FFA Working Papers

Natural Gas Storage Valuation Using Deep Reinforcement Learning

Masood Tadi, Milan Fičura, and Jiří Witzany

FFA Working Paper 3/2026

FACULTY OF FINANCE AND ACCOUNTING

Bibliographic information:

Tadi M., Fičura M., and Witzany J. (2026). Natural Gas Storage Valuation Using Deep Reinforcement Learning. FFA Working Paper 3/2026, FFA, Prague University of Economics and Business, Prague.

This paper can be downloaded at: wp.ffu.vse.cz

Contact e-mail: ffawp@vse.cz

Natural Gas Storage Valuation Using Deep Reinforcement Learning

Masood Tadi*[1,2], Milan Fičura[2], and Jiří Witzany[2]

[1]Department of Probability and Mathematical Statistics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
[2]Faculty of Finance and Accounting, Prague University of Economics and Business, Prague, Czech Republic

Abstract

We study natural gas storage valuation under a stochastic futures term structure using deep reinforcement learning (DRL). The storage problem is formulated as a continuous-state, continuous-action Markov Decision Process and solved using the Deep Deterministic Policy Gradient (DDPG) algorithm with Prioritized Experience Replay (PER) buffer and a constraint-aware policy network. We benchmark the approach against intrinsic and rolling intrinsic strategies and find that DRL consistently outperforms intrinsic valuation and achieves competitive performance relative to rolling intrinsic in markets with jumps and seasonality. The results show that DRL provides a practical valuation framework that captures additional extrinsic value under realistic market dynamics and operational constraints.

Keywords: Natural Gas Storage, Rolling Intrinsic Valuation, Deep Reinforcement Learning.

1 Introduction

Natural gas storage plays a key role in balancing seasonal fluctuations between supply and demand. Prices tend to be lower in summer and higher in winter, creating intertemporal trading opportunities by purchasing gas when prices are low and selling when prices are high, allowing operators to smooth spreads, hedge exposure, and enhance profitability (Honoré 2011). Consequently, natural gas storage valuation is a key problem in energy market arbitrage, hedging, and investment.

Commodity price models are typically based on mean-reverting processes with seasonality and jumps. The foundational work of Gibson and E. S. Schwartz (1990) introduced spot price and convenience yield, extended by multi-factor models in E. S. Schwartz (1997) and long-term/short-term decomposition in E. Schwartz and Smith (2000). Later developments incorporated stochastic volatility, jumps, and calibration techniques (Yan 2002; Peters et al. 2013). These motivate viewing gas storage as a real option, whose value arises from the flexibility to inject or withdraw gas under price uncertainty.

Classical valuation methods such as dynamic programming and Monte Carlo simulation can solve the stochastic control problem of optimal gas storage operation in theory but suffer from the curse of dimensionality (Thompson, Davison, and Rasmussen 2009; Felix 2012; Secomandi 2010; Lai, Margot, and Secomandi 2010). This has motivated machine learning frameworks capable of learning optimal policies from simulated or historical data (Bachouch et al. 2022; Curin et al. 2021).

The basic industry practice relies on intrinsic valuation, which exploits static arbitrage by selecting futures positions that maximize spread (Breslin et al. 2009; Felix 2012). It serves as a conservative benchmark (Löhndorf and Wozabal 2021; De Jong 2015) but ignores price path flexibility. Rolling intrinsic improves intrinsic valuation via periodic re-optimization (Breslin et al. 2009; Felix 2012) and guarantees non-decreasing value (Breslin et al. 2009) but may fail to identify intertemporal trade-offs (Felix 2012). Storage is also modeled as a basket of spread options (Breslin et al. 2009) based on swing-option theory Jaillet, Ronn, and Tompaidis (2004), providing quick approximations of extrinsic value (Löhndorf and Wozabal 2021; De Jong 2015).

*Corresponding author

Capturing full extrinsic value requires solving a stochastic control problem Geman (2005). Early approaches applied finite-difference methods and trees Felix (2012). Least-Squares Monte Carlo (LSMC) Longstaff and E. S. Schwartz (2001) was adapted by Boogert and De Jong (2006) and Jaillet, Ronn, and Tompaidis (2004) and refined in Warin (2012) and Malyscheff and Trafalis (2017). Approximate dynamic programming was introduced by Secomandi (2010), with improved performance when embedded in simulation (Lai, Margot, and Secomandi 2010). Further studies include Thompson, Davison, and Rasmussen (2009), Carmona and Ludkovski (2010), and Bjerksund, Stensland, and Vagstad (2011).

Reinforcement learning (RL) offers an alternative for high-dimensional control. Deep RL algorithms are increasingly applied in finance (Fischer 2018) and energy (Giannelos 2025), while battery storage studies show RL can learn arbitrage without specifying price models Sage, Campbell, and Y. F. Zhao (2024). In gas storage, Curin et al. (2021) introduce deep hedging–inspired models (Buehler et al. 2019), Bachouch et al. (2022) develop neural stochastic control, and Daluiso et al. (2020) apply DRL to swing options. Market-level modeling is explored by Bacaloni, Glielmo, and Taboga (2025), whereas this paper treats the operator as a price taker under an exogenous term structure. Beyond gas, DRL is applied in microgrids Jung et al. (2025) and compressor optimization X. Zhao et al. (2025).

This paper proposes a deep reinforcement learning (DRL) framework for gas storage valuation using a four-factor jump–diffusion model with seasonality following Yan (2002). The problem is formulated as a continuous-state, continuous-action Markov decision process and solved via a Deep Deterministic Policy Gradient (DDPG) algorithm with Prioritized Experience Replay (PER) and a constraint-aware autoregressive policy network.

Our contributions are: (1) a constraint-aware DRL framework enforcing feasibility without penalization methods, and (2) benchmarking against intrinsic and rolling intrinsic strategies under realistic futures dynamics.

2 Natural Gas Term-Structure Dynamics

We employ the four-factor Yan model (ibid.) with added seasonality factors under the risk-neutral measure.

1) The spot price S_t follows the jump-diffusion SDE

$$dS_t = (r_t - \delta_t - \lambda\mu_J)S_t\,dt + \sigma_s S_t\,dW_1 + \sqrt{v_t}S_t\,dW_2 + JS_t\,dq, \tag{1}$$

where r_t is the instantaneous risk-free rate, δ_t the convenience yield, λ the jump intensity, μ_J the average jump size, σ_s the base volatility, v_t the stochastic variance, W_1, W_2 Brownian motions, and dq a Poisson process with

$$\mathbb{P}(dq = 1) = \lambda dt, \tag{2}$$

and jump size

$$\ln(1+J) \sim \mathcal{N}\big(\ln(1+\mu_J) - \tfrac{1}{2}\sigma_J^2,\, \sigma_J^2\big)\,. \tag{3}$$

2) The short rate r_t follows a CIR process:

$$dr_t = (\theta_r - \kappa_r r_t)\,dt + \sigma_r\sqrt{r_t}\,dW_r. \tag{4}$$

3) The convenience yield δ_t follows an OU process:

$$d\delta_t = (\theta_\delta - \kappa_\delta\delta_t)\,dt + \sigma_\delta dW_\delta, \tag{5}$$

with correlation

$$dW_\delta dW_1 = \rho_1\,dt. \tag{6}$$

4) The variance v_t follows a jump-diffusion square-root process:

$$dv_t = (\theta_v - \kappa_v v_t)\,dt + \sigma_v\sqrt{v_t}\,dW_v + J_v dq, \tag{7}$$

where $J_v \sim \text{Exponential}(\theta)$ and

$$dW_v dW_2 = \rho_2\,dt. \tag{8}$$

Under the risk-neutral measure, futures prices $F(t,\tau)$ with maturity τ are

$$F(t,\tau) = \exp[\ln(S_t) + f_t + \beta_0(\tau) + \beta_r(\tau) r_t + \beta_\delta(\tau)\delta_t] \tag{9}$$

where f_t is a deterministic seasonality component and

$$\xi_r = \sqrt{\kappa_r^2 + 2\sigma_r^2}, \tag{10}$$

$$\beta_r(\tau) = \frac{2(1 - e^{-\xi_r \tau})}{2\xi_r - (\xi_r - \kappa_r)(1 - e^{-\xi_r \tau})}, \tag{11}$$

$$\beta_\delta(\tau) = \frac{-(1 - e^{-\kappa_\delta \tau})}{\kappa_\delta}, \tag{12}$$

and the drift term $\beta_0(\tau)$ is equal to

$$\begin{aligned} \beta_0(\tau) = {} & \frac{\theta_r}{\sigma_r^2}\left[2\ln\left(1 - \frac{(\xi_r - \kappa_r)(1 - e^{-\xi_r \tau})}{2\xi_r}\right) + (\xi_r - \kappa_r)\tau\right] \\ & + \frac{\sigma_\delta^2 \rho_1 \beta_\delta}{\kappa_\delta}\tau - \frac{(\sigma_s \sigma_\delta \rho_1 + \theta_\delta)}{\kappa_\delta^2}(1 - e^{-\kappa_\delta \tau}) \\ & + \frac{4\sigma_\delta^2 e^{-\kappa_\delta \tau} - \sigma_\delta^2 e^{-2\kappa_\delta \tau}}{4\kappa_\delta^3} + \frac{\sigma_s \sigma_\delta \rho_1 + \theta_\delta}{2\kappa_\delta^2} - \frac{3\theta_\delta^2}{4\kappa_\delta^3}. \end{aligned} \tag{13}$$

This closed-form expression allows direct calibration to market futures curves and efficient simulation. The model parameters are estimated from observed futures prices. Let $F(0,\tau;\Theta)$ denote the model futures price at $t = 0$ with maturity τ and parameter vector

$$\Theta = [S_0, \delta_0, r_0, \sigma_s, \sigma_r, \theta_r, \kappa_r, \sigma_\delta, \theta_\delta, \kappa_\delta, \lambda, \mu_J, \sigma_J, \rho_1, \rho_2, f_t]. \tag{14}$$

For observed futures prices $F^*(0,\tau)$ and maturities $\{\tau_j\}_{j=1}^n$, calibration solves

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{j=1}^{n} \big[F(0,\tau_j;\Theta) - F^*(0,\tau_j)\big]^2. \tag{15}$$

This nonlinear least-squares problem is typically solved by the Levenberg–Marquardt algorithm. Figure 1 illustrates the calibrated fit to TTF futures as of 31/12/2020.

3 Intrinsic and Rolling Intrinsic Valuation Methods

Natural gas storage valuation arises from solving an optimal control problem under uncertainty: at each point in time, the operator decides whether to inject gas (incurring a cost) or withdraw it (realizing profit), subject to physical and market constraints. We focus on two valuation techniques: intrinsic valuation, and rolling intrinsic valuation.

3.1 Intrinsic Valuation

Let $\mathbf{F} = [F_1, \ldots, F_n]^\top$ be the vector of monthly futures prices and $\mathbf{X} = [X_1, \ldots, X_n]^\top \in \mathbb{R}^n$ the vector of actions, with $X_i > 0$ (injection) and $X_i < 0$ (withdrawal). The intrinsic valuation problem chooses

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}}\big(-\mathbf{F}^\top \mathbf{X}\big) = \arg\min_{\mathbf{X}}\big(\mathbf{F}^\top \mathbf{X}\big), \tag{16}$$

subject to storage constraints (injection/withdrawal limits, final balance, and storage bounds):

$$\mathbf{A}\mathbf{X} \le \mathbf{b}, \quad \mathbf{A_e}\mathbf{X} = \mathbf{b_e}, \quad \mathbf{l} \le \mathbf{X} \le \mathbf{u}, \tag{17}$$

where

$$\mathbf{u} = I_{\max}\mathbf{1}_n, \quad \mathbf{l} = -W_{\max}\mathbf{1}_n, \quad \mathbf{A_e} = \mathbf{1}_n^\top, \quad \mathbf{b_e} = -V_0, \tag{18}$$

and

$$\mathbf{A} = \begin{bmatrix} \mathbf{L}_n \\ -\mathbf{L}_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} (V_{\max} - V_0)\mathbf{1}_n \\ -(V_{\min} - V_0)\mathbf{1}_n \end{bmatrix}, \tag{19}$$

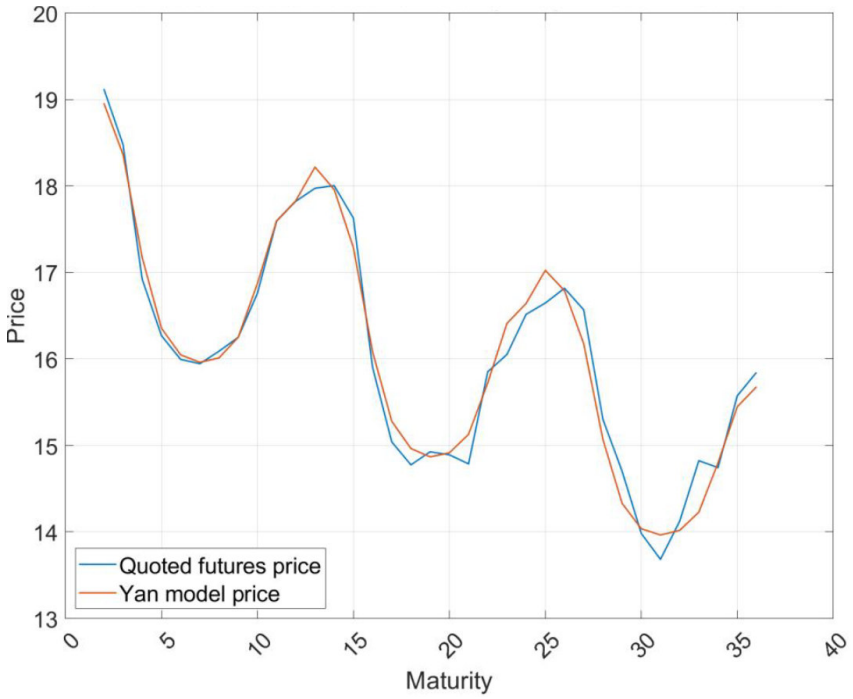

Figure 1: TTF futures term structure vs. Yan model fit – 31/12/2020

with $\mathbf{L}_n$ the $n \times n$ lower triangular matrix of ones. These constraints ensure

$$V_{\min} \leq V_0 + \sum_{k=1}^{i} X_k \leq V_{\max}, \quad i = 1, \ldots, n. \tag{20}$$

We assume monthly futures $n = 12$, constant bounds $V_{\max}, V_{\min}$, initial level $V_0 = 0$, final level $V_n = 0$, and fixed rates $I_{\max}, W_{\max}$.

3.2 Rolling Intrinsic Valuation (RIV)

The RIV improves upon intrinsic valuation by introducing price uncertainty and re-optimization. Instead of relying only on today's futures curve, RIV re-evaluates the intrinsic strategy over simulated futures term structures as they evolve over time. The RIV method proceeds through the following steps:

1. **Simulate futures term structures.** Simulate N independent scenarios. Each simulation j yields

$$\mathbf{F}_t^{(j)} = [F_{t,1}^{(j)}, \ldots, F_{t,n}^{(j)}], \quad t = 0, \ldots, T,$$

   with $\mathbf{F}_t^{(j)}$ shrinking as contracts expire.

2. **Initial intrinsic valuation.** At $t = 0 = \tau_0$, for each j solve

$$\hat{\mathbf{X}}_{\tau_0}^{(j)} = \arg\min_{\mathbf{X}} {\mathbf{F}_{\tau_0}^{(j)}}^{\top} \mathbf{X}$$

   subject to the intrinsic constraints (with V_0). The intrinsic value is

$$V_{I,\tau_0}^{(j)} = -{\mathbf{F}_{\tau_0}^{(j)}}^{\top} \hat{\mathbf{X}}_{\tau_0}^{(j)}.$$

3. **Rolling re-optimization.** For decision times $\tau_1, \ldots, \tau_n$ and each j:

   (a) Futures cash-flow:

$$CF_{F,\tau_i}^{(j)} = \left(\mathbf{F}_{\tau_i}^{(j)} - \mathbf{F}_{\tau_{i-1}}^{(j)}\right)^{\top} \hat{\mathbf{X}}_{\tau_{i-1}}^{(j)}.$$

(b) Updated intrinsic solution: solve again

$$\hat{\mathbf{X}}_{\tau_i}^{(j)} = \arg\min_{\mathbf{X}} \mathbf{F}_{\tau_i}^{(j)} \cdot \mathbf{X}$$

with the same constraints but with current storage level $V_{\tau_i}^{(j)}$ replacing V_0. Set

$$V_{I,\tau_i}^{(j)} = -\mathbf{F}_{\tau_i}^{(j)\top} \hat{\mathbf{X}}_{\tau_i}^{(j)}.$$

(c) Spot cash-flow:

$$CF_{S,\tau_i}^{(j)} = -F_{\tau_i,1}^{(j)} \hat{X}_{\tau_i,1}^{(j)} = -S_{\tau_i}^{(j)} \hat{X}_{\tau_i,1}^{(j)}.$$

(d) Total cash-flow:

$$CF_{\tau_i}^{(j)} = CF_{F,\tau_i}^{(j)} + CF_{S,\tau_i}^{(j)}.$$

4. **Value per simulation.**

$$RIV^{(j)} = \sum_{i=1}^{n} CF_{\tau_i}^{(j)}.$$

5. **Expected value.**

$$RIV = \frac{1}{N} \sum_{j=1}^{N} RIV^{(j)}.$$

4 Deep Reinforcement Learning (DRL) Method

Deep Reinforcement Learning (DRL) refers to a class of methods that integrate reinforcement learning with deep neural networks to approximate key components of the learning process. Depending on the algorithm, DRL may approximate value functions, policies, or both, enabling scalability to high-dimensional or continuous state and action spaces where traditional tabular methods are computationally infeasible.

A common approach in DRL is to approximate the action-value function $Q^\pi(s, a)$ using a parameterized neural network $Q(s, a; \phi)$, where $\phi \in \mathbb{R}^n$ represents the trainable parameters:

$$Q^\pi(s, a) \approx Q(s, a; \phi).$$

The parameters ϕ are typically optimized by minimizing the Temporal Difference (TD) error over observed transitions,

$$\mathcal{L}(\phi) = \mathbb{E}_{(s,a,r,s')} \left[\left(Q(s, a; \phi) - \left(r + \gamma \max_{a'} Q(s', a'; \phi^{\text{targ}}) \right) \right)^2 \right], \tag{21}$$

where ϕ^{targ} denotes the parameters of a target network used to stabilize training. Alternatively, policy gradient methods directly parameterize the policy $\pi_\theta(a|s)$ and optimize the expected return using gradient ascent. This approach follows from the stochastic policy gradient theorem (Sutton et al. 1999) and is especially effective in environments with continuous or high-dimensional action spaces, such as natural gas storage valuation. Finally, the actor-critic methods combine both approaches: an actor, representing the policy $\pi_\theta(a \mid s)$ or deterministic mapping $\mu_\theta(s)$; and a *critic*, which estimates a value function. The critic is trained using TD learning, and its estimates are used to guide the policy updates of the actor.

In this study, we adopt an actor–critic framework for natural gas storage valuation, specifically the Deep Deterministic Policy Gradient (DDPG) algorithm proposed by Lillicrap et al. (2015), which builds on the Deterministic Policy Gradient theorem developed by Silver et al. (2014). DDPG is well-suited for continuous control tasks due to its ability to learn deterministic policies in high-dimensional, continuous action spaces. It employs an actor network $\mu_\theta(s)$ to represent the deterministic policy, and a critic network $Q_\phi(s, a)$ to approximate the action-value function, with the critic updated via temporal difference learning and the actor updated via the deterministic policy gradient.

Input: initialize policy parameters θ and Q-function parameters ϕ, empty replay buffer $\mathcal{D}$
Set target parameters: $\theta_{\text{targ}} \leftarrow \theta$, $\phi_{\text{targ}} \leftarrow \phi$
repeat
  Reset the environment and observe initial state s
  for each step in episode **do**
    Select action $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{\text{Low}}, a_{\text{High}})$, $\epsilon \sim \mathcal{N}$
    Execute a, observe next state s', reward r, and terminal flag d
    Store (s, a, r, s', d) in $\mathcal{D}$
    if d **then**
      break
    end if
    $s \leftarrow s'$
    if $|\mathcal{D}|$ is sufficiently large **then**
      Sample batch $\{(s^{(i)}, a^{(i)}, r^{(i)}, s'^{(i)}, d^{(i)})\}_{i=1}^{B}$
      Compute targets:

$$y^{(i)} = r^{(i)} + \gamma(1 - d^{(i)}) Q_{\phi_{\text{targ}}}\Big(s'^{(i)}, \mu_{\theta_{\text{targ}}}(s'^{(i)})\Big)$$

      Update critic by minimizing:

$$\frac{1}{B}\sum_{i=1}^{B} \left(Q_\phi(s^{(i)}, a^{(i)}) - y^{(i)}\right)^2$$

      Update actor via gradient ascent:

$$\nabla_\theta \frac{1}{B}\sum_{i=1}^{B} Q_\phi\left(s^{(i)}, \mu_\theta(s^{(i)})\right)$$

      Update targets:

$$\theta_{\text{targ}} \leftarrow \rho\theta_{\text{targ}} + (1-\rho)\theta, \qquad \phi_{\text{targ}} \leftarrow \rho\phi_{\text{targ}} + (1-\rho)\phi$$

    end if
  end for
until convergence

Table 1: Deep Deterministic Policy Gradient (DDPG) Algorithm

Table 1 outlines the DDPG algorithm, where the Q-network parameters ϕ are updated via gradient descent and the policy parameters θ via gradient ascent. The parameter updates can be carried out using various optimization methods. A widely used method is the Adam optimizer introduced by Kingma (2014), which combines the benefits of momentum (Polyak 1964) and RMSProp (Hinton 2012), providing adaptive learning rates and faster convergence, and making it a popular choice for deep reinforcement learning. In our implementation of DDPG, we replace the standard uniform experience replay buffer with Prioritized Experience Replay (PER) for training the critic network. PER, introduced by Schaul (2015), prioritizes transitions based on their temporal-difference (TD) error, replaying more important experiences more frequently. In standard experience replay, transitions (s, a, r, s') are sampled uniformly from the replay buffer $\mathcal{D}$, whereas PER assigns higher sampling probabilities to transitions with higher TD error. For a transition (s_j, a_j, r_j, s'_j), the TD error in DDPG is

$$\delta_j = \left| r_j + \gamma\, Q(s'_j, \mu_{\theta^{\text{targ}}}(s'_j); \phi^{\text{targ}}) - Q(s_j, a_j; \phi) \right|, \tag{22}$$

and importance sampling weights are applied to correct the bias introduced by prioritized sampling during training. For further technical details, see Schaul (ibid.).

5 DRL for Natural Gas Storage Valuation: Model Setup

This section applies the DRL framework to natural gas storage valuation. We define the state and action spaces, the reward function, and the training setup used in the experiments.

5.1 State and Action Spaces

The state space $\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^{14}$ comprises the following components:

$$\mathbf{s} = \left(t, F_t^{(1)}, \ldots, F_t^{(12)}, V_t\right), \tag{23}$$

where $t \in \{0, \ldots, 12\}$, $F_t^{(k)}$ are futures prices generated from the Yan model using (9), and $V_t \in [V_{\min}, V_{\max}]$ is the inventory level. Expired contracts are set to zero: $F_t^{(k)} = 0$ for $k < t$.

Hyperparameters are listed in Tables 2 and 3. Figure 2 illustrates a sample simulated term structure.

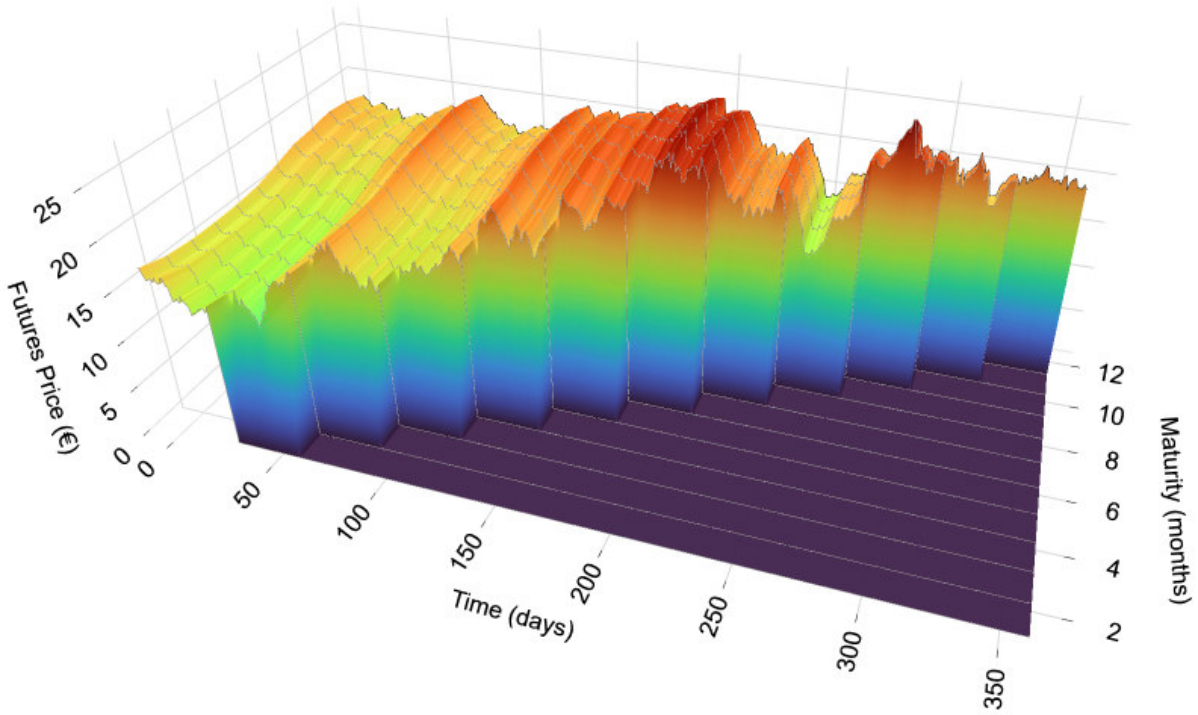

Figure 2: Simulated futures term structure for a single Monte Carlo path (front side)

Group	**Parameter**	**Value**
Storage	months n	12
	V_0, V_T	0.0, 0.0
	$V_{\min}$, $V_{\max}$	0.0, 1.0
	$I_{\max}$, $W_{\max}$	0.4, 0.4
Short rate r_t	θ_r	0.000588
	κ_r	0.492828
	σ_r	0.655899
	r_0	0.159586
Convenience δ_t	θ_δ	-0.213184
	κ_δ	1.177232
	σ_δ	1.036639
	δ_0	0.106417
Stochastic variance v_t	θ_v	0.050569
	κ_v	2.363092
	σ_v	0.825941
	v_0	0.024997
Spot factor	σ_S	0.791066
	S_0	19.065873
Jumps	λ	0.638842
	μ_J	0.013715
	σ_J	0.032046
	θ	0.006407
Correlations	$\rho_1 = \mathrm{Corr}(dW_1, dW_\delta)$	0.899944
	$\rho_2 = \mathrm{Corr}(dW_2, dW_v)$	-0.306811
Seasonality factors	$f_1, f_2, \ldots, f_{12}$	(see Table 3)

Table 2: Environment hyperparameters used in simulation

Month (k)	**Seasonality factor** (f_k)
1	−0.106616825
2	−0.152361004
3	−0.167724706
4	−0.167979840
5	−0.159526180
6	−0.139279435
7	−0.095340299
8	−0.047464680
9	−0.027862228
10	0.000000000
11	−0.008502635
12	−0.040963872

Table 3: Monthly seasonality factors

The action $\mathbf{a} \in \mathcal{A} \subset \mathbb{R}^{12}$ corresponds to a vector of injection or withdrawal decisions for each of the 12 remaining months:

$$\mathbf{a} = \mathbf{X_t} = (X_t^{(1)}, \ldots, X_t^{(12)}), \tag{24}$$

where

$$X_t^{(1)} \in [0,\, I_{\max}], \qquad X_t^{(i)} \in [-W_{\max},\, I_{\max}],\ i = 2, \ldots, 11, \qquad X_t^{(12)} \in [-W_{\max},\, 0]. \tag{25}$$

Each component $X_t^{(i)}$ reflects the flow in month i. The terminal balance constraint and cumulative storage bounds are not imposed directly at the action-space level, but are handled separately. Common approaches for handling these constraints include incorporating penalty terms into the reward function or projecting actions onto the feasible region through an optimization layer. However, penalty-based methods require careful tuning and often fail to produce feasible trajectories, while projection layers introduce substantial computational overhead.

To address these limitations, we enforce feasibility directly within the policy architecture using a constraint-aware parameterization. The policy network generates actions sequentially, conditioning each decision on the current state and the running inventory. This autoregressive (AR) design enforces all operational constraints by construction, remains end-to-end differentiable, and avoids both penalty tuning and computationally expensive projections.

The reward at each decision time t is defined as:

$$r_t = \begin{cases} (\mathbf{F_t} - \mathbf{F_{t-1}})^\top \mathbf{X_{t-1}}, & t = 1, \ldots, 11, \\ (\mathbf{F_t} - \mathbf{F_{t-1}})^\top \mathbf{X_{t-1}} \ - \ \mathbf{F_t}^\top \mathbf{X_t}, & t = 12. \end{cases} \tag{26}$$

The spot settlement is embedded since expired contracts transition to zero. This is mathematically equivalent to the rolling intrinsic cash-flow decomposition.

5.2 Architecture and Training

We use an actor–critic DDPG agent. Aside from the actor's autoregressive output layer, both networks are multilayer perceptrons (MLP) with four hidden layers $(512, 512, 256, 128)$, LeakyReLU activations, and layer normalization. The critic concatenates the action at the first hidden layer and outputs a scalar Q-value. Optimization uses Adam (Kingma 2014) with learning rates $\alpha_\pi = 3 \times 10^{-5}$, $\alpha_\phi = 5 \times 10^{-4}$ and gradient clipping at 1. Target networks are updated via Polyak averaging ($\tau = 0.0005$). A prioritized experience replay buffer (2×10^5 capacity, batch size 256; $\alpha = 0.6$, $\beta_0 = 0.1$ with annealing) is used.

Training begins after a warm-up phase of 200 batches, after which the algorithm alternates between critic updates, actor updates, and target network updates at each step. Exploration noise decays from 0.2 to 0.01 over 90% of 20,000 episodes per seed (five seeds). Evaluation is conducted without exploration noise. Hyperparameters are summarized in Table 4.

Component	Setting
Actor	4-layer MLP with hidden sizes $(512, 512, 256, 128)$; LayerNorm; LeakyReLU; i-th head conditions on C_{i-1} and squashes into $[\ell_i, u_i]$.
Critic	4-layer MLP with hidden sizes $(512, 512, 256, 128)$; LayerNorm on input; LeakyReLU.
Optimizers	Adam; actor LR $\alpha_\pi = 3 \times 10^{-5}$, critic LR $\alpha_\phi = 5 \times 10^{-4}$; gradient-norm clip $= 1$.
Targets & discount	Polyak averaging with $\tau = 0.005$; target updated every step; $\gamma = 0.99$.
Replay (PER)	Capacity 2×10^5; batch size 256; priority $\alpha = 0.6$; bias-correction $\beta_0 = 0.1$ with annealing (β rate $= 0.99992$).
Warm-up	$N_{\text{warm}} = 200$ batches (51,200 samples).
Exploration	Gaussian action noise; ratio decays $0.20 \to \approx 0.01$; last 10% episodes noise-free.
Episodes	20,000 per seed; across 5 seeds.
Evaluation	Greedy actor; $N = 100$ price paths; monthly decisions; one-year horizon.
Implementation	Last action clamp $[\texttt{env\_min}[-1], \texttt{env\_max}[-1]]$ with $\texttt{env\_max}[-1] = 0$.

Table 4: DRL agent hyperparameters and training protocol.

6 Empirical Results

Table 5 reports the summary statistics of simulated futures prices (mean €17.28; range €2.02–€68.84). Price dispersion increases with maturity since longer-dated contracts are observed over longer horizons and thus exhibit greater realized variability. Figure 3 shows the moving-average return during training

Maturity	Mean Price	Price Dispersion	Min Price	Max Price
M1	17.4262	2.6543	8.5299	29.2576
M2	16.4007	3.3063	7.3494	34.2528
M3	16.1047	3.6677	6.5271	38.5644
M4	15.9885	3.9307	5.0748	39.4628
M5	16.0067	4.1391	5.0999	49.1304
M6	16.2235	4.4687	4.2344	47.4047
M7	16.8290	4.9752	4.6584	55.7409
M8	17.5590	5.4007	4.5678	68.8405
M9	17.8656	5.6278	2.4381	63.8866
M10	18.2767	5.9182	2.0196	64.3367
M11	18.0205	5.8657	2.2569	58.9753
M12	17.3435	5.7266	2.4285	62.5863
Overall	**17.2813**	**5.3071**	**2.0196**	**68.8405**

Table 5: Summary statistics of simulated natural gas futures price levels by maturity (in euros)

and evaluation across different seeds. A sharp performance increase occurs at approximately episode 5,000, coinciding with the end of the replay-buffer warm-up phase and the onset of learning. Thereafter, both curves stabilize and fluctuate within a bounded range. Evaluation performance remains consistently above training performance due to the absence of exploration noise during evaluation. While strict convergence cannot be asserted from reward curves alone, the sustained plateau and bounded variability indicate that the learning dynamics reach a stable regime with no evidence of training collapse or performance degradation. The rapid improvement following the warm-up phase further suggests efficient utilization of prioritized experience replay and the autoregressive constraint-aware policy structure.

Figure 3: Moving-average episode returns during training (top) and evaluation (bottom).

After training, the learned policy is evaluated on out-of-sample scenarios by simulating independent one-year price paths under the Yan model. For each path, we compare:

1. the realized P&L generated by the trained DRL policy, and
2. the rolling intrinsic (RI) benchmark computed via repeated linear programming.

Figure 4 reports the pathwise realized P&L distributions for the DRL policy and the rolling intrinsic benchmark. The DRL agent achieves a higher average realized value across simulated paths than both the intrinsic and rolling intrinsic strategies. While rolling intrinsic responds to changes in the forward curve through repeated re-optimization, it does not condition decisions on the full simulated state in a forward-looking manner. In contrast, the DRL policy directly optimizes cumulative returns through interaction with the stochastic environment.

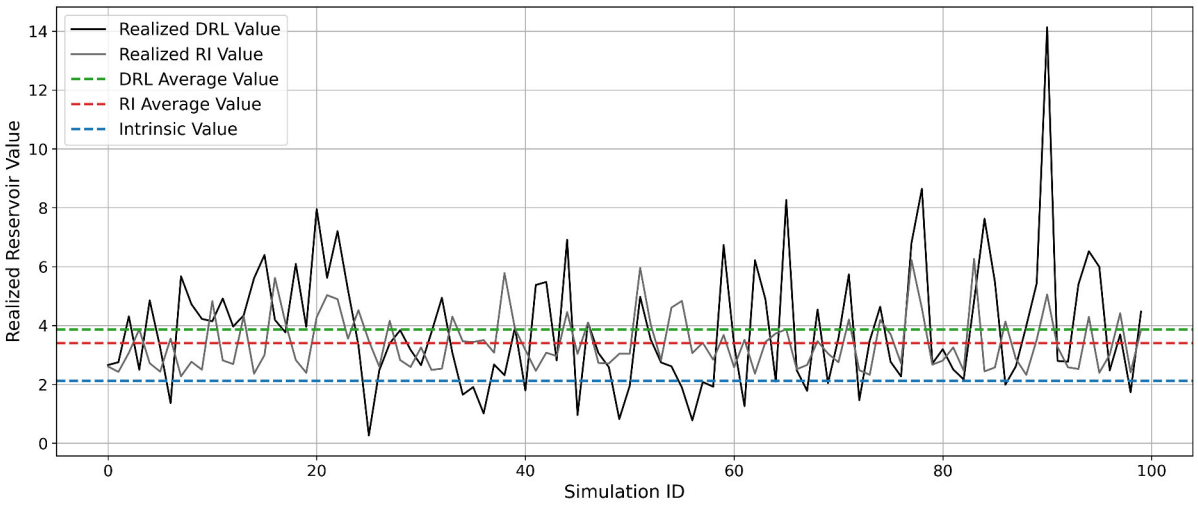

Figure 4: DRL policy value versus intrinsic and rolling intrinsic values.

Table 6 reports average performance across simulations. The DRL policy achieves a mean value of 3.9, compared with 3.4 for rolling intrinsic and 2.1 for intrinsic, representing improvements of approximately 86% and 15%, respectively. Although outcome variability is higher, this reflects optimization for expected return rather than variance and indicates effective exploitation of payoff convexity under jumps and stochastic volatility. Overall, DRL emerges as a competitive alternative to heuristic strategies in highly stochastic settings.

Method	**Intrinsic Value**	**RI Value**	**DRL Value**
Average Result (Standard Error)	2.1 (0.0)	3.4 (0.1)	3.9 (0.2)

Table 6: Average Performance Comparison

7 Conclusion

This paper presents a deep reinforcement learning (DRL) framework for valuing natural gas storage under a multi-factor stochastic futures model. By formulating the problem as a continuous-state, continuous-action MDP and applying DDPG with prioritized replay and a constraint-aware autoregressive policy, the proposed approach learns feasible and profit-maximizing strategies directly from simulated markets.

Empirically, the DRL policy outperforms intrinsic and rolling intrinsic benchmarks on average, while exhibiting higher dispersion consistent with an objective that prioritizes expected return over variance. In contrast to intrinsic lower bounds and limited re-optimization under rolling intrinsic, the DRL policy exploits both intertemporal flexibility and the evolving term structure, supporting reinforcement learning as a viable alternative to traditional valuation methods.

References

Bacaloni, Tiziano, Aldo Glielmo, and Marco Taboga (2025). "Natural-gas storage modelling by deep reinforcement learning". In: *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 829–837.

Bachouch, Achref et al. (2022). "Deep neural networks algorithms for stochastic control problems on finite horizon: numerical applications". In: *Methodology and Computing in Applied Probability* 24.1, pp. 143–178.

Bjerksund, Petter, Gunnar Stensland, and Frank Vagstad (2011). "Gas storage valuation: Price modelling v. optimization methods". In: *The Energy Journal* 32.1, pp. 203–228.

Boogert, Alexander and Cyriel De Jong (2006). "Gas storage valuation using a Monte Carlo method". In.

Breslin, John et al. (2009). "Gas storage: Rolling intrinsic valuation". In: *Energy Risk, January* 2009, pp. 61–65.

Buehler, Hans et al. (2019). "Deep hedging". In: *Quantitative Finance* 19.8, pp. 1271–1291.

Carmona, René and Michael Ludkovski (2010). "Valuation of energy storage: An optimal switching approach". In: *Quantitative finance* 10.4, pp. 359–374.

Curin, Nicolas et al. (2021). "A deep learning model for gas storage optimization". In: *Decisions in Economics and Finance* 44.2, pp. 1021–1037.

Daluiso, Roberto et al. (2020). "Pricing commodity swing options". In: *arXiv preprint arXiv:2001.08906*.

De Jong, Cyriel (2015). "Gas storage valuation and optimization". In: *Journal of Natural Gas Science and Engineering* 24, pp. 365–378.

Felix, Bastian (2012). "Gas Storage Valuation: A Comparative Simulation Study". In.

Fischer, Thomas G (2018). *Reinforcement learning in financial markets-a survey*. Tech. rep. FAU discussion papers in economics.

Geman, Hélyette (2005). *Commodities and commodity derivatives: modeling and pricing for agriculturals, metals and energy*. John Wiley & Sons.

Giannelos, Spyros (2025). "Reinforcement Learning in Energy Finance: A Comprehensive Review." In: *Energies (19961073)* 18.11.

Gibson, Rajna and Eduardo S Schwartz (1990). "Stochastic convenience yield and the pricing of oil contingent claims". In: *The journal of finance* 45.3, pp. 959–976.

Hinton, Geoffrey (2012). *Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude*. Coursera lecture slides. URL: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

Honoré, Anouk (2011). "European natural gas demand, supply, and pricing: Cycles, seasons, and the impact of LNG price arbitrage". In: *OUP Catalogue*.

Jaillet, Patrick, Ehud I Ronn, and Stathis Tompaidis (2004). "Valuation of commodity-based swing options". In: *Management science* 50.7, pp. 909–921.

Jung, Sang-Woo et al. (2025). "Multi-Agent Deep Reinforcement Learning for Scheduling of Energy Storage System in Microgrids". In: *Mathematics* 13.12, p. 1999.

Kingma, Diederik P (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Lai, Guoming, François Margot, and Nicola Secomandi (2010). "An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation". In: *Operations research* 58.3, pp. 564–582.

Lillicrap, Timothy P et al. (2015). "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971*.

Löhndorf, Nils and David Wozabal (2021). "Gas storage valuation in incomplete markets". In: *European Journal of Operational Research* 288.1, pp. 318–330.

Longstaff, Francis A and Eduardo S Schwartz (2001). "Valuing American options by simulation: A simple least-squares approach". In: *The review of financial studies* 14.1, pp. 113–147.

Malyscheff, Alexander M and Theodore B Trafalis (2017). "Natural gas storage valuation via least squares Monte Carlo and support vector regression". In: *Energy Systems* 8.4, pp. 815–855.

Peters, Gareth William et al. (2013). "Calibration and filtering for multi factor commodity models with seasonality: incorporating panel data from futures contracts". In: *Methodology and Computing in Applied Probability* 15.4, pp. 841–874.

Polyak, Boris T (1964). "Some methods of speeding up the convergence of iteration methods". In: *Ussr computational mathematics and mathematical physics* 4.5, pp. 1–17.

Sage, Manuel, Joshua Campbell, and Yaoyao Fiona Zhao (2024). "Enhancing battery storage energy arbitrage with deep reinforcement learning and time-series forecasting". In: *Energy Sustainability*. Vol. 87899. American Society of Mechanical Engineers, V001T01A007.

Schaul, Tom (2015). "Prioritized Experience Replay". In: *arXiv preprint arXiv:1511.05952*.

Schwartz, Eduardo and James E Smith (2000). "Short-term variations and long-term dynamics in commodity prices". In: *Management Science* 46.7, pp. 893–911.

Schwartz, Eduardo S (1997). "The stochastic behavior of commodity prices: Implications for valuation and hedging". In: *The Journal of finance* 52.3, pp. 923–973.

Secomandi, Nicola (2010). "Optimal commodity trading with a capacitated storage asset". In: *Management Science* 56.3, pp. 449–467.

Silver, David et al. (2014). "Deterministic policy gradient algorithms". In: *International conference on machine learning*. Pmlr, pp. 387–395.

Sutton, Richard S et al. (1999). "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems* 12.

Thompson, Matt, Matt Davison, and Henning Rasmussen (2009). "Natural gas storage valuation and optimization: A real options application". In: *Naval Research Logistics (NRL)* 56.3, pp. 226–238.

Warin, Xavier (2012). "Gas storage hedging". In: *Numerical methods in finance: Bordeaux, June 2010*. Springer, pp. 421–445.

Yan, Xuemin (2002). "Valuation of commodity derivatives in a new multi-factor model". In: *Review of Derivatives Research* 5, pp. 251–271.

Zhao, Xingyu et al. (2025). "Optimization of Start-up Strategies of Gas Injection Compressor in Underground Gas Storage Using Deep Reinforcement Learning". In: *Simulation Modelling Practice and Theory*, p. 103204.

FFA Working Paper Series

2019

1. Milan Fičura: Forecasting Foreign Exchange Rate Movements with k-Nearest-Neighbor, Ridge Regression and Feed-Forward Neural Networks.

2020

1. Jiří Witzany: Stressing of Migration Matrices for IFRS 9 and ICAAP Calculations.
2. Matěj Maivald, Petr Teplý: The impact of low interest rates on banks' non-performing loans.
3. Karel Janda, Binyi Zhang: The impact of renewable energy and technology innovation on Chinese carbon dioxide emissions.
4. Jiří Witzany, Anastasiia Kozina: Recovery process optimization using survival regression.

2021

1. Karel Janda, Oleg Kravtsov: Banking Supervision and Risk-Adjusted Performance in the Host Country Environment.
2. Jakub Drahokoupil: Variance Gamma process in the option pricing model.
3. Jiří Witzany, Oľga Pastiranová: IFRS 9 and its behaviour in the cycle: The evidence on EU Countries.

2022

1. Karel Janda, Ladislav Kristoufek, Binyi Zhang: Return and volatility spillovers between Chinese and U.S. Clean Energy Related Stocks: Evidence from VAR-MGARCH estimations.
2. Lukáš Fiala: Modelling of mortgage debt's determinants: The case of the Czech Republic.
3. Oľga Jakubíková: Profit smoothing of European banks under IFRS 9.
4. Milan Fičura, Jiří Witzany: Determinants of NMD Pass-Through Rates in Eurozone Countries.
5. Sophio Togonidze, Evžen Kočenda: Macroeconomic Responses of Emerging Market Economies to Oil Price Shocks: An Analysis by Region and Resource Profile.
6. Jakub Drahokoupil: Application of the XGBoost algorithm and Bayesian optimization for the Bitcoin price prediction during the COVID-19 period.

7. Karel Janda, Binyi Zhang: Econometric estimation of green bond premiums in the Chinese market.

8. Shahriyar Aliyev, Evžen Kočenda: ECB monetary policy and commodity prices.

9. Sophio Togonidze, Evžen Kočenda: Macroeconomic implications of oil-price shocks to emerging economies: a Markov regime-switching approach.

2023

1. Jiří Witzany, Milan Fičura: Machine Learning Applications for the Valuation of Options on Non-liquid Option Markets.

2. David Mazáček, Jiří Panoš: Key determinants of new residential real estate prices in Prague.

3. Milan Fičura: Impact of size and volume on cryptocurrency momentum and reversal.

4. Masood Tadi, Jiří Witzany: Copula-Based Trading of Cointegrated Cryptocurrency Pairs.

5. Pavel Jankulár, Zdeněk Tůma: Changes to Bank Capital Ratios and their Drivers Prior Integrating Flood Risk into House Price Models Using Expected Discounted Loss: Evidence from the Czech Republic in 2024

6. Teona Shugliashvili: The words have power: the impact of news on exchange rates.

7. Jiří Witzany, Milan Fičura: A Comparison of Neural Networks and Bayesian MCMC for the Heston Model Estimation *(Forget Statistics – Machine Learning is Sufficient!)*

2026

1. Marek Folprecht: Measuring Flood Risk in Czechia with Stress Testing and a Gumbel copula-based VaR

2. Marek Folprecht: Integrating Flood Risk into House Price Models Using Expected Discounted Loss: Evidence from the Czech Republic in 2024

3. Tadi M., Fičura M., and Witzany J. (2026). Natural Gas Storage Valuation Using Deep Reinforcement Learning. FFA Working Paper 3/2026, FFA, Prague University of Economics and Business, Prague.

All papers can be downloaded at: wp.ffu.vse.cz

Contact e-mail: ffawp@vse.cz

Faculty of Finance and Accounting, Prague University of Economics and Business, 2023

Winston Churchill Sq. 1938/4, CZ-13067 Prague 3, Czech Republic, ffu.vse.cz