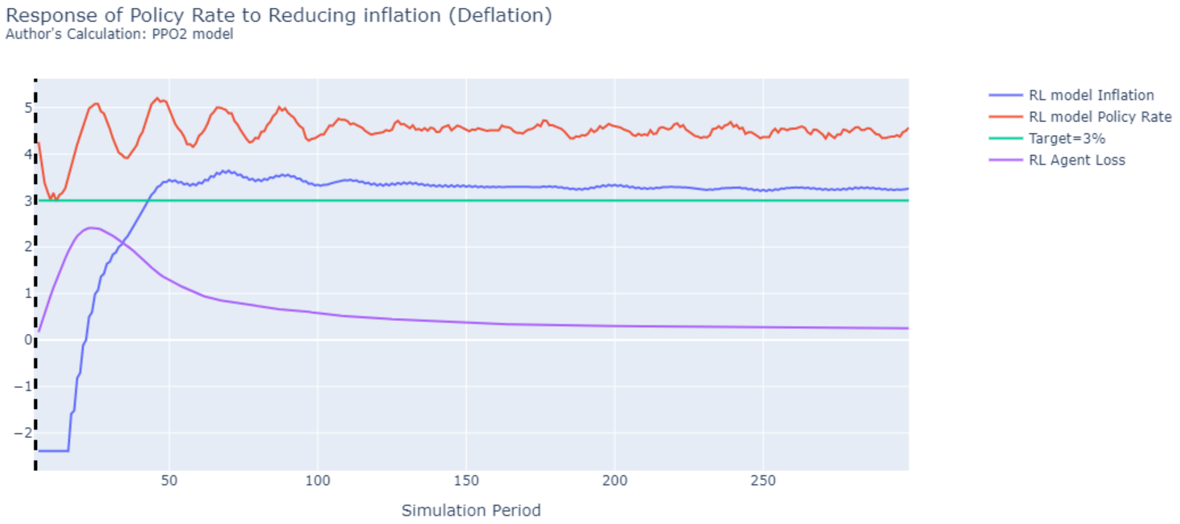


Figure 15 Response Of Policy Rate to reduction inflation (Deflation)

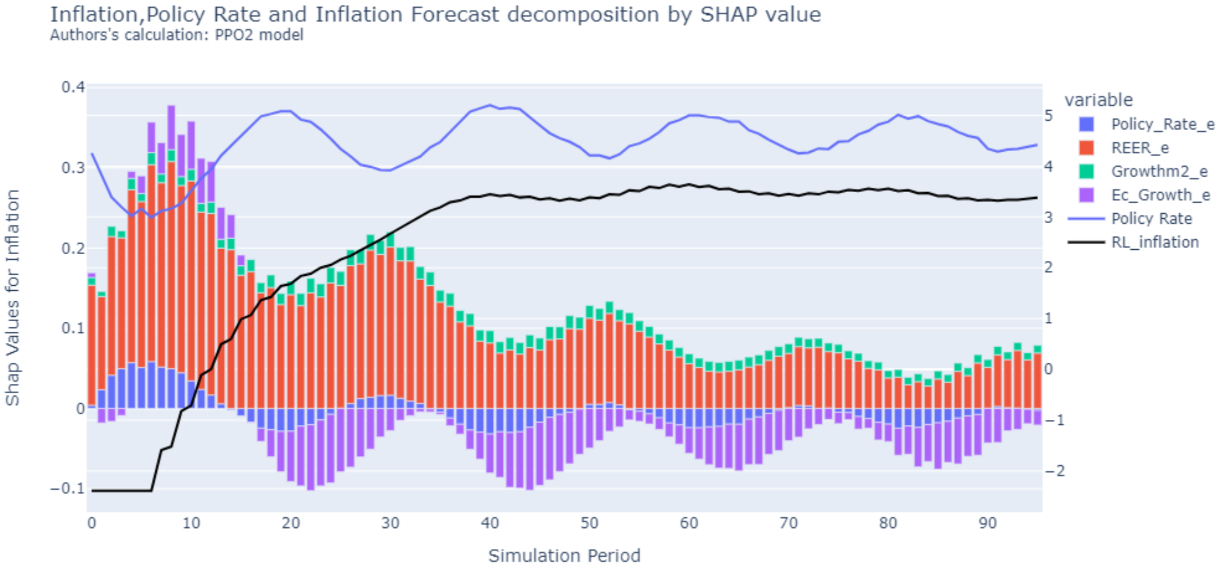


Figure 16 Inflation Forecast Decomposition by Shap value

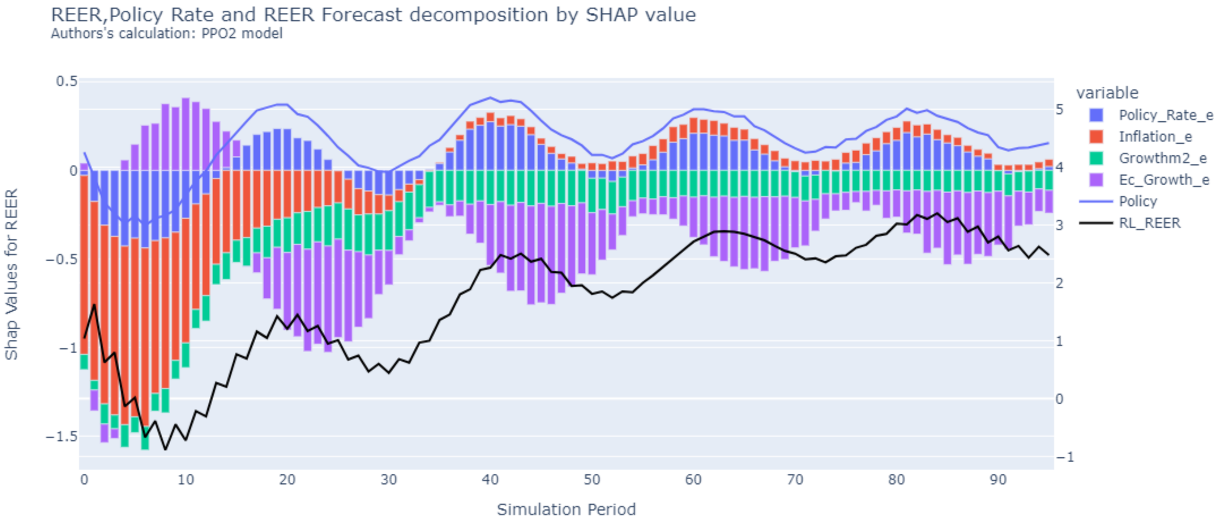


Figure 17 REER Forecast Decomposition by Shap value

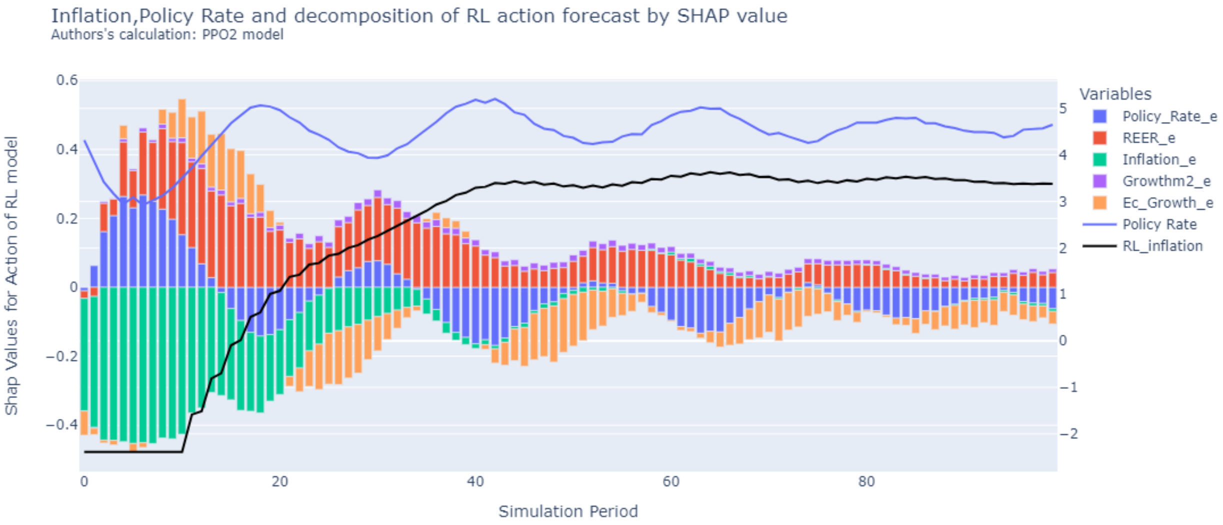


Figure 18 Action Of RL model Decomposition by Shap value

Inflationary Cost-Push Shock

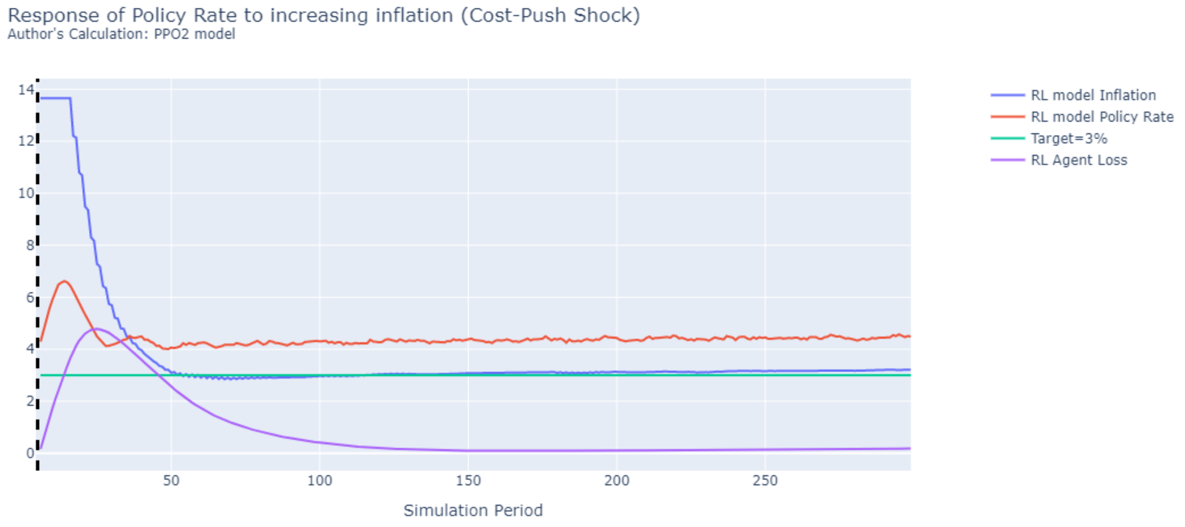


Figure 19 Response Of Policy Rate to Inflation Increase

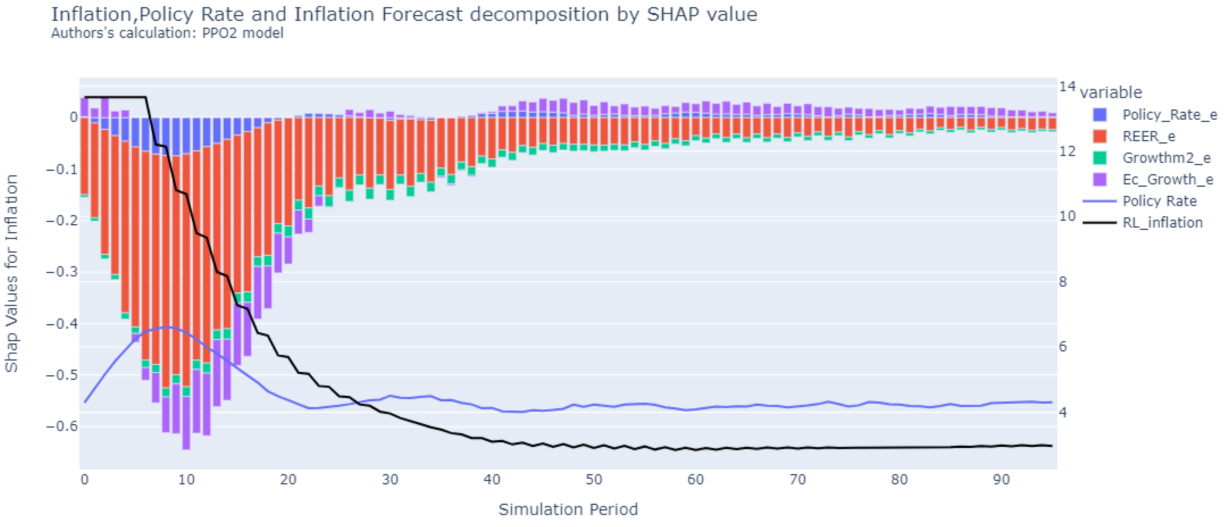


Figure 20 Inflation Forecast Decomposition by Shap value

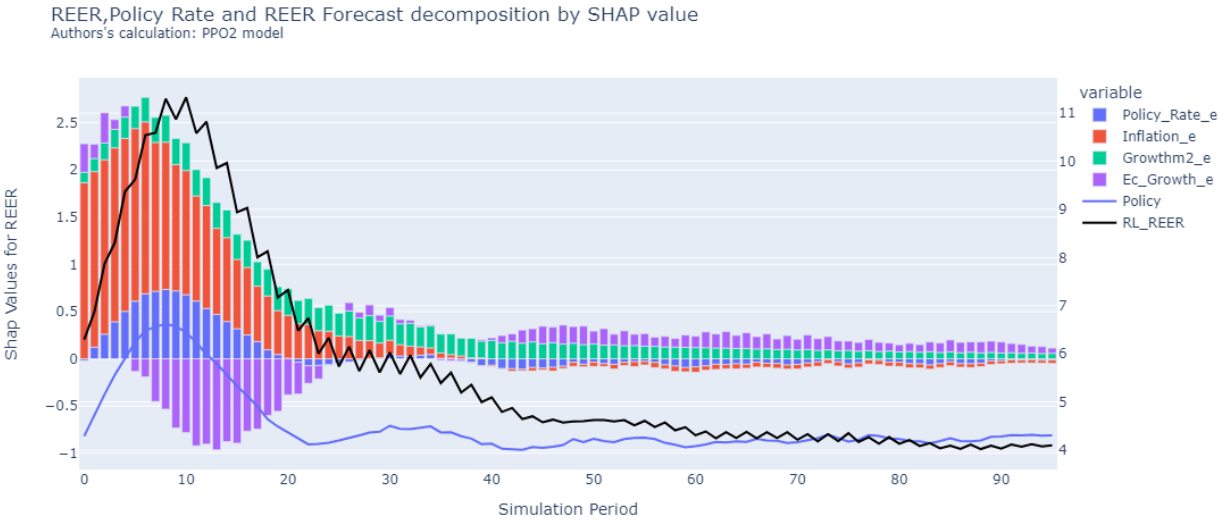


Figure 21 REER Forecast Decomposition by Shap value

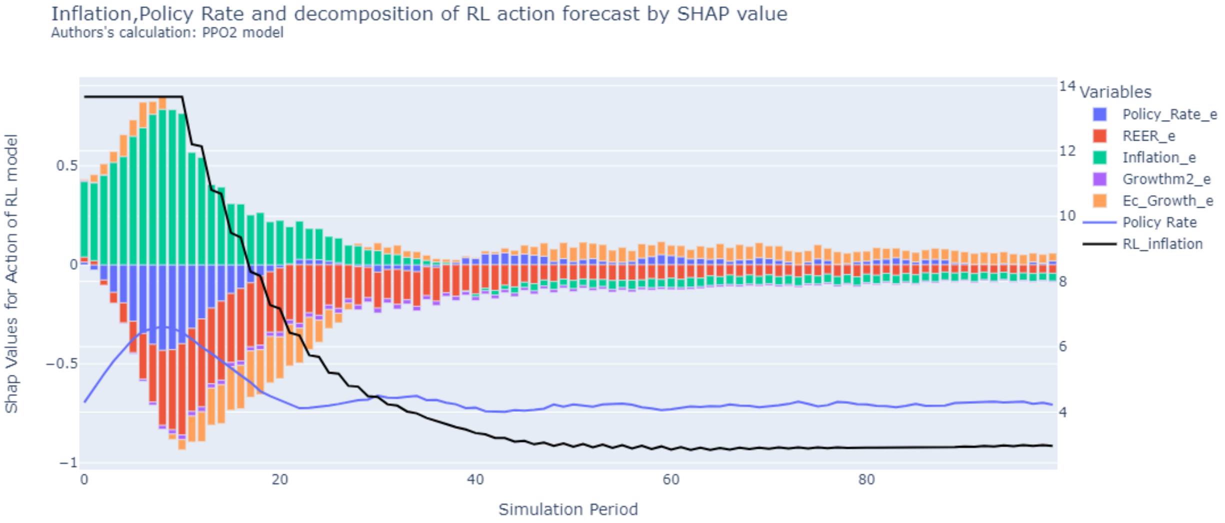


Figure 22 Action Of RL model Decomposition by Shap value

Nonlinearity of RL Agent Policy Reaction Function

We tested the agent against various shocks and environments. As a result, agents' behavior is compatible with economic understanding. The next step is to depict the nonlinearity of the macro agent. To achieve this goal the paper simulates a different environment. Agent samples action from the same environment, only inflation varies from -1 % to 6%. This helps us understand how the agent reacts to different level of inflation while other variables are in a steady state. It gives us a monetary policy reaction function against inflation. Considering additional variable with three different states will allow us to add another dimension. We plot three different monetary policy reaction functions: one for steady-state value of the additional variable, above and below steady state. The value above/below the steady state is obtained by adding/subtracting a quarter standard deviation of the variable. For better visualization, the paper uses the Locally Weighted Scatterplot Smoothing (Lowess) technique (Royston, 1992) to smooth relationship between action and inflation rate. . Figure 23 shows the relationship between macro agent and inflation rate with three different values of the policy rate, while other variables are in a steady state. The black horizontal lines show the decision bounds of the macro agent, which ranges from -0.5 and 0.5. This graph shows that when the inflation rate is low, the slope of the curve is small, while the slope increases when there is high inflation. Moreover, when the policy rate is greater than the steady state, the slope of the monetary reaction function at every point is larger than the other two reaction functions. Macro agent reacts more to high inflation and fears an inflationary shock compared to the low inflation situation, or the environment in the economy favors low inflation. The same conclusion can be made if we consider other variables (See. Figure 24,Figure 25,Figure 26). It is especially noticeable if we look at the graph concerning economic growth. When the economic growth is above its steady state, the monetary policy reaction function is less flattened relative to when economic growth is in a steady state or below. The curve is almost horizontal when inflation is less than its target value, and economic growth is less than its steady-state value. Meaning the action is almost constant and about -0.5. These four graphs shows that action behavior is very nonlinear, distinguishing the RL monetary policy reaction function from classical reaction functions.

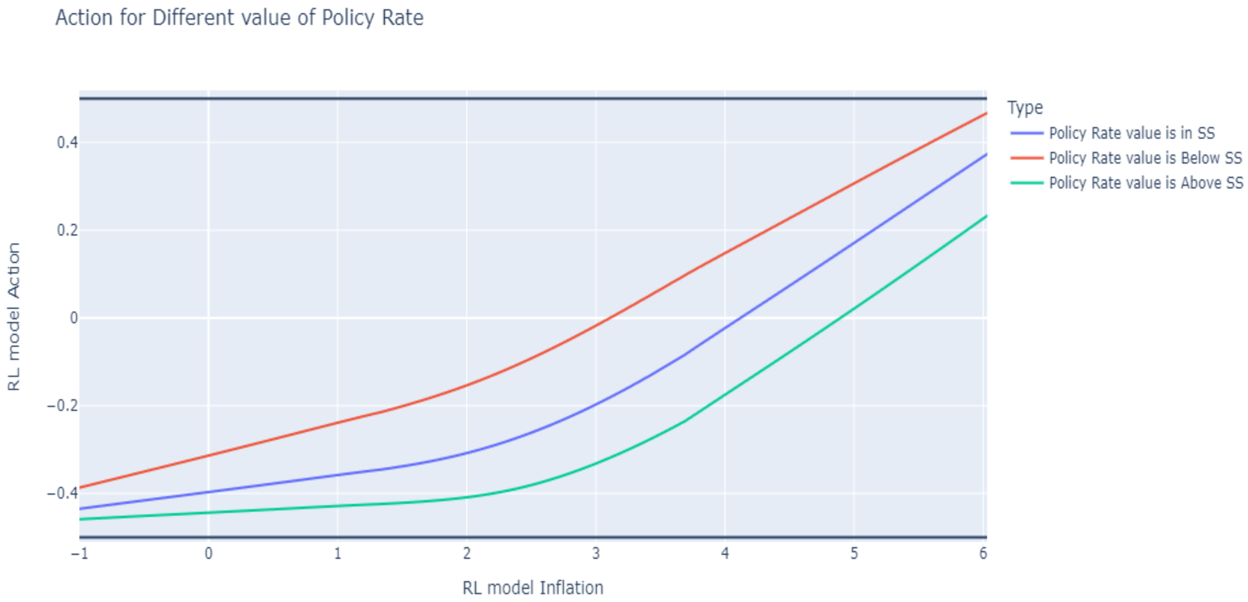


Figure 23 Monetary policy reaction with respect to inflation for three different value of Policy Rate: In Steady State, below steady state and above steady state

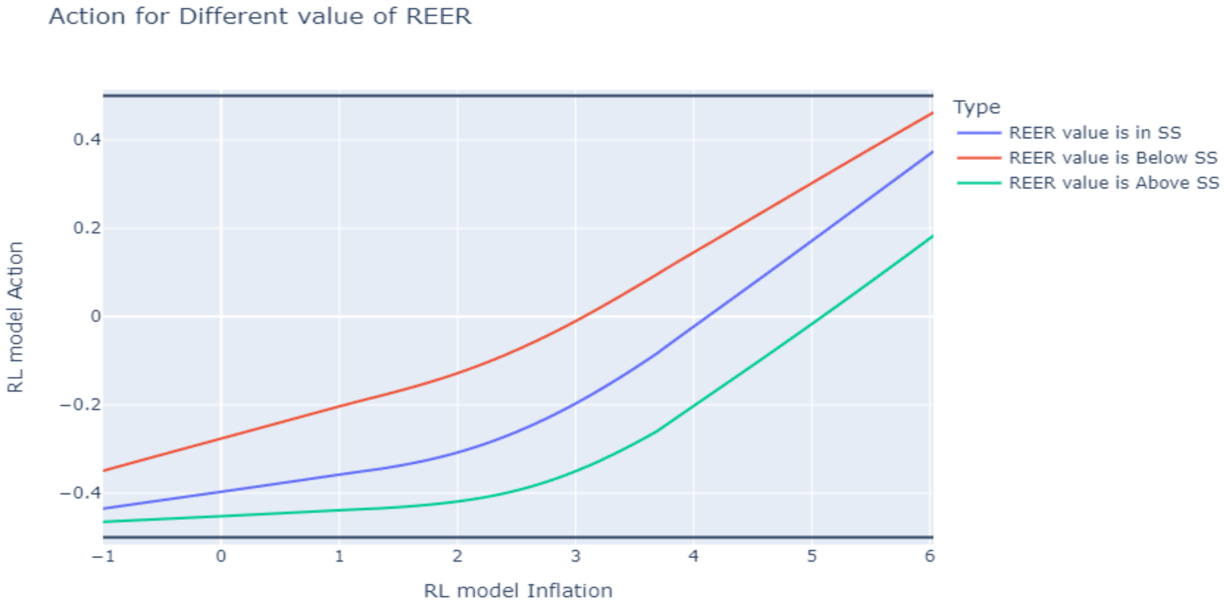


Figure 24 Monetary policy reaction with respect to inflation for three different value of REER: In Steady State, below steady state and above steady state

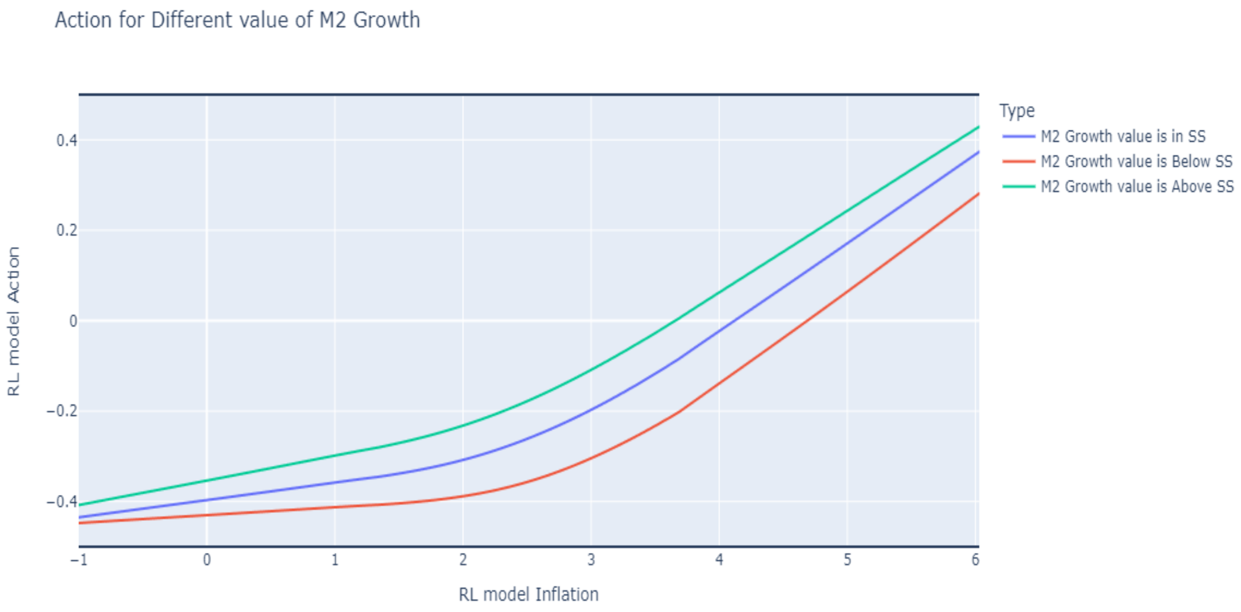


Figure 25 Monetary policy reaction with respect to inflation for three different value of M2 Growth: In Steady State, below steady state and above steady state

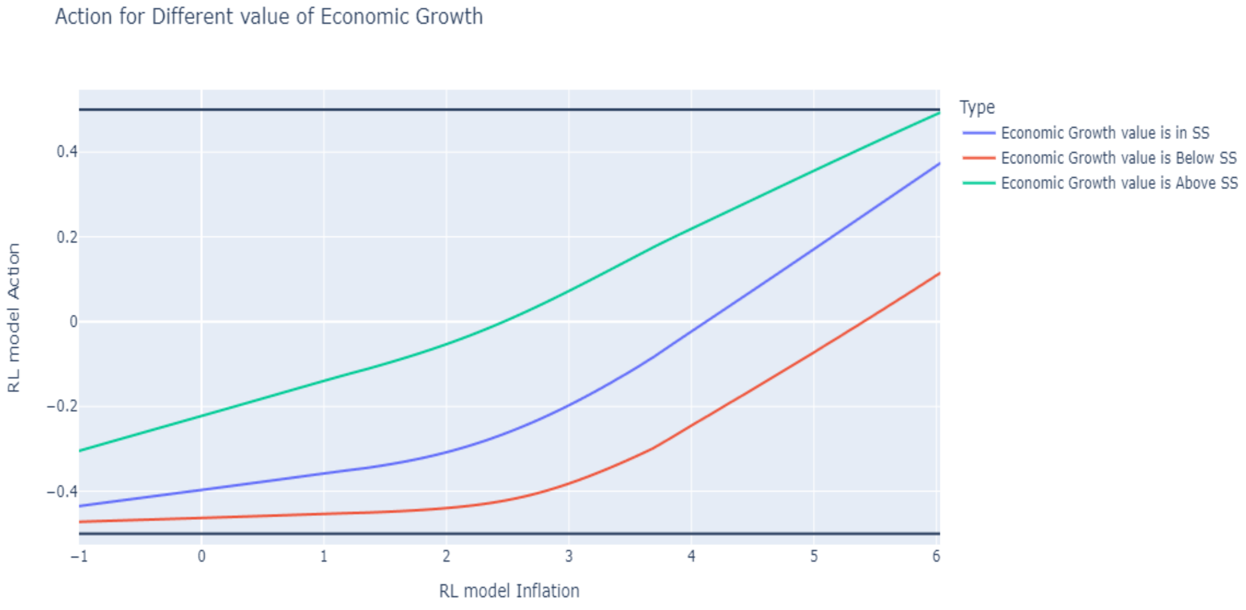


Figure 26 Monetary policy reaction with respect to inflation for three different value of Economic Growth: In Steady State, below steady state and above steady state

Counterfactual analysis[†††]

After testing the agents' behavior when the variables were in a steady state at the initial state, and the agents' reaction to structure shocks is adequate, the counterfactual analysis will be interesting. We tested the agent on actual data from 2010 July to 2022 March. We first tried to test the agent's one-step decision to conduct a counterfactual analysis. We compare at a specific time what decision the agent would make and what decision the central bank made. The agent makes a decision after looking at the actual data defined by S_t, but only for one period.

$$action_t = \pi(S_t) \ (28)$$

$$S_t = \left(policy_t, M2_{g_t}, REER_t, Inflation_t, Economic\_g_t\right) \ (29)$$

The next inflation rate that the agent would follow from this decision is given in Figure 27. The same picture also shows what the reaction of the central bank was and what would be the reaction of the agent. The agent was more rigid and made aggressive decisions. The same can be said from Figure 28, which shows the actual policy rate and the policy rate found due to the agent's decision. The vertical difference on the graph is a measure of the aggressiveness of the agent. The larger the vertical difference between the actual and the agent's policy rate, the more aggressive the agent is in most cases. Interestingly, in the last observation period, the agent says that the policy rate should be reduced, while inflation is high (Figure 27). The current high level of inflation, along with economic growth, tends to raise the policy rate, but a strongly anchored exchange rate and the policy rate itself tend to lower the policy rate (Figure 30).

In each period standard deviation of the inflation rate is calculated in order to evaluate RL agent performance, using the data available up to that point. If the Macro Agents' curve lies below actual inflation deviation curve, it means RL agents' inflation is stable compared to actual inflation. Figure 29 shows that RL macro agent outperforms actual policy rate decisions and achieves more stable inflation. It is noted that relatively stable inflation is partially due to a GRU model that helps us discover the next state. On the other side, the macro agent makes a decision only during one period.

[†††] Disclaimer: Comparisons with real data are of course not exact and are based on assumptions. During the actual decision making process, there could still be other variables and environment that our model cannot catch. An example of this is Covid-19, and there may be many more. Because of the above, this comparison may not be correct, but it is good for analyzing model results and for analytical visibility.

Macro agents’ goal is to maximize long-term rewards, not just one period . The decision made for one period may not be optimal in this particular period. Thus, one period action underestimates macro agents’ real performance.

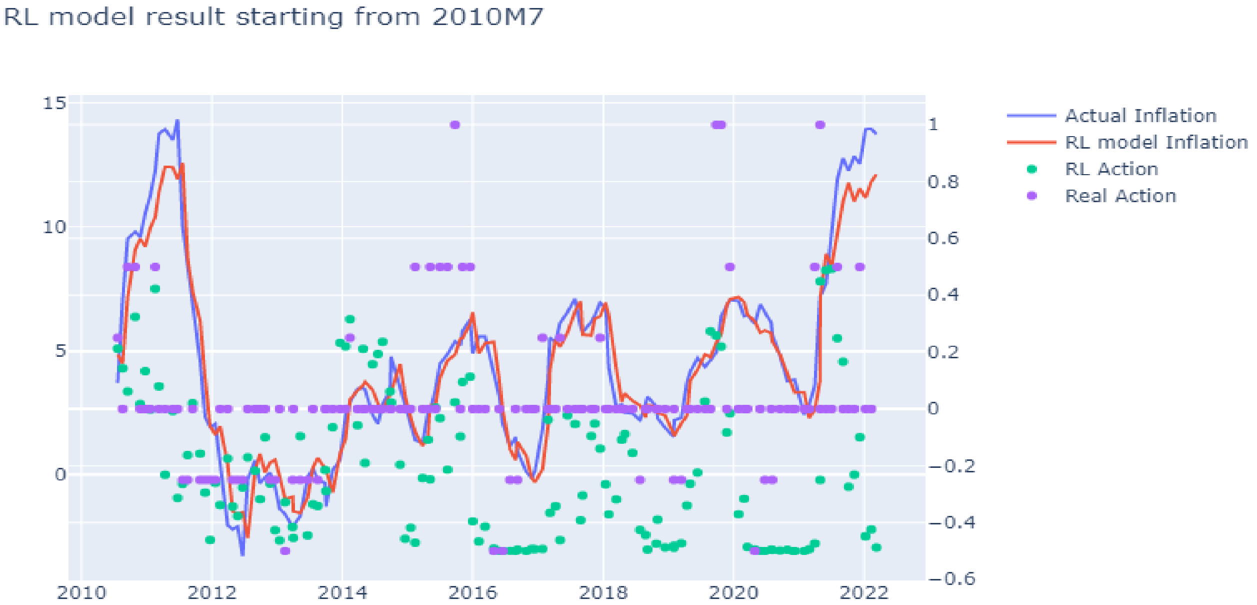


Figure 27 RL models and actual Inflation from 2010 to 2022

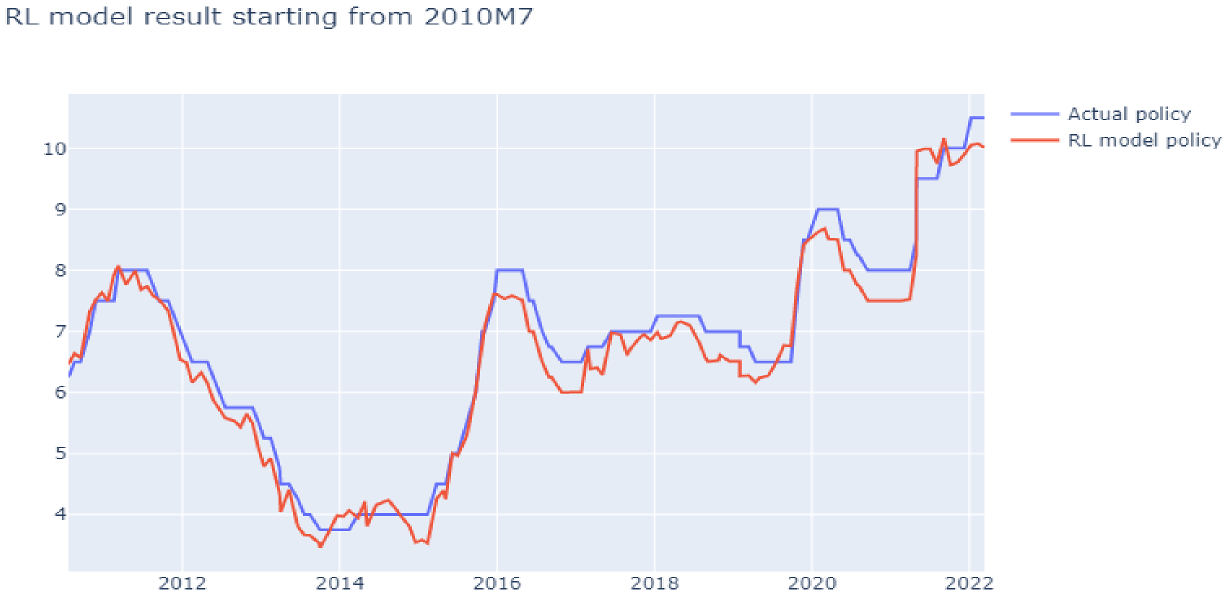


Figure 28 RL models and actual d policy rate from 2010 to 2022

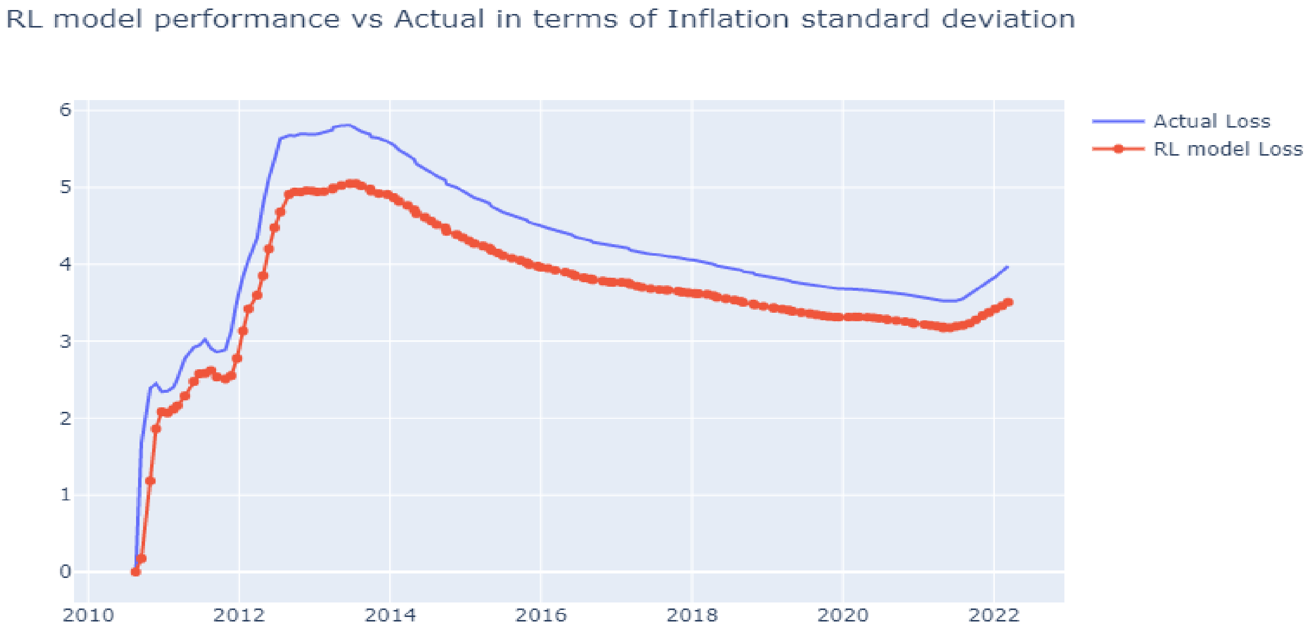


Figure 29 Testing RL agent Against Actual Decision

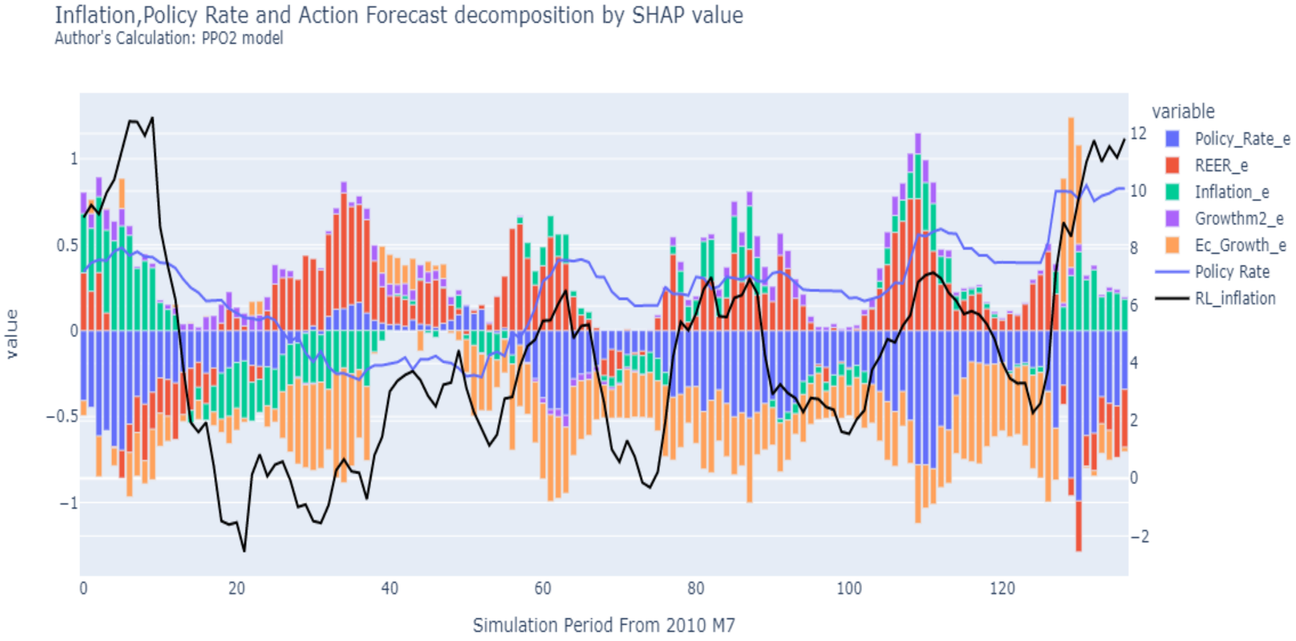


Figure 30 Action of RL model Decomposition by Shap value

To allow the agent to make multi-level decision-making without losing linkage with historical data, we also used the second, alternative approach to conduct counterfactual analysis. Results and method description can be found in appendix G.

Conclusion

This paper showed how to train a macro agent who conducts optimal monetary policy using an model-free Reinforcement learning algorithm. By interacting with estimated environment, a macro agent can form an model of macroeconomic dynamics. Testing the RL agents' behavior indicates that it can improve actual policy rate in terms of inflation stabilization. Moreover, the agent can adequately respond to economic shocks to bring inflation back to the target rate. The main finding of the paper is that the behavior of the RL agent is nonlinear and depends on complex factors. To address the inherent problem of transparency of AI models, proposed tools can make behavior of the AI based policy rules transparent enough to analyze it and contrast it with economic intuition.

The results indicate use the artificial intelligence tools to enhance monetary policy conduct. This can used as an additional toolkit to analyses complex nonlinear equations of a Dynamic general equilibrium model. In contrast, RL macro agent can learn nonlinear decision structures and handles the problem of high dimensions, which is another problem of nonlinear DGE models.

Future research should scale these results to automate monetary policy with AI macro agent. It can be done by estimating more precisely macroeconomic dynamics with the help of other models. Expanding the number of variables represented in the macroeconomic environment to make the macro agent decision based on many variables. For the agent to see many environments during the learning process, it is possible to add shocks to the environment randomly. The future extensions could consider training a similar agent using data from another country. Traning on larger, multicounty datasets could demonstrate the robustness of our RL approach in monetary policy.

The results so far indicate that at its minimum AI can be used by policy makers to aid their decision making and achieve more efficient results. In the future the approach could result in semi-autonomous policy conduct. This can go hand in hand with DLT based DeFi, CBDC projects and develop in MonetaryTech approach towards monetary policy. Many Central Banks who are actively pursuing design of CBDC projects could benefit from the robustness that MonetaryTech approach could bring by ensuring at least partial automation of the monetary policy. One practical implications of the research for the CBDC design are that: (i) it should be compatible with smart contacts to ensure at least semi-autonomous conduct of monetary policy; (ii) it should be oracle compatible and should have robust oracle governance. The later feature will allow AI to use wide range of economic feeds to achive some degree of autonomous monetary policy conduct.

References

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks, 5*(2), 157 - 166.

Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., & Guo, D. (2017). Real-Time Bidding by Reinforcement Learning in Display Advertising. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (pp. 661–670). ACM.

Chen, M., Joseph, A., Kumhof, M., Pan, X., Shi, R., & Zhou, X. (2021). *Deep Reinforcement Learning in a Monetary Model*. Retrieved from arxiv: https://arxiv.org/abs/2104.09368

Cho, K., Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). *On the Properties of Neural Machine Translation: Encoder–Decoder.* Association for Computational Linguistics.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. Retrieved from arxiv: https://arxiv.org/abs/1412.3555

Cleveland, F. R. (n.d.). *www.clevelandfed.org*. Retrieved from https://www.clevelandfed.org/our-reasearch/indicators-and-data/simple-monetary-policy-rules/: https://www.clevelandfed.org/our-reasearch/indicators-and-data/simple-monetary-policy-rules/

Curry, M., Trott, A., Phade, S., & Yu Bai, S. Z. (2022). *Finding General Equilibria In Many-Agent Economic Simulations Using Deep Reinforcement Learning*. Retrieved from openreview.net: https://openreview.net/forum?id=d5IQ3k7ed__

Edward S. Knotek II, R. J. (2016). *Federal Funds Rates Based on Seven Simple Monetary Policy Rules*. Retrieved from Federal Reserve Bank of Cleveland: https://www.clevelandfed.org/publications/economic-commentary/2016/ec-201607-federal-funds-rates-from-simple-policy-rules

Friedman, M. (1960). A Program for Monetary Stability. *New York: Fordham University Press*.

Friedman, M. (1968). The Role of Monetary Policy. *American Economic Review 58(1)*, 1-17.

Friedman, M., & Schwartz, A. J. (1963). A Monetary History of the United States. *Princeton University Press*, 1867-1960.

Giannoni , M., & Woodford, M. (2005). Optimal Inflation-Targeting Rules. In B. M. Bernanke. Chicago,IL: University of Chicago Press.

Hill, E., Bardoscia, M., & Turrell, A. (2021). *Solving Heterogeneous General Equilibrium Economic Models with Deep Reinforcement Learning*. Retrieved from arxiv.org: https://arxiv.org/abs/2103.16977

Hinterlang, N., & Tänzer, A. (2021). Optimal monetary policy using reinforcement learning. *Deutsche Bundesbank*.

Hochreiter, S. (1997). Long Short-term Memory. *Neural Computation 9(8)*.

Hu, J., & Wellman, M. (2003). Nash Q-Learning for General-Sum Stochastic Games. *Journal of Machine Learning Research*, 1039-1069.

Kober, J., Bagnell, J., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 1238–1274.

Littman, M. (1994). Markov games as a framework for multi-agent reinforcement. *Machine Learning Proceedings*, 157–163.

Lundberg, S., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions.* (I. G. Garnett, Ed.) Curran Associates.

Maskin, E., & Tirole, J. (1998). A theory of dynamic oligopoly, I: Overview and quantity competition with large fixed costs. *Econometrica, 56*, 549-569.

McCallum, B. T. (1989). *Monetary economics : theory and policy.* New York: Macmillan.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., . . . Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. *CoRR, abs/1602.01783*.

Monetary Policy Operations Manual. (2021, September 24). Retrieved from Nbg.gov.ge: nbg.gov.ge/fm/მონეტარული_პოლიტიკა/დოკუმენტები/mp-manual-final-eng221021.pdf?v=g1b27

Oriol Vinyals, I. B. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature volume 575*, 350-354.

Royston, P. (1992). Lowess Smoothing. *University College London*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2017). *Trust Region Policy Optimization*. Retrieved from arxiv.org: https://arxiv.org/abs/1502.05477

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. Retrieved from arxiv.org: https://arxiv.org/abs/1707.06347

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., . . . Hassabis, D. (2017). Mastering the Game of Go without Human Knowledge. *Nature 550*, 354–359.

Sinitskaya, E., & Tesfatsion, L. (2015). Macroeconomies as Constructively Rational Games. *J. of Economic Dynamics and Control 61*, 152-182.

Svensson, L. E. (2010). Inflation Targeting. In B. W. Friedman, *Handbook of monetary economics* (Vol. 3B). Amsterdam: Elsevier.

Taylor, J. B. (1993). Discretion versus Policy Rules in Practice. *Conference Series on Public Policy 39(1)*, (pp. 195–214).

Taylor, J. B., & Williams, J. C. (2011). Simple and Robust Rules for Monetary Policy. *Handbook of Monetary Economics, 3B*, 829-858.

Williams , J. (1999). *Simple Rules for Monetary Policy.* FEDS Working Paper 1999-12.

Woodford, M. (2010). Optimal monetary stabilization policy. In B. W. Friedman, *Handbook of monetary economics* (Vol. 3B). Amsterdam: Elsevier.

Zheng, S., Trott, A., Srinivasa, S., Naik, N., Gruesbeck, M., Parkes, D. C., & Socher, R. (2020). *The AI Economist:Improving Equality and Productivity with AI-Driven Tax Policies*. Retrieved from arxiv.org: https://arxiv.org/abs/2004.13332

Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C., & Socher, R. (2021). *The AI Economist: Optimal Economic Policy Design via Two-level Deep Reinforcement Learning*. Retrieved from arxiv.org: https://arxiv.org/abs/2108.02755

Appendix

Appendix A: GRU model Structure

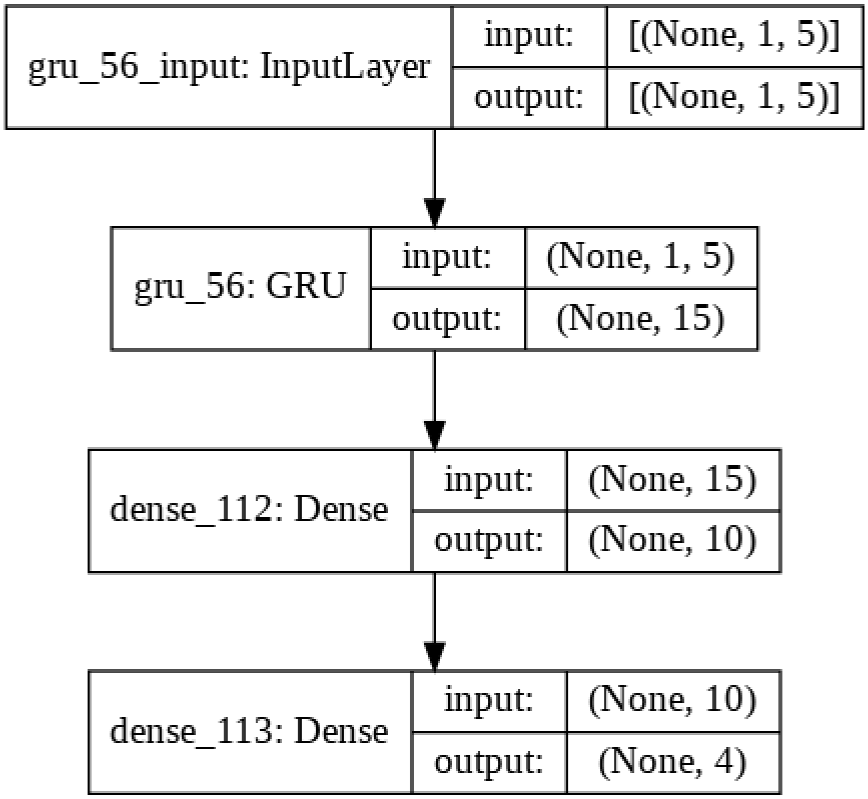


Figure 31 GRU model Structure

Appendix B: Hyperparameters of GRU model

Layer (GRU)	
activation	‘tanh’
batch_input_shape	(None, 1, 5)
bias_constraint	None
maxval	0.05
minval	-0.05
kernel_regularizer l2	0.009999999776482582
recurrent_activation	'tanh'
units	15
use_bias	True

recurrent_initializer	'Orthogonal'
return_sequences	False
Layer (Dense)	
activation	'tanh'
kernel_initializer	'RandomUniform'
units	10
use_bias	True
Layer (Dense)	
activation	'linear'
units	4
use_bias	True
Compile Information	
loss	mean_squared_error
metrics	mean_absolute_error
optimizer	‘Adam’
learning_rate	0.001
decay	1e-6
beta_1	0.9
beta_2	0.999
amsgrad	False
Epochs	500

Table 2 Hyperparameters of GRU model

Appendix C: GRU model performance

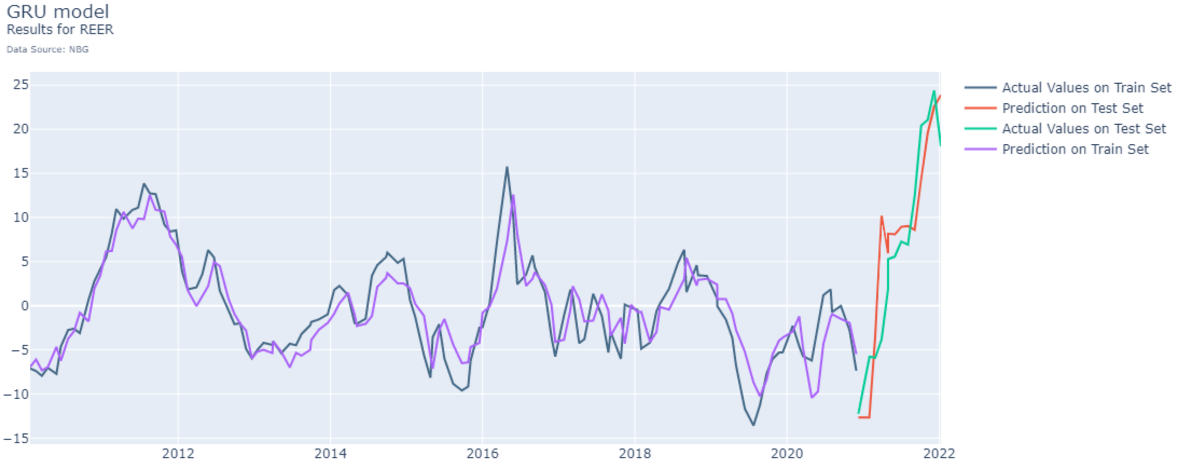


Figure 32 Real vs Predicted REER on training and testing set

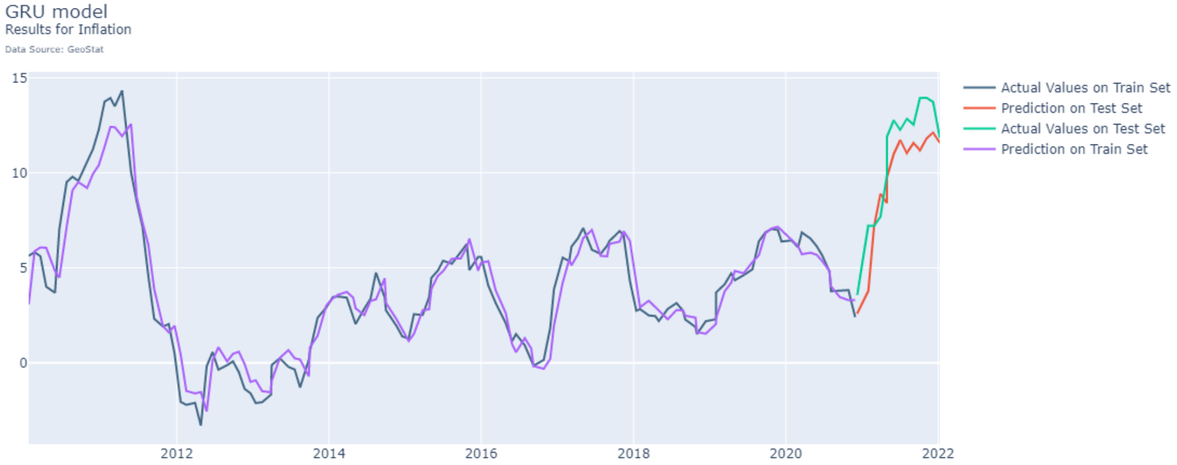


Figure 33 Real vs Predicted Inflation on training and testing set

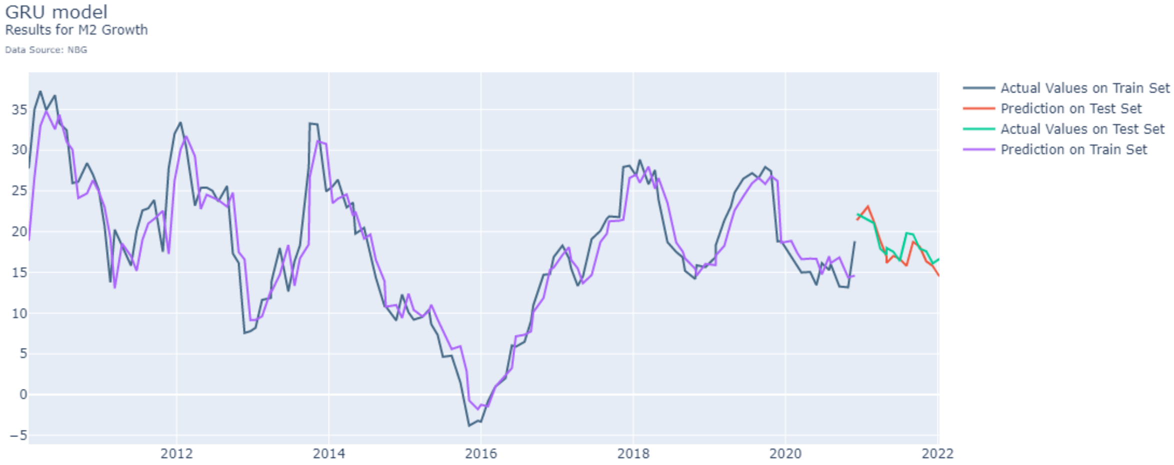


Figure 34 Real vs Predicted M2 Growth on training and testing set

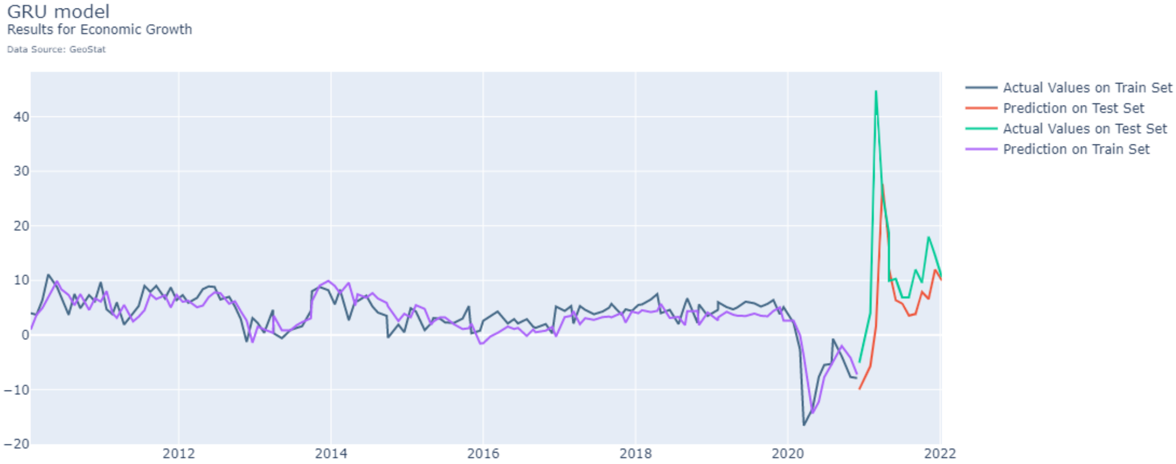


Figure 35 Real vs Predicted Economic Growth on training and testing set

Appendix D: Variable Importance For GRU Model based on SHAP Value

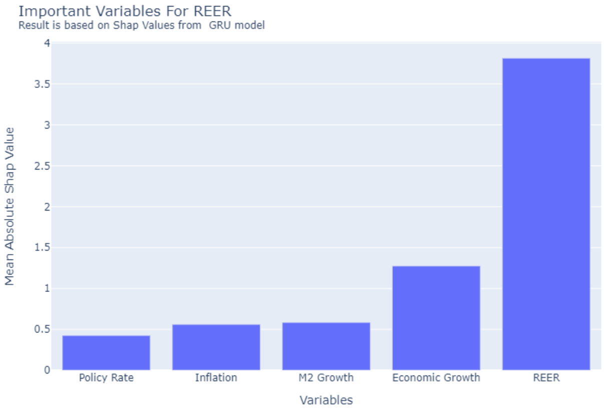


Figure 36 Variable Importance for REER

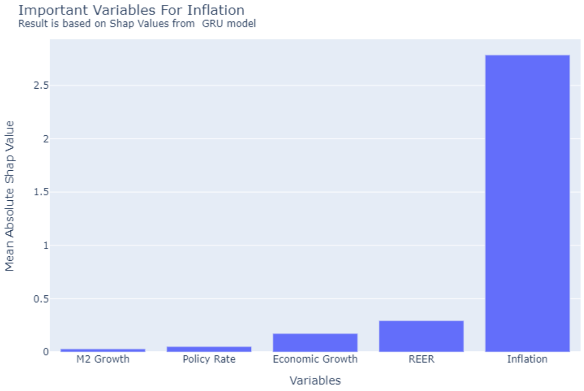


Figure 37 Variable Importance for Inflation

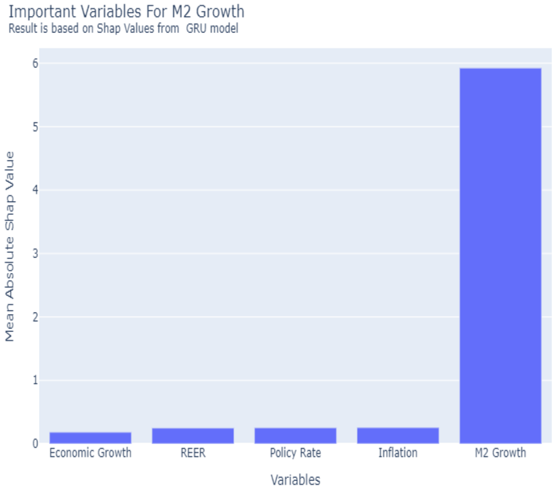


Figure 38 Variable Importance for M2 Growth

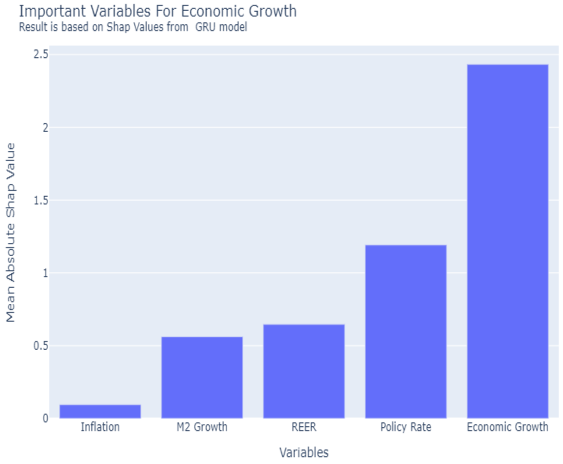


Figure 39 Variable Importance for Economic Growth

Appendix E: GRU model output explainability by SHAP Value

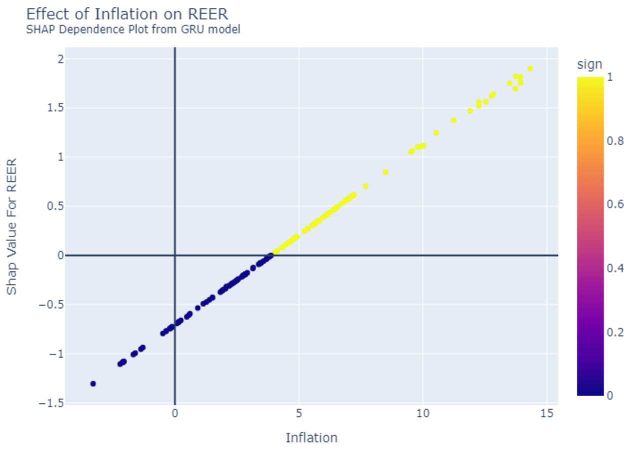


Figure 40 Effect of Inflation on REER Forecast

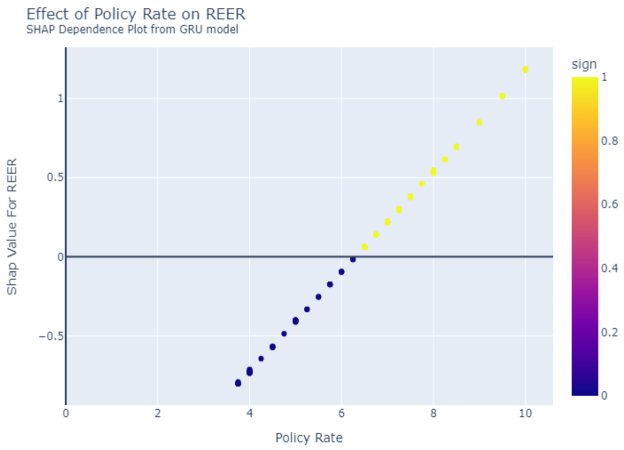


Figure 41 Effect of policy rate on REER Forecast

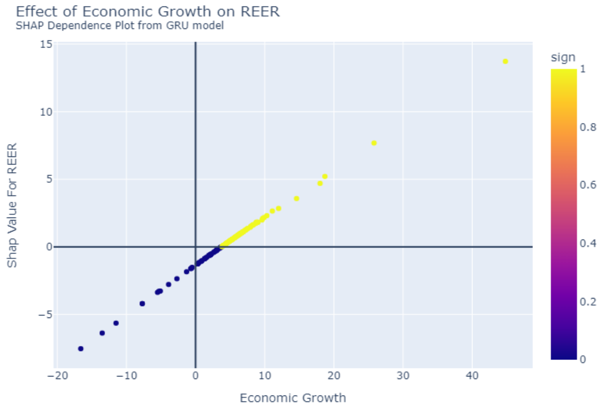


Figure 42 Effect of Economic Growth on REER Forecast

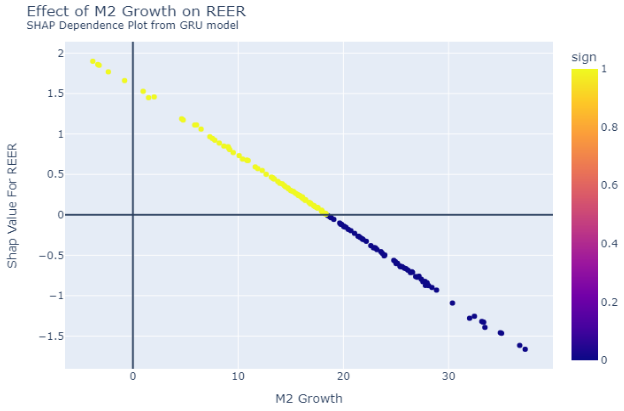


Figure 43 Effect of M2 Growth on REER Forecast

Explanation effect on REER Forecast

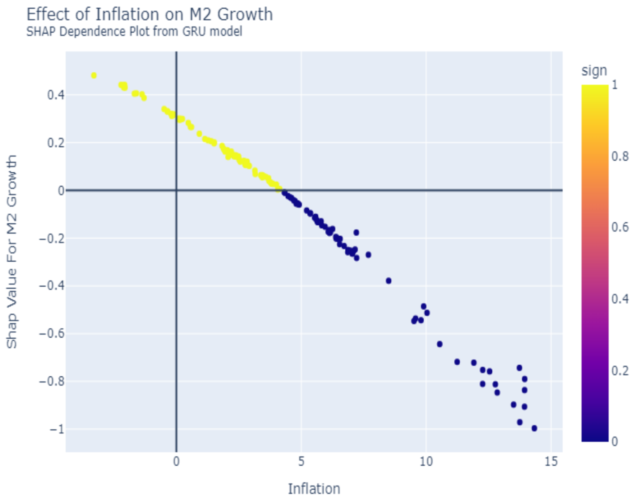


Figure 44 Effect of Inflation on M2 Growth Forecast

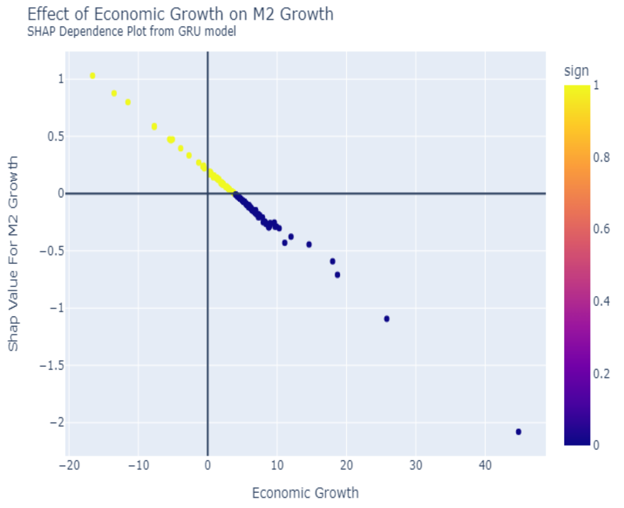


Figure 45 Effect of Economic Growth on M2 Growth Forecast

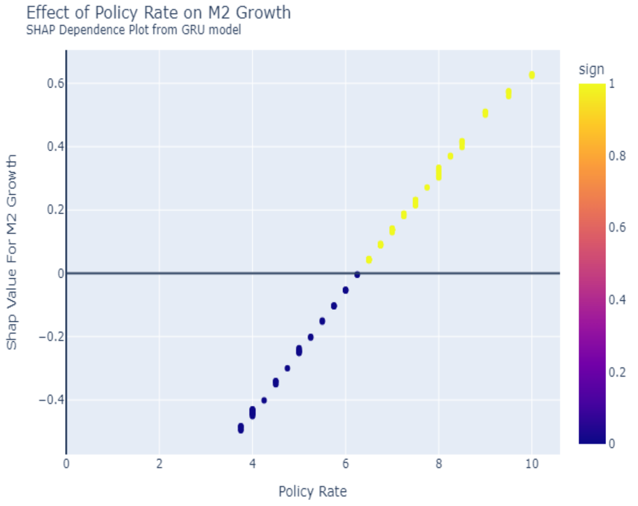


Figure 46 Effect of Policy Rate on M2 Growth Forecast

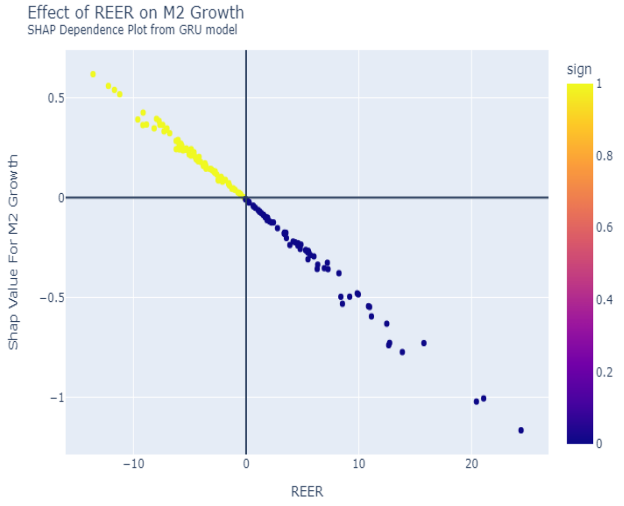


Figure 47 Effect of REER on M2 Growth Forecast

Explanation effect on M2 Growth Forecast

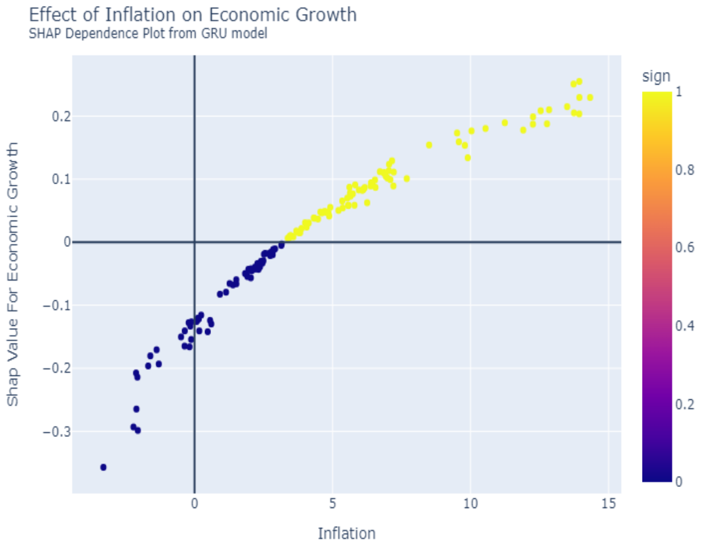


Figure 48 Effect of inflation on Economic Growth Forecast

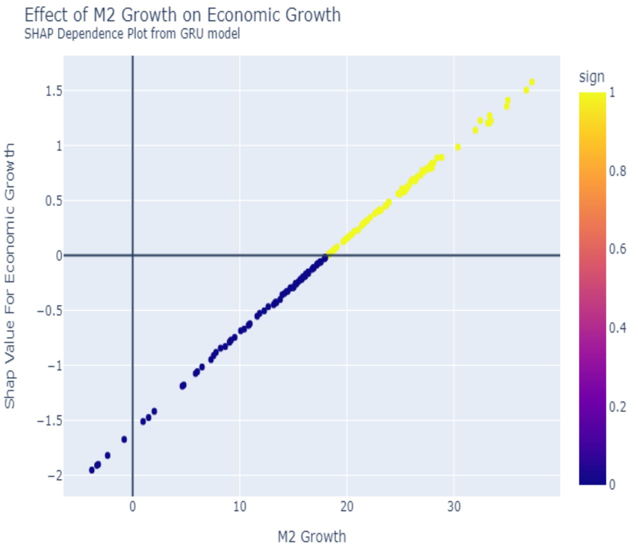


Figure 49 Effect of M2 Growth on Real Economic Growth Forecast

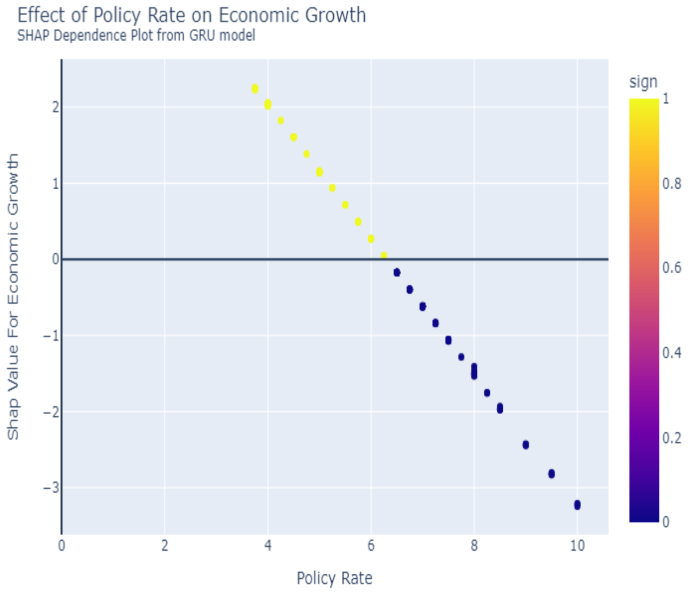


Figure 50 Effect of Policy Rate on Economic Growth Forecast

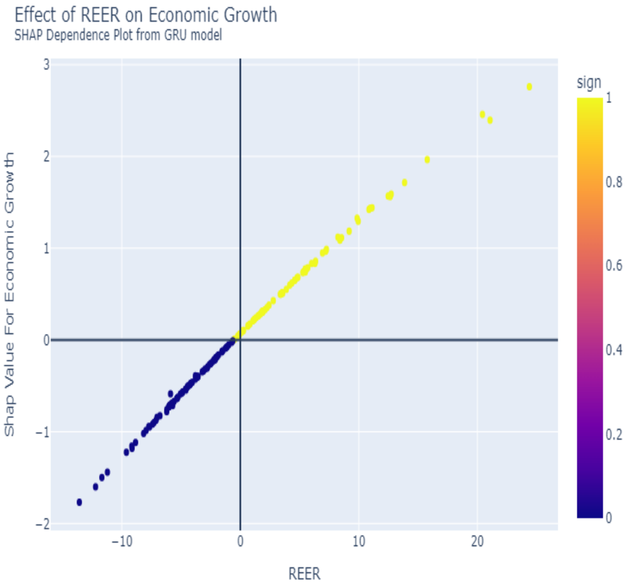


Figure 51 Effect of REER on Economic Growth Forecast

Explanation effect on Economic Growth Forecast

Appendix F: Testing Result against shocks

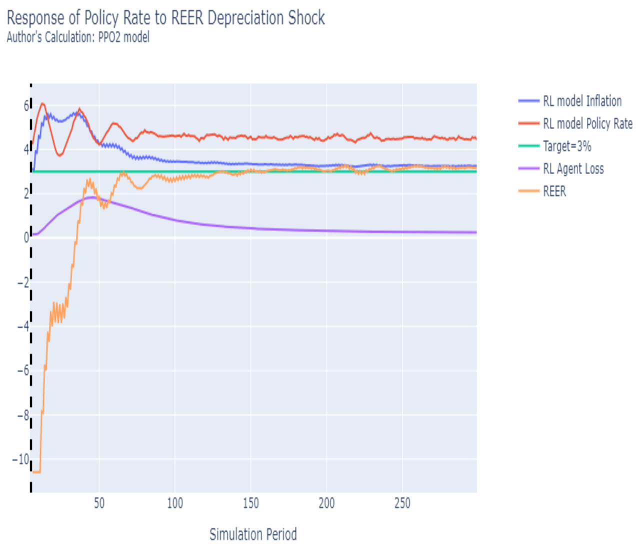


Figure 52 Response of Policy Rate to REER Depreciation shock

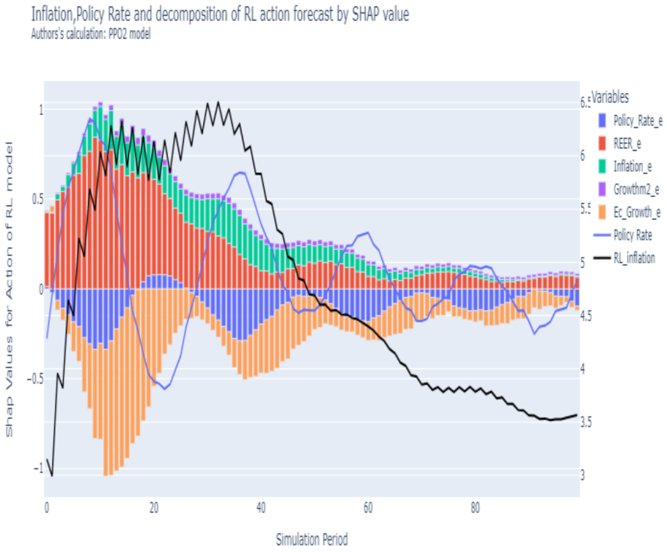
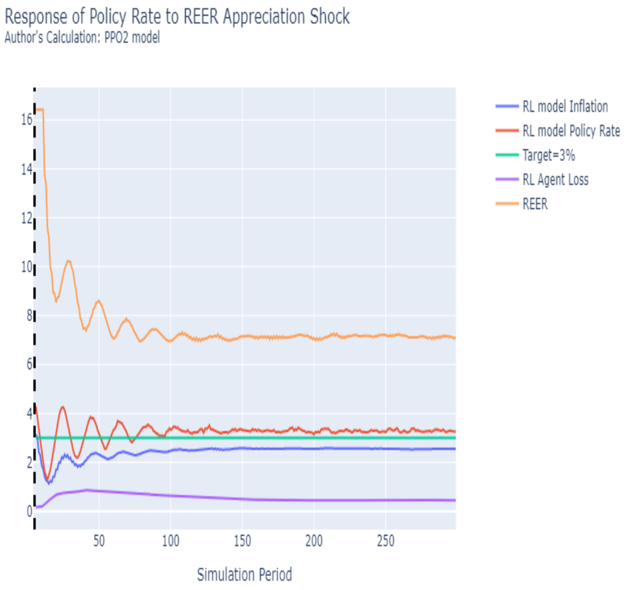


Figure 53 Decomposition of RL Action by Shap value

1. REER Depreciation shock



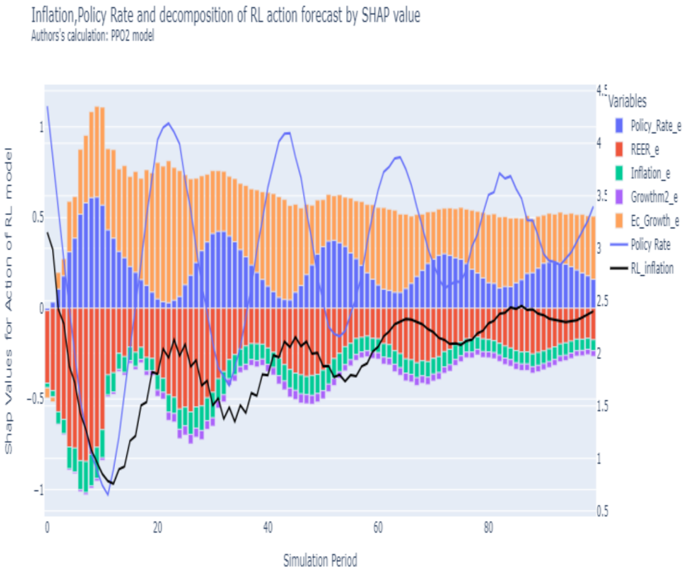
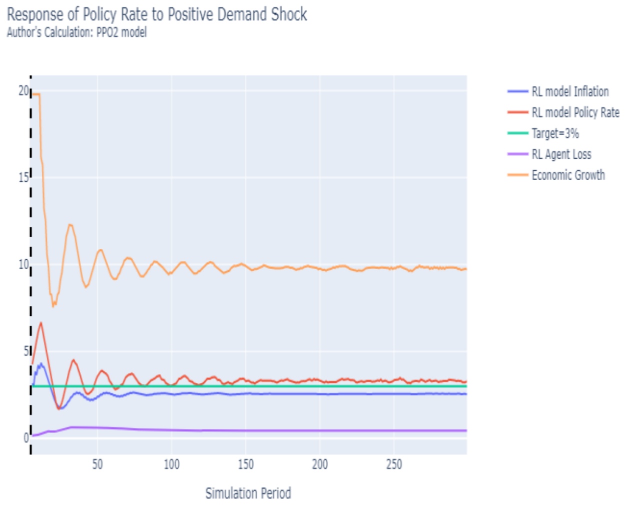


Figure 54 Response of Policy Rate to REER Appreciation shock

Figure 55 Decomposition of RL Action by Shap value

2. REER Appreciation shock



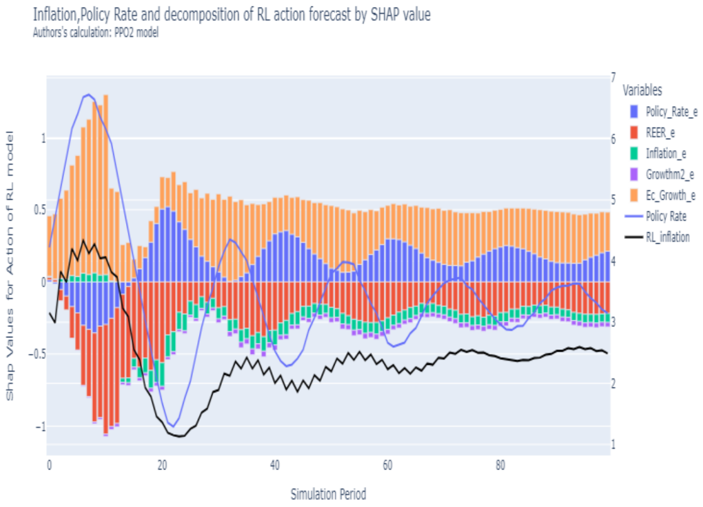
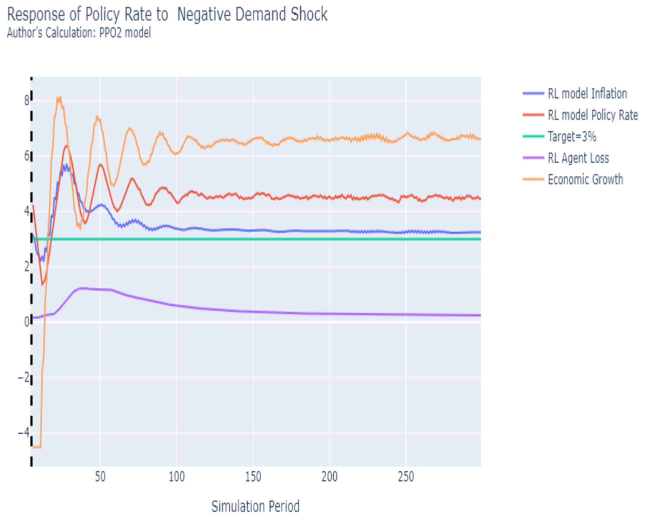


Figure 56 Response of Policy Rate to Positive Demand Shock

Figure 57 Decomposition of RL Action by Shap value

3. Positive Demand Shock



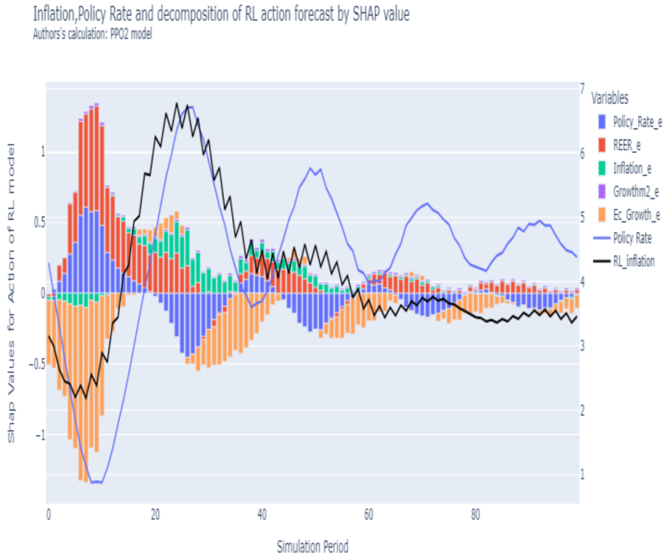
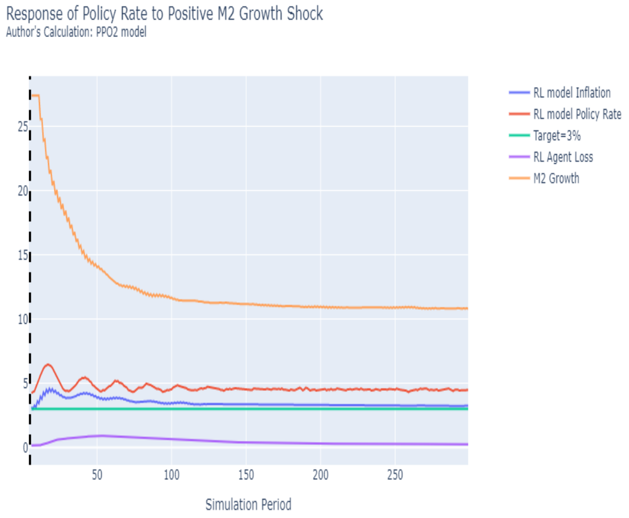


Figure 58 Response of Policy Rate to Negative Demand Shock

Figure 59 Decomposition of RL Action by Shap value

4. Negative Demand Shock



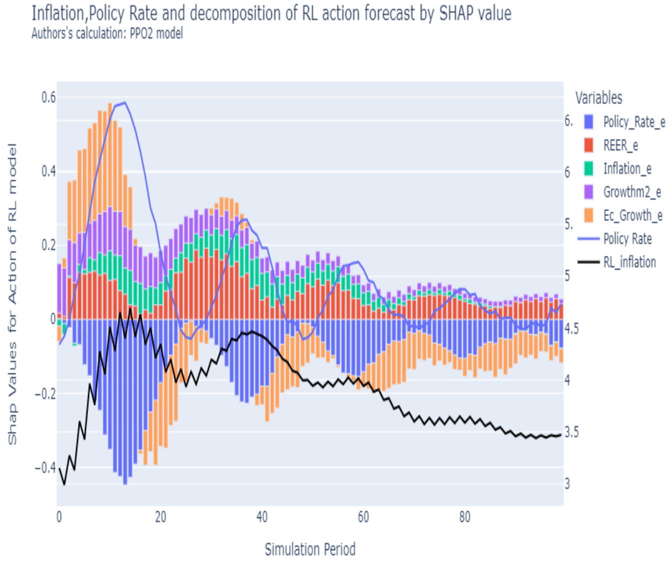


Figure 60 Response of Policy Rate to Positive M2 Growth Shock

Figure 61 Decomposition of RL Action by Shap value

5. Positive M2 Growth Shock

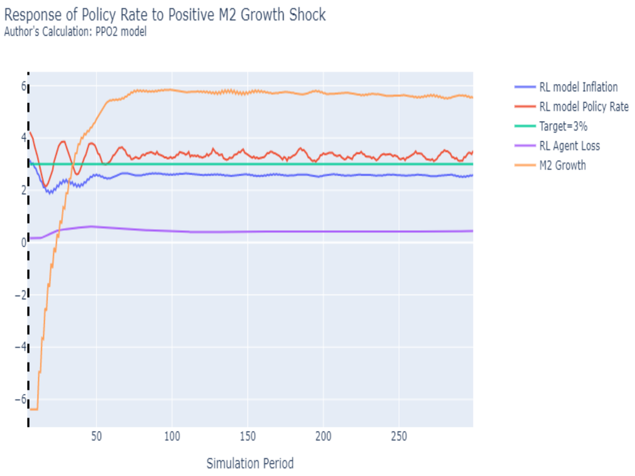


Figure 62 Response of Policy Rate to Negative M2 Growth Shock

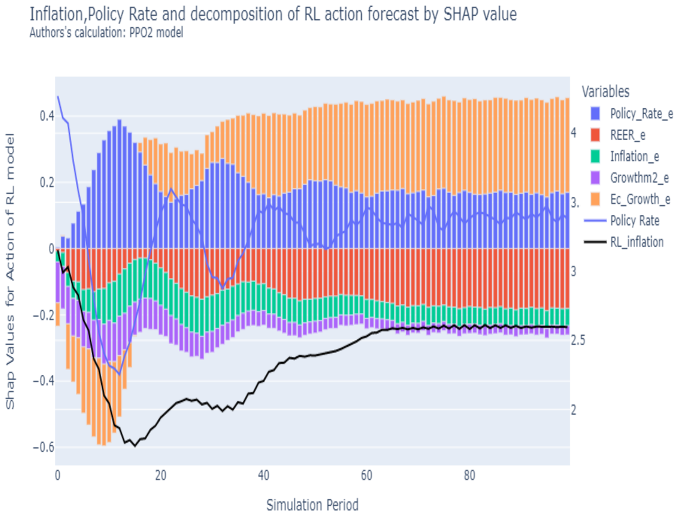


Figure 63 Decomposition of RL Action by Shap value

6. Negative M2 Growth Shock

Appendix G: Counterfactual analyses multi-level decision[‡‡‡]

The assumption is that at every period t, there is the shock to the environment defined by $f(x_t) - y_t$, which is residuals from GRU model. Thus, it means macro agent encountered the shock of the same size as the National Bank of Georgia faced at every period t (See Equation 30). Figure 66 shows that RL macro agent outperforms a decisions and achieves more stable inflation. In addition, macro agent achieves more stable inflation with a lower policy rate (see Figure 65).

[‡‡‡] Disclaimer: Comparisons with real data are of course not exact and are based on assumptions. During the actual decision making process, there could still be other variables and environment that our model cannot catch. An example of this is Covid-19, and there may be many more. Because of the above, this comparison may not be correct, but it is good for analyzing model results and for analytical visibility.

$$\aleph_{t+1} = f(\aleph_t) + (f(x_t) - y_{t+1}) \ (30)$$

Where, $\aleph_t = \left(\widehat{policy}_{t-4}, \widehat{M2_{g_t}}, \widehat{REER}_t, \widehat{Inflation}_t, Econ\widehat{omic}\_g_t\right)$ (31),

$x_t = \left(policy_{t-4}, M2_{g_t}, REER_t, Inflation_t, Economic\_g_t\right)$ (32),

$y_t = \left(M2_{g_t}, REER_t, Inflation_t, Economic\_g_t\right)$ (33)

f Represents functions learned by the GRU network

We used SHAP value decomposition techniques to explain the agent's decisions during counterfactual analyses. It is interesting to explain the part of the agent's decision where it strongly reduces the policy rate. At the beginning of the 2020 the policy rate starts to decline and already fell to 2% in early 2021 (See. Figure 65) Figure 67 shows that the main driver of such policy rate cuts was economic growth, which was very low due to the Covid-19 pandemic. In the following periods, we have a sharp increase in economic growth, pushing the policy rate to tighten. Decomposition with SHAP values allows us to adjust the agent's decision expertly, assuming that this variable's effect should not be considered when changing the policy rate.

Counterfactual Analysis: Agents' multi-step decision with real shocks

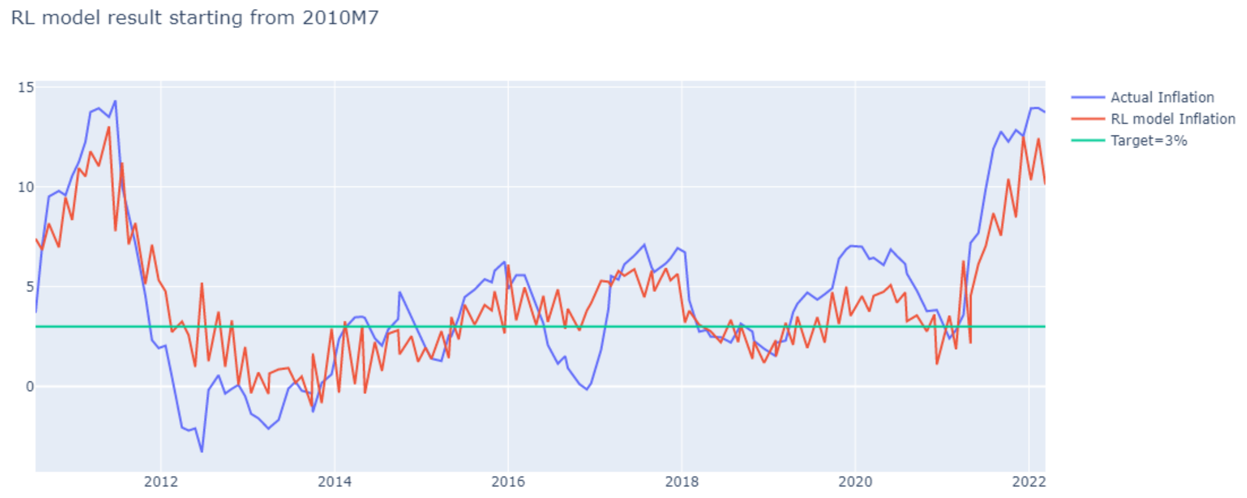


Figure 64 RL models and actual Inflation from 2010 to 2022

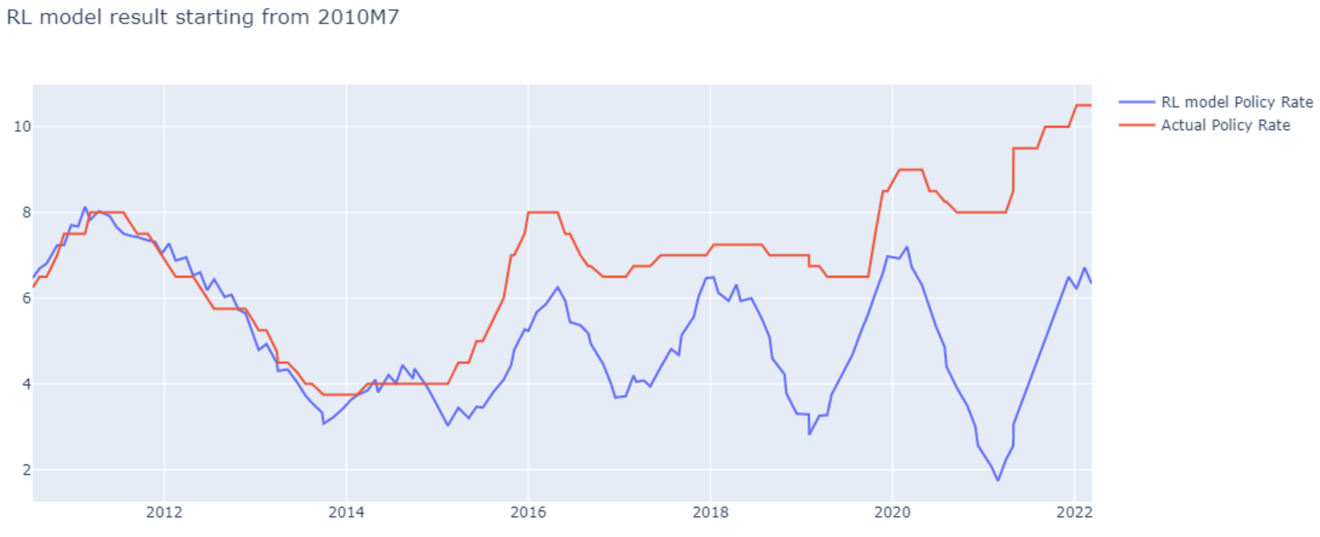


Figure 65 RL models and actual policy rate from 2010 to 2022

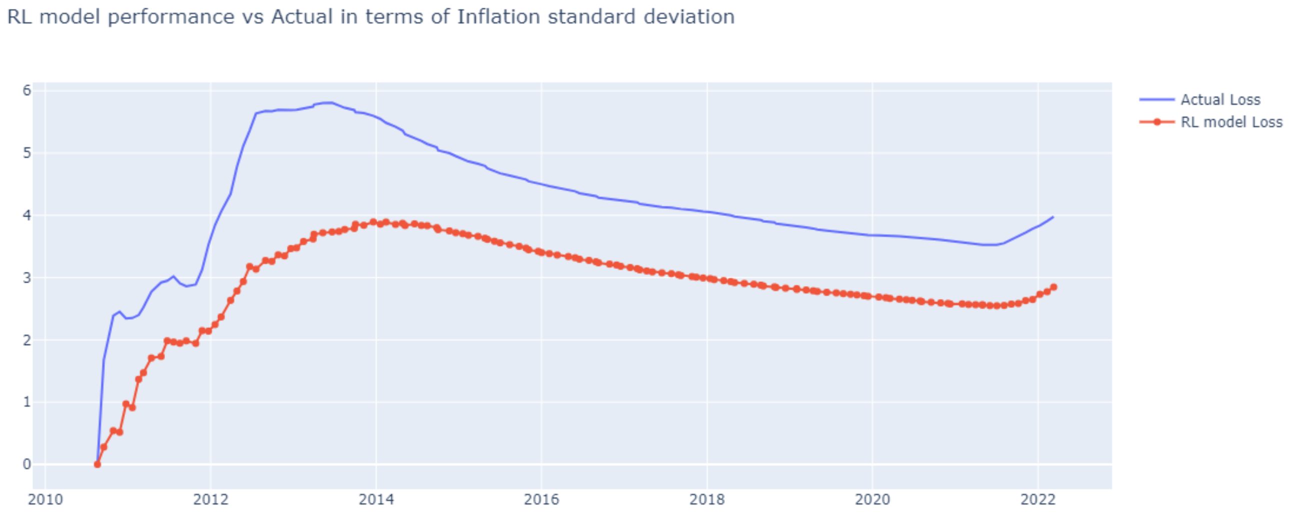


Figure 66 Testing RL agent Against Actual Decision

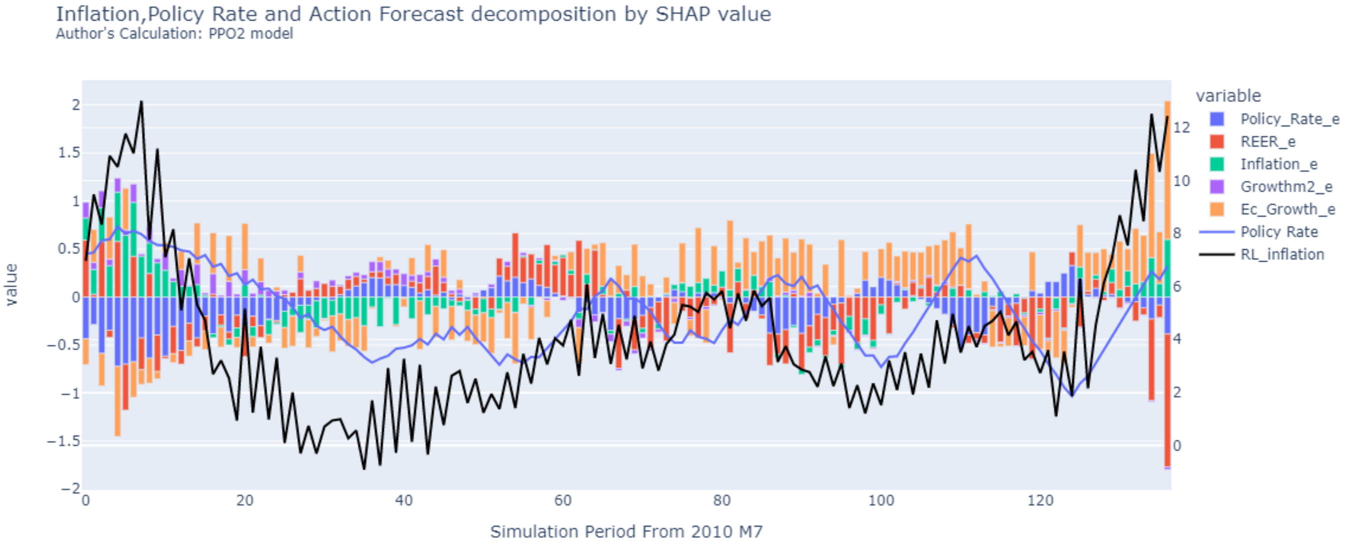


Figure 67 Action of RL model Decomposition by Shap value

Distributed by the National Bank of Georgia.

Available at www.nbg.gov.ge

National Bank of Georgia
Macroeconomic Research Division
1, Zviad Gamsakhurdia Embankment, 0114 Tbilisi
Phone: +995 32 2406278
www.nbg.gov.ge
Email: research@nbg.gov.ge

საქართველოს ეროვნული ბანკი
National Bank of Georgia