Mathematical Geosciences (2023) 55:673–692
https://doi.org/10.1007/s11004-023-10049-x

SPECIAL ISSUE

Multivariate Geostatistical Simulation and Deep Q-Learning to Optimize Mining Decisions

Sebastian Avalos[1] · **Julian M. Ortiz**[1]

Received: 25 July 2022 / Accepted: 29 January 2023 / Published online: 9 March 2023

Abstract
In open pit mines, the long-term scheduling defines how the mine should be developed. Uncertainties in geological attributes makes the search for an optimal scheduling a challenging problem. In this work, we provide a framework to account for uncertainties in the spatial distribution of grades in long-term mine planning using deep Q-Learning. Mining, processing and metallurgical constraints are accounted as restrictions in the reinforcement learning environment. Such environment provides a flexible structure to incorporate geometallurgical properties in production scheduling, as part of the block model. Geometric constraints (block precedence) and operational restrictions have been included as part of the agent-environment interaction. The effectiveness of the method is demonstrated in a controlled study case using a real multivariate drill-hole dataset, maximizing the net-present value of the project. The present framework can be extended and improved, to meet the particular needs and requirements of mining operations. We discuss on the current limitations and potential for further research and applications.

Keywords Mine planning · Deep learning · Q-learning · Geostatistics · Multivariate simulation

1 Introduction

In open pit mines, the long-term scheduling results from a detailed evaluation of alternative planning scenarios on how the mine should be developed. Conventionally, mine planners receive an estimated geometallurgical block model (GmBM) and deliver the optimum mine plan by processing the block model on a series of economical-

✉ Sebastian Avalos
sebastian.avalos@queensu.ca

✉ Julian M. Ortiz
julian.ortiz@queensu.ca

[1] The Robert M. Buchan Department of Mining, Queen's University, Kingston, Canada

operational scenarios in which a sequence of production is obtained by organizing the extraction of several pit-phases from a nested pit analysis.

A GmBM considers primary and secondary rock properties (Coward et al. 2009) relevant to the project development. The former are rock intrinsic attributes such as density, grades, alterations, and mineralogical characterization, independent of downstream processes, while the latter are responses of the ore such as throughput, grindability, recovery and specific energy consumption, to different downstream processes. The economical-operational scenarios consider forecasted prices and costs along with operational constraints such as maximum production and processing capacities, grades thresholds and ratios, ramp-up productions, processing plant expansions, stockpiling, blending constraints, among others. The pushbacks definition is an iterative process (Dowd 1994) that involves the use of an ultimate pit limit (UPL) optimizer, such as Lerchs-Grossman (Lerchs 1965) or Pseudoflow (Hochbaum 1998), to compute nested pits via revenue factor analysis (Whittle 1998; Whittle and Rozman 1991; Whittle 1997). The standard and prevailing mining software implementing such algorithms has been Whittle 4D (Whittle 1993).

As the conventional workflow deals only with a single instance of the rock property value, their intrinsic uncertainties are often neglected. Integrating the uncertainty on inputs yields more robust mine plans and enables risk-based decision-making. Although those benefits have been known for a long time (Dowd 1994), those uncertainties are not often considered due to the complexity of their integration in the conventional mine planning process (Monkhouse and Yeates 2018). As computer power grows, so too does the interest in uncertainty integration in mine plans as a research field (Jelvez et al. 2021; Gilani et al. 2020; Nelis et al. 2018). Dealing with hundreds of thousands of blocks, uncertainties in the primary and secondary rock properties, along with economic uncertainty and operational constraints makes the search for optimal mine plans an attractive problem for sequential decision-making algorithms (Koch and Rosenkranz 2020).

The benefits of integrating deep learning (Goodfellow et al. 2016) methods via deep neural network architectures as an additional tool for geoscientists have been studied in a wide range of earth science fields related to mining, such as mineral recognition (Chen et al. 2020; Liu et al. 2017), mineral prospectivity (Sun et al. 2020; Xiong et al. 2018), multiple-point statistics simulation (Avalos and Ortiz 2020; Bai and Tahmasebi 2020), and analysis of porous media and water resources (Santos et al. 2020; Tahmasebi et al. 2020; Yun et al. 2020). However, deep neural network architectures have yet to be incorporated into stochastic mine planning research.

One of the areas that naturally opens doors for research in stochastic mine planning is reinforcement learning (RL). Reinforcement learning in its core follows the principles of a Markov decision process (Sutton and Barto 2018). An optimal policy defines a sequence of decisions that lead to maximizing the total expected return. In mine planning, Lamghari and Dimitrakopoulos (2020) applied the RL principles to adjust parameters involved in heuristic approach selection for block scheduling. Paduraru and Dimitrakopoulos (2019) demonstrated the potential of training a feed-forward neural network with reinforcement learning to provide optimal material allocation policies. The input information corresponds to the block information and the output is the probability of destination. Kumar and Dimitrakopoulos (2021) proved the effectiveness of

RL in updating the short-term mine plan when new incoming information is available. The Monte Carlo tree search approach (Silver et al. 2016) played a fundamental role by creating new experiences to train the neural network-based agent.

In RL, the agent-environment framework sets the fundamentals for mapping states-and-actions to expected value, maximizing the long term total reward. When both spaces, states and actions, are small enough, the mapping function can be retrieved from look-up tables or parametrized as an approximation of the mapping function. However, in most RL problems, the environment is often incomplete, and the space of states and actions are non trackable or computationally unmanageable. Recent advances on deep learning (Goodfellow et al. 2016) have led to implement deep neural networks to approximate the mapping function, referred as deep Q-Learning (Watkins and Dayan 1992).

In this work, we focus on the integration of uncertainty in the spatial multivariate grade distribution into the long-term mine plan. Additional inputs, resulting from one or more transfer functions, such us period of extraction, block destination, revenue factor and obtained grades when blocks are mined out are incorporated as part of the geometallurgical block model. It is worth noting that the previous additional inputs are dynamic with respect to time. In other words, blocks in the GmBM will be initialized with a predefined value and they will be updated as the process takes place. No grade control or short-term production scheduling adaptation is considered. Processing and metallurgical costs and constraints as well as metal recovery as a function of grades are included in the environment in which the agent is trained to maximize the total net present value (optimal policy). The mine scheduling results from a sequence of decisions in which the agent selects the optimal block/set of blocks based on the current state of the mine and the experience learned via deep Q-Learning. The training process considers the entire life-of-mine over daily periods. As a result, the optimal policy accounts for economic time discount, geometric constraints, geological uncertainty, and metallurgical process responses. Section 2 describes the reinforcement learning model adopted in this work. Section 3 provides the context and details on the controlled case study. In Sect. 4 we analyze the main results discussion on the effectiveness and limitations on the method. Section 5 provides insights on further research and applications, while Sect. 6 summarizes the work by providing the main conclusions.

2 Model

As with any other reinforcement learning task, the resulting performance is driven by how the environment is built-in, the interaction between environment and agent in the form of states-actions, and the internal neural network structure of the agent. In this section, we present our approach, knowing that others can be created in a similar fashion, highlighting the benefits of employing this perspective.

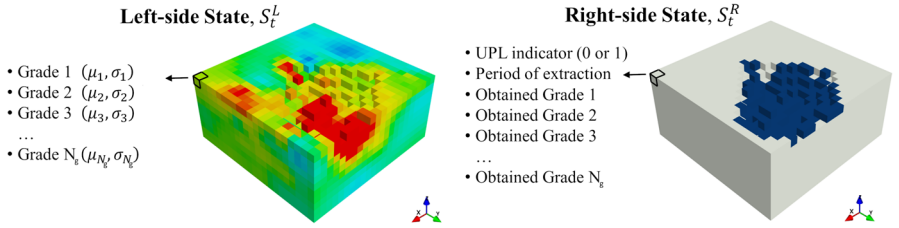

Fig. 1 Environment states. Left-side State S_L^t (left) and Right-side State S_R^t

2.1 Environment—Mining Context

In a conventional mine planning context, the GmBM usually carries information about primary and secondary rock properties in the form of estimated values. The resulting project parameters such as period of extraction, push-back labelling and block destination come as a result of a strategic mine plan. To account for geological uncertainty, geostatistical cosimulation of primary properties can be performed. These realizations can be summarized by the mean and standard deviation of the grades simulated at each block. If categorical attributes are available, such as alterations and lithologies, a similar approach can be applied, where the conditional distribution at each block could be summarized by its local proportions. For instance, Fig. 1 (left) illustrates a GmBM considering the mean and standard deviation of N_g grades (the mean of one of those elements is shown in the figure). We refer to the GmBM that contains the subset of attributes related to simulated primary and secondary rock properties as the left-side state of the environment. The principles on why doing this lays on how the left-side environment state is connected with the internal structure of the agent neural network, explained in Sect. 2.2.

The Right-side State of the environment (Fig. 1 (right)) corresponds to the GmBM containing the project parameters and the measured primary and secondary properties obtained when blocks are extracted. From the pool of project parameters, we consider the period in which each block has been extracted, the minimum revenue factor value at which the block is economically feasible for extraction, an ultimate pit limit indicator (one, if the block falls inside the final pit and zero, otherwise), and the N_g grades obtained at each block extracted in the past. All variables in the Right-side State are initialized to zero, except for the UPL indicator and the revenue factor value. It is worth noticing that air blocks are treated as zeros, and the entire GmBM is always considered.

A block b has coordinates $(i, j, k) \in \mathbb{N}^3$ with $i \in [1, N_x]$, $j \in [1, N_y]$, and $k \in [1, N_z]$, where N_x, N_y and N_z correspond to the total number of blocks in the X, Y and Z axes, respectively. The location (1, 1, 1) corresponds to the lowest (Z axis), southernmost (Y axis), and westernmost (X axis) block.

We have described the environment in which the agent interacts with and takes an action. The action is the selection of a block to be extracted, from the pool of *possible* blocks. As only surface blocks can be mined, the pool of possible blocks corresponds to all positions $(i, j) \in N_x \times N_y$. As only blocks inside the ultimate pit

are actually extracted, and blocks are subject to geometrical constraints, we define $\mathcal{B}_t \subseteq N_x \times N_y$ as the set of *feasible* blocks for extraction at each time step t. The geometrical constraints refer to the number (5 or 9) and position of preceding blocks to free a lower block for extraction, the bench height and overall pit angles. From here on, we assume a 5 blocks precedence (cross shape), one block benches, and one-block-over and one-block-down slope for simplicity.

Labelling the blocks in $\mathcal{B}_t$ as one and the blocks outside of $\mathcal{B}_t$ as zero, a matrix $\mathcal{M}_t$ of size $N_x \times N_y$ is automatically defined with zeros and ones. Let $s(\mathcal{M}_t)$ be the sum of all elements in the matrix, we have that $s(\mathcal{M}_0)$ corresponds to the initial number of feasible blocks, while $s(\mathcal{M}_t) = 0$ corresponds to the final state at time period T where all blocks, inside the pit limits, have been mined. To compute $\mathcal{M}_t$, the Right-side State of the environment S_t^R, and the position of the previously extracted block b_{t-1} are required. Using the three-dimensional block information, the integration of geometrical constraints into $\mathcal{M}_t$ is straightforward. We summarize the steps in the pseudo-code of Algorithm 1. Although the previous algorithm is efficient by not going over each location of $\mathcal{M}_t$, variations can be implemented for additional efficiency improvement.

Algorithm 1 Computation of feasible actions at time step t

Input Right-side state (S_t^R), Previous feasible actions ($\mathcal{M}_{t-1}$), Block previously extracted (b_{t-1}), Geometrical constraint.
Output Matrix of feasible actions ($\mathcal{M}_t$).
procedure COMPUTATION OF FEASIBLE ACTIONS ($S_t^R, \mathcal{M}_{t-1}, b_{t-1}$)
    $\mathcal{M}_t \leftarrow \mathcal{M}_{t-1}$ ▷ Initialize with previous feasible actions
    $\mathcal{M}_t(b_{t-1,x}, b_{t-1,y}) \leftarrow 0$ ▷ Location of b_{t-1} becomes unfeasible
    We loop over the blocks following the geometrical constraints
    for $x \leftarrow [-1, 0, 1]$ **do** ▷ Loop over the X axis on precedence blocks
        for $y \leftarrow [-1, 0, 1]$ **do** ▷ Loop over the Y axis on precedence blocks
            $(px, py) \leftarrow (b_{t-1,x} + x, b_{t-1,y} + y)$ ▷ Center of one of the precedence blocks
            If one of the coordinates px or py falls outside $\mathcal{M}_t$, the point is not considered
            if $\mathcal{M}_t(px, py) = 0$ **then** ▷ Analyzing if an unfeasible position becomes feasible
                $pz \leftarrow S_t^R(px, py)$ ▷ Position in the Z axis of the current surface block at position (px, py)
                if $pz \neq N_z$ **then** ▷ As long as the block does not belong to the top level of the block model
                    if The block in $S_t^R(px, py, pz)$ belongs to the final pit and has not been extracted yet **then**
                        if The precedence blocks at $pz + 1$ that belongs to the final pit have been mined **then**
                            $\mathcal{M}_t(px, py) \leftarrow 1$
                        end if
                    end if
                end if
            end if
        end for
    end for
end procedure

The reward plays a critical role in the agent learning experience (Matignon et al. 2006; Sutton and Barto 2018) and therefore, in the resulting mine scheduling. We relate the reward of the agent's action with the economic revenue obtained by processing

the selected block as:

$$G_t \approx r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot r_{t+3} + \cdots + \gamma^{T-t-1} \cdot r_T = \sum_{k=0}^{T} \gamma^k \cdot r_{t+k+1} \quad (1)$$

where $\gamma \in [0, 1[$ is a discount factor, T is the final step, and r_t and G_t are the reward and total reward at time step t, respectively. By defining $\gamma = (1 + d)^{-1}$ with $d > 0$ being the economic discount rate, the total reward G_t is equivalent to the net present value of all the future revenues starting at time step $t + 1$. Thus, training the agent to maximize the total reward would lead to a local maximum net present value. Notice that:

- We have imposed γ to reflect the time value of money as a function of the discount rate. Thus, the reward at time t must be equal to the economic revenue obtained at that period of time without discount.
- We could account for the discount rate d into the economic revenue r_t, leaving γ as an hyper-parameter of the training process unrelated to the time value of money.

The economic revenue of blocks, r_t, can include their processing performance, which may depend on the blend with other blocks. Notice that the sequence of extraction usually differs from the sequence of processing.

2.2 Agent—Deep Neural Network

The aim of the agent is to evaluate all possible actions with respect to the environment state, such that by taking the best possible action, the total reward is maximized. The agent could take the form of a simple function, an algorithm, or a complex process. In this work, we make use of the deep Q-Learning (Mnih et al. 2015; Watkins and Dayan 1992) strategy where the agent corresponds to a deep neural network architecture whose hyper-parameters are trained to perform the action evaluation. We provide details on the principles followed by the agent-environment interaction and the fundamentals of deep Q-Learning in Avalos and Ortiz (2021).

We know that convolutional neural networks (CNNs) stand out when working with regular grids. As the GmBM has a grid-like structure, it is reasonable to study the use of a deep convolutional neural network architecture. In brief, neural networks are parametrized by weights and biases, and the architecture represents how such parameters are interconnected and linked to additional functions, such as activation functions, max pooling and batch normalization. In CNNs, the weights correspond to filters or convolutional kernels. Activation functions transform the signal (information) via non-linear functions, such as sigmoid function and rectified linear unit (ReLU). Further details can be found in Goodfellow et al. (2016).

In this work, the agent architecture is made of convolutional blocks and feed forward neural networks. We decide to use convolutional blocks on the left-side and right-side states independently and then combine them via concatenation of hidden layers, as shown in Fig. 2. This decision is arbitrary and has been made under the assumption of having weights and biases related only to the primary/secondary rock properties,

others related only to project parameters, and other accounting for both in a higher level of the architecture. Exploring the architecture in detail, we have that:

- A convolutional block contains a convolutional function, followed by a leaky ReLU activation function, and a max pooling function.
- The environment is the input of the architecture. Two convolutional blocks are applied on the left-side state S_t^L, one after the other. Similarly, two other convolutional blocks are applied on the the right-side state S_t^R. Features (patterns) of the environment are extracted in these convolutional blocks.
- The resulting hidden layers on both sides are then concatenated and fed into a third convolutional block, for a refined feature extraction now considering features of the entire environment.
- The resulting hidden layer is reshaped into a one dimensional vector and fed into a fully connected layer, followed by a leaky ReLU activation function.
- The result is fed into another fully connected layer with a leaky ReLU activation function and then fed into the last fully connected layer, without activation function.
- The latter one-dimensional vector must have a $N_x \times N_y$ size. It is then reshaped into a $(N_x \times N_y)$ matrix, referred to as $\mathcal{Q}_t$, representing the estimated Q-value $q(s_t, a_t)$. Here, the state of the environment, reflected in the feature of the convolutional blocks and mapped into this last layer throughout the section of fully connected layers is reflected into the Q-values.
- The Algorithm 1 is applied on S_t^R, providing the matrix of feasible actions, $\mathcal{M}_t$.
- As $\mathcal{M}_t$ is a zeros-and-ones matrix, we compute the probability of extracting the block $b_{t,i,j}$ at time t, $p_{t,i,j} \in [0, 1]$, as:

$$p_{t,i,j} = \frac{e^{s_{t,i,j}}}{\sum_{i \in N_x, j \in N_y} e^{s_{t,i,j}}} \tag{2}$$

where

$$s_{t,i,j} = \mathcal{Q}_{t,i,j} \cdot \mathcal{M}_{t,i,j} - K \cdot (1 - \mathcal{M}_{t,i,j}) \tag{3}$$

with $K = 10^5$. Thus, $s_{t,i,j}$ takes the value of $-K$ when $\mathcal{M}_{t,i,j}$ is zero, and the value of $\mathcal{Q}_{t,i,j}$ otherwise. It leads to all unfeasible actions having a probability of extraction of zero and for feasible actions, a probability of extraction proportional to the estimated Q-value.

We rely on the ε-greedy approach for training the agent. At each iteration i, the epsilon value ε_i controls the trade-off between exploration ($\Pr = \varepsilon_i$), selecting a random block from the pool of feasible blocks, and exploitation ($\Pr = 1 - \varepsilon_i$), select the block with highest probability of being extracted. The epsilon value is adjusted by an epsilon decay parameter $\varepsilon_{\text{decay}}$ such that $\varepsilon_{i+1} \leftarrow \varepsilon_{\text{decay}} \cdot \varepsilon_i$. Although we explore the effects of different epsilon decay values, common values are $\varepsilon_0 = 1.0$, $\varepsilon_{\text{decay}} = 0.99$, and $\varepsilon_\infty = 0.05$. The latter as a baseline to avoid overfitting and promote random actions when exploiting the trained agent (Mnih et al. 2015).

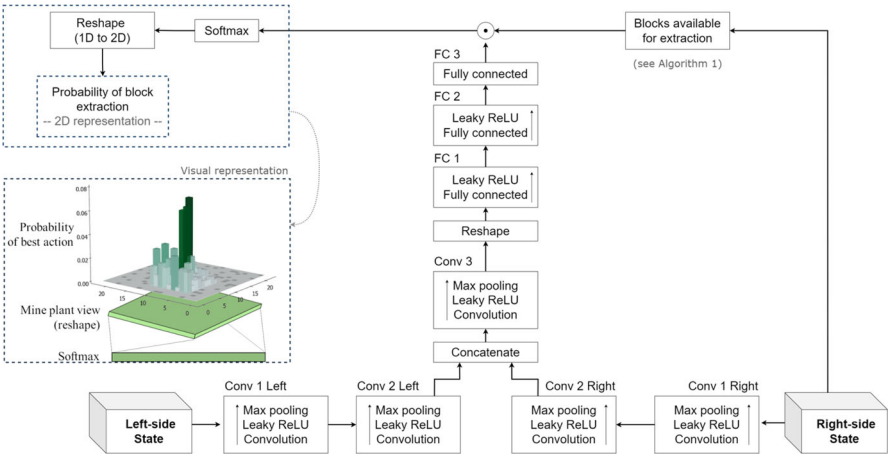

Fig. 2 Deep Q-network. The left and right state of the environment are fed separately into the neural network. The output corresponds to the probability of each block being selected for extraction

Table 1 Statistical summary of drill-hole samples

	Cu	Fe	S	C	Al	Na	As	K
Min	0.01	0.15	0.3	0.01	0.16	0.01	0.01	0.04
Mean	0.69	2.44	2.03	0.26	0.99	0.07	0.51	0.23
Max	5.55	14.77	10	4.99	3.47	0.26	2.94	0.94
Std Dev	0.52	1.2	1.34	0.56	0.67	0.03	0.59	0.1
p25	0.41	1.62	1.12	0.03	0.47	0.04	0.03	0.17
p50	0.58	2.2	1.71	0.08	0.71	0.06	0.26	0.2
p75	0.79	3.05	2.52	0.29	1.46	0.08	0.94	0.26

3 Case Study

We explore the potential of the agent-based mine planning methodology on a multivariate dataset of a porphyry copper deposit. This section describes how the mining context is formulated in terms of agent-environment interaction, the epsilon policies adopted to handle the exploration-exploitation dilemma and the hyper-parameters of the deep convolutional neural network used as agent.

3.1 Mining Context

The ore deposit model has dimensions of 1760 m by 1760 m by 800 m in the east, north and elevation directions, respectively. On 2460 drill-hole samples, the element grades of copper, iron, sulphur, carbon, aluminum, sodium, arsenic, and potassium are measured in percentage. Copper is the main element of interest. The main statistics over the elements are displayed in Table 1.

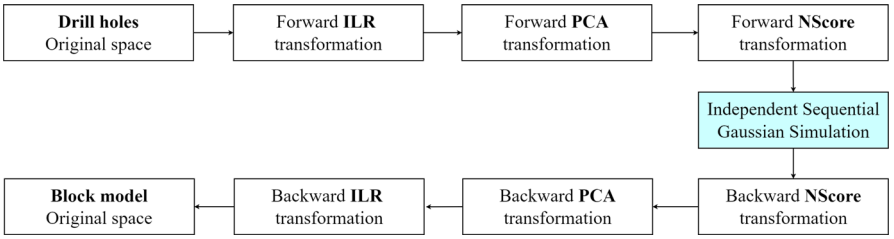

Fig. 3 Scheme of space transformations for multivariate simulation

Multivariate geostatistical simulation is not a trivial task. It aims to preserve the univariate, bivariate and multivariate statistics present in the sample datasets as well as their spatial structures, often measured via direct and cross variograms. Current methods, such as Direct Multivariate Simulation (de Figueiredo et al. 2021), Projection Pursuit Multivariate Transformation (Barnett et al. 2014), Flow Anamorphosis (Mueller et al. 2017), and Stepwise Conditional Transformation (Leuangthong and Deutsch 2003) seek to transform the original attribute space into a decorrelated multi-Gaussian space. Conventional geostatistical simulation techniques can then be performed independently on each transformed coordinate. Results are back transformed by reversing the mapping process/algorithmic steps. Other approaches, such as Linear Model of Coregionalization (Bourgault and Marcotte 1991), provide alternative solutions to the multivariate modeling problem without transforming the original space. It represents the multivariate random fields as a linear combination of independent univariate random functions, requiring direct and cross variogram models on the original space.

In this work, we have adopted a simple framework for multivariate decorrelation. To reproduce the joint distribution of all grades, a scheme of attribute transformation is performed over the original grades (Fig. 3): isometric log-ratios, principal component analysis, and normal score transformation is performed forward. Sequential Gaussian simulation is used to simulate the spatial distribution of the resulting normal score values. The simulated results are then back transformed to the original space. Grades are simulated on a block model of $110 \times 110 \times 50$ blocks of dimension $16 \times 16 \times 16$ m^3 at point support using 64 conditioning data points. One hundred realizations are generated to properly capture the deposit grade uncertainty.

In order to demonstrate how the approach can handle complex scenarios, the following criteria are imposed to define the mine plan:

- Mine capacity, as the amount of ore and waste handled, is 425,000 tonnes per day, around 40 blocks of $16 \times 16 \times 16\,\mathrm{m}^3$ per day.
- Unlimited crushing, grinding and processing capacities.
- Processing recovery (all elements in percentage):
  - $\mathrm{Rec}_{\mathrm{Cu}}^{\mathrm{OX}} = 93.4 + 0.7 \cdot \mathrm{Cu/S} - 20 \cdot \mathrm{As}$. Maximum: 95%. Minimum: 40%.
  - $\mathrm{Rec}_{\mathrm{Cu}}^{\mathrm{SUL}} = 80.0 + 5 \cdot \mathrm{Cu/S} - 10 \cdot \mathrm{As}$. Maximum: 95%. Minimum: 50%.
- Block destination: the ore is sent to the oxide plant subject to a maximum carbonate content of 0.5%, and a minimum ratio of Cu/S of 0.4, if and only if the obtained

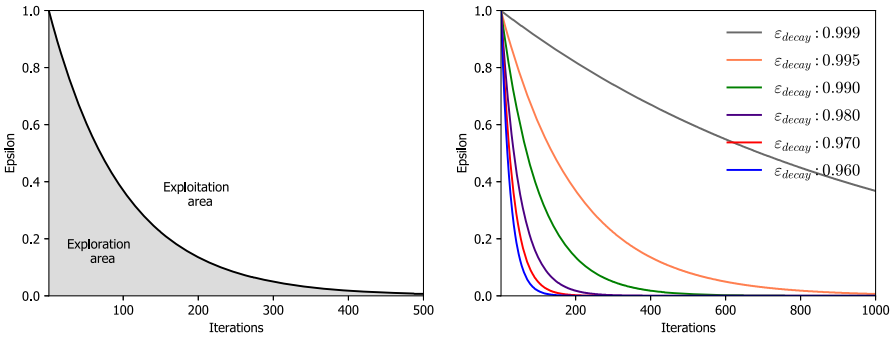

Fig. 4 Epsilon values. Exploration and exploitation trade-off (left), and continuous epsilon decay (right)

revenue is higher than the potential revenue at the sulphide plant. Otherwise, it is sent to the sulphide plant. If the block does not pay the processing costs, it is sent to the waste dump.

- Variable global acid consumption (GAC): for oxides, the cost of acid consumption is calculated as GAC $= 25.4 + 18.8 \cdot$ C in USD/ton.
- Copper price: 2.3 USD/lb. Mining cost: 6 USD/ton mined. Crushing & grinding cost: 15 USD/ton processed. Processing cost: 0.5 USD/lb, per pound of copper produced. Economic discount rate: 15 % yearly.

We have not imposed any additional operational restriction such as number of working faces, number of blocks to be mined in a certain area before moving into a new face or stripping ratio. The framework can be modified to include these constraints.

3.2 Exploration—Exploitation

We explore how different epsilon-greedy policies affect the trade-off between exploration and exploitation during training (Fig. 4 (left)). The initial and final epsilons are $\varepsilon_0 = 1.0$, and $\varepsilon_\infty = 0.0$, respectively. We define as continuous epsilon decay (Fig. 4 (right)) the polices that iterate 1000 times during training, whose epsilon decay parameters $\varepsilon_{\text{decay}}$ takes one of the values in [0.96, 0.97, 0.98, 0.99, 0.995, 0.999].

3.3 Scenarios

The Right-side State requires the actual element composition of the extracted blocks. Defining what would be the ground truth is arbitrary and not a straightforward decision. Thus, we study the cases in which the ground truth corresponds to:

- The E-type of all hundred realizations.
- A single realization from the pool of hundred realizations.

We combine the above possible ground truths with the epsilon-greedy policies to study the impact of both factors in the resulting optimal policy (mine scheduling). In practice the ground truth could be obtained with production data obtained from blasthole samples.

3.4 Agent

We define the agent's inner structure in this section. The input dimension of the first convolutional blocks on the Left-side and Right-side State must be equal to the number of attributes at each location in the three-dimensional GmBM. In this case, 16 (mean and standard deviation of 8 grades) and 11 (8 obtained grades, ultimate pit limit labelling, revenue factor value, and period of extraction), respectively. The output dimension is 32 on both convolutional blocks. For the rest of the structure, parameters can be found in Table 2. Notice that, in this case, the number of neurons of the last fully connected layer (FC 3) must be equal to $N_x \times N_y$. We recognize that model complexity, measured as the number of inner-parameters and architectural structure, has a direct impact on data overfitting/underfitting. The parameters used in our implementation have standard values and inner-parameter optimization is out of the scope of this work.

The main hyper-parameter to be tuned is the learning rate, on two different scenarios (ground truth). To accelerate the search of optimum hyper-parameters, we re-blocked the GmBM into $80 \times 80 \times 80\,\text{m}^3$. The obtained results are transferred into a re-blocked model of $32 \times 32 \times 32\,\text{m}^3$.

4 Results

We begin by displaying the probability plots between the sample database and 20 realizations on each grade (see Fig. 5). Reproducing the multivariate relationship is challenging and although Cu, Fe, As, and Al were statistically well reproduced, S, C, Na, and K show clear space for improvement. Isometric visualizations of Cu, S, As and C on a single realization and the average over one hundred realizations are displayed in Fig. 6. Note that we are displaying the re-blocked model at $32 \times 32 \times 32\,\text{m}^3$ (top) and $80 \times 80 \times 80\,\text{m}^3$ (bottom).

The Figs. 7 and 8 display the evolution of the net present value obtained as a function of the training process using the Etype and a single realization as ground truth, respectively. From Fig. 7 we observe that using $\varepsilon_{\text{decay}} = 0.995$ yields a better policy in terms of expected NPV. In the case of Fig. 8, the better policy is achieved with $\varepsilon_{\text{decay}} = 0.97$.

The difference in magnitude of obtained NPV after training, 3630 MUSD versus 4160 MUSD respectively, is due to the assumed ground truth that provides the obtained grades once extracted. The question on how different are both optimal policies arises. In order to understand if the selection of ground truth has an impact in the resulting mine scheduling, we process each realization following the block scheduling of the mine plans before and after training for each ground truth scenario. Figures 9 and 10 provide the distribution of resulting NPV using the Etype and realization as ground truth, respectively, with expected average NPV of 3260 MUSD and 3180 MUSD, respectively. We observe similar results, with a slight improvement using the Etype as ground truth. Thus, it appears that the ground truth used as reference does not play a critical role, as long as the general trends are learned. The previous results should

Table 2 Main parameters of the agent's structure

Conv 1—Left	In: 16, Out: 32, Kernel: $2 \times 2 \times 2$	Conv 1—Right	In: 11, Out: 32, Kernel: $2 \times 2 \times 2$
	Stride: $1 \times 1 \times 1$, Padding: $2 \times 2 \times 2$		Stride: $1 \times 1 \times 1$, Padding: $2 \times 2 \times 2$
Leaky ReLU	Slope: -0.01	Leaky ReLU	Slope: -0.01
Ma $\times$ Pooling	Kernel: 3×3	Ma $\times$ Pooling	Kernel: 3×3
Conv 2—Left	In: 32, Out: 16, Kernel: $2 \times 2 \times 2$	Conv 2—Right	In: 32, Out: 16, Kernel: $2 \times 2 \times 2$
	Stride: $1 \times 1 \times 1$, Padding: $2 \times 2 \times 2$		Stride: $1 \times 1 \times 1$, Padding: $2 \times 2 \times 2$
Leaky ReLU	Slope: -0.01	Leaky ReLU	Slope: -0.01
Max Pooling	Kernel: 3×3	Max Pooling	Kernel: 3×3
Conv 3	In: 32, Out: 8, Kernel: $2 \times 2 \times 2$		
	Stride: $1 \times 1 \times 1$, Padding: $2 \times 2 \times 2$	FC 1	In: 144, Out: 20
Leaky ReLU	Slope: -0.01	FC 2	In: 20, Out: 20
Max Pooling	Kernel: 3×3	FC 3	In: 20, Out

Total amount of hyper-parameters to be trained: 84,357

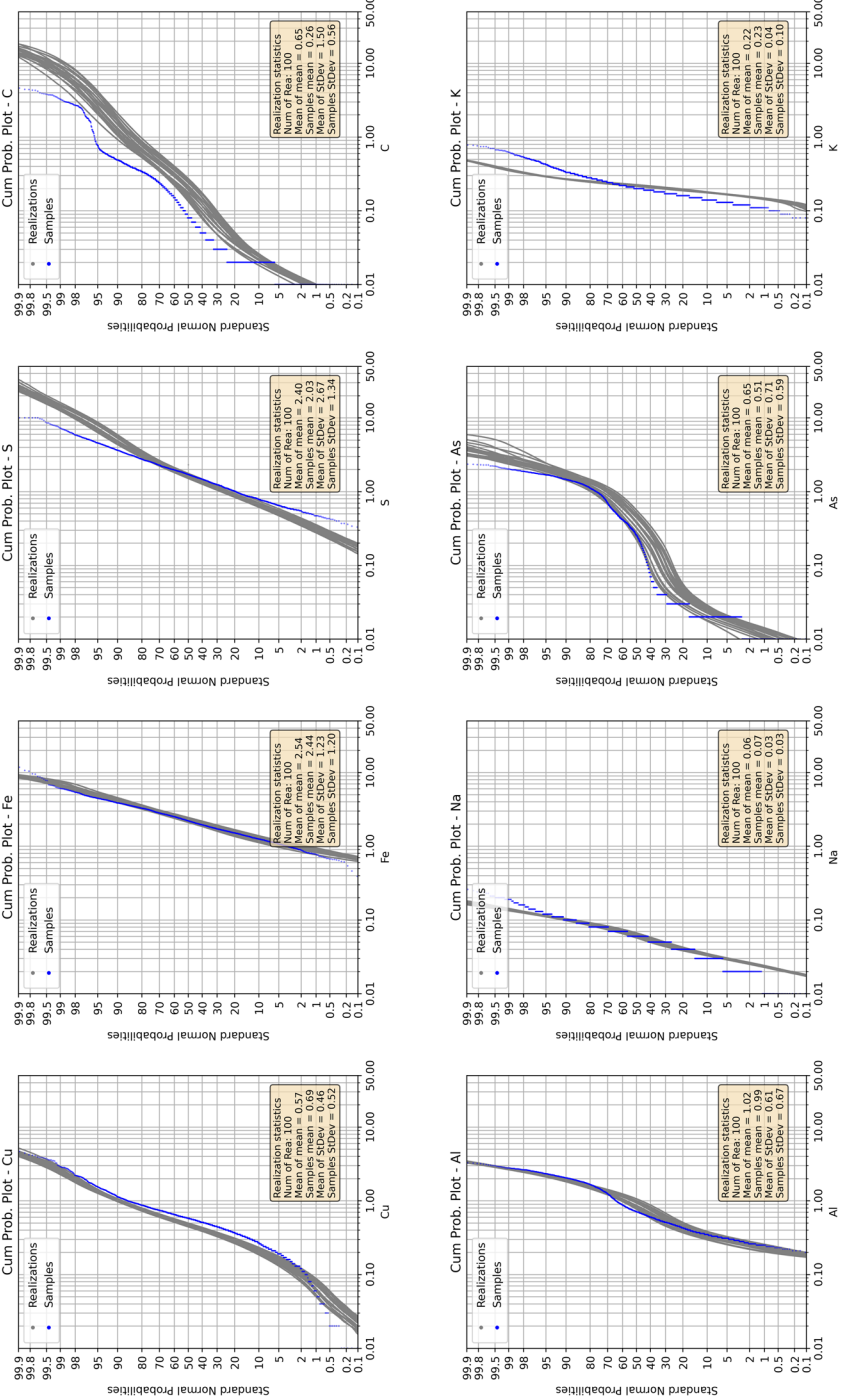

Fig. 5 Probability plots on the original grade space

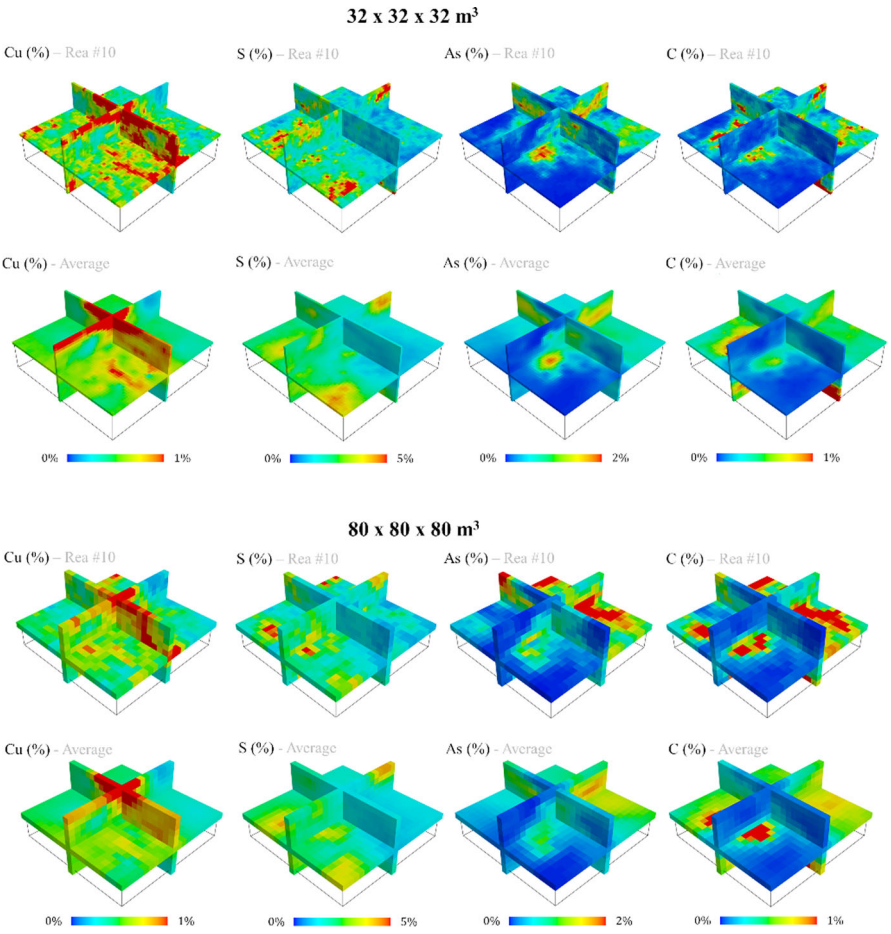

Fig. 6 Isometric views of one realization and Etype of Cu, S, As and C at $32 \times 32 \times 32\,\mathrm{m}^3$ (top) and $80 \times 80 \times 80\,\mathrm{m}^3$ block average

not be extended to other deposits, specially ones with high heterogeneity and lower density of samples.

The Fig. 11 displays slide visualization looking down (plan view) and east (cross section) showing the period of extraction of blocks before and after training, on a daily basis. From the plan views we observe that operational constraints are required to avoid the extractions of blocks that are not contiguous. In other words, we must consider how shovels operate as well as the number of shovels in operations. From the cross sections, we observe that geometrical constraints are well incorporated in Algorithm 1, providing a feasible representation of the possible blocks to extract.

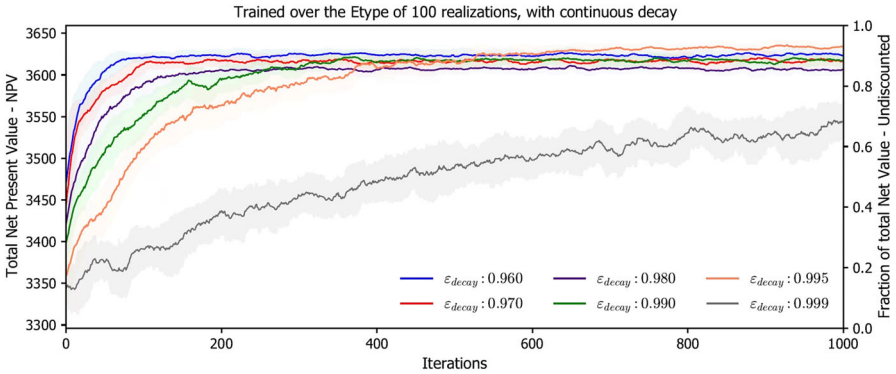

Fig. 7 Sensitivity on epsilon decay. Total NPV and Fraction of net value as a function of the training process. Ground truth: Etype. Trained on the $80 \times 80 \times 80\,\mathrm{m}^3$ block model. Shadow areas representing $\pm$ NPV Standard Deviation on a 15-iteration moving window

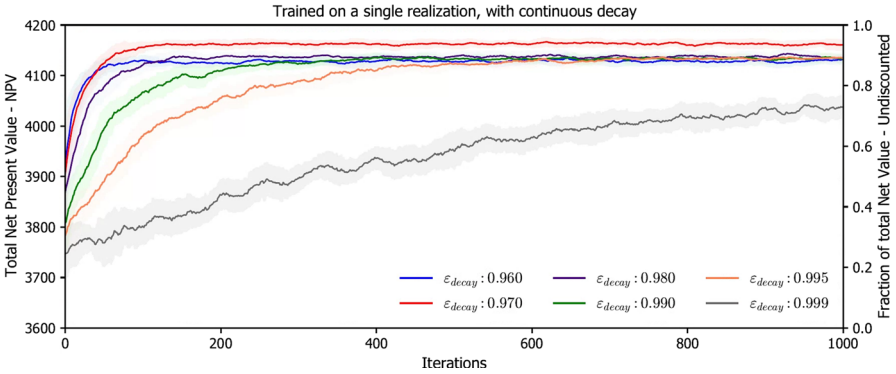

Fig. 8 Sensitivity on epsilon decay. Total NPV and Fraction of net value as a function of the training process. Ground truth: a single realization. Trained on the $80 \times 80 \times 80\,\mathrm{m}^3$ block model. Shadow areas representing $\pm$ NPV Standard Deviation on a 15-iteration moving window

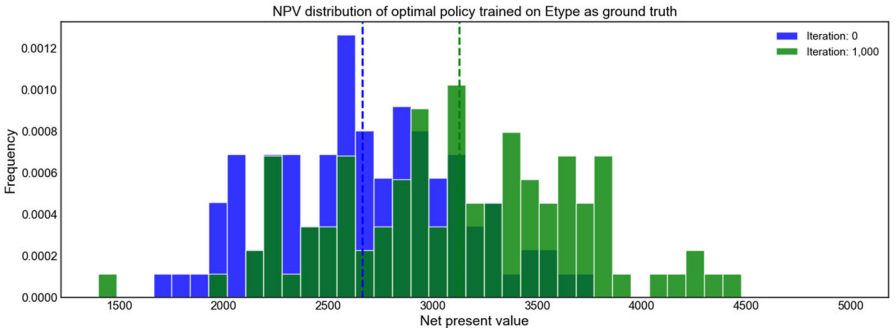

Fig. 9 NPV distribution over all realizations using a non-trained agent (iteration: 0, blue) and trained agent (iteration: 1000, green) using Etype as ground truth. Vertical lines represent the mean distribution value. Trained on the $32 \times 32 \times 32\,\mathrm{m}^3$ block model with $\varepsilon_{\mathrm{decay}} = 0.995$

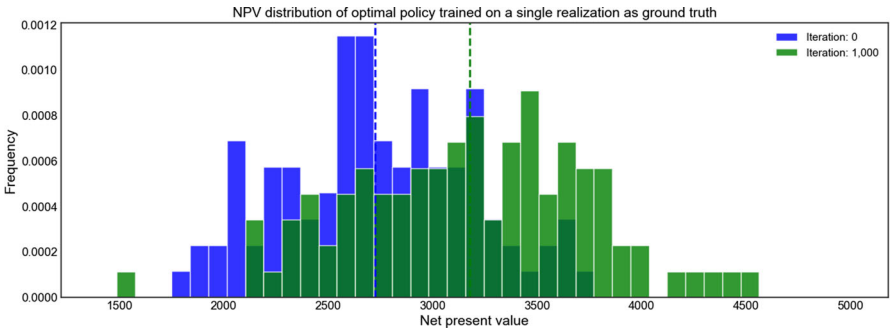

Fig. 10 NPV distribution over all realizations using a non-trained agent (iteration: 0, blue) and trained agent (iteration: 1000, green) using a single realization as ground truth. Vertical lines represent the mean distribution value. Trained on the $32 \times 32 \times 32\,\mathrm{m}^3$ block model with $\varepsilon_{\text{decay}} = 0.97$

5 Discussion

In this section, we discuss the potential for further research and applications.

- Agent-environment interaction. The environment could be improved by adding additional attributes that would guide the agents to a better scheduling strategy. A reward that does not only consider the block attributes in isolation but that also considers blending and material management implications would provide a more realistic assessment of the value of processing the blocks.
- Deep Q-network. While keeping the advantages of convolutional neural networks on the environmental state, incorporating the principles of recurrent neural networks would benefit the process of taking actions in a sequential manner. Thus, hybrid deep learning methods must be studied. Currently, transformer networks (Vaswani et al. 2017) and the attention mechanism (Tay et al. 2020) offer a potential starting point.
- Operability. Being able to extract a block, and at the next time step selecting another one several blocks away is an unfeasible assumption from the perspective of a single shovel operation. This could be improved by incorporating operational constraints, by imposing that shovels have fixed locations at their working faces and their movement is limited. Only blocks with at least two exposed faces are extracted unless the next horizontal level starts its extraction. A minimum area (number of consecutive blocks) must be imposed on each level before moving down into the next one to represent the minimum operating space. More than one face and one block per face can be mined at each time step. These and other constraints should be reflected in Algorithm 1.
- Hierarchical planning. We studied the impact of the epsilon decay value as well as the ground truth for the obtained grades using blocks of size $80 \times 80 \times 80$, which represents 125 times the volume of a unit block of $16 \times 16 \times 16$. Then, we moved into blocks of $32 \times 32 \times 32$, representing 8 times the unit block. However, the optimal sequence in the large scale is not propagated into the small scale, only the hyper-parameters of the training process. An interesting approach would be to train and use the same agent, taking care of the changing dimensions in their inner

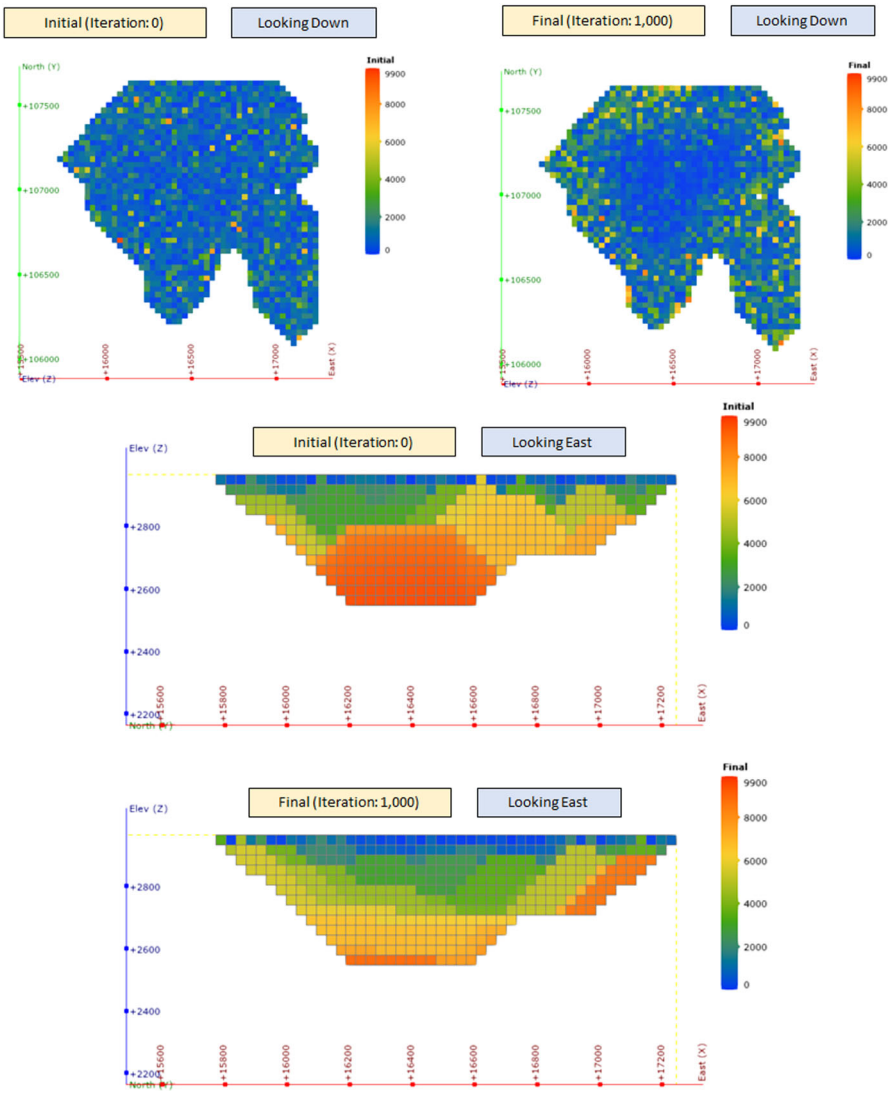

Fig. 11 Visualization of daily block sequence (period of extraction) before and after training. Looking down (top) and looking east (center-bottom). Optimal policy using the Etype as ground truth

layers, from a higher level to a lower level (ex: x125,x64,x27,x8,x1), transferring the optimal sequence into the right-side environment as accumulated knowledge.

- Underground mining. There is a great potential of extending the methodology to massive underground mining. On massive underground mining, such as block caving, a fundamental challenge is the updating process of the left-side state of the environment based on the action of removing material at the extraction points. The incorporation of geomechanical principles due to the ore flowing through extraction points is another relevant challenge.

6 Conclusions

The proposed framework uses geostatistical simulations obtained for multiple relevant attributes and trains a deep neural network via reinforcement learning to optimize mining decisions, such as long-term production scheduling. The sequence of block extraction corresponds to the long-term planning following a day-to-day sequence of extraction over all blocks in the final pit. The agent is made up of several deep neural networks trained in an environment that represents the expected uncertainties of multiple grades. Operational constraints, such as mining rates, processing throughput, limits on processing grades ratios, and so on, can be incorporated in the agent-environment interactions impacting the obtained reward as economic revenue. The present framework can be easily extended and improved, meeting the particular needs and requirements of mining operations. We demonstrated the feasibility of the method and discussed the potential for further improvements and applications.

Further research is required to improve the agent's learning mechanism, the evaluation of state-action pairs when selecting the best block for extraction, and the reproduction of operational practices. Some of the main benefits of the proposed framework is the simple incorporation of grade uncertainty by means of multivariate modeling, the possibility of updating transfer functions such as copper recovery and acid consumption, and the potential of being extended into underground mine projects.

List of Symbols

S_t or s_t	The state of the environment at time t
S_t^L, S_t^R	The Left-side and Right-side states of the environment at time t, respectively
a_t	Action at time t
r_t	Reward at time t
G_t	Total reward at time t
$\mathcal{B}_t$	Set of feasible blocks that can beccccccccc extracted at time t
$\mathcal{M}_t$	Matrix of possible and feasible actions (blocks for extraction)
$\mathcal{Q}_t$	A matrix of dimension (N_x, N_y), representing the inferred Q-value $q\ (s_t, a_t)$
γ	Discount factor
ε	Probability of chosen a random action
$\varepsilon_{\text{decay}}$	Updating rate of ε at each time step as $\varepsilon_{t+1} \leftarrow \varepsilon_t \cdot \varepsilon_{\text{decay}}$
$p_{t,i,j}$	Probability of extraction block at location (i, j) in the matrix $\mathcal{M}_t$ at time t
(N_x, N_y, N_z)	Number of blocks in the X, Y and Z axes
T	Total amount of periods (days)

Acknowledgements We acknowledge the support of the Mitacs Accelerate Program, and the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number RGPIN-2017-04200 and RGPAS-207-5057956.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Avalos S, Ortiz JM (2020) Recursive convolutional neural networks in a multiple-point statistics framework. Comput Geosci 141:104522

Avalos S, Ortiz JM (2021) Fundamentals of deep Q-learning. Predictive Geometallurgy and Geostatistics Lab, Queen's University, Annual Report 2021, paper 2021-02, 14–21

Bai T, Tahmasebi P (2020) Hybrid geological modeling: combining machine learning and multiple-point statistics. Comput Geosci 142:104519

Barnett RM, Manchuk JG, Deutsch CV (2014) Projection pursuit multivariate transform. Math Geosci 46(3):337–359

Bourgault G, Marcotte D (1991) Multivariable variogram and its application to the linear model of coregionalization. Math Geol 23(7):899–928

Chen Z, Liu X, Yang J, Little E, Zhou Y (2020) Deep learning-based method for SEM image segmentation in mineral characterization, an example from Duvernay Shale samples in Western Canada Sedimentary Basin. Comput Geosci 138:104450

Coward S, Vann J, Dunham S, Stewart M (2009) The primary-response framework for geometallurgical variables. In: Seventh international mining geology conference, pp 109–113

de Figueiredo LP, Schmitz T, Lunelli R, Roisenberg M, de Freitas DS, Grana D (2021) Direct multivariate simulation-a stepwise conditional transformation for multivariate geostatistical simulation. Comput Geosci 147:104659

Dowd P (1994) Risk assessment in reserve estimation and open-pit planning. Trans Inst Min Metall Sect A Min Ind 103:A148

Gilani S-O, Sattarvand J, Hajihassani M, Abdullah SS (2020) A stochastic particle swarm based model for long term production planning of open pit mines considering the geological uncertainty. Resour Policy 68:101738

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Hochbaum DS (1998) The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. In: International conference on integer programming and combinatorial optimization. Springer, pp 325–337

Jelvez E, Morales N, Ortiz JM (2021) Stochastic final pit limits: an efficient frontier analysis under geological uncertainty in the open-pit mining industry. Mathematics 10(1):100

Koch P-H, Rosenkranz J (2020) Sequential decision-making in mining and processing based on geometallurgical inputs. Miner Eng 149:106262

Kumar A, Dimitrakopoulos R (2021) Production scheduling in industrial mining complexes with incoming new information using tree search and deep reinforcement learning. Appl Soft Comput 110:107644

Lamghari A, Dimitrakopoulos R (2020) Hyper-heuristic approaches for strategic mine planning under uncertainty. Comput Oper Res 115:104590

Lerchs H (1965) Optimum design of open-pit mines. Trans CIM 68:17–24

Leuangthong O, Deutsch CV (2003) Stepwise conditional transformation for simulation of multiple variables. Math Geol 35(2):155–173

Liu J, Osadchy M, Ashton L, Foster M, Solomon CJ, Gibson SJ (2017) Deep convolutional neural networks for Raman spectrum recognition: a unified solution. Analyst 142(21):4067–4074

Matignon L, Laurent GJ, Fort-Piat NL (2006) Reward function and initial values: better choices for accelerated goal-directed reinforcement learning. In: International conference on artificial neural networks. Springer, pp 840–849

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533

Monkhouse P, Yeates G (2018) Beyond naive optimisation. In: Advances in applied strategic mine planning. Springer, pp 3–18

Mueller U, Boogaart K, Tolosana-Delgado R (2017) A truly multivariate normal score transform based on Lagrangian flow. In: Geostatistics Valencia 2016. Springer, pp 107–118
Nelis SG, Ortiz JM, Morales VN (2018) Antithetic random fields applied to mine planning under uncertainty. Comput Geosci 121:23–29
Paduraru C, Dimitrakopoulos R (2019) Responding to new information in a mining complex: fast mechanisms using machine learning. Min Technol 128(3):129–142
Santos JE, Xu D, Jo H, Landry CJ, Prodanović M, Pyrcz MJ (2020) PoreFlow-Net: a 3d convolutional neural network to predict fluid flow through porous media. Adv Water Resour 138:103539
Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489
Sun T, Li H, Wu K, Chen F, Zhu Z, Hu Z (2020) Data-driven predictive modelling of mineral prospectivity using machine learning and deep learning methods: a case study from southern Jiangxi Province, China. Minerals 10(2):102
Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press, Cambridge
Tahmasebi P, Kamrava S, Bai T, Sahimi M (2020) Machine learning in geo-and environmental sciences: from small to large scale. Adv Water Resour 142:103619
Tay Y, Dehghani M, Bahri D, Metzler D (2020) Efficient transformers: a survey. ACM Comput Surv (CSUR) 55(6):1–28
Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Proc Syst 30
Watkins CJ, Dayan P (1992) Q-learning. Mach Learn 8(3):279–292
Whittle J (1988) Beyond optimization in open pit design. In: canadian conference on computer applications in the mineral industries. Balkema Rotterdam, pp 331–337
Whittle J (1993) Four-D Whittle open pit opimisation software. User Manual, Whittle Programming Ltd, Melbourne, NSW, Aus
Whittle J, Rozman L (1991) Open pit design in 90's. Proceedings mining industry optimization conference, AusIMM, Sydney, Australia
Whittle J (1997) Optimization in mine design. WH Bryan Mining Geology Research Centre, Brisbane, Australia
Xiong Y, Zuo R, Carranza EJM (2018) Mapping mineral prospectivity through big data analytics and a deep learning algorithm. Ore Geol Rev 102:811–817
Yun W, Liu Y, Kovscek AR (2020) Deep learning for automated characterization of pore-scale wettability. Adv Water Resour 144:103708