Deep Reinforcement Learning for Inverse Inorganic Materials Design

Elton Pan[*] **Christopher Karpovich**,[*] **Elsa Olivetti**
Department of Materials Science and Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{eltonpan, ckarp, elsao}@mit.edu

Abstract

A major obstacle to the realization of novel inorganic materials with desirable properties is the inability to perform efficient optimization across both materials properties and synthesis of those materials. In this work, we propose a reinforcement learning (RL) approach to inverse inorganic materials design, which can identify promising compounds with specified properties and synthesizability constraints. Our model learns chemical guidelines such as charge and electronegativity neutrality while maintaining chemical diversity and uniqueness. We demonstrate a multi-objective RL approach, which can generate novel compounds with targeted materials properties including formation energy and bulk/shear modulus alongside a lower sintering temperature synthesis objectives. Using this approach, the model can predict promising compounds of interest, while suggesting an optimized chemical design space for inorganic materials discovery.

1 Introduction

The discovery of new materials with desirable properties is crucial to applications in areas such as energy storage and electronics manufacturing. However, virtual materials screening is still gated by time- and resource-consuming physics-based simulations and high-throughput experimentation. In recent years, machine learning (ML) has revolutionized inverse materials design, where given one or more target properties, a model is trained to generate candidate compounds that satisfy the desired constraints. Generative neural networks such as autoencoders have been used to optimize organic molecules with desirable properties over a learned latent space [1, 2]. However, optimizing over a non-convex objective function over a high-dimensional latent space is difficult [3]. Generative adversarial networks (GANs) have also seen success in molecular generation tasks. While they circumvent the need for latent space optimization, GANs suffer from issues such as training instability and mode collapse, which make it difficult to generate compounds with high diversity and validity [4]. RL approaches to molecular generation optimize molecules with respect to a reward function. Prior work includes both policy-gradient and Q-learning approaches to optimization of *organic* molecules and has demonstrated these methods to be sample efficient, stable during training, and high performance [5, 3, 6]. To the best of our knowledge, no prior work has leveraged RL for *inorganic* materials generation. Here, we propose a deep Q-network (DQN) approach for inorganic material composition generation. Our main contributions include:

- First known demonstration of RL for inorganic materials generation
- Multi-objective optimization of both inorganic materials synthesis and property constraints
- Generation of novel, chemically diverse inorganic compounds with desirable properties

[*] Equal contribution.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) AI4Mat Workshop.

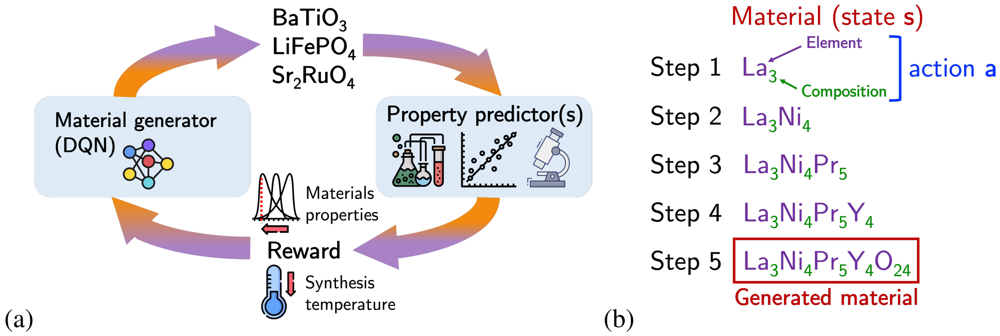

Figure 1: Schematic of (a) deep Q-network for material design where agent is assigned rewards based on predicted material properties in a feedback loop (b) material generation process. At each step of the generation sequence, state $\mathbf{s}$ is represented by material composition while action $\mathbf{a}$ constitutes the addition of an element (eg. La in step 1) and its corresponding composition (eg. 3 in step 1).

2 Methods

2.1 Deep Q-Network for materials design

Q-learning is a model-free reinforcement learning algorithm which trains an agent to behave optimally in a Markov process [7]. The agent performs actions to maximize the expected sum of all future discounted rewards given an objective function $J = \sum_{t=0}^{t_f} \gamma^t R_t(\mathbf{s}_t, \mathbf{a}_t)$ where $\gamma \in [0, 1]$ is the discount factor and R_t represents the reward at step t given values $\mathbf{s}_t$ and $\mathbf{a}_t$. For a policy π an action-value function can be defined as

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = R_{t+1} + \gamma \max_{\mathbf{a}_{t+1}} Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) \tag{1}$$

with $Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})$ being the expected sum of all the future rewards the agent receives in the resultant state $\mathbf{s}_{t+1}$.

In the context of materials design, the agent uses a policy π to maximize the expected future reward (material property) via a feedback loop by learning a deep Q-network (DQN) Q^π as shown in Fig. 1(a). The agent (material generator) generates materials, which are then evaluated by material property predictor(s), hence assigning the reward R to the agent. Here, we formulate the material generation process as a *sequence generation* task $\mathbf{s_0}, \mathbf{a_0}, R_0, \mathbf{s_1}, \mathbf{a_1}, R_1, ...$ as shown in Fig. 1(b) using DQN, where states $\mathbf{s}$ are represented by material compositions while actions $\mathbf{a}$ constitute the addition of an element and its corresponding composition to the existing incomplete material. The goal here is to learn a DQN material generator that maximizes expected rewards to generate compounds of desired properties. More details can be found in Appendices A.3. and A.4.

2.2 Constraint satisfaction for chemical validity

Without contraints a DQN may generate invalid materials through unbounded exploration. To address this, charge neutrality and electronegativity balance constraints are incorporated using a constraint oracle [8] (a trained function) $\hat{c}_{j,t}$ which is formulated as

$$\hat{c}_{j,t} = \max_{t' \geq t} \left[c_{j,t'}\right] \tag{2}$$

with $c_{j,t}$ being the jth constraint to be satisfied at step t, and the $\hat{c}_{j,t}$ is determined by the maximum level of violation to occur in all current and future steps t' in the generation process. The constraint oracles $C_j(\mathbf{x}, \mathbf{a})$ are learned functions (trained on $\hat{c}_{j,t}$) of the probability of an action satisfying a constraint in all future steps. Consequently, the *constrained* agent now follows a *modified* policy

$$\pi(\mathbf{s}) = \underset{\mathbf{a}}{\operatorname{argmax}} \left[Q^\pi(\mathbf{s}, \mathbf{a}) + \infty \cdot \min\Big(0, C_j(\mathbf{s}, \mathbf{a}) - T\Big) \right] \tag{3}$$

where $T \in [0, 1]$ is a threshold set at 0.5. This important modification to DQN ensures that actions predicted (by constraint oracle $C_j(\mathbf{s}, \mathbf{a})$) to generate invalid compounds are penalized by the last term in Eq. 3, increasing the probability of generating chemically valid compounds.

Table 1: Validity, diversity and properties of materials generated by DQN vs. baseline. **Tasks.** Form: formation energy, Bulk: bulk modulus, Shear: shear modulus, Sinter: sintering temperature, Random: random baseline. **Metrics.** Char. neut.: charge neutral, Elec. bal.: electronegativity balanced. Values and subscripts are mean and std of 10^3 generated materials, respectively.

	Validity		**Diversity**		**Material property**	
Task	% Char. neut. (↑)	% Elec. bal. (↑)	% Unique (↑)	ElMD (↑)	Property 1 (in Task)	Property 2 (in Task)
Form (↓)	41.0	60.1	100	$14.1_{7.6}$	$\mathbf{-2.54}_{0.72}$	-
Bulk (↑)	**85.3**	**65.8**	100	$7.5_{4.0}$	$\mathbf{5.35}_{0.21}$	-
Shear (↑)	**82.8**	**69.2**	100	$11.0_{5.0}$	$\mathbf{4.46}_{0.40}$	-
Sinter (↓)	**85.7**	**64.7**	100	$12.1_{7.4}$	$\mathbf{827}_{98}$	-
Sinter+Bulk (λ = 62.5)	85.9	61.6	100	$11.1_{6.2}$	826_{103}	$4.24_{0.77}$
Sinter+Bulk (λ = 125)	84.7	62.8	100	$10.1_{5.0}$	$\mathbf{886}_{102}$	$\mathbf{4.73}_{0.67}$
Sinter+Bulk (λ = 250)	92.6	63.9	100	$8.8_{4.4}$	$\mathbf{935}_{64}$	$\mathbf{5.22}_{0.36}$
Random (baseline)	59.9	42.8	100	$20.1_{9.8}$	$-1.37_{0.73}$ (Form) $4.29_{0.60}$ (Bulk) $3.65_{0.72}$ (Shear) 1011_{131} (Sinter)	

2.3 Probabilistic sampling of actions for chemical diversity

DQN agents typically take the greedy action that maximizes the expected return, rendering the policy deterministic in a static environment. However, for a materials generation task, a deterministic policy would be problematic as the agent would *always* take the same actions and generate the *same* material over and over again. This would not be optimal as materials generated by the agent must be chemically *diverse*. To address this, we implement another crucial modification to DQN. Instead of taking the greedy action, the policy π is given where the action is *sampled* from the top $n\%$ of actions such that $\pi(\mathbf{s}) = \text{random.sample}(\mathbf{a}_1^*, \ldots, \mathbf{a}_n^*)$ where $\mathbf{a}_1^*, \ldots, \mathbf{a}_n^*$ refer to the top $n\%$ of actions ranked by the Q^π and constraint oracles $C_j(\mathbf{s}, \mathbf{a})$. Related sampling strategies include Boltzmann sampling [9]. Consequently, this modification ensures that the DQN takes actions in a stochastic fashion, which enables the generation of diverse materials.

3 Experiments

We evaluate the DQN agents based on metrics related to real-world material discovery process. These metrics relate to stability and validity of compounds: formation energy per atom, charge neutrality, electronegativity balance [10]. In addition, performance-based material properties of interest include bulk modulus and shear modulus, which are measures of mechanical strength of a material. Since synthesis also plays a crucial role in real-world materials discovery, sintering temperature (a key synthesis parameter) of material is included. Finally, to evaluate the diversity of compounds generated, the element movers distance (ElMD) [11] and $\%$ uniqueness are employed.

The DQN agents are evaluated based on generating materials for 1) **single-property** 2) **multi-property** optimization.

3.1 Single-property material optimization

Setup. The task is to generate materials that optimize for a *single* property of interest. We generate 10^3 materials using the aforementioned method with hyperparameters presented in Appendix A.4. Subsequently, the properties of the generated materials are predicted using trained ML property predictor(s), and their charge neutrality and electronegativity balance are checked via the procedure in [10]. Details on dataset curation and ML property predictor training can be found in Appendices A.1 and A.2, respectively. We report the percentage of generated materials that satisfy each validity constraint. Here, the goal is to: **1)** Minimize formation energy per atom: to maximize stability of material **2)** Maximize bulk modulus and shear modulus: to maximize mechanical strength of material

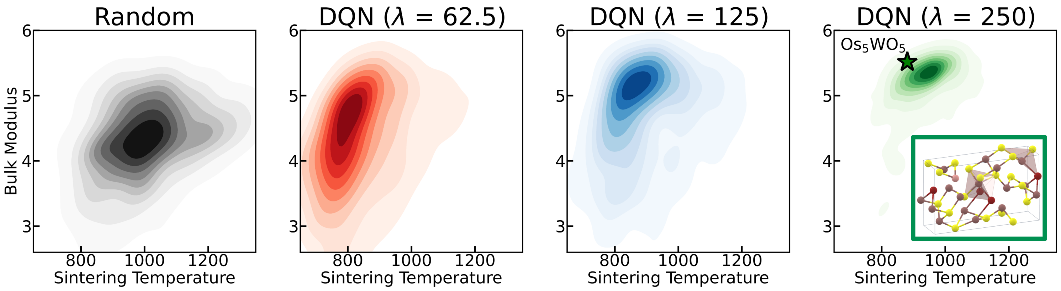

Figure 2: Property distributions of materials generated by DQN vs. baseline for the multi-property optimization task (*minimize* sintering temperature **and** *maximize* bulk modulus). DQN generates materials that have *simultaneous* improvements over the baseline. As λ increases, DQN prioritizes generating materials with high bulk modulus over those with low sintering temperature. Green star: Os_5WO_5, a *novel* DQN-generated Pareto-optimal material. Green box: its predicted crystal structure.

3) Minimize sintering temperature: lower temperatures are associated with lower energy consumption costs and greener synthesis pathways.

Results. For all tasks except for Form in Table 1, the DQN outperforms the baseline in terms of generating materials that are charge neutral and electronegativity balanced. This is due to the constraints enforced by the constraint oracles as described in Section 2.2, resulting in generation of valid materials. For uniqueness, the sampling strategy in Section 2.3 allows DQN to achieve 100$\%$ uniqueness. According to material properties in Table 1, DQN generates materials with significantly improved properties over the baseline: more negative formation energy, higher bulk/shear modulus and lower sintering temperature. Generally, the standard deviation (see subscripts) of DQN-generated material properties are lower vs. the baseline, which is supported by the lower ElMD for DQN. This could be explained by DQN prioritizing exploitation over exploration of chemical space, and is testament to the inherent trade-off between chemical diversity and optimal material property.

3.2 Multi-property material optimization

Setup. The task is to generate materials that jointly optimize for multiple properties of interest, which is more challenging and more closely represents real-world materials design. The generation protocol is similar to that of single-property task. Here, the goal is to *jointly* maximize bulk modulus and minimize sintering temperature. The reward function is now a linear combination $R = P_1 + \lambda P_2$ where P_1 and P_2 are rewards corresponding to individual material properties. λ is a hyperparameter to weigh the relative importance of individual material properties in the joint optimization. Here, we investigate the influence of varying λ on the properties of generated materials. The evaluation metrics are similar to that of the single-property task.

Results. In Fig. 2, as λ increases, the DQN prioritizes maximizing bulk modulus over minimizing sintering temperature (shifts up and right). When $\lambda = 250$ or 125, DQN generates materials that have simultaneous improvements in both material properties over the baseline (Table 1) as shown in the shift to the upper-left corner of the subplots in Fig. 2. Moreover, Table 1 clearly shows that DQN-materials have higher rates of validity (Char. neut. and Elec. bal.) but with lower diversity (ElMD). However, when $\lambda = 62.5$, it over-prioritizes minimizing sintering temperature at the expense of bulk modulus (Table 1). This shows the trade-off between the two properties, resulting in a characteristic Pareto front in multi-objective optimization, where λ is a parameter that could be tuned according to the application. We also show a *novel*, DQN-generated Pareto-efficient material Os_5WO_5 (green star in Fig. 2) with its crystal structure (space group: $Pmn2_1$) predicted by a template matching algorithm [12]. This approach has enabled us to suggest materials, but to assess them and understand their stability future work must involve validation using simulation-based quantum chemical methods.

4 Broader Impact

The ability to propose *novel*, *chemically valid* compounds with *desirable properties* in terms of synthesis and property constraints is an important step towards bridging experiment and computation to accelerate AI-guided materials design for tackling pressing modern-day challenges.

References

[1] Rafael Gómez-Bombarelli et al. “Automatic chemical design using a data-driven continuous representation of molecules”. In: *ACS central science* 4.2 (2018), pp. 268–276.

[2] Jaechang Lim et al. “Molecular generative model based on conditional variational autoencoder for de novo molecular design”. In: *Journal of cheminformatics* 10.1 (2018), pp. 1–9.

[3] Zhenpeng Zhou et al. “Optimization of molecules via deep reinforcement learning”. In: *Scientific reports* 9.1 (2019), pp. 1–10.

[4] Benjamin Sanchez-Lengeling et al. “Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC)”. In: (2017).

[5] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. “Deep reinforcement learning for de novo drug design”. In: *Science advances* 4.7 (2018), eaap7885.

[6] Marcus Olivecrona et al. “Molecular de-novo design through deep reinforcement learning”. In: *Journal of cheminformatics* 9.1 (2017), pp. 1–14.

[7] Christopher JCH Watkins and Peter Dayan. “Q-learning”. In: *Machine learning* 8.3-4 (1992), pp. 279–292.

[8] Elton Pan et al. “Constrained model-free reinforcement learning for process optimization”. In: *Computers & Chemical Engineering* 154 (2021), p. 107462.

[9] CJCH Watkins. “Models of delayed reinforcement learning”. In: *PhD thesis, Psychology Department, Cambridge University* (1989).

[10] Yabo Dan et al. “Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials”. In: *npj Computational Materials* 6.1 (2020), pp. 1–7.

[11] Cameron J Hargreaves et al. “The earth mover’s distance as a metric for the space of inorganic compositions”. In: *Chemistry of Materials* 32.24 (2020), pp. 10610–10620.

[12] Lai Wei et al. “TCSP: a Template-Based Crystal Structure Prediction Algorithm for Materials Discovery”. In: *Inorganic Chemistry* (2022).

[13] Anubhav Jain et al. “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation”. In: *APL materials* 1.1 (2013), p. 011002.

[14] Olga Kononova et al. “Text-mined dataset of inorganic materials synthesis recipes”. In: *Scientific data* 6.1 (2019), pp. 1–11.

[15] Dipendra Jha et al. “Elemnet: Deep learning the chemistry of materials from only elemental composition”. In: *Scientific reports* 8.1 (2018), pp. 1–13.

[16] Yashaswi Pathak et al. “Deep learning enabled inorganic material generator”. In: *Physical Chemistry Chemical Physics* 22.46 (2020), pp. 26935–26943.

[17] Rhys EA Goodall and Alpha A Lee. “Predicting materials properties without crystal structure: Deep representation learning from stoichiometry”. In: *Nature communications* 11.1 (2020), pp. 1–9.

[18] Logan Ward et al. “A general-purpose machine learning framework for predicting properties of inorganic materials”. In: *npj Computational Materials* 2.1 (2016), pp. 1–7.

[19] Volodymyr Mnih et al. “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).

[20] Long-Ji Lin. *Reinforcement learning for robots using neural networks*. Tech. rep. Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

A Appendix

A.1 Data acquisition and preprocessing

Datasets. To train the materials property predictor models, we leveraged a subset of inorganic materials and their computed properties contained within the Materials Project (MP) database [13]. Materials Project is a widely used inorganic materials database containing crystal structures and materials properties data calculated from high-throughput quantum mechanical calculations. To train the sintering temperature predictor model, we used a publicly released inorganic solid-state synthesis database text-mined from scientific literature using a combination of NLP and rule-based extraction techniques [14].

To preprocess the materials property data, we use a similar preprocessing strategy as [15, 10, 16]. For a formula with multiple reported formation energies, we choose the lowest one to select the most stable compound. We additionally removed all single-element compounds and any compounds with a formation energy outside the interval $[\mu - 5\sigma, \mu + 5\sigma]$, where μ and σ are the average and standard deviation of the formation energies of all compounds in the MP dataset. For each property, all entries which did not contain a computed property value were discarded. After preprocessing, we obtained datasets containing the formation energies for 68,708 compounds as well as the bulk and shear modulus values for 9,888 compounds. For bulk and shear moduli, property values were taken as their corresponding logarithm.

To preprocess the extracted sintering temperature data, reactions without at least one sintering step with a reported temperature were removed. Temperatures were converted into units of Celsius (°C) and limited to between 200 °C and 2000 °C. If sintering occurred more than once in a recipe, the last operation in chronological order was taken. For reactions with more than one reported occurrence in literature (same target and precursors), the ground truth reaction condition was taken to be the average of the reported conditions. From this dataset, we extracted the experimental sintering temperatures for 12,296 inorganic compounds.

A.2 ML property prediction models

We built ML property prediction models for both materials properties (formation energy, bulk/shear modulus) and synthesis objectives (sintering temperature). For the materials property prediction, we leveraged Roost [17], a graph-composition message passing neural network architecture which can predict materials properties from composition. Models were trained based on default hyperparameters presented in package using a random 90/10 train-test split.

For sintering temperature prediction, inorganic compounds were featurized using Magpie compositional features, which are physically motivated descriptors that take the form of a 145-dimensional embedding containing stoichiometric properties, elemental properties, electronic structure, and ionic compound features [18]. To predict sintering temperature from materials composition, we trained a random forest (RF) model and optimized hyperparameters using a 5-fold cross-validation.

Metrics of these prediction models are reported in Table 2.

Table 2: Performance metrics of ML property prediction models.

Task	Model	Dataset	Test R^2	Test MAE	Test RMSE
Formation energy per atom (eV)	Roost	MP	0.913	0.1367	0.3467
Bulk modulus (log MPa)	Roost	MP	0.753	0.2331	0.4323
Shear modulus (log MPa)	Roost	MP	0.701	0.3321	0.5135
Sintering temperature (°C)	RF	MP	0.854	52.09	96.63

A.3 Deep Q-Network

In Q-learning (or DQN), the Q-value is the expected discounted reward for a given state and action, and therefore the optimal policy π^* can be found using iterative updates with the Bellman equation (Eq. (1)). Upon convergence, the optimal Q-value Q^* is defined as:

$$Q^*\left(\mathbf{s}_t, \mathbf{a}_t\right) = \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[R_{t+1} + \gamma \max_{\mathbf{a}_{t+1}} Q^*\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}\right) \mid \mathbf{s}_t, \mathbf{a}_t\right] \tag{4}$$

$Q\left(\mathbf{s}_t, \mathbf{a}_t\right)$ can be represented by a function approximator. The Q-function is approximated with a deep Q-network (DQN) Q_θ parameterized by weights θ [19].

DQN architecture. The inputs are state $\mathbf{x}_t$ and action $\mathbf{a}_t$ at the corresponding step t. The architecture of the DQN can be found in Fig. 3. Leaky ReLU is used in all fully connected (fc) layers. The 4 vectors (vide infra) are then individually passed through fc layers, and their corresponding hidden

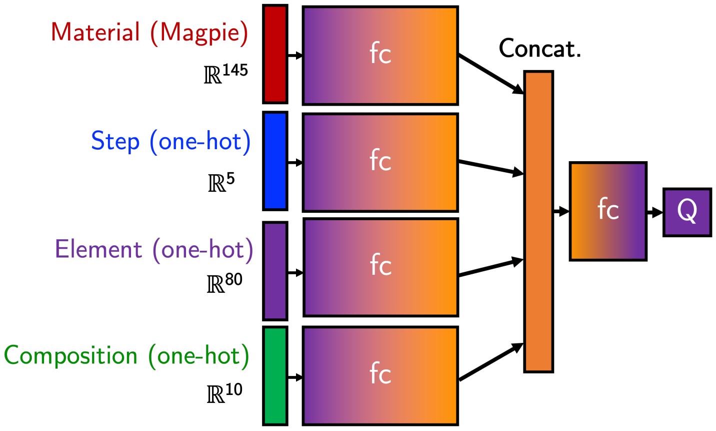

Figure 3: Architecture of deep Q-network.

states ($dim = 64$) are concatenated into a single vector ($dim = 256$) before passing through another fc layer to give the Q-value.

The state representation has two components:

1. Material composition: Each generated material (complete or incomplete) is featurized using Magpie to give a $\mathbb{R}^{145}$ vector as described in A.2.
2. Step: Step of generation. It has been reported in previous works [3] that the inclusion of the step (time-dependent policy) outperforms exclusion of such (time-independent policy). Since horizon is capped at 5 steps, this is represented by a one-hot encoding of $\mathbb{R}^{5}$.

The action representation has two components:

1. Element: The atomic species being added to the sequence. Since a element set consists of 80 elements, this is represented by a one-hot encoding of $\mathbb{R}^{80}$.
2. Composition: The numerical subscript of the atomic species that is added to the sequence. Since this ranges from 0-9, this is represented by a one-hot encoding of $\mathbb{R}^{10}$.

DQN training. The DQN is trained over 500 iterations with the use of a replay buffer of size 50000 that addresses the issue of correlated sequential samples [20]. In each iteration, 100 compounds were generated, stored in the replay buffer, and Q-network is trained on 100 samples randomly sampled from the replay buffer with smooth L1 loss function and Adam optimizer with learning rate of 0.01. Initial exploration is encouraged using an ϵ-greedy policy starting with high ϵ value of 0.99, which is decayed (by a factor of 0.99 after each iteration) over the course of training to ensure eventual exploitation and convergence to the optimal policy. Discount factor γ is set at 0.9. Models are trained on a single NVIDIA RTX A5000 GPU. Training a DQN agent generally takes 2-3 hours.

Table 3: Reward functions of optimization tasks. E = formation energy per atom, K = bulk modulus, G = shear modulus, S = sintering temperature, λ = weighing coefficient to weigh individual rewards.

Task	Reward
Formation energy per atom	$-E$
Bulk modulus	K
Shear modulus	G
Sintering temperature	$-S$
Sintering temperature + Bulk modulus	$-S + \lambda K$

A.4 Material generation

The material generation process is formulated as a *sequence generation* task $\mathbf{s_0}, \mathbf{a_0}, R_0, \mathbf{s_1}, \mathbf{a_1}, R_1, ...$ as shown in Fig. 1(b) with a horizon capped at 5 steps. At each step, an element (out of a set of 80 possible elements) and its corresponding composition (0-9) are appended to the existing (or empty) material. If the corresponding composition is 0, the agent does not add the element. This allows the agent to generate materials of any length of ≤ 5 elements (length of horizon) hence increasing chemical diversity. In all experiments in this work, we restrict to generating oxides, hence any element can be added in the first 4 steps, but only oxygen for the final step.

To increase the diversity of generated materials, the action is *sampled* from the top $n\%$ of actions ranked by the Q^{π} and constraint oracles $C_j(\mathbf{s}, \mathbf{a})$. Empirically, it was found that n can be increased from 0 up until 20 without a deterioration of material properties, hence n was set to 20 to maximize diversity of generated materials. Any larger value than n = 20 would further improve diversity, but at the expense of material properties. Rewards at all non-terminal are assigned zero until step 5 (terminal step) where the reward R is assigned according to Table 3.