Computer Methods in Applied Mechanics and Engineering 435 (2025) 117617

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Inverse material design using deep reinforcement learning and homogenization

V. Würz *, C. Weißenfels

Institute of Materials Resource Management, University of Augsburg, Germany
Center of Advanced Analytics and Predictive Sciences, University of Augsburg, Germany

ARTICLE INFO

Keywords:
Inverse material design
Deep reinforcement learning
Advantage actor–critic model
Homogenization
Finite element method
Micro-structural parameter optimization

ABSTRACT

This study presents an approach to solving an inverse problem through the application of Deep Reinforcement Learning (DRL) coupled with homogenization. The underlying objective is to determine the micro-structural parameters of a composite material, including particle radius, Young's moduli and Poisson's ratios in order to achieve a specific target bulk modulus at the macro-scale using DRL. This approach is later extended to a multi-objective task that also decreases total material weight by incorporating density and volume fraction as additional parameters. Employing homogenization under Periodic Boundary Conditions (PBCs), a 3D mesh is analyzed comprising a matrix and particles to identify a Representative Volume Element (RVE), thereby reducing computational complexity and allowing efficient Finite Element Method (FEM) calculations in subsequent steps. An Advantage Actor–Critic (A2C) model is employed, using the FEM analysis as feedback, to iteratively adjust the micro-structural parameters and incrementally approach the target properties. A Genetic Algorithm (GA) is implemented to fine-tune the hyper-parameters of the neural network, enhancing the ability of the model to effectively explore different parameter combinations. While tuning of hyperparameters is conducted in 2D, findings are transferred to 3D to verify this approach in more realistic scenarios. The DRL approach is compared in a partial manner with the Bayesian optimization, a well established algorithm type in the field of inverse material design, in order to give an idea of the different application circumstances. For A2C, the development of a specific reward function enables the DRL algorithm to approach the solution consistently leading to many different solutions of the inverse problem. In addition, the search space is decreased significantly without limiting the variety of determined micro-configurations by using effective balancing of exploration and exploitation. Extensive tuning of hyperparameters enables the adjustment of the algorithm to specific desired outcomes as the increase of sample efficiency or quantity of diverse solutions. Using this hands-on learning approach can unveil innovative material configurations by exploring a broader range of design scenarios, including those that might be overlooked or deemed non-intuitive by traditional methodologies. Hence. it is positioned to establish an alternative methodology for designing novel material combinations with tailored properties.

1. Introduction

In the rapidly evolving field of materials engineering, the ability to design complex heterogeneous materials with pre-determined macroscopic properties poses significant advantages. This process involves solving inverse problems, where the goal is to determine

* Corresponding author at: Institute of Materials Resource Management, University of Augsburg, Germany.
E-mail address: valentin.wuerz@uni-a.de (V. Würz).

https://doi.org/10.1016/j.cma.2024.117617
Received 30 July 2024; Received in revised form 19 November 2024; Accepted 26 November 2024
Available online 5 December 2024

the microscopic parameters that will result in desired macroscopic characteristics. Traditionally, these tasks constitute significant challenges due to their inherent complexity and the computational burden of exploring vast parameter spaces. In addition, the realization of such designed materials has not been trivial. However, advancements in manufacturing techniques, particularly 3D printing, have revolutionized the landscape by enabling the fabrication of complex-shaped and diverse compounded micro-structures. This technological progress necessitates a more sophisticated approach to inverse material design, as new micro-structural combinations can now be realized. In this context, it is necessary to develop algorithms capable of continuously exploring the expansive design space of possible microscopic property combinations. Brute force algorithms are unsuitable in this environment as they rely on finding better states by chance without a concept of reducing the overall state space.

Genetic algorithms (GAs) have been successfully applied to solve inverse material design problems with randomly dispersed particulates suspended in a homogeneous binding matrix. By mimicking the process of natural selection, GAs find optimal solutions within complex search spaces, as discussed in [1]. GAs have proven effective in addressing the non-convex optimization challenges characteristic of material design, benefiting from their ability to operate without derivatives and navigate complex, multidimensional search spaces. However, while GAs have a fast computation speed, they largely depend on initial states leading to likely local optima. Additionally, they struggle with complex and continuous state representations.

In [2] an approach of micro-structural materials design using generative adversarial networks (GANs) is proposed. This methodology successfully identifies design variables with desired dimensionality and is able to capture complex micro-structural material characteristics due to the special design of the neural network architecture. At the same time the process of generating enough accurate image samples for an effective training process, where both networks improve simultaneously, is notoriously difficult and unstable. Back and forth conversion steps from image representations to conventional simulation tools like finite element modeling (FEM) for evaluation purposes add additional complexity.

Some of the challenges that GANs face, like the lack of diversity and the disability to generate novel designs outside the domain of the training data are addressed in [3]. Applied to the example of real-world airfoil design a variant of GANs named “performance augmented diverse generative adversarial network” (PaDGAN) was developed. This approach is able to improve diversity and achieved higher mean quality samples in comparison to vanilla GANs.

An improvement over issues like mode collapse or oscillations that are common in GANs can also be found in Variational Autoencoders (VAEs). In [4] VAEs coupled with neural network surrogates as replacement for the FE-solver were used for topology optimization. This coupling proved successful in turning the high dimensional optimization problem into a low dimensional one, while offering computational efficiency with high accuracy in finding an optimal arrangement of materials within a design domain. The advantages of generative networks (GNs) like GANs and VAEs over GAs regarding the local optima problem or dependency on initial states is at cost of lower optimization efficiency. Both of these approaches require extensive pre-training on large datasets and in many cases the proposed samples are generated within its initial distribution regime. This is related to the issue of high sample correlation during the optimization process that many GNs are exposed to.

To address the mentioned challenges an alternative approach for material design was investigated by Sui et al. [5]. Here, Deep Reinforcement Learning (DRL) has been applied in the context of designing digital materials (DMs) as a mathematical representation for composites. In this approach two collaborative agents were able to optimize the distribution pattern of DMs based on interaction with FEM simulations. Major advantages of DRL in this context is the reduction of sample correlation, high sample efficiency and the lack of pre-training. DRL’s intelligent agents iteratively interact with a simulation environment, adjusting their search strategies based on immediate feedback and decreasing the search space significantly based on knowledge gathered in the simulation. This knowledge is stored intrinsically in the structure of the neural networks (NNs). Following this process enables efficient reduced exploration strategies of micro-configurations without neglecting promising states.

To solve the micro-structural design task in this work, DRL is coupled with a FEM simulation to inversely design continuous 3-dimensional composite materials in this work. It is additionally supported by homogenization breaking down the design complexity further by reducing the size of the material that is optimized utilizing the concept of a Representative Volume Element (RVE). In the realm of DRL, it is generally differentiated between model-free and model-based, value-based and policy-based, and off-policy and on-policy. For this work, the Advantage Actor–Critic (A2C) algorithm is used as it combines the benefits of both value-based and policy-based methods. It falls into the category of model-free, on-policy algorithms. As different studies in [6,7] show, A2C, due to its straightforward implementation of continuous action and state spaces and sample efficiency within increasingly complex environments, is assumed to outperform the more traditional Deep-Q Network when correctly tuned in this specific environment. Deep-Q Networks are explained in [8]. More information on the A2C algorithm can be found in [9,10]. A2Cs ability to efficiently balance exploration and exploitation decreases the search space significantly without limiting the variety of determined micro-configurations. Adjusting the hyperparameters systematically based on Genetic algorithms (GAs) facilitates specific desired outcomes as the increase of sample efficiency or quantity of diverse solutions. A specifically developed reward function additionally supports the algorithm in finding many different solutions of the inverse problem consistently. Bayesian optimization (BO) is another well established algorithm in the context of inverse material design, see [11,12]. To better understand how DRL and BO relate to each other in a computational manner applied to the underlying design objective, a partial comparison is conducted in 2D. In the following, the general concepts of homogenization including the determination of effective constitutive equation, proper boundary conditions and the discretization of the micro-equilibrium are explained. It is elaborated on the application of periodic boundary conditions in the context of the finite element modeling and as a consequence the approach of determining a Representative Volume Element is addressed. Afterwards the most important concepts of DRL including the role of the neural networks, reward function, loss calculation and back-propagation is elaborated. At the end it is worked through the results of the specified applied approach of inversely designing the micro-structure by determining the numerical RVE size in 2 and 3 dimensions and refining hyperparameters

to achieve tailored strategies. To address a more realistic scenario the findings are applied to a 3-dimensional list of real materials that are combined to not only achieve a specific target bulk modulus but also to decrease the overall weight of the material, rendering the algorithm from a single-objective to a multi-objective one. For this purpose, volume fraction of particles and material densities are considered additionally as training parameters.

2. Homogenization

2.1. Effective macroscopic material properties

The general idea of homogenization covers the determination of the effective constitutive equation to avoid the need for resolving the heterogeneous micro-structure in a large-scale simulation. Here, the effective macroscopic material properties are derived from a so called Representative Volume Element (RVE). This RVE is selected as a fraction of the material that in approximation represents the essential micro-structural characteristics so that microscopic stress and strain within this RVE represent the macroscopic material properties statistically [13]. To obtain the effective constitutive equations for material behavior, a boundary value problem is solved on the micro-scale. The partial differential equation derives from the balance of linear momentum. Applied to the material domain $\mathcal{B}$, qualifying as an RVE, every mass point $\mathbf{x} \in \mathcal{B}$ experiences internal stresses that, in absence of external and inertia forces, reduces to a stress tensor free of divergence [14] or [15]

$$\operatorname{div} \boldsymbol{\sigma}(\boldsymbol{x}) = 0 \quad \text{in } \mathcal{B} \tag{1}$$

with $\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$. For isotropic linear elastic materials the stress–strain relationship simplifies using Hooke's law to

$$\boldsymbol{\sigma}(\boldsymbol{x}) = 3\kappa(\boldsymbol{x})\boldsymbol{\epsilon}_{vol}(\boldsymbol{x}) + 2\mu(\boldsymbol{x})\boldsymbol{\epsilon}_{dev}(\boldsymbol{x}). \tag{2}$$

The left part denotes the volumetric stress $\boldsymbol{\sigma}_{vol} = \kappa(\boldsymbol{x})\boldsymbol{\epsilon}_{vol}$ and the right part the deviatoric stress $\boldsymbol{\sigma}_{dev}(\boldsymbol{x}) = 2\mu\boldsymbol{\epsilon}_{dev}$. Due to the heterogeneity the bulk modulus κ and the shear modulus μ can vary across $\mathcal{B}$. The averaged stress and strain tensor result from an integration of the corresponding quantity divided by the volume $|\mathcal{B}| = \int_{\mathcal{B}} dV$

$$\langle \boldsymbol{\sigma} \rangle_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \int_{\mathcal{B}} \boldsymbol{\sigma}(\boldsymbol{x})\, dV, \quad \langle \boldsymbol{\epsilon} \rangle_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \int_{\mathcal{B}} \boldsymbol{\epsilon}(\boldsymbol{x})\, dV. \tag{3}$$

The averaged quantities $\langle \boldsymbol{\sigma} \rangle_{\mathcal{B}}$ and $\langle \boldsymbol{\epsilon} \rangle_{\mathcal{B}}$ equal the effective properties on the macro-scale, i.e. $\boldsymbol{\sigma}^* = \langle \boldsymbol{\sigma} \rangle_{\mathcal{B}}$ and $\boldsymbol{\epsilon}^* = \langle \boldsymbol{\epsilon} \rangle_{\mathcal{B}}$. Based on these averaged stresses and strains in (3) the effective bulk κ^* and shear moduli μ^* on macro-scale results from (3) in the linear isotropic elastic case (2) exploiting

$$\kappa^* = \frac{\operatorname{tr}\langle \boldsymbol{\sigma} \rangle_{\mathcal{B}}}{3 \operatorname{tr}\langle \boldsymbol{\epsilon} \rangle_{\mathcal{B}}}, \quad \mu^* = \frac{\left\| \langle \boldsymbol{\sigma}_{dev} \rangle_{\mathcal{B}} \right\|}{2 \left\| \langle \boldsymbol{\epsilon}_{dev} \rangle_{\mathcal{B}} \right\|}. \tag{4}$$

More information on homogenization can be found in [16,17] or [18].

2.2. Boundary conditions and discretized micro-equilibrium

Originating from the requirement of fulfilling Hill's criterion [19], three different boundary conditions can be applied. In this work, Periodic Boundary Conditions (PBCs) are used with the strains and stresses being periodic at the level of the periodicity cell, defined as an RVE. Although a real heterogeneous material does not necessarily show perfect periodicity, it has been shown that PBCs can provide reasonable estimates on the effective moduli [20–22]. In addition, BCs are either driven by average stress or average strain enforcing $\boldsymbol{\sigma} = \boldsymbol{\Sigma}$ or $\epsilon = \boldsymbol{\mathcal{E}}$, respectively. In this work, the second option is applied. To establish PBCs for the analysis of an RVE, the boundary is divided into two counterpart segments, designated as $\partial\mathcal{B}^+$ and $\partial\mathcal{B}^-$ depending on the cutting direction. Together, these two parts form the complete boundary $\partial\mathcal{B}$ with $\boldsymbol{x}^+$ on $\partial\mathcal{B}^+$ having a corresponding point $\boldsymbol{x}^-$ on $\partial\mathcal{B}^-$. Importantly, the unit normal vectors at these pairs are oriented such that $\boldsymbol{n}^-$ is the negative of $\boldsymbol{n}^+$ in relation to the coordinate system. In case of average strain driven micro-structure two requirements must be enforced at the boundary

$$\boldsymbol{u}^+ - \boldsymbol{u}^- = \langle \boldsymbol{\epsilon}(\boldsymbol{x}^+ - \boldsymbol{x}^-) \rangle_{\partial\mathcal{B}} \quad \text{and} \quad \boldsymbol{t}^- = -\boldsymbol{t}^+ \quad \text{on} \quad \partial\mathcal{B}. \tag{5}$$

Due to Cauchy stress theorem $\boldsymbol{t}^{\pm} = \boldsymbol{\sigma}\boldsymbol{n}^{\pm}$. These conditions ensure that the displacement field $\boldsymbol{u}$ is periodic across the boundary, while the stress field $\boldsymbol{t}$ exhibits antiperiodicity [22]. While linear displacement boundary conditions only enforce periodicity in displacement and traction boundary conditions antiperiodicity in traction, PBCs enforce both [23]. The analyzed structure is composed of a matrix embedded with spherical particles each characterized by distinct Young's moduli and Poisson's ratios as illustrated in Fig. 1(left). The particles, with a predefined radius, are distributed randomly within the matrix. To solve the boundary value problem the FEM is applied in which the micro-structure is subdivided into individual elements. There are different ways of constructing the mesh and resolving heterogeneous materials. As opposed to irregular meshes that link elements conforming to the micro-structure, this work uses regular meshes in the form of 8-node hexahedral elements to discretize the RVE. This non-conforming approach does not require the boundaries of the finite elements to match with material interfaces and enables the adaptive refinement or coarsening in regions of interest without needing to adjust the entire mesh, which decreases the computational effort [24]. The rapid generation of structured internal meshes lead to the absence of finite element distortion caused by the microstructure [18]. In [25] the different meshing approaches are compared in numerical studies. Without a conforming mesh, material discontinuities are located within the element. A convergence study regarding the necessary number of finite elements

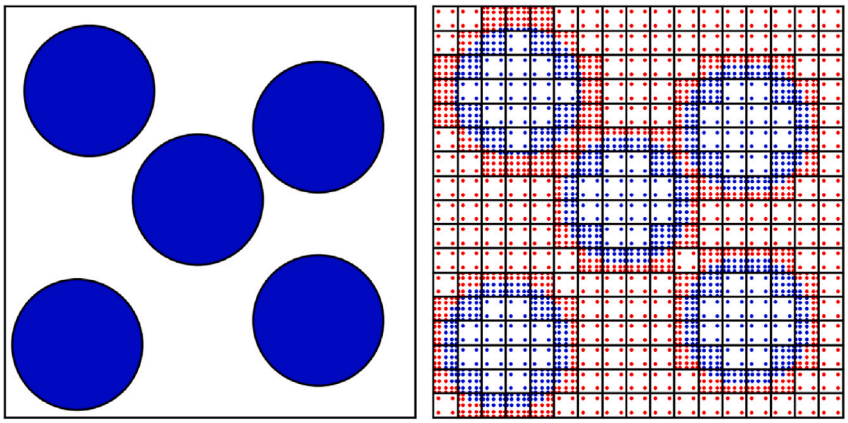

Fig. 1. RVE in 2D with particles in matrix (left) and discretization (right) including Gauss integration points (red: matrix; blue: particles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

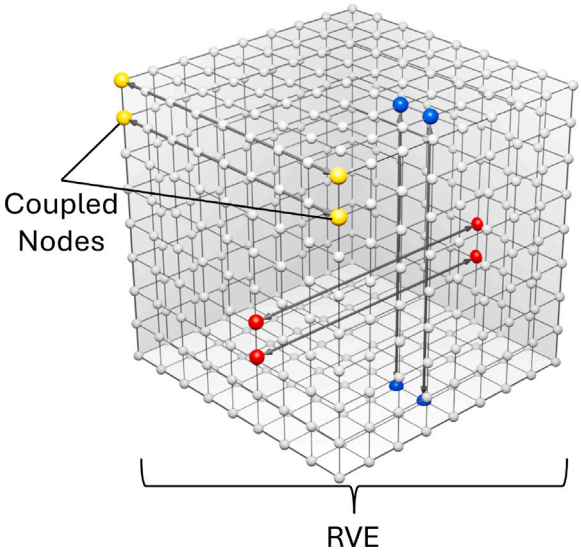

Fig. 2. RVE mesh with boundary-coupling implementation of PBCs.

is conducted in 4.1. To better capture the geometry and material behavior a higher rank of quadrature needs to be employed. Elements without particle intersections use two Gauss points in each dimension, so $N_g = 2$ in case of linear shape functions for numerical integration. Conversely, interface elements with intersecting particles require higher-order integration, utilizing $N_g = 5$ Gauss points per direction (Fig. 1) as recommended in [26]. Various approaches exist for imposing PBCs. First, the pairing nodes must be determined. The description from [27] was used in this work by pairing nodes at opposite planes of the RVE's surface as illustrated in Fig. 2. To fulfill the conditions in (5), the Lagrange multiplier [28,29], penalty methods or direct elimination [30] are possible. A more efficient option can be found in [31] which bases the PBCs on a division into dependent and independent degrees of freedom due to periodicity. This allows the linear system of equations to be condensed only retaining the independent degrees of freedom.

2.3. Determination of the RVE

To determine the RVE an ensemble averaging and a sample scaling must be conducted [24,32]. The first estimation involves analyzing small samples that, individually, may not exhibit isotropy. For a pre-defined sample size, ensemble averaging over N

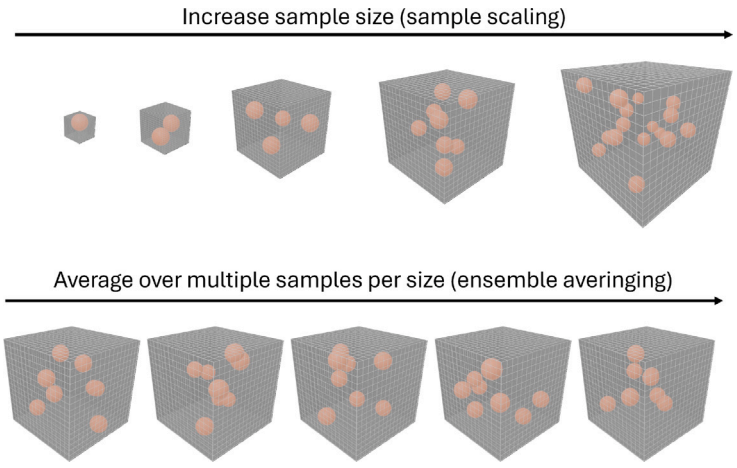

Fig. 3. Considering successively larger samples (sample scaling) and multiple samples per sample size (ensemble averaging) to determine RVE size.

samples is conducted, where each sample shows a different distribution of particles. This continues until the calculated averaged macroscopic properties of κ and μ

$$\langle \kappa \rangle \equiv \frac{1}{N} \sum_{i=1}^{N} \kappa_i, \quad \langle \mu \rangle \equiv \frac{1}{N} \sum_{i=1}^{N} \mu_i. \tag{6}$$

converge under a specific convergence tolerance

$$\left| \frac{P^{k+1} - P^k}{P^k} \right| \leq \text{TOL}. \tag{7}$$

The symbol P^K denotes either κ or μ for a given sample k. In the second step of estimation, the size of the domain $\mathcal{B}$ is increased, and its response is once again evaluated for convergence within a predefined tolerance threshold, as specified in (7). By progressively scaling these samples and accounting for the distribution of randomly dispersed particles within the matrix, the inherent anisotropy of individual samples in the heterogeneous material is reduced. This approach allows the statistical RVE to capture a broad range of potential mechanical responses reflective of the material's global structure. As noted in [24], smaller sample RVEs often exhibit anisotropic effects, leading to what is known as 'isotropic inconsistency'. However, by enlarging these samples and analyzing the response of multiple averaged ensembles with random inclusions, the level of anisotropy can be diminished to an acceptable threshold, facilitating a more representative statistical RVE. In Fig. 3 this scheme of sample scaling (top) and ensemble averaging (bottom) over multiple samples per sample size is illustrated for 3-dimensional RVEs. For a general explanation on the computation of bounds on effective properties using harmonic averaging see [13,33]. Comprehensive insights into the statistical approaches of defining the RVE size and its derivation from the Statistical Volume Element can be found in [34] or [35].

3. Micro-structure design using actor–critic reinforcement learning

Inverse problems, like the underlying, infer unknown parameters or functions from observed data and are typically ill-posed because they frequently lack one or more of the following properties: existence, uniqueness, and stability of the solution [36]. This instability arises naturally in these problems due to the indirect and often noisy relationship between the measured data and the sought parameters as explained in detail in [37]. The effective macroscopic behavior of the heterogeneous material depends on the size and number of particles, as well as on Young's modulus and Poisson's ratio of both the matrix and particles. Using an A2C reinforcement learning approach these properties are adjusted based on the outcome of Neural Networks (NNs). These NNs are trained based on rewards gathered in an environment including an FE-simulation of the current micro-configuration. More information on NNs in the context of deep learning can be found in [38–40].

3.1. Role of neural networks and forward-propagation

In the context of this work, two NNs, the actor and the critic network, construct the agent to predict, execute and evaluate the learned strategy (see Fig. 4). Detailed information of their structure and activation functions can be found in Section 4.2 Table 1. A detailed explanation of activation functions is given in [41,42]. The actor network outputs a probability distribution related to the current best modification of the micro-configuration. The critic provides an evaluation of the quality of the current strategy that leads to this probability distribution. A tuple of the current state $\boldsymbol{S}_t$ in the episode step t that consists of Young's modulus E and Poisson's ratio ν of particles and matrix as well as the particle radius r are the inputs of both NNs.

$$\boldsymbol{S}_t = [E_p, E_m, \nu_p, \nu_m, r_p]. \tag{8}$$

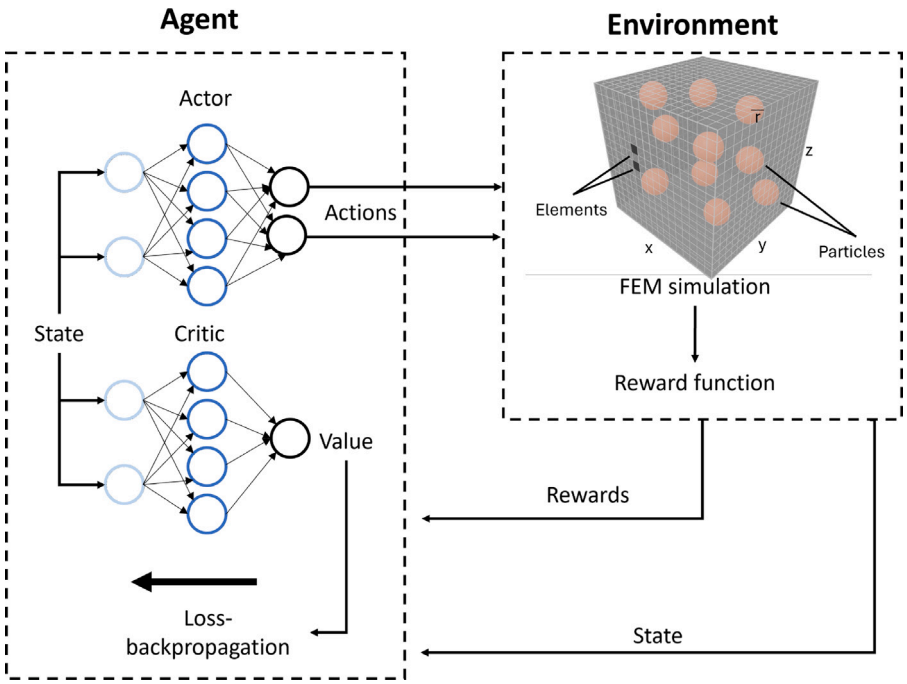

Fig. 4. Basic A2C scheme.

Table 1
Network architecture for Actor and Critic networks.

Network	Layer type	Number of neurons	Activation function
Actor	Input layer	4	–
	Dense layer 1	64	GeLU
	Dense layer 2	128	GeLU
	Dense layer 3	256	GeLU
	Output layer	10	Linear
Critic	Input layer	4	–
	Dense layer 1	128	GeLU
	Dense layer 2	256	GeLU
	Dense layer 3	512	GeLU
	Dense layer 4	256	GeLU
	Output layer	1	Linear

These five design variables represent the most basic form of parameterization of our state. In Section 4.6 the state vector is complemented with additional entries, see (22). An episode in this context is defined as multiple sequential steps $\boldsymbol{S}_t$ of forward propagation, action execution and received reward. As output $\boldsymbol{o}_t$ the actor network in this work does not only estimate a mean μ for all potentially best adjustments of each action but also its standard deviation σ

$$\boldsymbol{o}_t^{\mu} = [\mu_E^p, \mu_E^m, \mu_\nu^p, \mu_\nu^m, \mu_r^p], \quad \boldsymbol{o}_t^{\sigma} = [\sigma_E^p, \sigma_E^m, \sigma_\nu^p, \sigma_\nu^m, \sigma_r^p]. \tag{9}$$

The action comprises a change of the Young's moduli, Poisson's ratios and radii. Using the Box–Muller formula [43]

$$Z = \sqrt{-2\,\ln(U_1)}\cos\left(2\pi U_2\right) \tag{10}$$

with randomly generated variables $U_1, U_2 \in \{0, 1\}$ standard normal random variables $\boldsymbol{Z}$ are calculated for every entry in the action space that are derived in approximation of normal distributed variables $\boldsymbol{Z} \sim \mathcal{N}(\mu, \sigma)$ with $\mu = 0$ and $\sigma = 1$. A discrete action tuple is sampled using different variables $\boldsymbol{Z}$ and the means and standard deviations from the output (9)

$$\boldsymbol{A}_t = \boldsymbol{o}_t^{\mu} + \boldsymbol{o}_t^{\sigma} * \boldsymbol{Z}. \tag{11}$$

Due to the scaling of standard normal samples by σ and shifting by μ stochastic actions are obtained according to the learned policy parameters. The adjusted micro-configuration defines the basis for a new FEM-computation together with the determination of the macroscopic properties according to (4) (see also Fig. 4). The fundamental objective of the algorithm is the proper adjustment of state $\boldsymbol{S}_t$ that leads to the desired macroscopic properties. The primary reason for using NNs to achieve the above mentioned relationship is their ability to learn and represent complex, high-dimensional mappings from inputs to outputs. Unlike for example

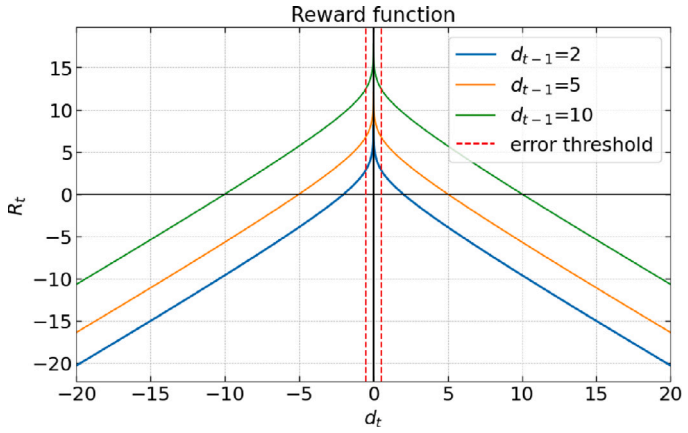

Fig. 5. Reward function in dependency of previous-distance values d_{t-1}.

tabular methods that store values for every possible state–action pair, NNs approximate these values, making them very useful in environments with continuous state spaces like the underlying micro-configurations in this material design task. Ultimately, NNs are able to provide a compact, generalized and scaleable way of representing the connection between current micro-configuration and its ideal adjustment to reach the target macroscopic properties.

3.2. Reward engineering

For an efficient learning process a proper reward must be defined guiding the algorithm closer to the target bulk modulus κ_{target}. In this work, the agent collects positive or negative reward signals on interaction with the FE-solver based on the quality of the current action and its influence on the mechanical properties of the structure. The simulation returns a positive or negative absolute distance to the target bulk modulus d_t in the current step t compared to the distance in the previous adjustment d_{t-1} as result of the current micro-mechanical configuration of the material

$$d_t = \kappa_{target} - \kappa_t, \quad d_{t-1} = \kappa_{target} - \kappa_{t-1}. \tag{12}$$

To guide the agent in the direction of the desired bulk modulus a reward is developed

$$R_t = -a \cdot \ln\left(\frac{|d_t| + \epsilon}{|d_{t-1}| + \epsilon}\right) + (|d_{t-1}| - |d_t|). \tag{13}$$

This formulation ensures an exponential increase of the reward if the distance to the target bulk modulus decreases as plotted in Fig. 5. It can be observed that the distance d_t needs to be smaller than the distance d_{t-1} in order to gain rewards. This behavior provides the agent with an incentive to further refine the achieved bulk modulus, even though it has already received a significant amount of rewards for reducing the distance from the initial configuration to the current configuration. The additional numerical parameters a and ϵ allow to adjust the ascent of the natural logarithmic function while still being defined at every value of d_t including zero. For this purpose ϵ is chosen to be a very small number unequal to zero and $a = 0.7$, however different values may be used dependent on specific reward preferences. In addition, the agent gains another 10 reward when it is able to reach the target bulk modulus within a pre-defined error threshold of 0.33% (Fig. 5). In this case, the task is considered solved and a micro-configuration is found that achieves the defined macroscopic property in a sufficient manner.

3.3. Loss calculation and back-propagation

To effectively adjust the parameters of the NNs in a way that achieves the target macro properties, a loss function is minimized or maximized together with back-propagation of the NN at the end of each episode, respectively. The general scheme of the last step of one episode is illustrated in Fig. 4. As mentioned before, A2C uses two NNs, one that determines which action to take (actor) and another that evaluates the action (critic). By taking both into account, the model learns more effectively from the discrepancies between the predicted value of the critic and the actual return from the environment, which is also called the advantage $A(\boldsymbol{S}_t, \boldsymbol{A}_t)$

$$A(\boldsymbol{S}_t, \boldsymbol{A}_t) = Q^{\pi}(\boldsymbol{S}_t, \boldsymbol{A}_t) - V(\boldsymbol{S}_t). \tag{14}$$

The value-function $V(\boldsymbol{S}_t)$ is the estimated quality of the state or micro-configuration predicted by the critic network. The action–value function $Q^{\pi}(\boldsymbol{S}_t, \boldsymbol{A}_t)$ represents the expected return that is received starting from state $\boldsymbol{S}_t$ and taking action $\boldsymbol{A}_t$ following policy π

$$Q^{\pi}(\boldsymbol{S}_t, \boldsymbol{A}_t) = R_t + \gamma V^{\pi}(\boldsymbol{S}_{t+1}). \tag{15}$$

The action–value function is an objective measure of the quality of the action or the modification of the micro-mechanical properties starting from the current configuration or state of the environment. R_t defines the immediate reward received for executing the action. The action–value function is formulated including the value-function in the next step $t+1$. It needs to be noted that $V^{\pi}(\boldsymbol{S}_{t+1})$ is different to $V(\boldsymbol{S}_t)$ formulated in (14). The value-function is not estimated by the critic but it represents the cumulative summed and discounted returns in all prior steps of the current episode

$$V^{\pi}(\boldsymbol{S}_t) = \left[\sum_{k=0}^{n} \gamma^k R_{t+k+1} \mid \boldsymbol{S}_t\right]. \tag{16}$$

The vertical bar | indicates that each discounted reward in the summation loop starts in state $\boldsymbol{S}_t$ followed by forward propagation, action output, action execution and reward return for each step in the episode. For this all received rewards must be stored. The discount factor balances how future rewards are valued compared to immediate rewards. In addition, it avoids an infinite growth of future rewards and leads to a finite and stable expected return improving the stability and convergence of the learning process. The advantage $A(\boldsymbol{S}_t, \boldsymbol{A}_t)$ is calculated at the end of each episode. In addition, both the actor and the critic have their own loss functions, with which the weights of both neural networks are updated. Among different ways, see for instance [44], the critic loss is computed by the mean squared error (MSE) of the advantage $A(\boldsymbol{S}_t, \boldsymbol{A}_t)$ given in (14) is used

$$L_{\text{critic}} = \frac{1}{N}\sum_{t=1}^{N} A(\boldsymbol{S}_t, \boldsymbol{A}_t)^2. \tag{17}$$

Here, N is the number of steps within one episode. This ensures that the model learns from the entire trajectory of that episode.

In environments with continuous action spaces like in this work the actor network outputs parameters of a probability distribution π that define the policy (see (9)). The actor loss is calculated by taking the logarithm of this likelihood multiplied by the advantage function

$$L_{\text{actor}} = -\frac{1}{N}\sum_{t=1}^{N} \ln\big(\pi(\boldsymbol{A}_t|\boldsymbol{S}_t)\big) A(\boldsymbol{S}_t, \boldsymbol{A}_t), \tag{18}$$

where $\pi(\boldsymbol{A}_t|\boldsymbol{S}_t)$ is the policy's probability of taking action $\boldsymbol{A}_t$ in state $\boldsymbol{S}_t$. This term is then averaged over all prior steps t of the episode. Taking actions based on this probability distribution balances exploration and exploitation naturally based on the current weights of the actor network. The loss function for the actor is designed to encourage actions that lead to higher returns. Similar to the critic loss the actor loss is also taken as the mean over all time steps within one episode. In the optimization process actor and critic loss are combined together with the weighted entropy to form the total loss L_{total}

$$L_{total} = w_c L_{critic} + w_a L_{actor} - \beta \mathcal{H}_{sum}. \tag{19}$$

The entropy term $\mathcal{H}$ corresponds to the average uncertainty in the outcomes of the continuous distribution returned from the actor network (9), where i iterates through all entries of the standard deviation tuple

$$\mathcal{H}_i(\sigma_i) = \frac{1}{2}\ln\big(2\pi\epsilon\sigma_i^2\big), \quad \mathcal{H}_{sum} = \sum_{i=1}^{I} \mathcal{H}_i. \tag{20}$$

In (19) $\mathcal{H}$ is scaled by the entropy weight β, which regulates how much importance the policy should put on discovering new micro-mechanical configurations as compared to exploit the already learned correlations that are stored implicitly in the artificial neural networks. A more detailed explanation of entropy and exploration–exploitation balancing in A2C is given in [7,45]. The weights w_c and w_a in (19) can be balanced based on the specific environment. In this work w_a is always set to 1 while critic loss and entropy weight are considered as hyperparameters. To optimize these weights the stochastic and gradient-based optimizer Adam is chosen. This algorithm computes individual adaptive learning rates for different parameters based on the first and second moments of the gradients [46]. The Adam optimizer has proven to be very robust and works well with different gradient scales due to its dynamic learning rate [47,48]. During the adjustment of micro-mechanical properties the macroscopic behavior is expected to fluctuate significantly (see (10)), which leads to very volatile rewards. As gradients are calculated directly from rewards, robustness is improved. Using the respective gradients both the actor and the critic networks are updated simultaneously. The updates are scaled by a learning rate α, which controls the step size in the gradient descent/ascent process

$$\theta^a_{new} = \theta^a_{old} - \alpha_a \nabla^a_\theta L_{total}, \quad \theta^c_{new} = \theta^c_{old} - \alpha_c \nabla^c_\theta L_{total}. \tag{21}$$

The learning rate for actor and critic can be chosen independently. A more profound explanation on optimizing NNs is given in [49,50]. After this optimization step the gradients are cleared and the new episode is started with the updated network parameters.

3.4. General process flow

The algorithm starts with an initialization of the actor and critic network as well as the setup of the FEM environment. Starting with the first training loop the initial micro-configuration is defined. In the first step, forward propagation through actor and critic network are performed on initial state. The first action tuple is derived from estimated mean and standard deviation for all adjustable micro-properties (given in (9)). This action is scaled and executed within the environment, changing the initial micro-configuration to a new state. The FEM calculates the mechanical response of the new RVE and returns the difference to the desired target bulk modulus in the form of positive or negative rewards (5). The current reward is stored within a batch. This process is repeated in

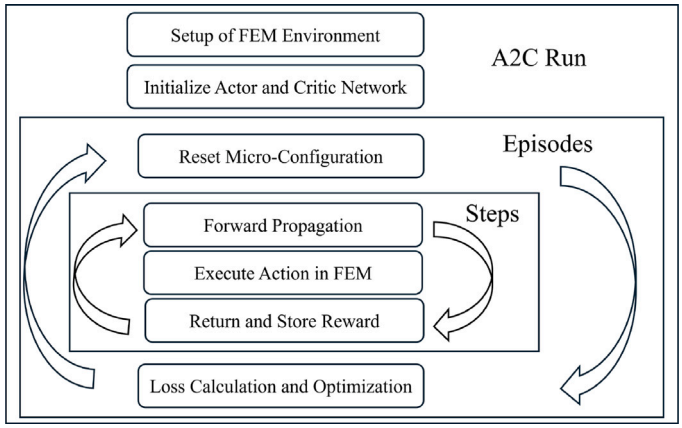

Fig. 6. General A2C process flow.

the next steps until one of the following termination conditions is met (Fig. 5). Each episode is terminated either after a defined number of steps or after the desired precision of target bulk modulus is achieved. After termination the NNs are optimized using (19) and a new episode is initiated. This process is repeated until the agents are able to reach the desired micro-configuration that leads to the target macro properties consistently within every episode. While typical DRL use-cases cover tasks, where every step in the process is dependent on the previous step, in this work a sequential learning should not be necessary. As the action space is continuous and the state is independent of prior states, all actions in all possible micro-adjustments can be of any desired scale and are able to reach the goal at any step of the episode. In addition, the potential of finding micro-configurations that conform with the desired macro properties do not need to follow a specific trajectory. This leads to the hypothesis that multiple steps in every episode are not necessary and one step is sufficient in reaching the target as shown in Section 4.3.

4. Results and discussion

This chapter presents the key findings of the study, beginning with the determination of the Representative Volume Element (RVE) size as a foundational step. The implementation and basic setup of the Advantage Actor–Critic (A2C) algorithm are then discussed, followed by a sensitivity analysis of the Deep Reinforcement Learning (DRL) model to key hyperparameters, allowing the algorithm to be tailored to specific objectives. A comparative analysis between Bayesian Optimization (BO) and A2C relates their process in achieving target bulk modulus values and computational efforts. Furthermore, the study extends into a three-dimensional application and explores a more complex example that incorporates a 3-dimensional material list as search space and extends the adjustable material parameters with density and volume fraction. This extension enables a transition to a multi-objective optimization framework, which balances material weight alongside bulk modulus, broadening the applicability of the proposed model.

4.1. Numerical RVE size

At first a convergence study regarding the necessary number of finite elements is conducted to verify that the inclusions and their contributions to the overall stress progression is resolved with sufficient accuracy. In Fig. 7, the convergence of the bulk modulus is depicted as a function of the number of elements per dimension and the number of elements per inclusion along each dimension. The results demonstrate that the bulk modulus achieves convergence within an acceptable tolerance when using between 8 and 16 elements per dimension which corresponds to 2 to 6 elements per dimension for the inclusions, in the case considered. Thus, a resolution for each particle of 2–4 is maintained in the following. As described in Section 3.3, ensemble averages of multiple random distributions of particles in the matrix are calculated for individual sample sizes (Fig. 6). Based on (7) the tolerance is calculated and the ensemble averages together with the sample sizes are monitored for convergence under a specific tolerance threshold. As later in the reinforcement runs, the radius of the distributed particles are allowed to change between 1 and 2, it needs to be assured that the material is homogenized for all possible radii. Hence, homogenization is only conducted for the lowest, highest and intermediate radius. It is observed that a sample size of 18 elements per direction and an ensemble average of 10 samples is enough to statistically represent the material.

4.2. Basic setup of A2C algorithm

To initialize the NNs, their hyperparameters and net parameters need to be specified based on educated guess and prior testing. The layout and activation functions of each network are presented in Table 1. In this work, the input is normalized using the

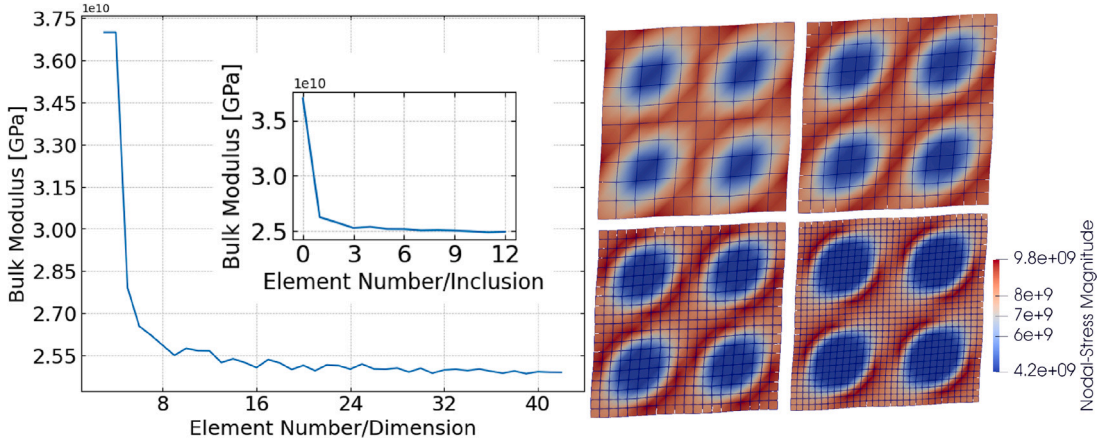

Fig. 7. Convergence of bulk modulus for different element numbers per dimension (left-outer) and inclusions (left-inner) and nodal stress for selected resolutions (right).

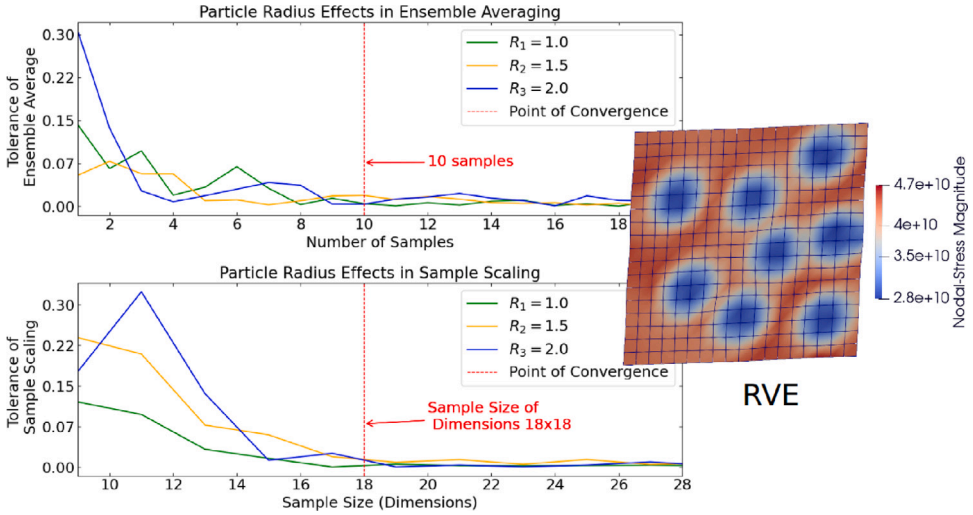

Fig. 8. Convergence of ensemble averages (top) and sample scaling (bottom) based on (6), (7) and Fig. 3, RVE stress magnitude (right).

MinMax normalization scheme [51]. In addition, gradient clipping and weight decay is performed on the actor and critic gradients and weights like mentioned in [52,53]. The theoretical design is constrained regarding Young's modulus and Poisson's ratio. In the state space $\boldsymbol{S}_t$, the Young's moduli for matrix and particles are constrained between 0 and 200 GPa and Poisson's ratios between 0.0 and 0.5. The radius of the particles are bounded between 1 and 2. The initial values at the start of each episode are chosen to be in the middle of each state space. The output is normalized to an appropriate range that corresponds to the specific property being optimized. The scale of each step in the different micro-property adjustments is chosen so that the limits mentioned above are not exceeded. If the achieved bulk modulus falls within the allowed range of ± 0.5 GPa of the target bulk modulus, the episode is terminated and started again. Otherwise, the episode terminates after a fixed number of steps. The first A2C run is conducted using hyperparameters based on experience and literature (Fig. 9). While the algorithm is able to decrease the distance to the target bulk modulus effectively and reaches its goal within the allowed margin of error multiple times (see for instance Fig. 9), there is still potential for improvements.

4.3. Sensitivity analysis of the deep reinforcement learning model

Time efficient learning and general model performance depends on the algorithm's hyperparameters. While intuition and empirical testing already leads to good results as shown in the previous section in Fig. 9, a study on the refinement of hyperparameters is conducted by applying a GA. Initially, multiple sets of potentially successful parameters sets are defined. These sets include the learning rate, discount factor, entropy coefficient, episode duration, and critic loss weight. The initial population of the GA will combine these sets to create potentially superior configurations. A two-stage GA is used to optimize parameter settings, aiming to

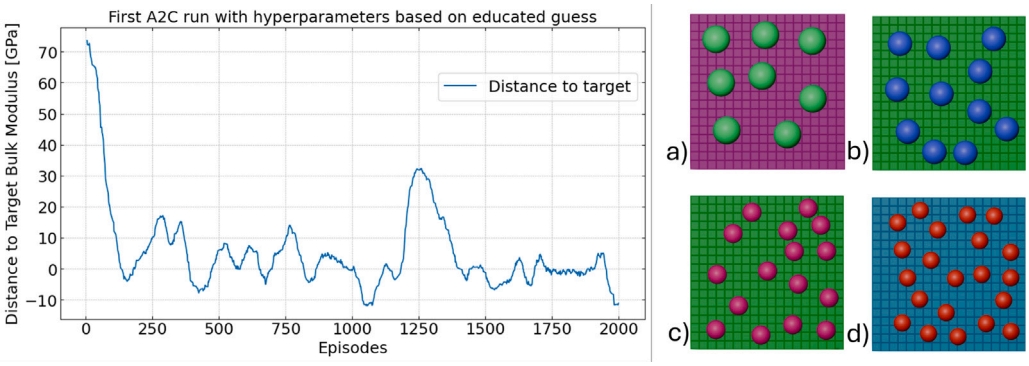

Fig. 9. First A2C run using following hyperparameters: [learning-rate = 0.0001, entropy-weight = 0.01, discount-factor = 0.95, episode-duration = 10, critic-loss-weight = 0.5](left), excerpt of achieved micro-configurations with bulk modulus of 12 GPa (right (a)–(d)).

minimize the distance to a target bulk modulus. A detailed introduction and implementation scheme of different tuning algorithms like Bayesian optimization, GAs and grid search is given in [54].

4.3.1. Evaluating general learning capabilities

The objective of the GA is designed to find hyperparameter combinations that minimize the cumulative distance to the target bulk modulus at the end of each RL episode. By minimizing the summed distance to the target bulk modulus, the algorithm is incentivized to not only approach but also maintain proximity to the smallest possible distance. Initially, the parameter sets are chosen broadly, allowing the algorithm to explore a wide range of possible micro-structures. This broad selection is critical as it enables an extensive initial search within the parameter space. After the first GA run, the outcome is analyzed to identify promising sets. The second stage involves refining these parameters through a new GA run, employing a more focused search to fine-tune the parameters more effectively. The upper diagram in Fig. 8 displays the first stage of the GA tuning process, taking into account a very broad range of hyperparameters. In the lower diagram, the hyperparameter range is refined based on the gathered knowledge. For illustration purposes only a few of all runs are depicted. For each run the absolute distance to the target bulk modulus at the last step of each episode is plotted against the episode. Among different hyperparameters, learning rate, episode duration, and entropy weight posses the strongest influence on the overall performance of the A2C. In combination with the reward function explained in Section 3.4 the episode duration seems to primarily influence the variance of the distance at the end of the episode. The runs in the upper diagram in Fig. 8 have multiple steps per episode ranging from 2 to 10 steps while the refined runs only consider a single step per episode. The variance is much higher in the single step approach. This however does not have noticeable influence on the models overall performance. The single-step runs also decrease variance significantly over time. Using more steps, the computational effort also increases. For this reason in the following each episode includes only one step. It can be noted that higher learning rates, like Conf. 1 and Conf. 3 in Fig. 10, lead very quickly to allowed micro-configurations. However, such hyperparameters lead to a very unstable learning as these runs move away from the target later on or fluctuate significantly. In comparison, medium or small learning rates as depicted by Conf. 2 and Conf. 4 approach the small absolute distances slower, but tend to be more stable in the monitored time-frame. Learning rates in the range of 1 to 5×10^{-4} and entropy values of 0.1 to 0.3 show the most stable learning behavior and approach low distance values rather quickly. Hence, only values in this range were allowed in the second GA tuning process. This refined approach is shown in the lower diagram in Fig. 9. It can be observed that the general ability to converge to low distance values from top to bottom in Fig. 10 could be improved significantly. Conf. 7, Conf. 5 and Conf. 8 all show high conformance with small distances to the target bulk modulus after a learning process of around 1000 episodes. Converging to distance values around zero is a strong indication that the algorithm understands how the micro-configuration needs to be adjusted in order to achieve the target bulk modulus. The longer the algorithm is able to stay at low distance values the higher is the probability that the desired target bulk modulus is reached multiple times. This makes it possible to find multiple different configurations of the micro-structure that have the desired macroscopic properties. While the run with the highest learning rate (Conf. 6 in lower diagram of Fig. 8) is able to reach small distance values at first, it is not able to stay in the desired range over the whole time frame. This is consistent with the findings in the first GA approach. To get a deeper understanding of the influence of entropy weight and learning rate on the models performance and efficiency the best configurations were tested using variations of the mentioned parameters.

4.3.2. Algorithm sensitivity to entropy weight and learning rate

As the entropy weight and the learning rate are the most sensitive, these properties were investigated in more detail. In the upper diagram in Fig. 11 an excerpt of different entropy weights is shown. High values (blue and red curve) lead to a rather fluctuating performance at larger episodes. Medium values (yellow curve) and low values (green curve) have superior properties regarding volatility of performance. The lower diagram verifies that by adjusting the learning rate either the goal of early achieved targets or of consistent learning can be achieved. However, it is not possible to achieve both by only adjusting this parameter. Higher learning rates (red and blue curves) reach a small distance earlier compared to the green or yellow curves. However, the latter have a lot more

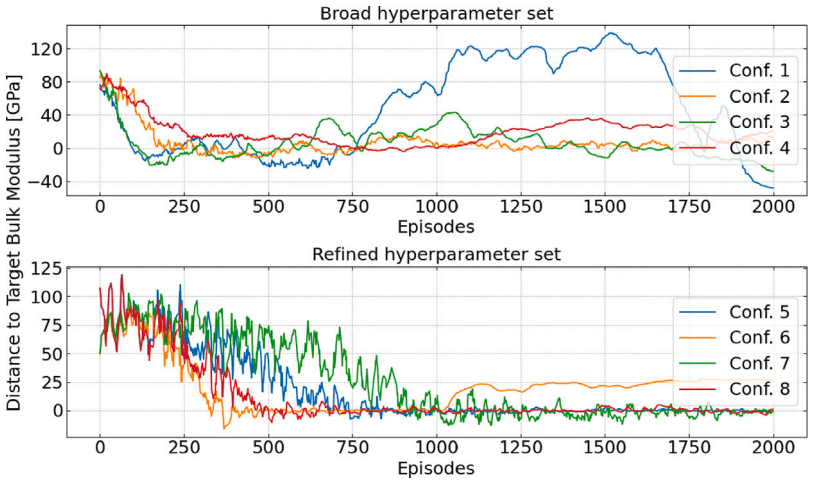

Fig. 10. A2C runs with GA tuning, broad set (top), refined set (bottom) for given hyperparameters listed in Table 2.

Table 2
Specification of used hyperparameters of Fig. 10.

Runs	Learning rate	Entropy weight	Discount factor	Episode duration	Critic weight
Conf. 1	0.000079	0.202	0.976	6	0.683
Conf. 2	0.000052	0.148	0.975	2	0.845
Conf. 3	0.000070	0.298	0.978	4	0.546
Conf. 4	0.000028	0.299	0.971	8	0.810
Conf. 5	0.000022	0.114	0.982	1	0.703
Conf. 6	0.000045	0.254	0.986	1	0.719
Conf. 7	0.000018	0.127	0.978	1	0.737
Conf. 8	0.000037	0.139	0.990	1	0.779

stable performance over the whole time-frame. Rewinding the findings of the analysis it can be concluded that the A2C algorithm is sensitive to hyperparameter tuning. To achieve a consistent learning, low variance and early achieved target especially entropy and learning rate need to be balanced within certain ranges. Consistently good performance was achieved with one-step episodes using learning rates between 2.5 to 3.5×10^{-4} and entropy weights of around 0.1 to 0.3. Beyond its role in managing exploration and exploitation, the entropy term also serves to enhance the diversity of achievable material configurations. While the reward structure illustrated in Fig. 5 directs the algorithm toward minimizing the distance to the target, the entropy term counterbalances this, enabling exploration across a broader range of the search space. This dual effect intentionally prevents convergence of the material properties in (8) within the episode range, aligning with the goal of identifying multiple viable configurations rather than a single unique solution.

4.3.3. Tailoring algorithm performance to specific objectives

Key performance indicators (KPIs) represent metrics to quantify the quality of the algorithm. In literature regarding machine learning and statistics it is often differentiated between performance, efficiency, sample efficiency and robustness metrics. Here, performance is understood as the accuracy to achieve the desired bulk modulus as well as the ability of reaching desired states as often as possible. Efficiency evaluates the time required to reach the target. Robustness measures the ability to achieve the goals reliably and stable over multiple episodes without high fluctuations.

While so far consistent learning and low volatile convergence to the target bulk modulus was the focus, an insight into different hyperparameter configurations that each serve one specific purpose better than the others is now presented. In Fig. 12, Conf. 2 has high efficiency as only 125 steps are needed to find a proper micro-configuration that achieves the desired bulk modulus. Conf. 5 has the smallest cumulative total distance to zero and therefore shows high performance and robustness.

Hence, this configuration reaches close distances very early on and stays within a close range for a relatively long time. This behavior is desired in order to maintain a high likelihood of reaching the target bulk modulus as the tested micro-configurations all lie within close proximity to the margin of accepted error. While this run indeed reaches a high number of different accepted micro-configurations, i.e. 157, Conf. 1 and Conf. 4 result in the highest number of target bulk modulus, i.e. 345 and 338. The configuration also indicate the highest performance due to the ability to learn the intrinsic mechanical properties quickly and also very accurately. The absolute distances stay within the region of accepted margin of error for a long consecutive time and multiple accepted configurations can be accumulated.

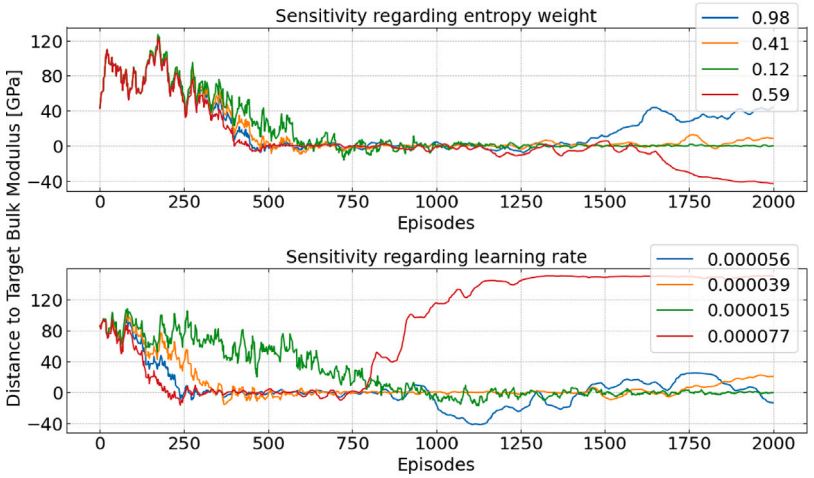

Fig. 11. Sensitivity to entropy weight (top) and learning rate (bottom), hyperparameters: [learning rate, entropy coefficient, discount factor, episode duration, critic weight]; top: [0.000026, see legend, 0.99, 1, 0.5]; bottom: [see legend, 0.23, 0.99, 1, 0.5]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

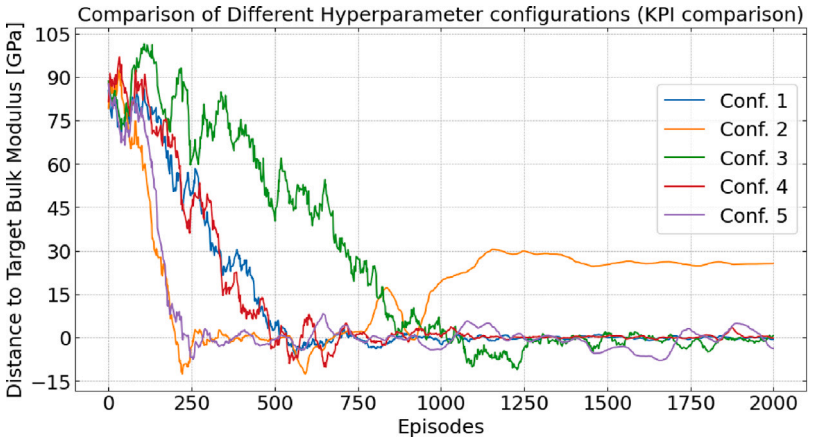

Fig. 12. KPI comparison, hyperparameters: [learning rate, entropy coefficient, discount factor, episode duration, critic weight]; Configurations 1–5 (discount factor, episode duration and critic weight stay constant): [see Table 3, see Tables 3, 0.99, 1, 0.5].

Table 3
List of hyperparameter configurations for each curve in Fig. 12.

Runs	Learning rate	Entropy weight	Cumulative distance	Steps until achieved target	Number of achieved targets
Conf. 1	0.000033	0.14	2904	435	338
Conf. 2	0.000077	0.15	4247	**125**	80
Conf. 3	0.000015	0.33	6059	460	57
Conf. 4	0.000030	0.17	3034	358	**345**
Conf. 5	0.000061	0.28	**1790**	159	157

4.4. Comparing A2C based material design with Bayesian optimization

An alternative and already well established optimization algorithm in material design is Bayesian Optimization (BO) as a probabilistic model-based approach for optimizing complex functions that are expensive to evaluate [11]. While reinforcement learning (RL) methods like A2C rely on iterative strategy optimization through trial-and-error interactions, adjusting the neural network's structure and refining strategies, Bayesian Optimization (BO) builds a probabilistic model to guide each step toward the target more efficiently, representing a strategy grounded in optimization. Although comparing these algorithms directly is challenging due to their distinct methodologies, their progress in reducing the distance to the target bulk modulus can be analyzed. Fig. 13 shows that BO effectively minimizes the distance to the target bulk modulus in a simple 2-dimensional case similarly to the A2C approach. Additionally, both algorithms successfully identify between 100 and 400 accepted material configurations within

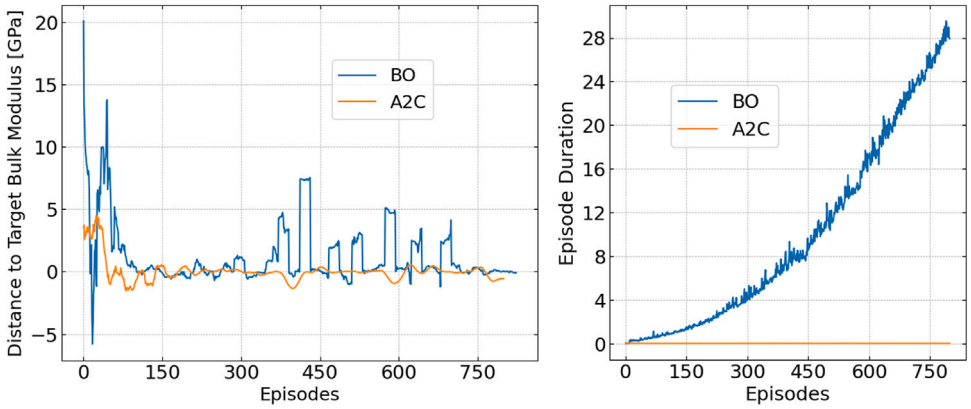

Fig. 13. Comparison of A2C and BO in terms of cumulative number of achieved target bulk modulus (left) and duration of every episode (right).

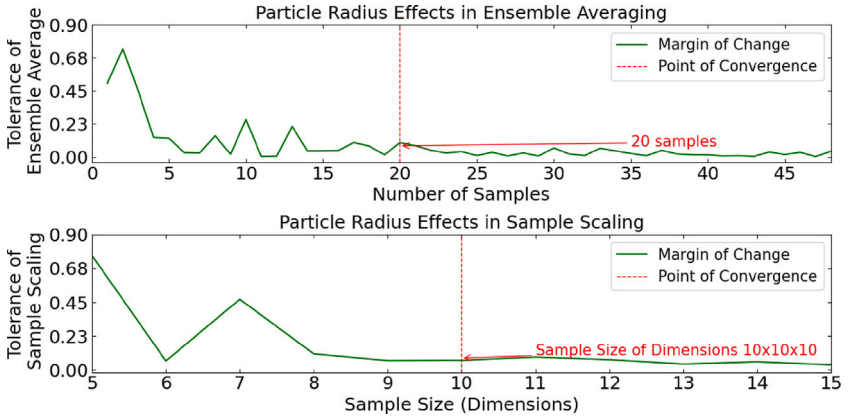

Fig. 14. Ensemble averages (top) and sample scaling (bottom) to define an RVE in a 3-dimensional case.

800 episodes. While these numbers vary significantly across different runs, the general learning trend depicted as the reduction in the distance to the target bulk modulus remains consistent. When examining the duration of episodes (Fig. 13), a notable difference emerges during the training process. Initially, the episode durations for both algorithms are comparable. However, as training progresses, BO's episode duration increases with each iteration step, whereas A2C's durations remain relatively stable. While the absolute values of episode durations hold limited significance due to differences in the technical implementation of the two algorithms, the relative trends are noteworthy. BO's increasing iteration duration aligns with theoretical expectations, as its underlying probabilistic model becomes more complex with additional episodes, see [55,56]. In contrast, A2C maintains a consistent model complexity over time and consequently a constant iteration time.

4.5. 3-dimensional inverse material design

Having successfully implemented a fine-tuned 2D-A2C algorithm the developed approach is also applied for a 3-dimensional test case. First the RVE is determined (Fig. 14). The upper diagram shows the margin of change when applying ensemble averaging over multiple same size RVE samples with random particle distributions. At a sample size of $10 \times 10 \times 10$ and 20 samples convergence is observed (Fig. 14). The developed A2C algorithm is applied to achieve a specific target bulk modulus of 150 GPa by adjusting the microscopic material properties, i.e. Poisson's ratios, Young's moduli and particle radii. The hyperparameters are selected based on the conclusions for the 2D case. In Fig. 15 the three KPIs of performance, efficiency and robustness are compared. It is observed that the training behavior in a 3D environment closely matches the 2D findings regarding influence of learning rate and entropy weight. However, the absolute sample efficiency, performance and robustness differs. To learn underlying interrelationships consistently seems to be slightly more challenging for the agents.

However, the hyperparameters can still be adjusted effectively to the different KPIs. Conf. 1 achieved the highest sample efficiency reaching the target after 147 steps. Conf. 2 and 3 show the highest robustness, fluctuating in close proximity to the target very consistently over the time frame. Conf. 3 achieved the highest performance, i.e. 125 allowed micro-configurations conforming with the target. Hence, it can be concluded that findings in the 2-dimensional case can be transferred to the 3-dimensional case effectively. The A2C algorithm can be adjusted precisely to the desired strategy and reaches the target within the error threshold as desired.

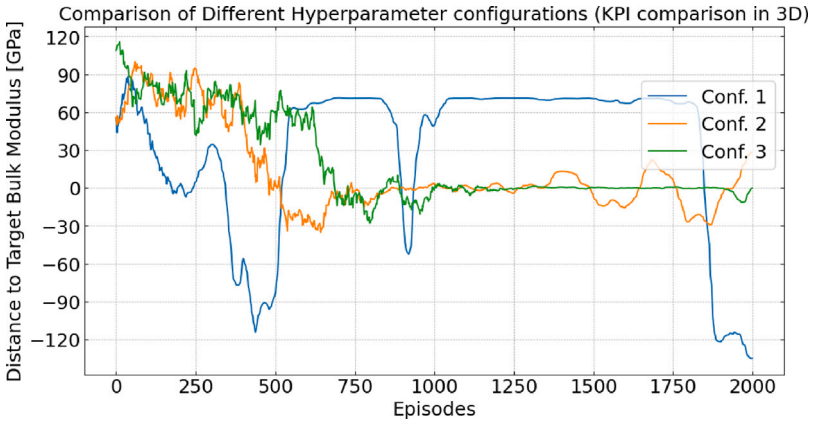

Fig. 15. KPI comparison, lowest cumulative distance to target (Conf. 2), least steps until achieved target (Conf. 1), number of achieved target (Conf. 3), hyperparameters in Table 4.

Table 4
Hyperparameter configurations of Fig. 15.

Runs	Learning rate	Entropy weight	Discount factor	Episode duration	Critic weight
Conf. 1	0.00005	0.15	0.99	1	0.5
Conf. 2	0.00001	0.25	0.99	1	0.5
Conf. 3	0.000007	0.33	0.99	1	0.5

4.6. Material list and density optimization

As the general capabilities of the A2C algorithm applied to a theoretical problem of artificial continuous materials are now studied, these findings are applied in a more realistic scenario. In this context a material list is created consisting of around 8000 materials extracted from the online database matweb.com. The properties of Young's modulus, Poisson's ratio and density are structured in a way that the RL algorithm is able to navigate within this discrete space and select material combinations based on the mechanical responses. It is important to implement a logical structuring to enable successive decisions while only a fraction of the theoretical, wide-ranging design space needs to be explored. The material properties are ranked and structured from lowest to highest value. In each step the algorithm places a theoretical material point within the boundaries of all existing materials both for particle and for matrix material based on the weights of the current NNs. The distances to all possible discrete materials within the design space are calculated efficiently using the "Compressed KD-Tree" algorithm [57]. The spatially nearest discrete material is used for the next evaluation within the environment. Based on feedback from the FEM algorithm the RL algorithm is again able to navigate the design space in a way that leads to multiple material combinations conforming with the desired target properties. In this version the reward structure is extended. Rewards are not only given for getting closer to the target bulk modulus but also when the total weight of the material based on the density of matrix and particles is decreased. In addition to particle radius this version includes the volume fraction of particles in the matrix as adjustable and learnable topological parameter. Higher volume fractions necessitate a more sophisticated approach for random micro-structure generation. For this purpose an algorithm is created that calculates repulsive forces between overlapping inclusions that are applied to separate them until no overlap remains as depicted in Fig. 17. This way the theoretical upper limit for reachable volume fractions is between 70 and 80 percent. However, for faster micro-structure generation the volume fraction is capped in this example at 60 percent. On micro-structure generation the particles can also move through the boundaries onto the other side of the RVE. This periodicity represents a theoretically infinite structure. In Fig. 17 the 2D version of the micro-structure generation is illustrated. The 3D version is equivalent to it in terms of periodicity and handling of overlaps. The state representation represents now eight entries compared to (8)

$$\boldsymbol{S}_t = [E_m^r, \nu_m^r, \rho_m^r, E_p^r, \nu_p^r, \rho_p^r, r_p, f_p] \tag{22}$$

incorporating the three coordinates in the 3-dimensional material list of ranked properties of Young's modulus E, Poisson's ratio ν, density ρ, and the particle radius and volume fraction f_p of particles as additional trainable parameters. Consequently, also the output of the NN's must be adjusted accordingly to (9). As shown in Fig. 18 over the course of all episodes, the number of configurations meeting the desired threshold steadily increases. By providing an added incentive for lower weights, the material configurations become progressively lighter while still achieving a larger number of configurations with the required mechanical properties. The balance between these two objectives can be controlled by adjusting their respective reward weights w_T and w_W in

$$R_t^{total} = w_T * R_t^T + w_W * R_t^W \tag{23}$$

where the first term corresponds to the weighted reward received for decreasing the distance to the target bulk modulus as given in (23). The second reward term R_t^W is calculated as the difference between the weight in the current episode t and the weight

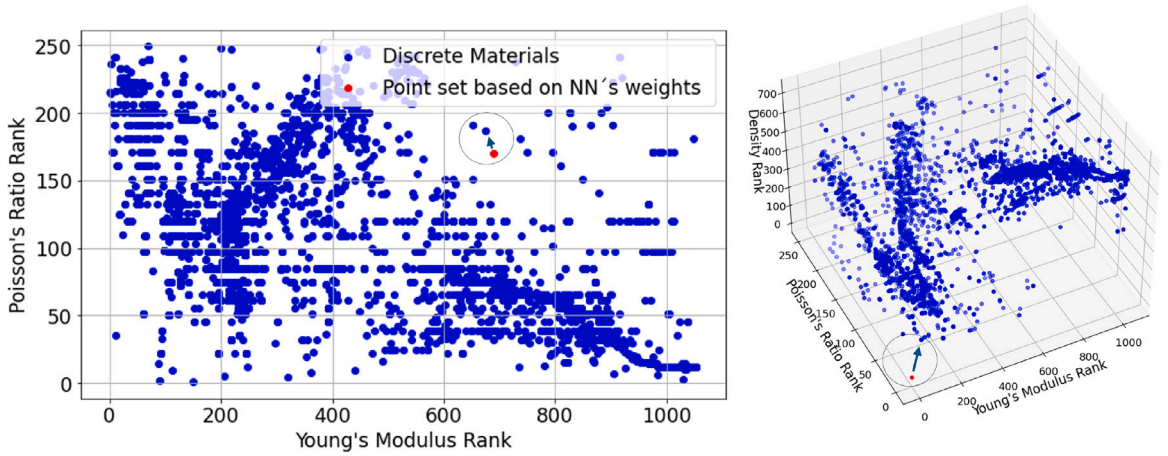

Fig. 16. Material list based discrete materials with Young's modulus and Poisson's ratio (left) and additional density (right) ranked from highest to lowest.

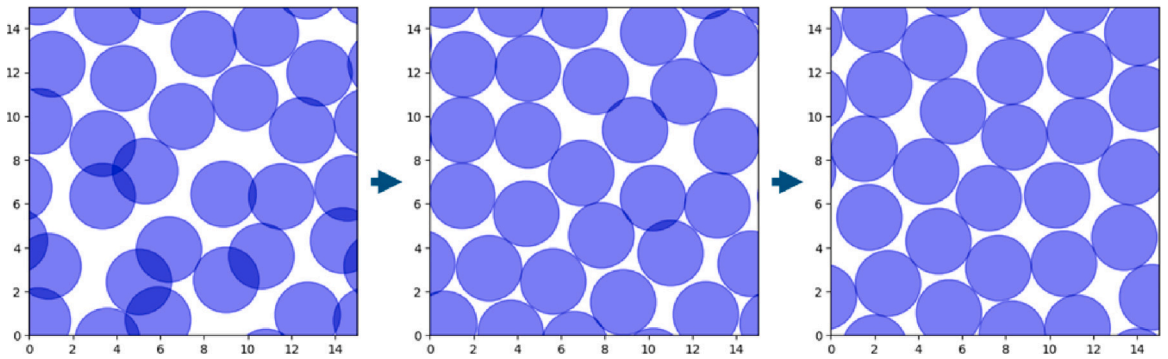

Fig. 17. Process of decreasing overlap in randomly generated micro-structure from left to right.

Table 5
Microscopic geometric scales and numbers of particles, volume fractions and material parameters leading to similar macroscopic target bulk moduli and corresponding lower and upper Hashin–Shtrikman bounds of discrete material combinations listed in Table 6.

	E_m [GPa]	E_p [GPa]	ν_m	ν_p	VF	κ [GPa]	HS_{lower} [GPa]	HS_{upper} [GPa]
(a)	207.0	3.4	0.32	0.30	0.28	87.5	367.7	7.17
(b)	117.0	380.0	0.045	0.18	0.58	90.9	90.9	85.8
(c)	82.85	/	0.344	/	0.0	88.5	88.5	88.5
(d)	89.75	16.9	0.43	0.345	0.25	88.1	233.0	39.6
(e)	16.9	114.0	0.345	0.45	0.17	88.2	19.8	211.0

in the previous episode. In Tables 5 and 6 a selection of achieved material combinations from the material list is given that all achieve the desired target bulk modulus of 90 GPa in a margin of error of ±3 GPa. The goal of this study is to identify a diverse range of material configurations. However, it is important to note that the algorithm does not explicitly exclude the possibility of an arbitrary homogeneous solution where $E_m = E_p$ and $\nu_m = \nu_p$, as listed in Tables 5 and 6 with solution (c). To verify that the effective moduli of the computed material configurations are within physically realistic and theoretically predicted ranges based on the properties and volume fractions of its constituents they were evaluated against the Hashin–Shtrikman bounds [58]. Generally, the determined material combinations are not necessarily compatible or realizable, but it is shown that the inverse design scheme based on RL is applicable to more realistic scenarios with real discrete materials using a structured 3-dimensional list to support the RL algorithm in designing discrete material combinations effectively. It is also shown that this method can be effectively extended to multi-objective tasks by giving additional rewards for reducing the overall weight from one episode to the next. Maintaining the reward structure as formulated in (13) the search space is kept in close proximity to solutions conforming with the target bulk modulus. At the same time the reward given for reduced weight drags the search space to configurations that have lower overall weight. As a result the achieved accepted configurations tend to decrease material weight during the training process (see Figs. 16 and 19).

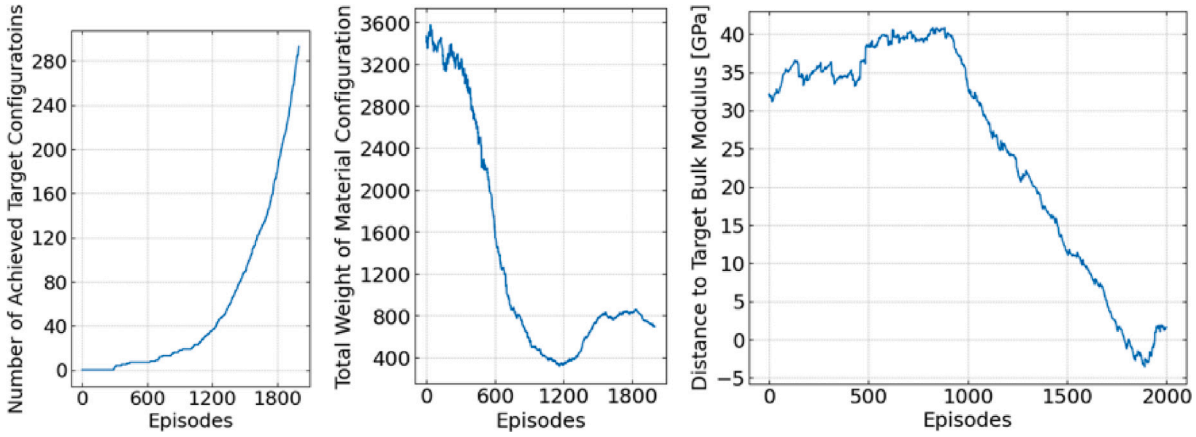

Fig. 18. A2C run of discrete material design with number of achieved target configurations per episode (left), total weight of corresponding material configuration (middle), and distance to target bulk modulus in GPa (right).

Table 6
Discrete matrix and particle material combinations with density, particle radius and number and resulting total weight of configurations.

	ρ_m [g/cc]	ρ_p [g/cc]	R_p	N_p	m_{total}
(a)	6.04	0.81	1.83	11	4575.6
(b)	1.2	2.56	1.80	24	1969.6
(c)	1.31	/	/	0	1279.9
(d)	1.21	0.07	1.15	40	925.0
(e)	0.07	1.21	1.40	15	257.7

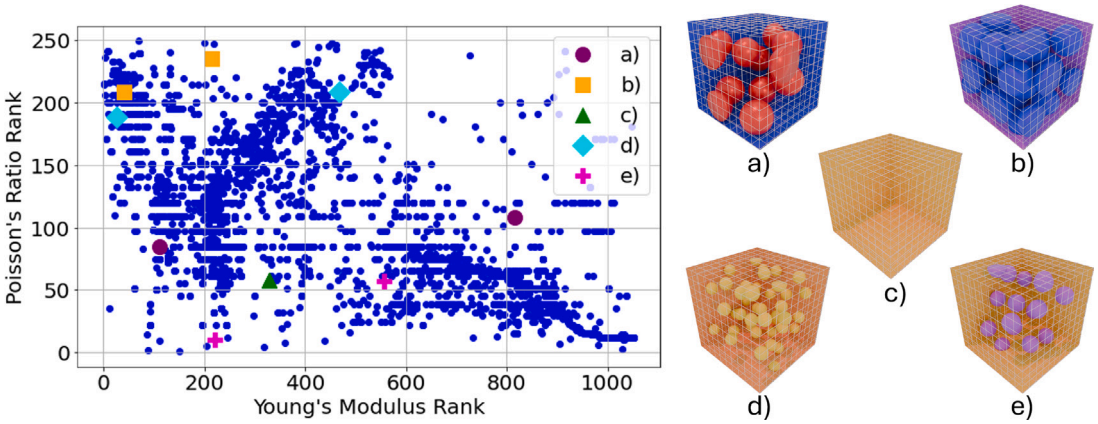

Fig. 19. Excerpt of different material micro-configurations (a)–(e) leading to a similar bulk modulus of 90 GPa in materiallist (left) and visualized (right).

5. Summary

This study demonstrates the potential of using Deep Reinforcement Learning in conjunction with homogenization for solving inverse material design problems. By utilizing an A2C model, a methodology was successfully established that links macroscopic material properties to micro-structural parameters. Different KPIs such as achieving the target as early as possible (efficiency) or detecting as many micro-configurations as possible that align with the macroscopic target (performance) were investigated. It was shown that the applied methodology is highly flexible and can be adjusted to different strategies by tuning the hyperparameters carefully. The framework effectively optimizes micro-structural features such as particle radius,volume fraction, Young's modulus, and Poisson's ratio to achieve a target bulk modulus at the macro-scale. The findings from the 2D investigation were successfully extended to a more complex and realistic 3D scenario. In this case, real-world materials were combined using a 3D material list that also incorporated material density. This extension transformed the original single-objective focused on minimizing the distance to the target bulk modulus into a multi-objective function that simultaneously reduces the overall weight of the structure. The proposed approach showcases the robustness and adaptability of the DRL-based model in reliably identifying multiple distinct solutions

to the inverse problem through interactions with the FEM environment. This highlights its significant potential for designing microstructures. Future research could expand the current focus on macroscale isotropic materials to include anisotropic materials by enabling the RL framework to incorporate orientation and particle topology as additional trainable parameters. Consequently, the multi-objective function could be extended to account for various loading scenarios, broadening its applicability.

CRediT authorship contribution statement

V. Würz: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **C. Weißenfels:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing 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.

Data availability

Data will be made available on request.

References

[1] T.I. Zohdi, Constrained inverse formulations in random material design, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 192 (28) (2003) 3179–3194, http://dx.doi.org/10.1016/S0045-7825(03)00345-1, URL https://www.sciencedirect.com/science/article/pii/S0045782503003451.
[2] Z. Yang, X. Li, L.C. Brinson, A.N. Choudhary, W. Chen, A. Agrawal, Microstructural materials design via deep adversarial learning methodology, J. Mech. Des. (ISSN: 1050-0472) 140 (11) (2018) 111416, http://dx.doi.org/10.1115/1.4041371.
[3] W. Chen, F. Ahmed, PaDGAN: Learning to generate high-quality novel designs, J. Mech. Des. (ISSN: 1050-0472) 143 (3) (2020) 031703, http://dx.doi.org/10.1115/1.4048626.
[4] R.J. Gladstone, M.A. Nabian, V. Keshavarzzadeh, H. Meidani, Robust topology optimization using variational autoencoders, 2021, ArXiv, abs/2107.10661.
[5] F. Sui, R. Guo, Z. Zhang, G.X. Gu, L. Lin, Deep reinforcement learning for digital materials design, ACS Mater. Lett. (ISSN: 2639-4979) 3 (10) (2021) 1433–1439, http://dx.doi.org/10.1021/acsmaterialslett.1c00390.
[6] M. Wang, L. Wu, J. Li, L. He, Traffic signal control with reinforcement learning based on region-aware cooperative strategy, IEEE Trans. Intell. Transp. Syst. 23 (2021) 6774–6785, http://dx.doi.org/10.1109/TITS.2021.3062072.
[7] K. Alibabaei, P.D. Gaspar, E. Assunção, S. Alirezazadeh, T.M. Lima, V.N.G.J. Soares, J.M.L.P. Caldeira, Comparison of on-policy deep reinforcement learning A2C with off-policy DQN in irrigation optimization: A case study at a site in Portugal, Computers (ISSN: 2073-431X) 11 (7) (2022) http://dx.doi.org/10.3390/computers11070104.
[8] M. Sewak, Deep Reinforcement Learning: Frontiers of Artificial Intelligence, Springer, Singapore, ISBN: 9789811382857, 2019.
[9] Z. Ding, Y. Huang, H. Yuan, H. Dong, Introduction to reinforcement learning, in: H. Dong, Z. Ding, S. Zhang (Eds.), Deep Reinforcement Learning, in: Springer eBook Collection, Springer Singapore and Imprint Springer, Singapore, ISBN: 978-981-15-4094-3, 2020, pp. 47–123, http://dx.doi.org/10.1007/978-981-15-4095-0_.
[10] V. François-Lavet, P. Henderson, R. Islam, M.G. Bellemare, J. Pineau, An introduction to deep reinforcement learning, Foundations and Trends® in Machine Learning (ISSN: 1935-8237) 11 (3–4) (2018) 219–354, http://dx.doi.org/10.1561/2200000071.
[11] P.I. Frazier, J. Wang, Bayesian optimization for materials design, in: T. Lookman, F.J. Alexander, K. Rajan (Eds.), Information Science for Materials Discovery and Design, Springer International Publishing, Cham, ISBN: 978-3-319-23871-5, 2016, pp. 45–75, http://dx.doi.org/10.1007/978-3-319-23871-5_3.
[12] P. Honarmandi, V. Attari, R. Arroyave, Accelerated materials design using batch Bayesian optimization: A case study for solving the inverse problem from materials microstructure to process specification, Comput. Mater. Sci. (ISSN: 0927-0256) 210 (2022) 111417, http://dx.doi.org/10.1016/j.commatsci.2022.111417, URL https://www.sciencedirect.com/science/article/pii/S0927025622001859.
[13] C. Huet, Application of variational concepts to size effects in elastic heterogeneous bodies, J. Mech. Phys. Solids (ISSN: 0022-5096) 38 (6) (1990) 813–841, http://dx.doi.org/10.1016/0022-5096(90)90041-2, URL https://www.sciencedirect.com/science/article/pii/0022509690900412.
[14] S.P. Timoshenko, J.N. Goodier, H.N. Abramson, Theory of elasticity (3rd ed.), J. Appl. Mech. 37 (3) (1970) 888–888.
[15] Y.C. Fung, Foundations of Solid Mechanics, Prentice-Hall, ISBN: 0133299120, 1965, URL http://www.worldcat.org/isbn/0133299120.
[16] C. Liu, C. Reina, Discrete averaging relations for micro to macro transition, arXiv: Mater. Sci. (2015) http://dx.doi.org/10.1115/1.4033552.
[17] V. Kouznetsova, W.A.M. Brekelmans, F.P.T. Baaijens, An approach to micro-macro modeling of heterogeneous materials, Comput. Mech. (ISSN: 1432-0924) 27 (1) (2001) 37–48, http://dx.doi.org/10.1007/s004660000212.
[18] T.I. Zohdi, P. Wriggers, An introduction to computational micromechanics, Lecture Notes in Applied and Computational Mechanics, Springer Berlin Heidelberg, ISBN: 9783540228202, 2004.
[19] R. Hill, On constitutive macro-variables for heterogeneous solids at finite strain, Proc. R. Soc. A (ISSN: 0080-4630) 326 (1565) (1972) 131–147, http://dx.doi.org/10.1098/rspa.1972.0001.
[20] O. van der Sluis, P. Schreurs, W. Brekelmans, H. Meijer, Overall behaviour of heterogeneous elastoviscoplastic materials: effect of microstructural modelling, Mech. Mater. (ISSN: 0167-6636) 32 (8) (2000) 449–462, http://dx.doi.org/10.1016/S0167-6636(00)00019-3.
[21] K. Terada, M. Hori, T. Kyoya, N. Kikuchi, Simulation of the multi-scale convergence in computational homogenization approaches, Int. J. Solids Struct. (ISSN: 0020-7683) 37 (16) (2000) 2285–2311, http://dx.doi.org/10.1016/S0020-7683(98)00341-2.
[22] C. Miehe, Computational micro-to-macro transitions for discretized micro-structures of heterogeneous materials at finite strains based on the minimization of averaged incremental energy, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 192 (5) (2003) 559–591, http://dx.doi.org/10.1016/S0045-7825(02)00564-9.
[23] A. Denisiewicz, M. Kuczma, K. Kula, T. Socha, Influence of boundary conditions on numerical homogenization of high performance concrete, Mater. (Basel, Switzerland) (ISSN: 1996-1944) 14 (4) (2021) http://dx.doi.org/10.3390/ma14041009.
[24] I. Temizer, T.I. Zohdi, A numerical method for homogenization in non-linear elasticity, Comput. Mech. (ISSN: 1432-0924) 40 (2) (2007) 281–298, http://dx.doi.org/10.1007/s00466-006-0097-y.

[25] T.I. Zohdi, M. Feucht, D. Gross, P. Wriggers, A description of macroscopic damage through microstructural relaxation, Internat. J. Numer. Methods Engrg. (ISSN: 0029-5981) 43 (3) (1998) 493–506, http://dx.doi.org/10.1002/(SICI)1097-0207(19981015)43:3<493::AID-NME461>3.0.CO;2-N.
[26] T.I. Zohdi, P. Wriggers, Computational micro-macro material testing, Arch. Comput. Methods Eng. (ISSN: 1886-1784) 8 (2) (2001) 131–228, http://dx.doi.org/10.1007/BF02897871.
[27] D. Garoz, F. Gilabert, R. Sevenois, S. Spronk, W. Van Paepegem, Consistent application of periodic boundary conditions in implicit and explicit finite element simulations of damage in composites, Composites B (ISSN: 1359-8368) 168 (2019) 254–266, http://dx.doi.org/10.1016/j.compositesb.2018.12.023.
[28] C. Miehe, A. Koch, Computational micro-to-macro transitions of discretized microstructures undergoing small strains, Arch. Appl. Mech. (ISSN: 1432-0681) 72 (4) (2002) 300–317, http://dx.doi.org/10.1007/s00419-002-0212-2.
[29] D. Zäh, C. Miehe, Computational homogenization in dissipative electro-mechanics of functional materials, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 267 (2013) 487–510, http://dx.doi.org/10.1016/j.cma.2013.09.012, URL https://www.sciencedirect.com/science/article/pii/S0045782513002399.
[30] P. Henyš, J. Březina, L. Capek, Comparison of current methods for implementing periodic boundary conditions in multi-scale homogenisation, Eur. J. Mech. A Solids 78 (2019) http://dx.doi.org/10.1016/j.euromechsol.2019.103825.
[31] V. Kouznetsova, Computational homogenization for the multi-scale analysis of multi-phase materials, Softw. Process: Improv. Pract. - SOPR (2002).
[32] D. Savvas, G. Stefanou, M. Papadrakakis, Determination of RVE size for random composites with local volume fraction variation, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 305 (2016) 340–358, http://dx.doi.org/10.1016/j.cma.2016.03.002.
[33] T. Zohdi, P. Wriggers, C. Huet, A method of substructuring large-scale computational micromechanical problems, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 190 (43–44) (2001) 5639–5656, http://dx.doi.org/10.1016/S0045-7825(01)00189-X.
[34] T. Kanit, S. Forest, I. Galliet, V. Mounoury, D. Jeulin, Determination of the size of the representative volume element for random composites: statistical and numerical approach, Int. J. Solids Struct. (ISSN: 0020-7683) 40 (13) (2003) 3647–3679, http://dx.doi.org/10.1016/S0020-7683(03)00143-4, URL https://www.sciencedirect.com/science/article/pii/S0020768303001434.
[35] M. Ostoja-Starzewski, Material spatial randomness: From statistical to representative volume element, Probab. Eng. Mech. (ISSN: 0266-8920) 21 (2) (2006) 112–132, http://dx.doi.org/10.1016/j.probengmech.2005.07.007, URL https://www.sciencedirect.com/science/article/pii/S0266892005000433.
[36] J. Hadamard, Sur les problèmes aux dérivés partielles et leur signification physique, Princet. Univ. Bull. 13 (1902) 49–52.
[37] S. Kabanikhin, N. Tikhonov, V. Ivanov, M. Lavrentiev, Definitions and examples of inverse and ill-posed problems, J. Inverse and Ill-Posed Probl. - J. INVERSE ILL-POSED PROBL. 16 (2008) 317–357.
[38] S. Kollmannsberger, D. D'Angella, M. Jokeit, L. Herrmann, Deep learning in computational mechanics: An introductory course, Studies in computational intelligence, vol. 977, Springer, Cham, ISBN: 9783030765866, 2021.
[39] S. Jang, Y. Son, Empirical evaluation of activation functions and kernel initializers on deep reinforcement learning, in: 2019 International Conference on Information and Communication Technology Convergence, ICTC, 2019, pp. 1140–1142, http://dx.doi.org/10.1109/ictc46691.2019.8939854.
[40] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, second ed., Adaptive computation and machine learning series, The MIT Press, Cambridge Massachusetts, ISBN: 9780262039246, 2018.
[41] A. Apicella, F. Donnarumma, F. Isgrò, R. Prevete, A survey on modern trainable activation functions, Neural Netw. (ISSN: 0893-6080) 138 (2021) 14–32, http://dx.doi.org/10.1016/j.neunet.2021.01.026.
[42] M. Lee, GELU activation function in deep learning: A comprehensive mathematical analysis and performance, 2023, arXiv:2305.12073.
[43] D.W. Scott, Box–muller transformation, WIREs Comput. Stat. 3 (2) (2011) 177–179, http://dx.doi.org/10.1002/wics.148.
[44] H. Dong, Z. Ding, S. Zhang (Eds.), Deep reinforcement learning, in: Springer eBook Collection, Springer Singapore and Imprint Springer, Singapore, ISBN: 978-981-15-4094-3, 2020.
[45] F.A. Galatolo, M.G.C.A. Cimino, G. Vaglini, Solving the scalarization issues of advantage-based reinforcement learning algorithms, Comput. Electr. Eng. (ISSN: 0045-7906) 92 (2021) 107117, http://dx.doi.org/10.1016/j.compeleceng.2021.107117.
[46] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017, arXiv:1412.6980.
[47] E. Amid, R. Anil, C. Fifty, M.K. Warmuth, Step-size adaptation using exponentiated gradient updates, 2022, ArXiv, abs/2202.00145.
[48] Z. Li, S. Bhojanapalli, M. Zaheer, S. Reddi, S. Kumar, Robust training of neural networks using scale invariant architectures, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 12656–12684.
[49] S. Amari, Backpropagation and stochastic gradient descent method, Neurocomputing (ISSN: 0925-2312) 5 (4) (1993) 185–196, http://dx.doi.org/10.1016/0925-2312(93)90006-O.
[50] S. Ruder, An overview of gradient descent optimization algorithms, 2016, CoRR, abs/1609.04747.
[51] D. Kim, Normalization methods for input and output vectors in backpropagation neural networks, Int. J. Comput. Math. 71 (2) (1999) 161–171, http://dx.doi.org/10.1080/00207169908804800.
[52] J. Zhang, T. He, S. Sra, A. Jadbabaie, Why gradient clipping accelerates training: A theoretical justification for adaptivity, 2020, arXiv:1905.11881.
[53] A. Krogh, J. Hertz, A simple weight decay can improve generalization, Adv. Neural Inf. Process. Syst. 4 (1991).
[54] H. Alibrahim, L. Hussain, A. Simone, Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization, in: 2021 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2021, pp. 1551–1559.
[55] A. Candelieri, A gentle introduction to Bayesian optimization, in: 2021 Winter Simulation Conference, WSC, 2021, pp. 1–16, http://dx.doi.org/10.1109/WSC52266.2021.9715413.
[56] L. Gongjin, J.M. Tomczak, D.M. Roijers, A.E. Eiben, Time efficiency in optimization with a bayesian-evolutionary algorithm, Swarm Evol. Comput. (ISSN: 2210-6502) 69 (2022) 100970, http://dx.doi.org/10.1016/j.swevo.2021.100970, URL https://www.sciencedirect.com/science/article/pii/S2210650221001322.
[57] G. Gutiérrez, R. Torres-Avilés, M. Caniupán, cKd-tree: A compact kd-tree, IEEE Access 12 (2024) 28666–28676, http://dx.doi.org/10.1109/ACCESS.2024.3365054.
[58] Z. Hashin, S. Shtrikman, Note on a variational approach to the theory of composite elastic materials, J. Franklin Inst. (ISSN: 0016-0032) 271 (4) (1961) 336–341, http://dx.doi.org/10.1016/0016-0032(61)90032-1, URL https://www.sciencedirect.com/science/article/pii/0016003261900321.