INSTITUTE
OF ECONOMIC STUDIES
Faculty of Social Sciences
Charles University

GRADIENT-BASED REINFORCEMENT LEARNING FOR DYNAMIC QUANTILE

Lukáš Janásek

IES Working Paper 12/2025

Institute of Economic Studies,
Faculty of Social Sciences,
Charles University in Prague

[UK FSV – IES]

Opletalova 26
CZ-110 00, Prague
E-mail : ies@fsv.cuni.cz
http://ies.fsv.cuni.cz

Institut ekonomických studií
Fakulta sociálních věd
Univerzita Karlova v Praze

Opletalova 26
110 00 Praha 1

E-mail : ies@fsv.cuni.cz
http://ies.fsv.cuni.cz

Bibliographic information:
Janásek L. (2025): " Gradient-Based Reinforcement Learning for Dynamic Quantile " IES Working Papers 12/2025. IES FSV. Charles University.

Gradient-Based Reinforcement Learning for Dynamic Quantile

Lukáš Janásek

[1]Institute of Economic Studies, Charles University, Prague, Czech Republic
E-mail: lukas.janasek@fsv.cuni.cz

July 2025

Abstract:
This paper develops a novel gradient-based reinforcement learning algorithm for solving dynamic quantile models with uncertainty. Unlike traditional approaches that rely on expected utility maximization, we focus on agents who evaluate outcomes based on specific quantiles of the utility distribution, capturing intratemporal risk attitudes via a quantile level $\tau \in (0, 1)$. We formulate a recursive quantile value function associated with time consistent dynamic quantile preferences in Markov decision process. At each period, the agent aims to maximize the quantile of a distribution composed of instantaneous utility combined with the discounted future value, conditioned on the current state. Next, we adapt the Actor-Critic framework to learn τ-quantile of the distribution and policy maximizing the τ-quantile. We demonstrate the accuracy and robustness of the proposed algorithm using an quantile intertemporal consumption model with known analytical solutions. The results confirm the effectiveness of our algorithm in capturing optimal quantile-based behavior and stability of the algorithm.

JEL: C61, C63
Keywords: Dynamic programming, Quantile preferences, Reinforcement learning

1 Introduction

Dynamic decision-making is fundamental to economic analysis, playing a central role in numerous fields and applications. Classical approaches to sequential decision-making typically revolve around maximizing the expected sum of discounted utility, providing a convenient and analytically tractable framework. However, a growing body of literature has proposed quantile maximization as a compelling alternative. Initially introduced by Manski (1988) and later axiomatized by Chambers (2009) and Rostek (2010), quantile preferences shift the focus from expected utility to specific points in the distribution of utility. As emphasized by Rostek (2010), quantile-based preferences offer attractive features: robustness to extreme outcomes and invariance to ordinal transformations of utility. With quantile preferences, the decision maker's attitude toward risk is captured by the quantile level $\tau \in (0,1)$, with lower values reflecting greater aversion to downside risk. The theoretical foundations for dynamic quantile preferences have been rigorously developed in recent work. de Castro and Galvao (2019) and de Castro and Galvao (2022) establish key properties of recursive quantile models, including time consistency and analytic tractability. Building on this, de Castro et al. (2025) further extends the framework to incorporate state-dependent decisions. Notably, as discussed in de Castro and Galvao (2019), the dynamic quantile model distinguishes between two dimensions of risk: intertemporal risk, shaped by the curvature of the utility function, and intratemporal risk, governed by the choice of the quantile level τ.

In parallel with these theoretical contributions, quantile preferences have been applied in various economic contexts. Giovannetti (2013) explored their implications for asset pricing. Long et al. (2021) used equal-quantile rules to design resource allocation mechanisms under uncertainty. In financial econometrics, Baruník and Čech (2021) developed a panel quantile regression model to capture systemic tail risks, and Baruník and Nevrla (2022) introduced the Quantile Spectral Beta to analyze risk across investment horizons. He et al. (2021) examined portfolio selection under median and quantile criteria, offering alternatives to expected return-based strategies. Finally, de Castro et al. (2022) provided experimental evidence on the extent to which individual behavior aligns with quantile maximization.

Despite theoretical advancements, quantile-based dynamic models present significant computational challenges. Classical numerical solution methods for dynamic decision problems are typically designed for expected utility models and are not directly applicable to the quantile setting (Taylor and Uhlig, 1990; Rust, 1996, 2016; Gaspat and L. Judd, 1997; Christiano and Fisher, 2000; Aruoba et al., 2006; Den Haan, 2010; Kollmann et al., 2011; Miao, 2013; Maliar and Maliar, 2014). A recent contribution by de Castro et al. (2023) introduced a quantile-based value function iteration algorithm for solving dynamic quantile models. However, traditional methods such as value function iteration rely on full discretization of the state and action spaces, as well as the transition dynamics. This makes them computationally infeasible in high-dimensional settings, namely when the state or action space is multidimensional or composed of multiple variables.

In this paper, we propose a novel gradient-based reinforcement learning algorithm designed for solving dynamic quantile models. Our approach is based on reinforcement learning combined with neural networks and offers several advantages over existing methods such as value iteration (de Castro et al., 2023). First, the algorithm scales naturally to high-dimensional state and action spaces, as neural networks are well-suited for processing complex, high-dimensional inputs. This allows the algorithm to handle large-scale models without requiring discretization of the model's structure. Second, reinforcement learning can operate in model-free environments, enabling our algorithm to be applied not only in theoretical settings but also in empirical or simulated environments where transition probabilities are unknown or difficult to specify.

Several recent studies have explored the application of reinforcement learning in economics. Zhou et al. (2025) used payoff-based reinforcement learning to study liquidity provision in limit order markets. He (2023) proposed a gradient-based reinforcement learning method for modeling belief-based equilibria in repeated games. Wu and Li (2024) applied an Actor-Critic algorithm to portfolio selection under regime uncertainty. Tahvonen et al. (2022) used reinforcement learning to solve high-dimensional forest management problems, while Bekiros (2010) introduced a fuzzy reinforcement learning model for forecasting financial market dynamics. These contributions collectively highlight the potential of reinforcement learning methods to address the computational complexity in dynamic models.

While reinforcement learning methods are traditionally designed to find policies that maximize the expected cumulative reward, a growing body of literature has focused on learning the distribution of cumulative rewards—a field known as distributional reinforcement learning. Bellemare et al. (2017) introduced this idea through the distributional Bellman equation, enabling the learning of the full distribution of cumulative rewards. Building on this, Dabney et al. (2018) employed quantile regression to model the reward distribution. However, in both cases, the learned distribution was ultimately used to compute its expectation, and the resulting policy aimed to maximize the expected return. In these approaches, modeling the distribution serves primarily to stabilize the learning of the expected value. A more closely related work to our work is that of Jiang et al. (2022), which proposes a method that not only learns the distribution of cumulative rewards but also directly optimizes the policy for a specific quantile. Yet the study does not address the issue of time-consistent quantile preferences as developed in de Castro et al. (2025).

The contribution of this paper is the following. First, we formalize dynamic quantile preferences within a Markov decision process (MDP) framework, generalizing the dynamic quantile preference approach presented in de Castro et al. (2025). Within this MDP framework, we define a recursively formulated value function consistent with dynamic quantile preferences. At each period, the agent aims to maximize the quantile of a distribution composed of instantaneous utility combined with the discounted future value, conditioned on the current state. The decision maker's intratemporal risk preference is characterized by selecting a quantile level $\tau \in (0, 1)$, indicating their attitude toward risk within each period.

Second, we propose a numerical solution to the dynamic quantile model. Our solution relies on a functional approximation of value function and the decision maker's policy using neural networks. Utilizing the reinforcement learning technique, we train the networks to approximate the theoretical value function and policy by a repetitive interaction with the underlying economic model. Specifically, we adapt Actor-Critic algorithm from the domain of deep reinforcement learning (Sutton and Barto, 2018). We modify the algorithm so that the Critic network estimates the τ-quantile of the relevant distribution, while the Actor network learns actions that maximize this quantile. Using neural networks enables us to approximate the value of any given state and identify optimal actions directly, resulting in a complete description of the solution. We present the algorithm for both finite and

infinite time horizons.

We evaluate the proposed algorithm on an intertemporal consumption-based dynamic model introduced by de Castro et al. (2025). In this model, a decision-maker chooses how to allocate wealth between current consumption and investment in a risky asset over time, aiming to maximize a recursively defined quantile-based utility function. The model is characterized by three key parameters: the discount factor (β), the risk attitude parameter (τ), and the elasticity of intertemporal substitution (determined by parameter γ). de Castro et al. (2025) provide explicit analytical solutions for the value function, optimal consumption, and asset allocation, enabling us to reliably assess the accuracy and efficiency of our numerical approach. For consistency and direct comparability, our numerical experiments adopt the same specification as de Castro et al. (2023), who previously used a value iteration algorithm to solve this consumption model. The results show high accuracy and convergence of the proposed algorithm. Moreover, we confirm the robustness of our algorithm through repeated simulations with different initializations, indicating reliable and stable convergence to optimal policy and value function estimates.

The paper is structured as follows. section 2 reviews the Markov decision process and fundamental concepts related to quantile preferences and establishes a quantile value function in the dynamic setting. section 3 introduces the gradient-based reinforcement learning algorithm, detailing the update rules for Critic and Actor network and the resulting algorithm for finite and infinite horizon. section 4 presents numerical experiments evaluating the algorithm against analytical benchmarks using the intertemporal consumption model. Finally, section 5 concludes by summarizing the main findings and suggesting directions for future research.

2 Dynamic quantile model

In this section, we establish the dynamic quantile model. First, we review the Markov decision process as a framework for decision-making in stochastic and dynamic setting. Next, we present basic theoretical concepts related to quantile preferences in a static case and, finally, we introduce quantile preferences in the dynamic setting fitting into the Markov decision process.

2.1 Markov decision process framework

In this paper, we focus on a class of stochastic dynamic economic models that can be formulated as a Markov decision process (MDP) Rust (1996). The MDP framework is a widely used and flexible tool for modeling discrete-time stochastic dynamic systems. It imposes no specific structure on the states, actions, or transition dynamics—both states and actions may be discrete or continuous, and can be either unidimensional or multidimensional. The core assumption of an MDP is the Markov property: the transition dynamics depend only on the current state and action, and not on the history or the path that led to the current state. Importantly, this assumption does not restrict decision-making to be based solely on the "present" in a narrow sense. The state itself can be defined to include relevant historical information, thus preserving memory and capturing path dependence when necessary. For example, consider an $AR(p)$ time series process. The state can be represented by the last p realizations of the series since the next value depends only on these. This structure satisfies the Markov property because the future depends solely on the current state, even though that state is composed of past observations. In this way, the MDP framework can capture persistence and cyclical dynamics common in economic modeling. Another appealing feature of MDPs is their ability to accommodate cases where actions directly determine future states. For instance, a savings decision made today determines the wealth available in the next period, making the action an explicit component of the future state.

Broadly, a vast majority of economic models that involve dynamic programming or sequential decision-making can be described in terms of MDP. By relying on the MDP framework, we can capture a large variety of economic models such as intertemporal consumer choice, firm behavior involving inventory and production planning, or labor supply. In macroeconomics, we can consider models of economic growth, business cycles, or fiscal and monetary policy models. In financial economics, the framework captures models of portfolio selection and asset pricing, where decisions and outcomes evolve over time in response to stochastic shocks. The application of the MDP framework in economic modeling is substantial. In the next paragraphs, we review the MDP framework.

Let $\mathcal{S}$ be a set of possible states and let $\mathcal{A}$ denote the space of possible actions available to the decision maker. In a period $t = 0, 1, \ldots$, the decision maker finds herself in a fully

observable state $s_t \in \mathcal{S}$ and chooses an action a_t. After taking the action a_t, the decision maker transitions to a new state s_{t+1} and receives a flow utility $u_{t+1} = u(s_t, a_t)$. The utility u_{t+1} is also commonly referred to as a reward, yet, we will use the term utility throughout the paper. In addition, we will assume that the utility $u(s_t, a_t)$ is a random variable. The state transition is characterized by a probability measure $\mathcal{P}_s$ and exhibits Markov property. That is the state s_t contains all relevant information about the transition to the next state and the transition history s_{t-1}, s_{t-2}... is not relevant for the future transition:

$$p(s_{t+1}|s_t) = p(s_{t+1}|s_0, s_1, \ldots, s_t) \tag{1}$$

Similarly, the flow utility $u(s_t, a_t)$ (its distribution) depends only on the current state s_t and the action a_t taken. The actions nor the states in the previous periods do not influence the utility or its distribution.

The dynamic decision-making in the MDP setting is represented by policy π. The policy is a mapping from state space $\mathcal{S}$ to the probability of selecting an available action $a_t \in \mathcal{A}$. We denote $\pi(a_t|s_t)$ as the probability of selecting action a_t in state s_t. A standard goal of MDP-based models is to find an optimal policy π^* that maximizes a value function of an initial state. Typically, for a specified policy π, a value function associated with an initial state s_0 is defined as the expected sum of discounted utilities:

$$v_\pi(s_0) = \mathbb{E}\left[\sum_{t=0}^{T-1} \beta^t u(a_t, s_t)\right] \tag{2}$$

where β is a discount factor and T is the number of periods in the model. For infinite horizon problems, the value function becomes the expected sum of infinite series of discounted flow utilities.

A central concept in solving for the optimal policy is the *Bellman operator*, which provides a recursive formulation of the value function. For any given policy π, the Bellman operator $\mathcal{T}_\pi$ maps a value function $v : \mathcal{S} \to \mathbb{R}$ to a new function $\mathcal{T}_\pi v$ defined as:

$$(\mathcal{T}_\pi v)(s_t) = \mathbb{E}\left[u(a_t, s_t) + \beta \sum_{s_{t+1} \in \mathcal{S}} p(s_{t+1}|s_t, a_t) v(s_{t+1})\right] \tag{3}$$

for all $s_t \in \mathcal{S}$, where $p(s_{t+1}|s_t, a_t)$ is the transition probability from state s_t to state s_{t+1} given action a_t. The Bellman operator expresses the expected value of following policy π

starting from state s_t, by combining the immediate utility with the discounted continuation value. In the case of an optimal policy, the corresponding Bellman optimality operator $\mathcal{T}$ is defined as:

$$(\mathcal{T}v)(s_t) = \max_{a \in \mathcal{A}} \mathbb{E}\left[u(a, s_t) + \beta \sum_{s_{t+1} \in \mathcal{S}} p(s_{t+1}|s_t, a)v(s_{t+1})\right] \tag{4}$$

The fixed point of this operator yields the optimal value function v^*, i.e., $v^* = \mathcal{T}v^*$, and the corresponding optimal policy π^* can be derived by selecting actions that achieve the maximum in the Bellman equation. Finally, we can express optimal value function v^* associated with optimal policy π^* in the following recursive way:

$$v^*(s_t) = \max_{a \in \mathcal{A}} \mathbb{E}\left[u(a, s_t) + \beta v^*(s_{t+1})\middle| s_t\right] \tag{5}$$

2.2 Static quantile preferences

We now shift from the standard expected utility approach to focus on quantile preferences. We first introduce these preferences in a static case and then extend them to the dynamic case in subsection 2.3.

We define τ-quantile of a random variable Z with cumulative density function F as:

$$\mathbf{Q}_\tau[Z] = inf\{z \in \mathbb{R}, F(z) \geq \tau\} \tag{6}$$

where $\tau \in (0, 1)$. Analogously, we define a quantile function F^{-1}, which is an inverse of F as:

$$F^{-1}(\tau) = \mathbf{Q}_\tau[Z] \tag{7}$$

For a utility function $u : \mathbb{R} \to \mathbb{R}$, we define τ-quantile preference over two random variables Z_1 and Z_2 as follows:

$$Z_1 \succeq Z_2 \iff \mathbf{Q}_\tau[u(Z_1)] \geq \mathbf{Q}_\tau[u(Z_2)] \tag{8}$$

The decision maker with τ-quantile preference chooses a variable Z_1 over Z_2 if a τ-quantile of utility stemming from variable Z_1 exceeds τ-quantile of utility stemming from variable Z_2. Compared to the standard expected utility case, we replace the expectations of the utility distribution with a quantile of the distribution. In a static case, the quantile prefer-

ence definition could even be simplified. de Castro and Galvao (2019) notes that for any strictly increasing and continuous function u, the quantile preference remains unaffected by the choice of utility function, due to the invariance of quantiles to monotonic transformations. Therefore, in the static case, the utility function can be omitted, allowing for a direct comparison of the quantiles of Z_1 and Z_2:

$$Z_1 \succeq Z_2 \iff \mathbf{Q}_\tau[u(Z_1)] \geq \mathbf{Q}_\tau[u(Z_2)] \tag{9}$$

$$\iff u(\mathbf{Q}_\tau[Z_1]) \geq u(\mathbf{Q}_\tau[Z_2]) \tag{10}$$

$$\iff \mathbf{Q}_\tau[Z_1] \geq \mathbf{Q}_\tau[Z_2] \tag{11}$$

Note that the quantile level τ reflects a decision maker's risk attitude (Manski, 1988; Rostek, 2010). Lower τ implies greater risk aversion. Quantile preferences also enable a meaningful separation of risk into intertemporal and intratemporal dimensions. This separation is valuable in dynamic settings, where uncertainty exists both across time and within each period. Quantile preferences isolate the within period risk attitude, encoded in τ, from the intertemporal trade-offs. Overall, quantile preferences possess several desirable properties: robustness, invariance under ordinal transformations, and a clear characterization of risk attitudes (Rostek, 2010). This makes quantile preferences an appealing choice for modeling decision makers' risk aversion.

2.3 Dynamic quantile preferences

We now extend the static quantile preference framework described in subsection 2.2 into a dynamic setting consistent with Markov decision processes, subsection 2.1. Unlike the standard expected utility framework where the decision maker aims to maximize the expected sum of discounted utilities, under quantile preferences the decision maker instead evaluates policies based on the quantile stemming from the discounted utilities. This shift in the objective function leads to significant modeling and analytical implications.

The dynamic quantile preferences were introduced by de Castro and Galvao (2019) and further studied by de Castro et al. (2022) and de Castro et al. (2025). An important consideration when working with dynamic quantile preferences is time consistency. In contrast to expected utility preferences, quantile-based preferences are not generally dynamically consistent - the decisions made to optimize a future quantile may no longer be optimal

when re-evaluated from an updated state. This inconsistency arises because the quantile is not a linear operator, which means we cannot rely on the law of iterated expectations. de Castro and Galvao (2019) illustrates this point with a clear example, showing that optimizing a quantile of the sum of discounted utilities in a sequential decision-making setting leads to time inconsistency. This occurs because the decision-maker may have an incentive to deviate from the plan chosen in the first period when making decisions in the second period. As a consequence, simply replacing the expectations operator in Equation 2 by a quantile leads to a *time inconsistent* value function

$$v_\pi^\tau(s_0) = \mathbf{Q}_\tau \left[\sum_{t=0}^{T-1} \beta^t u(a_t, s_t) \right] \tag{12}$$

Instead, de Castro and Galvao (2019) studied a recursive formulation of the quantile preference. The recursive structure was further extended by de Castro et al. (2025) into a more general setting who introduced a state conditioning. The recursive formulation combined with the state conditioning was shown to lead to dynamically consistent quantile preference. We build on the time consistent specification introduced in de Castro et al. (2025). For some policy π, we define a τ-quantile specific value function $v_\pi^\tau : \mathcal{S} \to \mathbb{R}$ that ensures time consistency with the following recursive equation:

$$v_\pi^\tau(s_t) = \mathbf{Q}_\tau \left[u(a_t, s_t) + \beta v_\pi^\tau(s_{t+1}) \middle| s_t \right] \tag{13}$$

where $\beta \in (0,1)$ is the discount factor, action a_t is chosen from policy π and s_{t+1} refers to a state following state s_t. The optimal value function $v^{*\tau}$ then satisfies the following recursive specification

$$v^{*\tau}(s_t) = \max_{a \in \mathcal{A}} \mathbf{Q}_\tau \left[u(a, s_t) + \beta v^{*\tau}(s_{t+1}) \middle| s_t \right] \tag{14}$$

Note that the equation has the same form as the recursive definition of the value function in the expected case in Equation 5. We only replaced the expectation operator with the quantile operator. This similarity provides a direct link between the dynamic quantile preference and the expected utility maximization theory. We can replace the expected operator with the quantile operator, but only in the recursive specification. Through the law of iterated expectations, the standard expected utility maximization theory then allows

to write the objective as the expectations over the discounted sum of utilities as in Equation 2. However, no such law is available for the quantile operator and we must thus use the recursive specification following Equation 14.

From Equation 14, we can see that the decision maker with τ-quantile preference chooses the action that maximizes the quantile of the sum of current period utility and the discounted next period value conditioned on the current period state. Let denote the sum as follows

$$\mathbf{y}^* \stackrel{d}{=} u(a, s_t) + \beta v^{*\tau}(s_{t+1}) \Big| s_t, \quad \text{where } a \sim \pi^*(\cdot|s_t) \tag{15}$$

Both components of the sum in distribution from Equation 15 are random variables, and they together determine the distribution whose quantile is being maximized. The current period utility from an action is a random variable from the definition of MDP. The next period value $v^{*\tau}(s_{t+1})$ is a random variable since there is uncertainty about the next period state s_{t+1}. Yet the uncertainty about the next period state is conditioned on the current state. This ensures the dynamic consistency of the decision-making. Each state is assigned a value through the value function and the dynamic quantile preference reflects only the instantaneous utility and the next state transition. By choosing the actions, the agent influences the instantaneous utility and the probability of transitioning from state s_t to a new state s_{t+1}. The agent thus controls the distribution in Equation 15 directly by choosing utility distribution and indirectly through the probability of the occurrence of the next state s_{t+1}. The balance between the instantaneous utility and the future value makes the decision-making dynamic with τ-quantile level capturing the risk preference.

Unfortunately, the recursive specification leads to computational difficulties. We can expand the recursive structure as a sequence of nested conditional quantiles:

$$v^{*\tau}(s_t) = \max_{a_t} \mathbf{Q}_\tau \Big[u(a_t, s_t) + \beta v^{*\tau}(s_{t+1}) \,\Big|\, s_t \Big] \tag{16}$$

$$= \max_{a_t} \mathbf{Q}_\tau \Big[u(a_t, s_t) + \beta \max_{a_{t+1}} \mathbf{Q}_\tau \big[u(a_{t+1}, s_{t+1}) + \beta v^{*\tau}(s_{t+2}) \,\big|\, s_{t+1} \big] \,\Big|\, s_t \Big] \tag{17}$$

$$= \max_{a_t} \mathbf{Q}_\tau \Big[u(a_t, s_t) + \beta \max_{a_{t+1}} \mathbf{Q}_\tau \Big[u(a_{t+1}, s_{t+1}) + \beta \max_{a_{t+2}} \mathbf{Q}_\tau \Big[\cdots \tag{18}$$

$$+ \beta \max_{a_{t+n-1}} \mathbf{Q}_\tau \big[u(a_{t+n-1}, s_{t+n-1}) + \beta v^{*\tau}(s_{t+n}) \,\big|\, s_{t+n-1} \big] \cdots \Big| s_{t+1} \Big] \Big| s_t \Big] \tag{19}$$

Yet, searching for the optimal value function requires a recursive evaluation of the con-

ditional nested quantiles. This makes finding an analytical solution of dynamic quantile models generally difficult. As a remedy, we present a numerical solution in section 3 which is the main contribution of this paper.

3 Gradient-based reinforcement learning algorithm

In this section, we present a numerical solution to the dynamic quantile model from section 2.

3.1 Preliminaries

Dynamic economic problems have traditionally been approached using methods such as value iteration. These techniques are well-established and provide a reliable numerical solution to dynamic stochastic optimization problems. A standard approach is to fully discretize both the state space and the action space, along with a possible discretization of the model's transition dynamics to match the state and action grid. The value function is represented as a matrix of real numbers each corresponding to a state and action grid cell. A Bellman operator is then used to iteratively update the value function estimates for individual cells until the value function estimates converge. The policy is then derived from the final value function estimates simply as an action leading to the highest value function for each state grid. This approach was used by de Castro et al. (2023).

However, these methods can be applied as long as the full discretization of state space is feasible. For problems, where the state is not represented with a single variable but a collection of multiple variables or a vector, a full discretization of the n-dimensional state space complicates. As the number of variables representing the state space grows, discretization becomes computationally prohibitive and, in many cases, infeasible. This phenomenon is also known as the curse of dimensionality.

We propose an alternative approach based on function approximation that does not require any discretization of states, actions or the transition dynamics and works in high dimension settings. Rather than representing the value function as a matrix of real numbers over a grid, we instead directly approximate the theoretical value and policy functions. For the value function, we aim to find a parametrized mapping $\boldsymbol{V}_w$: $\mathcal{S} \to \mathbb{R}$ that will return a value function estimate $\boldsymbol{V}_w(s)$ for any state s. For the policy, we aim to find a mapping

π_θ: $\mathcal{S} \rightarrow \mathcal{A}$ that will return an action a for any state s mimicking the optimal theoretical policy of the decision maker.

Generally, the functional form of the value function and the policy is unknown, and putting any assumptions on the functional form is cumbersome. To avoid any issues with the specification of functional form, we use neural networks that can serve as a universal approximator, Hornik (1991). Another advantage of neural networks is their ability to process high dimensional inputs. Specifically, we use multilayer perceptron which is a computational model composed of layers of nodes. Each node takes inputs, multiplies them by weights, sums the product, and applies a non-linear activation function to the sum. Each layer in the perceptron repeats this process, passing its outputs as inputs to the next layer nodes. Ultimately, the resulting network is a non-linear mapping that projects the network's input space to the output space.

Our solution relies on two neural networks. We use network $\boldsymbol{V_w} : \mathcal{S} \rightarrow \mathbb{R}$ to approximate the theoretical optimal value function v_π^τ defined in Equation 14. The network consists of trainable weights $\boldsymbol{w} \in \mathbb{R}^m$ and the non-linear activation functions. The network takes a state s as its input, transforms the input and returns an estimate of the theoretical value function. We refer to the network $\boldsymbol{V_w}$ as Critic network. Next, we use network $\boldsymbol{\pi_\theta}$ that represents the decision maker's policy. The network consists of trainable weights $\boldsymbol{\theta} \in \mathbb{R}^{m'}$ and the non-linear activation functions. It processes state s and returns a probability of choosing action a when making a decision in state s. We denote the probability as $\boldsymbol{\pi_\theta}(a|s)$ and refer to the network $\boldsymbol{\pi_\theta}$ as Actor. The specific output layer of Actor depends on the type of action. For discrete actions, Actor network returns a vector of probabilities $\boldsymbol{\pi_\theta}(a|s)$ for individual actions. For economic models where the actions are continuous, we represent the policy with a parametrized density function. In the continuous case, Actor returns the parameters of the density function and $\boldsymbol{\pi_\theta}(a|s)$ represents the value of the density function. The choice of the density function depends on the nature of the actions, e.g. the range of possible actions.

Our objective is to find optimal weights $\boldsymbol{w}^*$ and $\boldsymbol{\theta}^*$ so that Critic and Actor networks approximate the theoretical value function from Equation 14 and the respective optimal policy as precisely as possible. To find the optimal weights, we adopt a method from the

reinforcement learning framework, which is well-suited for modeling dynamic decision-making in MDP-like environments. In reinforcement learning, an agent or a decision maker interacts with an environment over a sequence of time steps, makes decisions based on observed states, receives feedback in the form of instantaneous utility, and updates her decision-making or belief about the value function according to the received instantaneous utility. Unlike traditional dynamic programming methods that rely on complete knowledge of the transition probabilities, reinforcement learning methods can operate also in model-free settings, where the agent does not require an explicit specification of the environment's dynamics. This is particularly advantageous for economic applications, where such information is difficult to derive analytically, or available only through simulation. For a detailed introduction to reinforcement learning, we refer the reader to Sutton and Barto (2018).

In our setting, we train Critic and Actor network by a repetitive interaction with the underlying economic model and update the weights with each interaction. This process leads to a gradual improvement of the policy and the value function estimate towards its theoretical counterparts. In theory, we can simulate an infinite number of interactions with the underlying model and update the weights to any arbitrary degree of precision. Hence, we can train the networks to fit the theoretical model almost perfectly as we can simulate an unlimited amount of interactions from which the networks can learn. In the next sections, we detail the gradient-based update rules for Critic and Actor weights reflecting the quantile preferences and the training algorithm itself.

3.2 Quantile value function update rule

Under some policy π, the Critic network $\boldsymbol{V}_{\boldsymbol{w}}$ represents an estimate of τ-quantile value function given by Equation 13. Our goal is to learn the τ-quantile of the distribution

$$u(a_t, s_t) + \beta v_\pi^\tau(s_{t+1}) \Big| s_t \tag{20}$$

which is the sum of the instantaneous utility and the discounted future certainty equivalent of the value function conditioned on the current state. Since the value function of the next state s_{t+1} is unknown, we will use the Critic network estimate in state s_{t+1} instead.

Consequently, we aim to learn the τ-quantile of the following distribution denoted as $\mathbf{y}_{\boldsymbol{w}}$:

$$\mathbf{y}_{\boldsymbol{w}} \stackrel{d}{=} u(a_t, s_t) + \beta \boldsymbol{V}_{\boldsymbol{w}}(s_{t+1}) \Big| s_t \tag{21}$$

Now, consider the following transition in the MDP. The decision maker chooses action $a_t \sim \pi(\cdot|s_t)$ in state s_t, receives utility u_{t+1} and transition to a new state s_{t+1}. For this transition, we calculate a target value which is a sum of the flow utility u_{t+1} and the discounted certainty equivalent as estimated by Critic in state s_{t+1}:

$$y_t = u_{t+1} + \beta \boldsymbol{V}_{\boldsymbol{w}}(s_{t+1}) \tag{22}$$

Notice that the target y_t is a single sample from the distribution in Equation 21. Since the transition realized from state s_t, the distribution of the instantaneous utility and the future certainty equivalents is already conditioned on state s_t. Thus, the target variable y_t represents a single realization from the distribution for which the decision maker has a τ-quantile preference.

In order to learn the τ-quantile of the distribution in Equation 21, we use quantile regression loss over y_t realization. We calculate the temporal difference error of the Critic estimate in state s_t compared to the target y_t as follows:

$$\delta_t = y_t - \boldsymbol{V}_{\boldsymbol{w}}(s_t) \tag{23}$$

Following Koenker (2005), the τ-quantile regression loss is given by the following expression:

$$L_Q(\delta_t) = |\tau - \mathbb{I}\{\delta_t < 0\}| \cdot |\delta_t| \tag{24}$$

where $\mathbb{I}$ is an indicator function with value 1 if $\delta_t < 0$ and 0 otherwise. We perform a gradient descent step to Critic weights minimizing the quantile regression loss from Equation 24, which gives us the following update rule for Critic network:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha_{\boldsymbol{w}} \nabla_{\boldsymbol{w}} L_Q(\delta_t) \tag{25}$$

where $\alpha_{\boldsymbol{w}}$ is a small positive learning rate. By performing a single update from Equation 25, Critic's weights are updated in the direction minimizing the quantile loss from Equation 24 and the Critic estimate becomes closer to the theoretical value function from Equation 13.

3.3 Policy update rule

Now we can turn to the gradient update rule for the Actor network π_{θ} representing decision makers policy. Our goal is to gradually update weights so that Actor favors actions leading to higher value function. Recall that the Actor network returns the probability of actions and the resulting policy is stochastic. Our objective is thus to increase the probability of actions leading to higher value function. We now derive the Actor update rule borrowing from the methodology proposed by Jiang et al. (2022).

Assume that the decision maker is facing distribution from Equation 21 in state s_t with a cumulative density function $F_Y(y, \boldsymbol{\theta})$, with Y being a random variable following distribution $\mathbf{y}_w$. The distribution depends on policy π and thus on policy parameters $\boldsymbol{\theta}$. Further assume, that there exists an inverse function F_Y^{-1}, so that we can express the τ-quantile of variable Y conditioned on the policy parameters as:

$$\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}] = F_Y^{-1}(y, \boldsymbol{\theta}) \tag{26}$$

Our goal is to derive the gradient of the τ-quantile with respect to the policy parameters $\nabla_{\boldsymbol{\theta}}\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}]$ so that we can find Actor's weights maximizing the τ-quantile. From definition, the following expression holds:

$$F_Y(\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}],\ \boldsymbol{\theta}) = \tau \tag{27}$$

Taking the derivative of both sides with respect to parameters $\boldsymbol{\theta}$ in point $\boldsymbol{\theta}_0$ and applying a chain rule for multivariate functions we obtain the following relationship:

$$\nabla_y F_Y\big|_{y=\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}_0],\ \boldsymbol{\theta}=\boldsymbol{\theta}_0} \cdot \nabla_{\theta}\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} + \nabla_{\theta} F_Y\big|_{y=\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}_0],\ \boldsymbol{\theta}=\boldsymbol{\theta}_0} = 0 \tag{28}$$

from which we can express the derivative of the τ-quantile with respect to policy parameters:

$$\nabla_{\boldsymbol{\theta}}\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}]\big|_{y=\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}_0],\ \boldsymbol{\theta}=\boldsymbol{\theta}_0} = -\frac{\nabla_{\boldsymbol{\theta}} F_Y}{f_Y}\Big|_{y=\mathbf{Q}_{\tau}[Y|\boldsymbol{\theta}_0],\ \boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{29}$$

where f_Y denotes the density of variable Y.

Since f_Y is positive and univariate, it does not affect the direction of the gradient in Equation 29. We now solve for the numerator $\nabla_{\boldsymbol{\theta}} F_Y$. We can express F_Y with the expecta-

tions over an indicator function (Jiang et al., 2022):

$$F_Y(y, \boldsymbol{\theta}) = \mathbb{E}\left[\mathbb{I}\{Y \leq \mathbf{Q}_\tau[Y|\boldsymbol{\theta}]\}\right] \tag{30}$$

The value of quantile $\mathbf{Q}_\tau[Y|\boldsymbol{\theta}]$ is unknown. Nevertheless, we use Critic network to provide an estimate of its value

$$\mathbf{Q}_\tau[Y|\boldsymbol{\theta}] \approx \boldsymbol{V}_{\boldsymbol{w}}(s_t) \tag{31}$$

Then we can express the gradient $\nabla_{\boldsymbol{\theta}} F_Y$ in terms of the Actor network as follows:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} F_Y(y, \boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}\left[\mathbb{I}\{Y \leq \mathbf{Q}_\tau[Y|\boldsymbol{\theta}]\}\right] && (32)\\
&\approx \nabla_{\boldsymbol{\theta}} \mathbb{E}\left[\mathbb{I}\{Y \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\}\right] && (33)\\
&= \nabla_{\boldsymbol{\theta}} \mathbb{E}\left[\sum_a \boldsymbol{\pi}_{\boldsymbol{\theta}}(a|s_t)\mathbb{I}\{Y \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\}\right] && (34)\\
&= \mathbb{E}\left[\sum_a \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(a|s_t)\mathbb{I}\{Y \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\}\right] && (35)\\
&= \mathbb{E}\left[\sum_a \boldsymbol{\pi}_{\boldsymbol{\theta}}(a|s_t)\nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\pi}_{\boldsymbol{\theta}}(a|s_t))\mathbb{I}\{Y \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\}\right] && (36)\\
&= \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\pi}_{\boldsymbol{\theta}}(a_t|s_t))\mathbb{I}\{Y \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\}\right] && (37)
\end{aligned}$$

We estimate the last term using the transition s_t, a_t, s_{t+1} and u_{t+1} as

$$\mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\pi}_{\boldsymbol{\theta}}(a_t|s_t))\mathbb{I}\{Y \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\}\right] \approx \nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\pi}_{\boldsymbol{\theta}}(a_t|s_t))\mathbb{I}\{u_{t+1} + \beta \boldsymbol{V}_{\boldsymbol{w}}(s_{t+1}) \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\} \tag{38}$$

Building on Equation 29 and Equation 38, the update rule for Actor network becomes the following:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_\theta \nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\pi}_{\boldsymbol{\theta}}(a_t|s_t))\mathbb{I}\{r_{t+1} + \beta \boldsymbol{V}_{\boldsymbol{w}}(s_{t+1}) \leq \boldsymbol{V}_{\boldsymbol{w}}(s_t)\} \tag{39}$$

where α_θ is a learning rate. The update rule can be interpreted as minimizing the log-likelihood of actions that result in lower value than the benchmark given by the Critic network in state s_t. Repeatedly applying this rule reinforces actions that lead to policies with higher value than the Critic's benchmark.

Table 1. Summary of Notation

π	policy
$\pi(a\|s)$	probability of action a in state s under policy π
v_π^τ	value function for τ-level preference under policy π
$\pi^{*\tau}$	optimal policy for τ-level preference
$v^{*\tau}$	value function for τ-level preference under optimal policy
$\boldsymbol{\pi_\theta}$	Actor network with weights $\boldsymbol{\theta}$
$\boldsymbol{V_w}$	Critic network with weights $\boldsymbol{w}$

3.4 Time consistent Quantile Actor-Critic algorithm

We build the resulting algorithm on the update rules for Critic network Equation 25 and Actor network Equation 39. We present an algorithm both for finite time horizon and infinite time horizon.

The algorithm for the finite horizon is depicted in Algorithm 1. The algorithm is based on a standard Actor-Critic algorithm, see Sutton and Barto (2018), with adjusted update rules matching the time consistent quantile preference. We repeatedly interact with the underlying model and train the Actor and Critic network on individual transitions in the model. When the agent reaches a terminal state, we disregard the continuation value and start from the initial state again. By repeating this process, the Actor and Critic network weights converge to a local optimum.

Algorithm 1 Finite horizon Actor-Critic for time consistent dynamic quantile preferences

Set $\tau \in (0,1)$ preference level
Randomly initialize Actor network $\boldsymbol{\pi_\theta}$ with weights $\boldsymbol{\theta}$
Randomly initialize Critic network $\boldsymbol{V_w}$ with weights $\boldsymbol{w}$
while not done **do**:
  Get initial state s
  while s' is not terminal **do**:
    Sample action a in current state s: $a \leftarrow \boldsymbol{\pi_\theta}(\cdot|s)$
    Take action a, observe new state s' and utility u
    Calculate target value: $y \leftarrow u + \beta \cdot \boldsymbol{V_w}(s') \cdot is\_terminal$
    Calculate error of value function: $\delta \leftarrow y - \boldsymbol{V_w}(s)$
    Update critic weights $\boldsymbol{w}$ with update rule from Equation 25:
      $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha_w \nabla_{\boldsymbol{w}} L_Q(\delta)$
    Update actor weights $\boldsymbol{\theta}$ with update rule from Equation 39:
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_\theta \nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\pi_\theta}(a|s)) \mathbb{I}\{\delta \leq 0\}$
    Set $s \leftarrow s'$
  end while
end while

The algorithm for the infinite horizon is depicted in Algorithm 2. Since there is no terminal state in the infinite horizon setting, we always use the Critic network as an estimate of the next state value. This leads to a potential instability during the training since a large error in the next value estimate can introduce a large error in the current value estimate leading to an even higher error in the next value estimate and eventually to a divergence. To stabilize the training, we use a Critic target network providing next state estimates (Mnih et al., 2015). The target network is a copy of Critic network but its weights are updated with a delay. Specifically, after each update to the Critic network, the target network parameters are adjusted using a soft update rule: $\boldsymbol{w}' \leftarrow \rho \boldsymbol{w} + (1-\rho)\boldsymbol{w}'$, where $\rho \in (0,1]$ controls the update rate. This ensures that the target values change slowly, reducing the risk of divergence.

Algorithm 2 Infinite horizon Actor-Critic for time consistent dynamic quantile preferences

Set $\tau \in (0,1)$ preference level
Randomly initialize Actor network $\boldsymbol{\pi_\theta}$ with weights $\boldsymbol{\theta}$
Randomly initialize Critic network $\boldsymbol{V_w}$ with weights $\boldsymbol{w}$
Set Critic target network $\boldsymbol{V'_{w'}}$, weights to Critic network weights
while not done **do**:
  Get initial state s
  Sample action a in current state s: $a \leftarrow \boldsymbol{\pi_\theta}(\cdot|s)$
  Take action a, observe new state s' and utility u
  Calculate target value: $y \leftarrow u + \beta \cdot \boldsymbol{V'_{w'}}(s')$
  Calculate error of value function: $\delta \leftarrow y - \boldsymbol{V_w}(s)$
  Update critic weights $\boldsymbol{w}$ with update rule from Equation 25:
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha_w \nabla_{\boldsymbol{w}} L_Q(\delta)$
  Update actor weights $\boldsymbol{\theta}$ with update rule from Equation 39:
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_\theta \nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\pi_\theta}(a|s)) \mathbb{I}\{\delta \leq 0\}$
  Update target network weights: $\boldsymbol{w}' \leftarrow \rho \boldsymbol{w} + (1-\rho)\boldsymbol{w}'$
  Set $s \leftarrow s'$
end while

4 Assessing the reinforcement learning algorithm

In this section we assess the proposed algorithm on an intertemporal consumption model with dynamic quantile preferences studied by de Castro et al. (2025). Since there exists an analytical solution to the model, we compare the numerical results with their theoretical counterpart. This allows us to access the convergence and precision of the algorithm. We also conduct an experiment where we repeatedly train the individual networks

with different initialization of weights to evaluate the stability of the solution.

4.1 Intertemporal consumption

To assess our reinforcement learning algorithm, we adopt the intertemporal consumption-saving model with dynamic quantile preferences studied by de Castro et al. (2025). This model is particularly useful for evaluating the numerical algorithm, as it provides explicit analytical solutions for the value function, optimal consumption, and asset allocation under certain conditions. Hence, it serves as an effective benchmark to test convergence and accuracy. We choose the same parametrization as de Castro et al. (2023), so our results are directly comparable with the numerical method presented by de Castro et al. (2023).

Consider a decision maker who, at the beginning of period t, possesses an amount $x_t \in X \subseteq \mathbb{R}+$ of a risky asset. This asset yields a stochastic return denoted by $z_t \in Z \subseteq \mathbb{R}++$. Thus, the agent starts the period with total wealth equal to $x_t z_t$. Each period, the agent decides how much wealth to consume immediately, denoted by c_t, and how much to carry forward as asset holdings for the next period, x_{t+1}. The consumption and asset holdings are related by the following relationship:

$$c_t = x_t z_t - x_{t+1}. \tag{40}$$

In each period t, the agent receives a flow CRRA utility from the consumption:

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma} \tag{41}$$

The agent's dynamic preferences translate to solving the following recursive dynamic quantile preference problem for the value function, de Castro et al. (2025):

$$v(x_t, z_t) = \max_{x_{t+1} \in [0, x_t z_t]} \left\{ \frac{(x_t z_t - x_{t+1})^{1-\gamma}}{1-\gamma} + \beta v(x_{t+1}, z_{t+1}) \middle| z_t \right\} \tag{42}$$

where $\gamma > 0$, $\gamma \neq 1$ measures risk aversion, $\beta \in (0,1)$ is the discount factor, and $\tau \in (0,1)$ characterizes the agent's quantile-based risk preference.

de Castro et al. (2025) showed that when asset returns z_t are independently and identically distributed (i.i.d.), the model has the following analytical solution. Letting

$$a_{\tau,\gamma} = \beta^{1/\gamma} (Q_\tau[z])^{(1-\gamma)/\gamma} \tag{43}$$

the explicit analytical solutions for consumption, asset allocation, and value function are the following:

$$c_t = (1 - a_{\tau,\gamma}) x_t z_t, \tag{44}$$

$$x_{t+1} = a_{\tau,\gamma} x_t z_t, \tag{45}$$

$$v(x_t, z_t) = \frac{(1 - a_{\tau,\gamma})^{-\gamma}}{1 - \gamma} (x_t z_t)^{1-\gamma} \tag{46}$$

These closed-form expressions allow us to directly assess the numerical accuracy and convergence properties of the proposed reinforcement learning algorithm.

4.2 Training setup

We start with framing the intertemporal consumption problem in the Markov decision process setting. The state s_t is represented by an asset holding x_t and its return z_t. The state representation then becomes

$$s_t = (x_t, z_t) \tag{47}$$

Knowing the asset holding and the return, the agent decides about the consumption level c_t. We represent the action as consumption share from the available wealth:

$$a_t = \frac{c_t}{x_t z_t} \tag{48}$$

After choosing the consumption level, the agent receives a flow utility $u(c_t)$:

$$u_{t+1} = u(c_t) \tag{49}$$

The transition dynamics depend on the distribution of the next period return z_{t+1}. Following the example in de Castro et al. (2023), we assume that the returns are i.i.d with possible values $\{0.9, 0.95, 1, 1.05, 1.15\}$. We assign use probabilities $\{0.3, 0.10, 0.15, 0.3, 0.15\}$ for the individual shock values.

We restrict the approximation to an interval of asset holding $x_t \in [0.1, 2]$. We sample the initial value x_0 from this interval. Since it can happen that the asset holding diverges from this interval during the training we use boundaries 0.1 and 2. Whenever the asset holding crosses the boundaries we use a naive guess of the next period value function. The naive guess ensures that we have some value function estimates even outside the region of

interest. Specifically, we use the following guess:

$$v_{\text{guess}} = \frac{(c_{\text{share}} \cdot wQ_z)^{1-\gamma}}{(1-\gamma)[1-\beta[(1-c_{\text{share}})Q_z]^{1-\gamma}]} \tag{50}$$

The guess corresponds to a value function when the decision maker having wealth $w = x_t z_t$ will choose a static consumption share c_{share} under constant return $Q_z = \mathbf{Q}_\tau(z)$ for the all next periods. After 50 time steps, we sample a new initial state x_0, and the training proceeds from the initial state. The reason for this is that the intertemporal consumption model is an infinite time steps model, and we cannot simulate the interaction to the end.

Next, we set up the Critic and Actor networks. The Critic consists of two hidden layers with 20 neurons each, using hyperbolic tangent (tanh) and leaky ReLU activation functions respectively. The output layer has no activation function and provides the estimate of the value function. The Actor features two hidden layers, each with 11 neurons and leaky ReLU activations. We model the policy with Beta distribution density. The Beta distribution is specifically chosen because it naturally models continuous actions constrained within the interval $[0, 1]$. Hence, we can sample the consumption share from the Beta distribution density and ensure that its values are between 0 and 1. The output layer of Actor thus consists of two output neurons representing α and β parameters of Beta distribution. To ensure both coefficients are positive, we use the softplus activation function on the Actor's output layer.

We use a discount factor of 0.95. For the learning rates, we use polynomial decay - we start with a higher learning rate so that both Critic and Actor can calibrate quickly and then gradually decrease the learning rate to get more precise estimates. For Critic, we start with a learning rate of 0.01 and end with 0.001. For Actor, we start with 0.001 and end with 0.0001. We set the speed of adjustment for the Critic target $\rho = 0.02$. To stabilize the training, we parallelize the interactions. We let the networks interact simultaneously with 1024 independent realizations of the underlying model. In addition, we collect the transitions into a batch and update the network weights every 4 time steps. As a result, a single update of Critic's and Actor's weight is made on 4096 transitions.

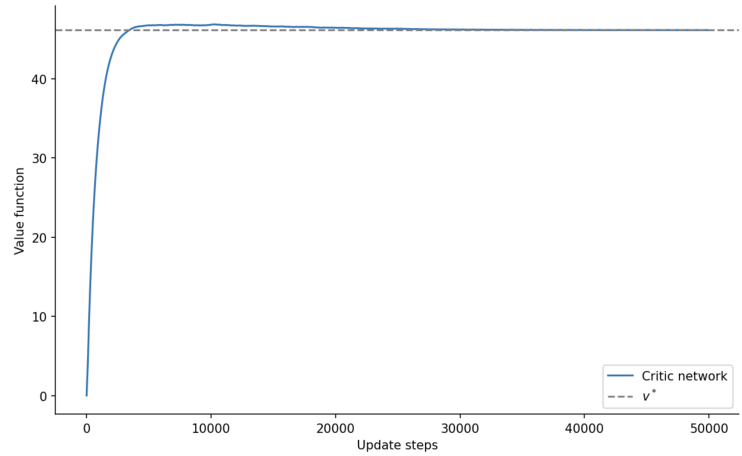


Figure 1. Convergence of Critic network value function estimate for $\tau = 0.5$ in state $x = 1$ and $z = 1$

The gray dashed line, v^*, refers to the theoretical optimal value function. The blue solid line corresponds to the predictions of Critic network in state $x = 1$ and $z = 1$ after the respective number of updates to its weights.

4.3 Results

First, we present the results for the training process of individual networks for risk preference $\tau = 0.5$. Figure 1 depicts the convergence of value function estimate in state with asset holdings $x = 1$ and return $z = 1$. The prediction of the Critic network prior to the training is 0 and uninformed. After approximately the first 5000 updates the estimate quickly approaches the theoretically optimal counterpart with a slight overshooting. After the 30000 updates, the numerical estimate becomes indistinguishable from the theoretical value function. The chart confirms the convergence of Critic network in the selected state. The convergence of Actor network is depicted in Figure 2. The figure shows densities of Beta distribution parametrized by Actor network predictions after the respective number of updates. The first sub-figure depicts a flat density on the interval $[0, 1]$. It illustrates that the starting policy is uninformed with an expected value consumption share of 0.5. With a higher number of updates, the density shifts towards the theoretically optimal policy depicted by the vertical green dashed line. After 5000 updates, the density starts to concentrate around the optimal consumption share with the expected value (vertical blue dashed line) overlapping the optimal consumption share. After 50000 updates, the density is narrowly distributed around the optimal consumption share. The expected value of the density changes negligibly compared to the density after 5000, but the variance of the density reduces significantly. Together Figure 1 and Figure 2 illustrate a joint convergence of

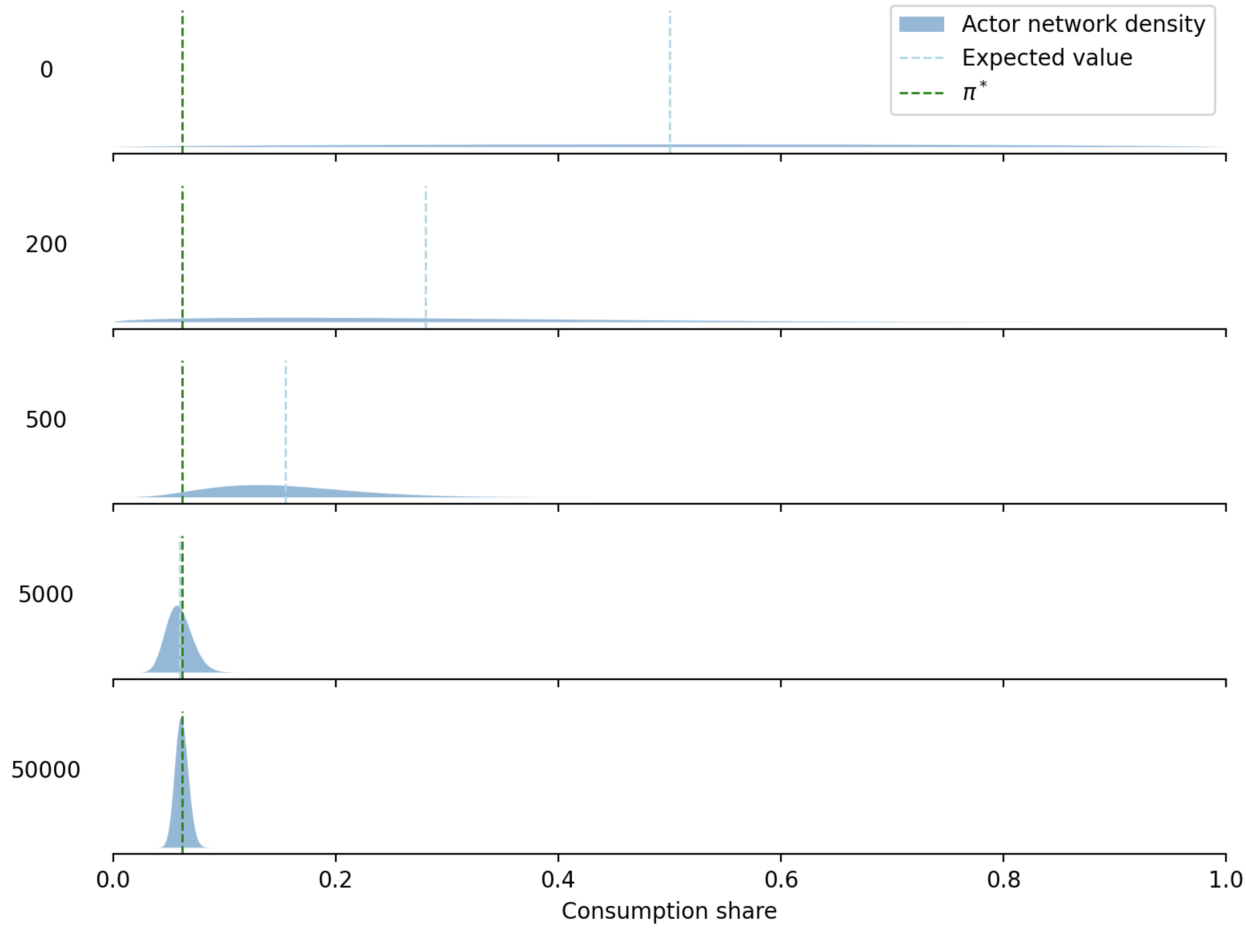


Figure 2. Convergence of Actor network policy for $\tau = 0.5$
The blue area depicts the density of Beta distribution as predicted by Actor network after the respective number of updates. The blue dashed line corresponds to the expected value associated with the density. The green dashed line, π^*, refers to the theoretical optimal policy for the consumption share.

Critic and Actor networks. After approximately 5000 updates, the prediction of networks is already relatively accurate. Yet, more updates provide more accurate estimates.

Next, we present the value function and policy for different values of asset holding x for various risk preference levels. Figure 3 depicts results for three Critic networks, each trained for a respective risk preference level $\tau = 0.25, 0.5, 0.75$. On the selected interval of asset holding $x \in [0.1, 2]$, the Critic networks exhibit an almost perfect fit to the theoretical value function for all risk preference levels. This result gives us the confidence that the proposed algorithm can provide very precise estimates of the value function for various risk preference levels. Recall, that we have theoretically an infinite number of transitions on which the Critic network can be trained. Therefore, a network with sufficient capacity should be capable of a perfect fit on a selected interval. Figure 4 compares the policy

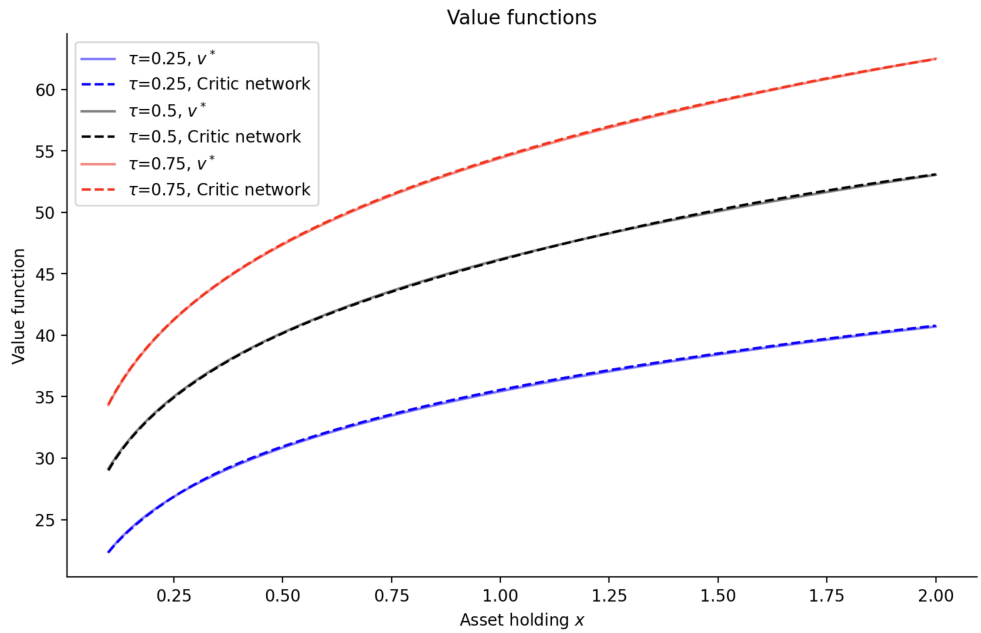


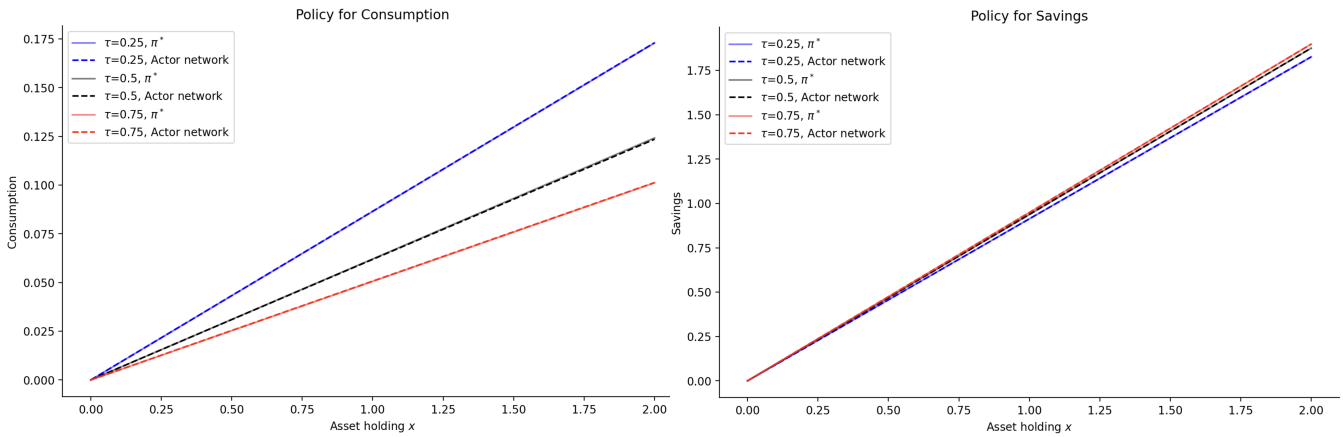
Figure 3. Comparison between Critic network prediction and theoretical optimal value function for risk preference levels $\tau = 0.25, 0.5, 0.75$ and $z = 1$

based on Actor network against its theoretical optimal counterpart. The left part of Figure 4, compares the policies for consumption level. The dashed lines represent an optimal consumption level for a specified asset holding level given for the three risk preference levels. The solid lines represent policy derived from the Actor network for the respective risk level. We derive the consumption level policy by taking the expected value of Beta distribution:

$$c = \frac{\alpha}{\alpha + \beta} \cdot x \tag{51}$$

where coefficients α and β are predicted by the Actor network. The policy for saving level is then simply the difference between the predicted consumption level and the asset holding $x - c$. From the left part of the figure, we can see that the consumption level derived from the expected value of the Beta distribution almost perfectly matches the optimal consumption level. This chart implies that the Actor network managed to learn the optimal policy for all three risk preference levels. The right part of the figure illustrates the approximation of the optimal saving level.

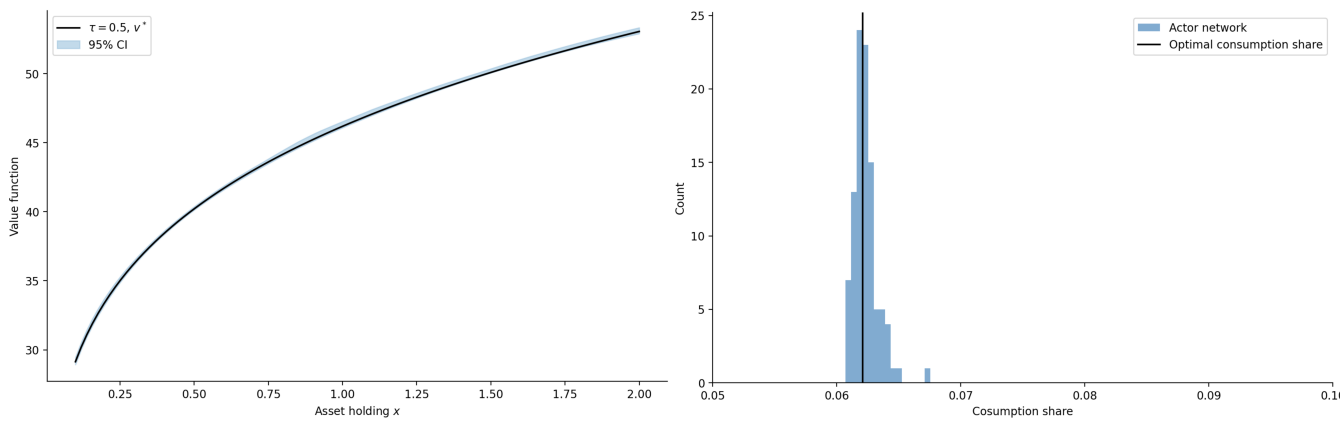
Since our algorithm is based on a gradient descent optimization, the algorithm converges to locally optimal weights $\boldsymbol{w}$ and $\boldsymbol{\theta}$. The resulting solution is thus dependent on the weight initialization as it does guarantee to find globally optimal weights. Consequently,



(a) **(b)**

Figure 4. Comparison of Actor network based policy with theoretical optimal policy for consumption and saving for risk preference levels $\tau = 0.25, 0.5, 0.75$ and $z = 1$

each training can produce different results. For this reason, we evaluate the stability of the results. We are agnostic to the stability of the network weights themselves, our interest is in the stability of the networks' output. As long as the network provides a good approximation of the theoretical value function and policy, the weighs themselves are not relevant. To evaluate the stability of network outputs, we run an experiment where we train 100 pairs of Critic and Actor networks each with differently initialized weights. We train each pair with the exact same setting except for using the differently initialized weights at the beginning of the training. The results of the experiment are depicted in Figure 5. The



(a) **(b)**

Figure 5. Results of the experiment measuring the uncertainty of Critic and Actor network convergence for risk preference $\tau = 0.5$

left chart in the figure depicts the stability of Critic network output trained for $\tau = 0.5$.

The black line represents the theoretical optimal value function. The blue area around the value function represents the band that contains 95% of the Critic network outputs for the given asset holding level on the x-axis. The chart illustrates the strong stability of the Critic network output. The algorithm found a suitable solution regardless of the weight initialization. The right chart in the figure depicts the stability of Actor network. It plots a histogram of the expected value of Actor policies around the theoretical optimal consumption share. The resulting policies are tightly distributed around the optimal consumption share suggesting stability of the algorithm also for the Actor network.

5 Conclusion

This paper develops a numerical solution method for dynamic quantile preference models, where a decision-maker maximizes a stream of future utilities evaluated through the τ-quantile, for $\tau \in (0,1)$. We first formalize the quantile preference framework within a Markov decision process and establish a recursive formulation of the value function that guarantees time consistency. The key contribution of the paper is the development of a novel gradient-based reinforcement learning algorithm using neural networks that solves the dynamic quantile optimization problem. By adapting the Actor-Critic method, we propose an algorithm in which the Critic network estimates the τ-quantile value function and the Actor network optimizes actions to maximize this value. The approach is scalable to high-dimensional models and applicable in both model-based and model-free environments.

To evaluate the algorithm, we study an intertemporal consumption model with risky asset returns, for which a closed-form analytical solution is available. This allows us to benchmark the accuracy and convergence of the algorithm across various quantile-based risk attitudes. Our results demonstrate that the proposed algorithm provides highly accurate approximations of the theoretical value and policy functions, and converges reliably across different initializations. Moreover, we confirmed the robustness of our algorithm through repeated simulations with different initializations, indicating reliable and stable convergence to optimal policy and value function estimates.

Overall, the developed gradient-based reinforcement learning framework significantly expands the applicability of dynamic quantile models, enabling economists and practition-

ers to solve complex, high-dimensional economic problems previously considered computationally intractable. This approach offers a promising direction for future research, extending quantile-based decision-making models into broader, more realistic economic environments such as dynamic models with partial observability, general equilibrium settings, or multi-agent interactions.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to refine and polish the written text. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

Aruoba, S. B., J. Fernández-Villaverde, and J. F. Rubio-Ramírez (2006). Comparing solution methods for dynamic equilibrium economies. *Journal of Economic Dynamics and Control 30*(12), 2477–2508.

Baruník, J. and M. Nevrla (2022, 06). Quantile spectral beta: A tale of tail risks, investment horizons, and asset prices. *Journal of Financial Econometrics 21*(5), 1590–1646.

Baruník, J. and F. Čech (2021). Measurement of common risks in tails: A panel quantile regression model for financial returns. *Journal of Financial Markets 52*, 100562.

Bekiros, S. D. (2010). Heterogeneous trading strategies with adaptive fuzzy actor–critic reinforcement learning: A behavioral approach. *Journal of Economic Dynamics and Control 34*(6), 1153–1170.

Bellemare, M. G., W. Dabney, and R. Munos (2017). A distributional perspective on reinforcement learning. In *International conference on machine learning*, pp. 449–458. PMLR.

Chambers, C. P. (2009, April). An Axiomatization Of Quantiles On The Domain Of Distribution Functions. *Mathematical Finance 19*(2), 335–342.

Christiano, L. J. and J. D. Fisher (2000). Algorithms for solving dynamic models with occasionally binding constraints. *Journal of Economic Dynamics and Control 24*(8), 1179–1232.

Dabney, W., M. Rowland, M. Bellemare, and R. Munos (2018). Distributional reinforce-

ment learning with quantile regression. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 32.

de Castro, L. and A. F. Galvao (2019). Dynamic quantile models of rational behavior. *Econometrica 87*(6), 1893–1939.

de Castro, L. and A. F. Galvao (2022). Static and dynamic quantile preferences. *Econ Theory 73*, 747–779.

de Castro, L., A. F. Galvao, and A. Muchon (2023). Numerical Solution of Dynamic Quantile Models. *Journal of Economic Dynamics and Control 148*(C).

de Castro, L., A. F. Galvao, C. N. Noussair, and L. Qiao (2022). Do people maximize quantiles? *Games and Economic Behavior 132*, 22–40.

de Castro, L. I., A. F. Galvao, and D. d. S. Nunes (2025, January). Dynamic economics with quantile preferences. *Theoretical Economics 20*(1).

Den Haan, W. J. (2010). Comparison of solutions to the incomplete markets model with aggregate uncertainty. *Journal of Economic Dynamics and Control 34*(1), 4–27. Computational Suite of Models with Heterogeneous Agents: Incomplete Markets and Aggregate Uncertainty.

Gaspat, J. and K. L. Judd (1997). Solving large-scale rational-expectations models. *Macroeconomic Dynamics 1*(1), 45–75.

Giovannetti, B. (2013). Asset pricing under quantile utility maximization. *Review of Financial Economics 22*(4), 169–179.

He, X. D., Z. Jiang, and S. Kou (2021). Portfolio selection under median and quantile maximization.

He, Z. L. (2023). A gradient-based reinforcement learning model of market equilibration. *Journal of Economic Dynamics and Control 152*, 104670.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks 4*(2), 251–257.

Jiang, J., Y. Peng, and J. Hu (2022). Quantile-based policy optimization for reinforcement learning. In *2022 Winter Simulation Conference (WSC)*, pp. 2712–2723. IEEE.

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press.

Kollmann, R., S. Maliar, B. A. Malin, and P. Pichler (2011). Comparison of solutions to the multi-country real business cycle model. *Journal of Economic Dynamics and Control 35*(2),

186–202. Computational Suite of Models with Heterogeneous Agents II: Multi-Country Real Business Cycle Models.

Long, Y., J. Sethuraman, and J. Xue (2021). Equal-quantile rules in resource allocation with uncertain needs. *Journal of Economic Theory 197*, 105350.

Maliar, L. and S. Maliar (2014). Chapter 7 - numerical methods for large-scale dynamic economic models. In K. Schmedders and K. L. Judd (Eds.), *Handbook of Computational Economics Vol. 3*, Volume 3 of *Handbook of Computational Economics*, pp. 325–477. Elsevier.

Manski, C. F. (1988). Ordinal utility models of decision making under uncertainty. *Theory and Decision 25*, 79–104.

Miao, J. (2013). Economic dynamics: Discrete time. *MIT Press*.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis (2015). Human-level control through deep reinforcement learning. *Nature 518*(7540), 529–533.

Rostek, M. (2010). Quantile maximization in decision theory. *The Review of Economic Studies 77*(1), 339–371.

Rust, J. (1996). Chapter 14 numerical dynamic programming in economics. Volume 1 of *Handbook of Computational Economics*, pp. 619–729. Elsevier.

Rust, J. (2016). *Dynamic Programming*, pp. 1–26. London: Palgrave Macmillan UK.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement Learning: An Introduction*. The MIT Press.

Tahvonen, O., A. Suominen, P. Malo, L. Viitasaari, and V.-P. Parkatti (2022). Optimizing high-dimensional stochastic forestry via reinforcement learning. *Journal of Economic Dynamics and Control 145*, 104553.

Taylor, J. B. and H. Uhlig (1990). Solving nonlinear stochastic growth models: A comparison of alternative solution methods. *Journal of Business & Economic Statistics 8*(1), 1–17.

Wu, B. and L. Li (2024). Reinforcement learning for continuous-time mean-variance portfolio selection in a regime-switching market. *Journal of Economic Dynamics and Control 158*, 104787.

Zhou, X., S. Lin, and X.-Z. He (2025). Reinforcement learning and rational expectations equilibrium in limit order markets. *Journal of Economic Dynamics and Control 172*, 104991.

IES Working Paper Series

2025

1. Kseniya Bortnikova, Josef Bajzik, Evzen Kocenda: *How Do Event Studies Capture Impact of Macroeconomic News in Forex Market? A meta-Analysis*
2. Zuzana Meteláková, Adam Geršl: *Does Bank Regulation and Supervision Impact Income Inequality? Cross-Country Evidence*
3. Tersoo David Iorngurum: *Inflation Differentials in the African Economic Community*
4. Lorena Skufi, Adam Gersl: *Does FX Hedge Mitigate the Impact of Exchange Rate Changes on Credit Risk? Evidence from a Small Open Economy*
5. Meri Papavangjeli: *From Skies to Markets Implications of Extreme Weather Events for Macroeconomic and Financial Imbalances in CESEE Countries*
6. Matej Opatrny, Milan Scasny: *Bridging the Gap: A Novel M2/LIHC Hybrid Indicator Unveils Energy Poverty Dynamics - Case Study of the Czech Republic*
7. Vojtěch Mišák: *Temperature and Productivity in Soccer*
8. Klara Kantova, Tomas Havranek, Zuzana Irsova: *The Elasticity of Substitution between Native and Immigrant Labor: A Meta-Analysis*
9. Suren Karapetyan, Matej Bajgar: *Unicorn Exits and Subsequent Venture Capital Investments*
10. Jonáš Čekal, Adam Geršl: *The Effects of Crisis Management Measures on the Economy: Evidence from Past Crises*
11. *Samuel Fiifi Eshun, Evzen Kocenda, Princewill Okwoche, Milan Ščasný: Price and Income Elasticities of Industrial Energy Demand in New EU Member States*
12. Lukáš Janásek: *Gradient-Based Reinforcement Learning for Dynamic Quantile*

All papers can be downloaded at: http://ies.fsv.cuni.cz

Univerzita Karlova v Praze, Fakulta sociálních věd
Institut ekonomických studií [UK FSV – IES] Praha 1, Opletalova 26
E-mail : ies@fsv.cuni.cz http://ies.fsv.cuni.cz