

## ORIGINAL RESEARCH ARTICLE

# A multi-agent deep reinforcement learning framework for the generative design of alloys and processing routes

**Bilal Muhammed<sup>1</sup>**, **Akash Bhattacharjee<sup>2</sup>**, **B. P. Gautham<sup>3\*</sup>**, and **Amol Joshi<sup>4</sup>**

TCS Research and Innovation, Tata Consultancy Services, Pune, Maharashtra, India

## Abstract

The design of alloys and their manufacturing processes requires extensive exploration of a broad design space comprising various compositional and processing variables, many of which remain inadequately explored in practice. The existence of multiple viable processing routes for achieving desired alloy properties further complicates the design process. This paper presents a multi-agent deep reinforcement learning (DRL) framework for the *in silico* design of alloys and their processing routes/conditions tailored to specific property targets. The framework consists of distinct decentralized DRL agents, each responsible for making decisions regarding composition selection and the individual manufacturing steps involved in the process. These agents interact with their respective environments, which represent the assigned processes, and share responsibilities related to both process-specific outcomes and overall property satisfaction, as governed by the reward functions. The reward functions integrate considerations of sustainability, cost, and manufacturability into the decision-making process. A generative design step is proposed to leverage the capabilities of the trained DRL agents to produce multiple design alternatives for a given requirement. The framework is applied to the design of a hot-rolled steel sheet, exploring two feasible processing routes: Conventional casting and thin slab casting, resulting in several alternatives for each route. The framework's performance is evaluated on two experimental cases from the literature, indicating its success in biasing the sample toward the preferred solution space. A benchmark study is conducted to evaluate the framework's performance against designs produced by materials engineers for three distinct use cases, demonstrating the superior performance of the proposed framework.

### \*Corresponding author:

B. P. Gautham  
(bp.gautham@tcs.com)

**Citation:** Gautham BP. A multi-agent deep reinforcement learning framework for the generative design of alloys and processing routes. *Int J AI Mater Design*. 2026;3(1):46-68. doi: 10.36922/IJAMD025480050

**Received:** November 26, 2025

**Revised:** December 24, 2025

**Accepted:** January 5, 2026

**Published online:** January 28, 2026

**Copyright:** © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

**Publisher's Note:** AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Keywords:** Alloy and processing design; *In silico* design; Multi-agent systems; Deep reinforcement learning; Manufacturing process routes

## 1. Introduction

Alloy design is a complex, resource-intensive, and costly process aimed at achieving specific target properties required for a class of applications. Composition selection is closely linked to the manufacturing route, as processing directly affects final properties. During manufacturing, the composition of an alloy influences both desired properties and potential issues in intermediate stages of manufacturing. Therefore, alloys are often co-designed with their processing route. The vast array of compositional and process variables, together with multiscale phenomena from atomic to macro levels, further

complicates the design. In addition, the growing emphasis on sustainability is driving the need for new, efficient, and eco-friendly alloys.

Conventionally, alloy design has relied on empirical methods, including phase diagrams, tacit domain knowledge, and trial-and-error experimentation, to develop acceptable alloys. This approach is costly and limits exploration of the design space, introducing bias toward regions familiar to practitioners.<sup>1</sup> The Calculation of Phase Diagrams method has accelerated alloy design by predicting phase formation and the likelihood of meeting certain target properties in multi-component systems.<sup>2-4</sup> Similarly, many computational process modeling tools are used for optimizing the processes and realizing some target properties.<sup>5,6</sup> Frameworks, such as integrated computational material engineering (ICME) and the materials genome initiative integrate physics-based models, computational tools, and informatics to support studies across entire processing routes.<sup>7,8</sup> However, these frameworks are most often used for a few forward “what-if” studies and applying them to inverse design remains challenging due to the number of variables, the many-to-one nature of forward models, and high computational cost.

The ascent of machine learning (ML) and artificial intelligence (AI) has fundamentally transformed alloy design by enabling the optimization of compositions and manufacturing processes through the utilization of vast experimental and simulation data.<sup>9-12</sup> Initially focused on forward predictions, ML is increasingly applied to inverse design, where alloys are created to fulfill specific property requirements via algorithmic automation. Noh *et al.*<sup>13</sup> classified inverse design approaches into three strategies: High-throughput virtual screening (HTVS), global optimization, and generative models.<sup>14-18</sup> HTVS, often combined with ML, accelerates the discovery of alloys through computational screening and experimental validation.<sup>19,20</sup> Optimization algorithms, such as NSGA II<sup>21</sup> work alongside forward ML or computational models, while Bayesian methods,<sup>22-25</sup> coupled with active learning,<sup>26,27</sup> enhance inverse design by reducing simulation runs. Among generative models, techniques, such as generative adversarial networks (GAN)<sup>28-32</sup> and variational autoencoders (VAE)<sup>33</sup> are the most prominent, along with recurrent neural networks,<sup>34</sup> generative inverse design networks,<sup>35,36</sup> and deep reinforcement learning (DRL). These generative techniques specifically address the one-to-many nature of inverse problems, often producing solutions that lie outside the distribution of the source data used for training.

DRL methods have been effectively employed in materials design, optimizing compositions, structures, or processing

conditions. Early compositional design studies in drug and molecular design leveraged reinforcement learning (RL) for simplified molecular-input line-entry system (SMILES) generation, guiding molecular compositions toward desired physicochemical or biological objectives and establishing a template for discrete composition space exploration.<sup>37</sup> Subsequent work benchmarked RL against other generative approaches, such as VAEs and GANs, illustrating that RL can effectively bias sampling while preserving novelty.<sup>38</sup> Another study extends RL to multi-objective rewards that couple property satisfaction with synthesizability for inverse inorganic materials design.<sup>39</sup> In 2023, AlphaFlow introduced an AlphaGo-style RL controller for multi-step chemistry in a fluidic lab, enabling sample-efficient discovery of synthesis pathways.<sup>40</sup> More recently, DRL has been applied to multi-component alloy design, successfully navigating large compositional spaces while integrating experimental validation, outperforming Bayesian optimization under equivalent budgets.<sup>41</sup> For structural design, DRL has been utilized in digital materials design,<sup>42</sup> multi-phase microstructure design,<sup>43</sup> stretchable kirigami design for two-dimensional materials,<sup>44</sup> and microstructural optimization of silica aerogels.<sup>45</sup> Across these diverse applications, common practice relies on physics-based simulators, analytical models, or surrogate models to provide rapid feedback that guides an agent’s exploration of structural configurations. Similarly, DRL has been employed in processing design, utilizing structure-guided optimization that leverages microstructure feedback to navigate multi-step forming processes,<sup>46</sup> and in energy-efficient heat treatment studies, where an RL agent controls furnace temperature profiles to achieve target microstructures under simultaneous energy and performance constraints.<sup>47</sup>

In summary, these works have predominantly employed single-agent DRL approaches, independently applied to either compositional design, structural design, or processing condition optimization. The AtomAgents framework, while not based on DRL, demonstrates a multi-agent approach using LLM-based reasoning and simulation agents to explore alloy compositional spaces.<sup>48</sup> However, multi-agent DRL approaches have yet to be applied to materials design. To the best of authors’ knowledge, DRL has also not been applied to the joint problem of coupling compositional exploration with multi-stage processing design, nor has an approach exploring multiple feasible manufacturing process routes been considered. Recent reviews on ML-assisted manufacturing quality control highlight multi-physics effects, data scarcity, and transferability as recurring challenges, which motivate scalable learning-based process design approaches.<sup>49</sup>

This paper presents a multi-agent DRL framework for the generative design of alloys and their processing

routes for a given set of target properties and performance goals/constraints. This framework assigns different agents to distinct stages of the design process, allowing each to focus on specific tasks within their own environment simulators, which is not common in current materials design literature. That is, the agents work in a sequential fashion with each deciding on alloy composition or the following process conditions, each interacting with distinct environments representing the composition space or manufacturing process space. A two-level reward system, which provides a local reward to an agent for process-specific outcome and a global reward that rewards the overall outcome in terms of requirements satisfaction, is utilized in the framework. These are multi-objective reward functions which enable the agents to work in collaboration while incorporating sustainability, cost, and manufacturability considerations alongside process-specific constraints and target properties. Methodology to utilize available past experimental/computational design data to train the DRL agents is also discussed. A generative design step utilizes the trained DRL agents to predict multiple alloy designs for a given requirement and constraints. This framework is demonstrated using a case of design of a hot-rolled steel sheet with two feasible manufacturing routes (conventional and thin slab casting). The performance of the developed framework on experimental data from literature and a benchmark study with materials engineers are discussed.

The paper is structured as follows: Section 2 details the framework's components and methodology, including how to formulate various decision problems as DRL problems, train DRL agents, and generate design alternatives. Section 3 applies the framework to a case study on hot-rolled steel sheet design, presenting results and performance analysis. Section 4 concludes the work.

## 2. Data and methods

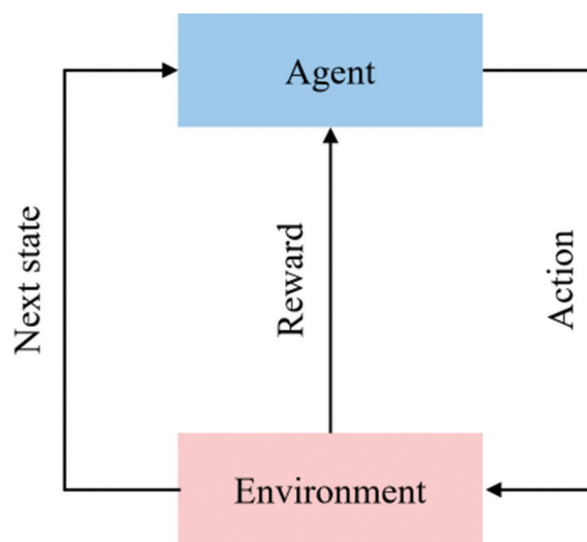
The objective of this work is to develop a framework that systematically explores composition and processing variables to generate high-performing, diverse solutions. These solutions assist decision-makers in selecting the most suitable option, potentially informed by factors beyond those directly included in the generative design process. Alloy and processing route design begins with composition selection, followed by multiple manufacturing steps. Each step refines the microstructure (through representative feature parameters), ultimately determining the final properties. To effectively predict designs that meet specified requirements, it is essential to capture the composition–process–structure–property relationships. Furthermore, each manufacturing step imposes practical constraints on the end-state parameters of both itself

and the preceding steps, underscoring the importance of efficiently modeling these processes.

Previous studies have applied single-agent DRL methods to compositional design or processing condition optimization independently, demonstrating scalability and effectiveness in large design spaces. However, extending single-agent DRL to simultaneously address composition and processing route design is challenging due to the distinct and different sets of decision variables in each processing step. Consequently, a multi-agent DRL approach is necessary. DRL combines RL with deep neural networks, enabling agents to learn optimal strategies in complex environments.<sup>50,51</sup> In a standard RL framework, the agent interacts with an environment modeled as a Markov decision process (MDP), consisting of a state space ( $S$ ), action space ( $A$ ), transition probability ( $P(s'|s,a)$ ), reward function ( $R(s,a)$ ), and discount factor ( $\gamma$ ). At each time step, the agent observes its state ( $s_t$ ), selects an action ( $a_t$ ) based on a policy ( $\pi$ ), receives a reward ( $r_t$ ), and transitions to a new state ( $s_{t+1}$ ) (Figure 1). The agent's objective is to determine the optimal policy ( $\pi$ ) that maximizes its expected total reward over time.

In multi-agent DRL, multiple autonomous agents learn and act within a shared environment. Each agent has its own policy and can interact either cooperatively or competitively with others. However, applying multi-agent DRL to inverse alloy and processing design faces several interrelated challenges<sup>52,53</sup>:

- (i) Non-stationarity: Simultaneous policy updates by multiple agents create a dynamic environment that disrupts the Markov property assumed in single-agent RL, destabilizing convergence.



**Figure 1.** Schematic representation of the working of a reinforcement learning agent

- (ii) Credit assignment: In cooperative settings, determining which agent's actions contribute to the team's success or failure becomes non-trivial, leading to inefficient learning when naïve approaches assign the same team reward to all agents.
- (iii) Scalability: As the number of design variables and agents increases, the joint action–state space grows combinatorially, hindering efficient policy learning in high-dimensional alloy design settings.
- (iv) Partial observability: In most multi-agent settings, each agent has only local observations and cannot directly observe the full environment or other agents' internal states/actions. This creates a partially observable MDP for each agent, requiring them to infer missing information or communicate, which increases complexity and can degrade performance if not properly addressed.

The problem is further complicated in alloy design due to sparse, delayed rewards, as final alloy properties often become evident only after multiple manufacturing steps, making it difficult to attribute outcomes to individual actions. The framework consists of several approaches that address the requirements of alloy and processing design while tackling common challenges in multi-agent RL described above.

## 2.1. Framework overview

The framework comprises a set of collaborative agents that operate within distinct environments representing composition selection and various processing steps, as illustrated in Figure 2. A two-level reward system is adopted, incorporating a local reward for process-specific outcomes and a global reward for the collective efforts of the agents, facilitating effective collaboration while addressing the credit assignment issue. Desired materials behaviors, such as sustainability, cost, and manufacturability, are incorporated into the reward functions to guide agents toward preferred sampling spaces.

The framework operates in two stages: Training and prediction. During training, each agent interacts with its respective environment and learns from the outcomes of its actions. Given the distinct agent–environment pairs, this approach proves more scalable than scenarios involving a joint environment. Each processing step's environment comprises computational simulators that capture the physics of the process and reward models that provide local rewards based on specific outcomes. The state space of each environment incorporates previous decisions or outcomes from preceding agents, allowing agents to be informed of earlier actions. A reverse sequential training

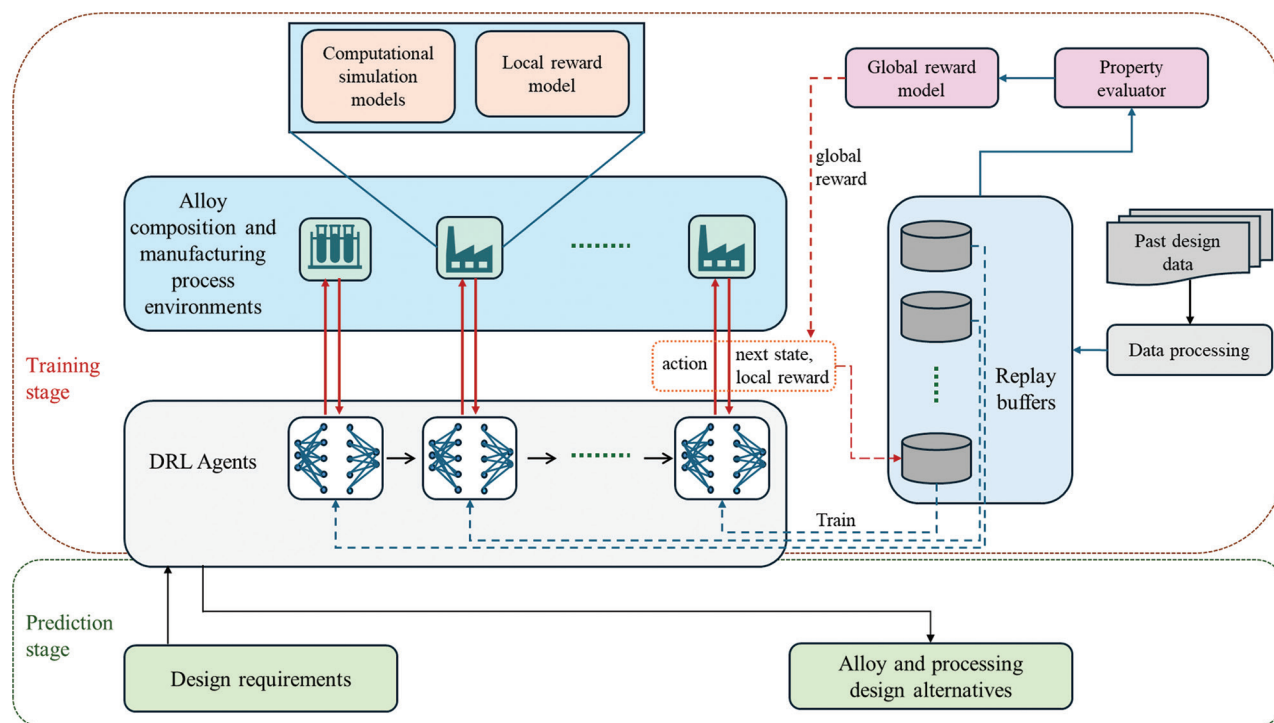


Figure 2. Components of the multi-agent deep reinforcement learning framework



approach is proposed in which the last agent is trained first, followed by the others, enabling agents to be aware of the performance and behavior of subsequent agents.

During the prediction stage, the framework employs a methodology to generate multiple solution options through a generative design step. The DRL agents function synergistically, with each model making conditional predictions informed by the output of its predecessor, ensuring a coherent and sequential flow of information and decision-making. The components of the framework are detailed in the following subsections.

## 2.2. Two-level reward setup

The reward system for a given DRL agent used for composition or process variables estimation consists of two components: Local and global rewards. Local rewards are based on an agent's performance in satisfying processing step-specific goals and constraints, while the global reward is determined by the aggregate performance of all DRL agents in the framework to meet overall requirements. This reward structure allows individual DRL agents to account for both process-specific constraints and goals, as well as final property requirements, simulating the scenario of multiple design experts collaborating to achieve the overall goal.

For the composition selection problem, the local reward may include the following:

- (i) Sustainability indices, such as elemental extraction energy and carbon footprint, which depend on the alloy composition.
- (ii) Cost of the alloy, solely dependent on its composition.
- (iii) Preliminary estimates of alloy properties solely derived from composition.

For a processing step, the local rewards may include the following:

- (i) Process-specific contributions toward sustainability indices, such as energy consumption during a rolling process.
- (ii) Cost incurred by the specific manufacturing processing step.
- (iii) Process end-state constraints, including:
  - Microstructure state constraints, such as the maximum grain size after a given processing step.
  - Process/equipment-related constraints, such as the maximum rolling force in a rolling operation.
  - Process end-state variable constraints, referring to parameters observed at the end of a manufacturing process, guided by experience, heuristics, or other factors.

The global reward reflects the satisfaction of property targets, estimated by a global reward estimator based on

achieved and target property values. The final properties of the designed alloy are evaluated by a property evaluator based on composition and attained microstructure. A weighted approach is adopted to capture the relative importance of each design output and make the rewards of each agent a competing mix of local and global objectives.

## 2.3. DRL environment setup

The environment serves as a playground for the DRL agent to learn, which interprets the agent's actions, evaluates the corresponding next state and rewards, and determines the end of a learning episode. The environment of a composition selection or processing step consists of computational simulation models, which can be analytical, empirical, AI/ML-based, or physics-based models. These models evaluate various functions that dictate state transitions and rewards.

State represents the present situation or environment that the DRL agent observes and uses to make decisions. The state of a composition selection/processing step DRL environment comprises the following elements (refer to Figure 3):

- (i) Composition or process variables already selected from preceding processing steps in the processing route.
- (ii) Process evaluation metrics, including sustainability indices, cost, process/equipment constraints, and microstructural state constraints.
- (iii) Microstructure state variables, representing the current microstructural state, depending on the chosen microstructure representation.
- (iv) Target properties, which are the final desired properties for the designed alloy.

In a composition selection/processing step DRL problem, the actions are the design variables that the DRL agent should learn to optimize. For composition selection, actions correspond to the composition values of alloy elements, while for a processing step, they represent the process variables. This generic problem structure allows the same DRL agents to be trained to make decisions for a processing step across different manufacturing routes, thereby increasing flexibility.

## 2.4. Agent training approach

The presence of multiple DRL agents, each with competing process-specific and overall objectives, necessitates a novel training approach. In this methodology, the DRL agents responsible for composition selection and various processing step problems are trained in reverse sequence: The last processing step DRL agent is trained first, followed by the preceding agents. During the training of each DRL

agent, subsequent agents in the process route are employed as trained prediction models operating in evaluation mode (refer to Figure 4). This implies that only one agent is trained at a time. Owing to the conditional nature of this training, the trained DRL agents must perform well across a broad range of state conditions, as their performance directly impacts the achievable performance of the agent currently in training. Consequently, it is recommended to enhance the training of each DRL agent by utilizing any available past design data.

The DRL agents undergo online training within their respective environments, during which experiences from learning episodes are stored in the agent's memory, referred to as the replay buffer. The deep neural networks within these DRL agents learn from batches of experiences stored in this replay buffer, formatted as 'state, reward, next state, done' tuples. However, relying solely on online training of the DRL agents can be computationally expensive, as it necessitates executing simulation models

to determine the outcomes of each set of actions selected for a given processing step during every learning episode. Past design data, typically gathered from experimental or computational model-based studies, include selected composition and process parameters and the resulting output parameters at the end of each processing step. This data are transformed into learning experiences for the inverse design problem by setting random target conditions and evaluating the corresponding local and global rewards using reward models. These learning experiences are then added to the replay buffer. This approach allows the DRL agents to learn from both computational simulations and past design data, thereby accelerating the learning process.

In sequential process chains, such as alloy and processing design, the final processing step most directly determines the final microstructure and properties, whereas upstream agent actions influence outcomes through a longer causal path. Training the last agent first, therefore, (i) yields a denser learning signal due to shorter

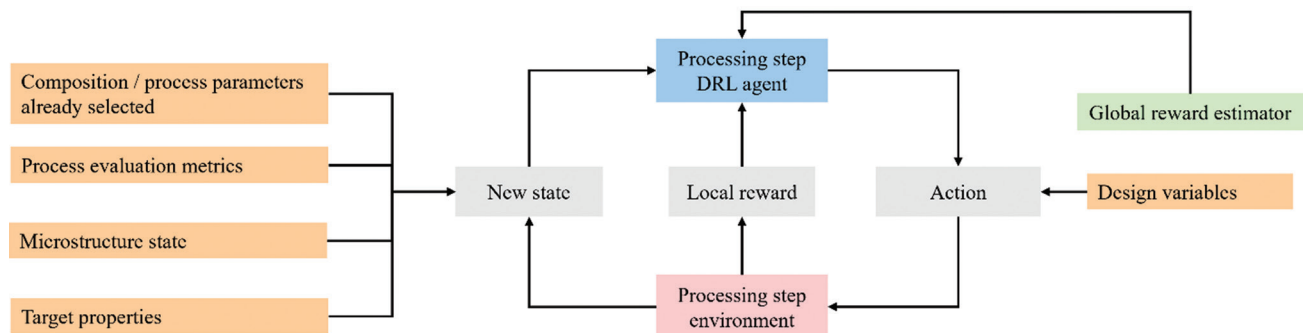


Figure 3. Environment setup of a CS/PS RL problem

Abbreviations: CS: Composition selection; DRL: Deep reinforcement learning; PS: Processing step; RL: Reinforcement learning.

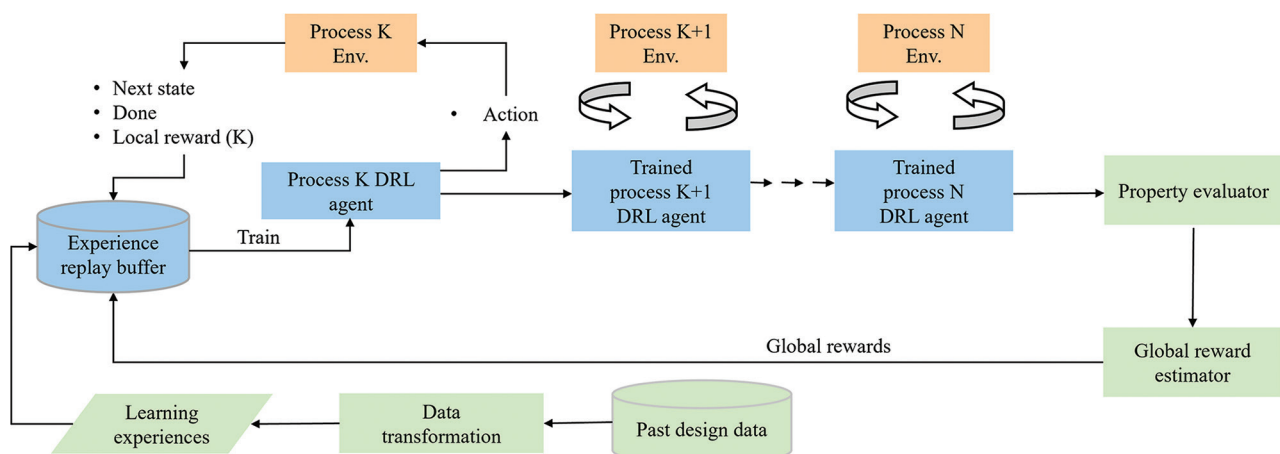


Figure 4. The training strategy adopted for deep reinforcement learning agents showing the training of an agent that keeps the succeeding trained agents in evaluation mode

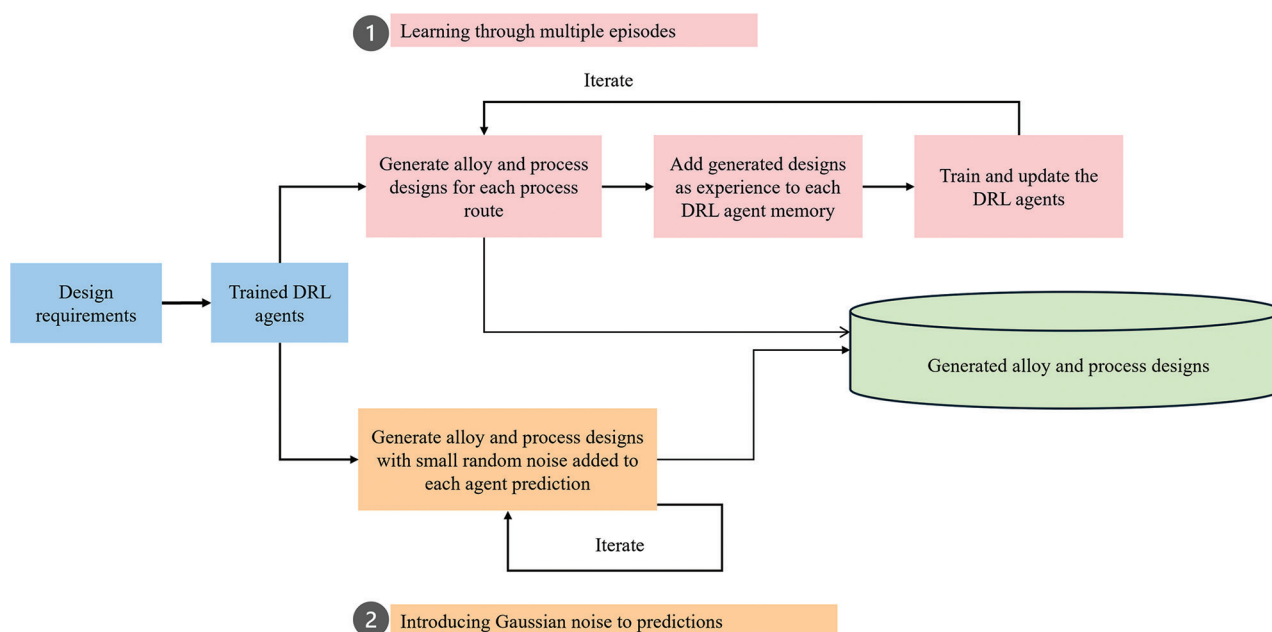
reward delay, and (ii) provides a stable downstream policy when training upstream agents, reducing non-stationarity because only one agent updates at a time while others are frozen in evaluation mode. Compared to forward sequential training, reverse training prevents early agents from learning against untrained or highly noisy downstream behavior. Joint multi-agent training is less attractive in this agent architecture as it optimizes a much larger coupled design-variable space simultaneously, which typically increases computational requirements and convergence time under ICME-based environments. Centralized-critic approaches are also not well aligned with the modular formulation adopted, since agents have heterogeneous action spaces and distinct process-specific state definitions/environments.

## 2.5. Generative design step for multiple solution generation

Generative design is a broad term that covers approaches to automatically generate multiple design options for a design problem using algorithms.<sup>54</sup> The generative design step, which is used to generate multiple designs of alloys and their manufacturing processes for a given design requirement, is illustrated in Figure 5. The workflow begins with gathering design requirements from the designer, primarily involving target values for various identified properties of the alloys. In the subsequent step, the trained DRL agents are loaded. Some DRL agents may be consumed in multiple processing routes if the processing

step is common. The two approaches used to generate multiple design solutions for the given requirement are explained below:

- (i) Learning through multiple episodes: In this approach, the DRL agents predict the composition and process variables for identified processing routes separately. After each prediction, the agents update their memory with relevant experiences and are subsequently retrained with this additional data. This iterative process, repeated for a prescribed number of times, leads to improved predictions and generates multiple designs. This method leverages the understanding of reward landscapes of each individual DRL agent and the effect of sequential agent predictions to produce multiple solutions.
- (ii) Introducing Gaussian noise to predictions: In this approach, a small random Gaussian noise is added to each agent's prediction. Multiple designs are generated through sequential predictions by the respective agents for each processing route, repeated for a prescribed number of times. This method generates multiple design solutions by slightly augmenting the preferred design space of the learned agents, along with the effect of sequential agent predictions. This approach is applicable to DRL algorithms with deterministic policies. If the algorithm selected for the DRL agents has stochastic policies, the selection of actions is always probabilistic, and therefore, this approach may not be necessary.



**Figure 5.** Generative design step used to generate multiple alloy and processing designs for a given requirement  
Abbreviation: DRL: Deep reinforcement learning.

### 3. Application to hot-rolled steel sheet design

The proposed design framework is applied to the case of a hot-rolled steel sheet design. The hot-rolled steel sheets are manufactured via a hot strip mill, which includes a casting strand, a reheating furnace, a hot rolling mill, and a cooling and coiling strand. There are two different types of casting routes considered, conventional casting (refer to Figure 6) and thin slab casting (refer to Figure 7).

In conventional casting, thick slabs (typically 200-mm-thick) of rectangular cross-section are cast and then heated in a reheating furnace before hot-deformed to achieve the final sheet thickness through rough and finish rolling. Major deformation is achieved in rough rolling, and after the finish rolling, the steel sheets are water-cooled in a run-out table before coiling.

In contrast, thin slab casting involves casting slabs with a thinner section (typically 50 mm thick), which requires significantly less reduction in the rolling mill. Thin slab casts are directly deformed in the finish rolling mill, thereby eliminating the need for reheating from a cooled state as well as the rough rolling step.

The final properties of the steel sheet are governed by the microstructure of the sheet and the alloy composition. The microstructure changes from the cast billet to steel

sheet at each processing step and hence to capture the final microstructure, microstructure evolution across the process chain must be captured. Along with the microstructure there are certain environmental parameters that can be captured to evaluate the efficiency of the processing steps and can be optimized to achieve the objectives. The following section captures the different computational tools used to simulate the process chain.

#### 3.1. ICME framework

An ICME framework is developed to model microstructure evolution throughout the process chain. This enables the prediction of the final mechanical properties of the steel sheet. In addition to microstructural aspects, the framework incorporates environmental considerations, such as energy consumption, cost, and carbon footprint. To simplify the problem at the initial stage and expedite the simulation process, a set of key parameters is considered in the framework, ensuring a more focused and manageable exploration. Figure 8 shows the different input parameters considered in the ICME framework and various microstructure, environmental, and property parameters evaluated.

Accurate physics-based modeling at each processing stage typically requires high-fidelity computational models, which can demand several days or even weeks

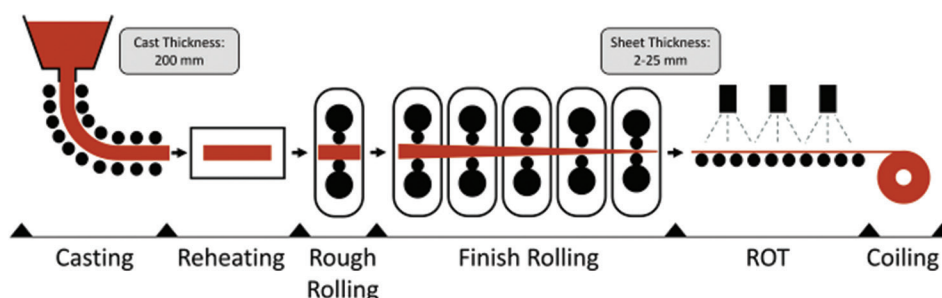


Figure 6. Conventional casting manufacturing process route for hot-rolled steel sheet  
Abbreviation: ROT: Run-out table.

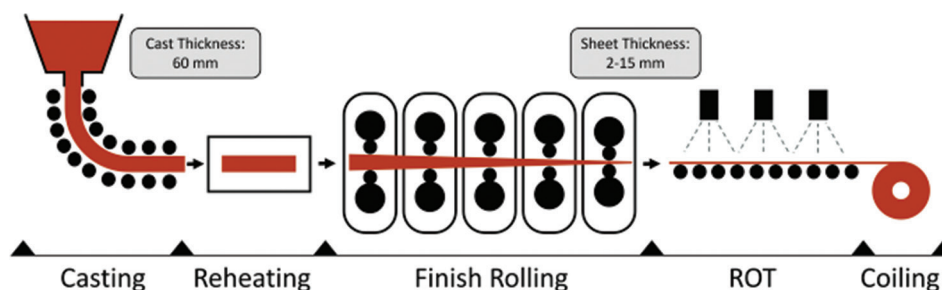
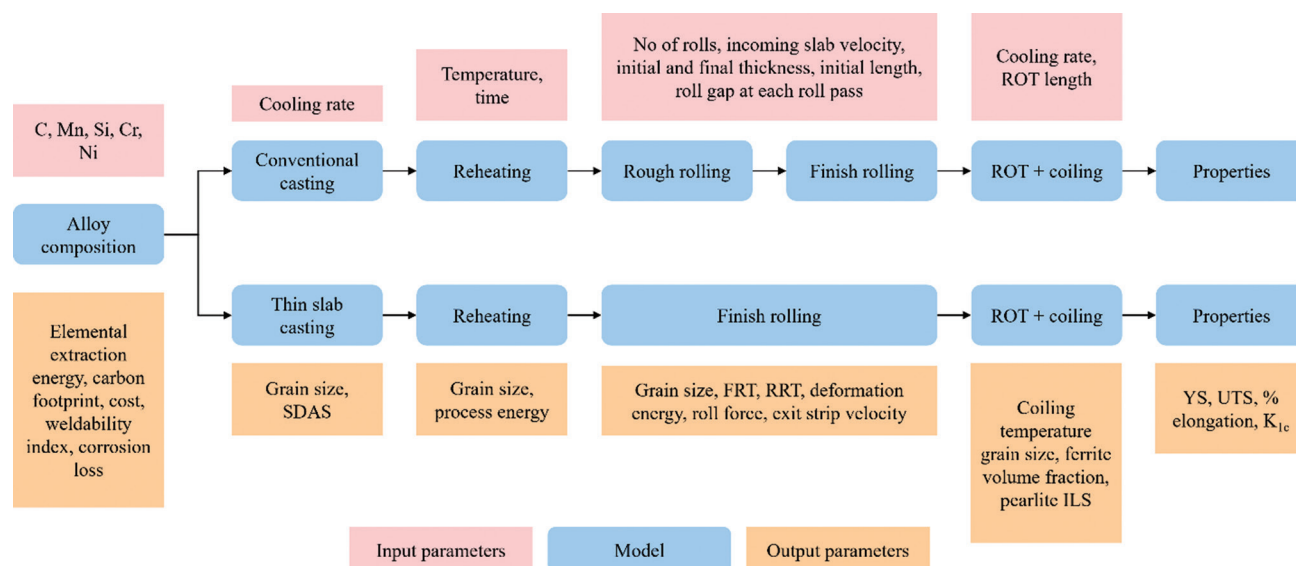


Figure 7. Thin-slab casting manufacturing process route for hot-rolled steel sheet  
Abbreviation: ROT: Run-out table.





**Figure 8.** ICME framework developed for hot-rolled steel sheet design problem

Abbreviations: FRT: Finish rolling temperature; ICME: Integrated computational material engineering; ILS: Inter-lamellar spacing;  $K_{Ic}$ : Fracture toughness; ROT: Run-out table; RRT: Rough rolling exit temperature; SDAS: Secondary dendrite arm spacing; UTS: Ultimate tensile strength; YS: Yield strength.

for a single end-to-end simulation. To address this challenge, this study employs low-fidelity empirical models that prioritize computational efficiency—defined as significantly reduced simulation time and resource consumption—while retaining essential physical trends. For example, while a full ICME simulation may require several days, the adopted low-fidelity models provide approximate results in seconds, making them suitable for early-stage design-space exploration and optimization. These models sacrifice some precision but capture dominant process–microstructure relationships within acceptable accuracy limits for the intended application. This trade-off is justified when rapid decision-making or iterative optimization is required, and predictions are validated within the process window. Details of the models used for different stages are provided in Appendix. The abbreviations of various parameters considered in the design problem are described in Table 1.

### 3.2. Problem formulation

**Given:** Target values for yield strength (YS), ultimate tensile strength (UTS), elongation (%), and sheet thickness.

**Find:** Alloy and processing designs of hot-rolled steel sheets, achieved through conventional or thin slab casting.

**Bounds:**

- Feasible ranges of alloy composition and processing parameters (independent of casting route):
  - $0.05 \leq \%C \leq 0.2$ ;  $0.5 \leq \%Mn \leq 1.5$ ;  $0.04 \leq \%Si \leq 0.25$

- $0 \leq \%Cr \leq 0.5$ ;  $0 \leq \%Ni \leq 0.5$ ;
- $1 \leq \text{Roll passes (finish)} \leq 8$ ;  $0.25 \leq \text{Strip velocity (finish)} \leq 1.5$ ;
- $50 \leq \text{run-out table (ROT) length} \leq 150$ ;  $1 \leq \text{ROT cooling rate} \leq 20$ ;
- $1 \leq \text{Sheet thickness} \leq 25$
- Bounds specific to conventional casting route:
  - $1 \leq \text{Casting cooling rate} \leq 5.0$ ;  $1100 \leq \text{Reheating temperature} \leq 1200$
  - $1800 \leq \text{Reheating time} \leq 18000$ ;  $1 \leq \text{Roll passes (rough)} \leq 8$
  - $1 \leq \text{Slab velocity (rough)} \leq 2$ ;  $70 \leq \text{Reduction (rough)} \leq 90$
- Bounds specific to the thin-slab casting route:
  - $5.0 \leq \text{Casting cooling rate} \leq 10.0$ ;  $1000 \leq \text{Reheating temperature} \leq 1100$
  - $600 \leq \text{Reheating time} \leq 1800$

**Constraints:**

- Casting:
  - $GS_{\text{cast}} < 350 \mu\text{m}$ ;  $SDAS_{\text{cast}} < 100 \mu\text{m}$
- Reheating:
  - $GS_{\text{rh}} < 450 \mu\text{m}$
- Rough rolling:
  - $GS_{\text{rr}} < 120 \mu\text{m}$ ;  $RRT > 1050^\circ\text{C}$
- Finish rolling:
  - $GS_{\text{fr}} < 120 \mu\text{m}$ ;  $FRT > 850^\circ\text{C}$ ;  $v_{\text{fr}} < 15 \text{ m/s}$
- ROT and coiling:
  - $GS_{\text{rot}} < 25 \mu\text{m}$ ;  $T_{\text{coil}} < 600^\circ\text{C}$ ;  $X_{\text{fer}} < 0.5$ ;  $ILS < 150 \text{ nm}$

Table 1. Abbreviations of model parameters used in hot-rolled steel sheet design

Parameter	Description	Parameter	Description
$LR_i$	Local reward	$wt_i$	Weight in the local or global reward
$EEE$	Extraction energy of the alloy	$CF$	Carbon footprint associated with the alloy
$Cost$	Cost of the alloy	$CE$	Carbon equivalent
$CR_{loss}$	Corrosion loss	$ul$	Upper limit
$ll$	Lower limit	$\alpha_f$	Fuzzy limit
$GR$	Global reward	$TR$	Total reward
$C_C$	Carbon composition in the alloy in wt %	$C_{Mn}$	Manganese composition in the alloy in wt %
$C_{Si}$	Silicon composition in the alloy in wt %	$C_{Cr}$	Chromium composition in the alloy in wt %
$C_{Ni}$	Nickel composition in the alloy in wt %	$A, B$	Material parameters dependent on alloy composition for corrosion loss calculation
$t$	Time	$\lambda$	Secondary dendritic arm spacing
$T_\gamma$	Cooling rate	$d_\gamma$	Austenite grain size
$T_\gamma$	Maximum temperature of the fully austenitic structure	$GS_{cast}$	Austenite grain size after casting
$SDAS_{cast}$	Secondary dendritic arm spacing after casting	$p, q_i$	Composition multiplier for the grain growth calculation
$R$	Universal gas constant	$Q_{rh}$	Energy required by the slab to reach the reheating temperature
$m$	Mass of the slab	$c_p$	Specific heat
$T_{rh}$	Reheating temperature	$T_i$	Initial temperature of the slab before reheating
$GS_{rh}$	Austenite grain size after reheating	$Q_{rh}$	Energy required during reheating
$Q_{hr}$	Energy required for reduction during the rolling process	$Q_{rr}$	Energy required during rough rolling
$\rho$	Density	$GS_{rr}$	Austenite grain size after rough rolling
$RRT$	Rough rolling the exit temperature	$RF^{RR}$	Roll force during rough rolling
$FRT$	Finish rolling the exit temperature	$Q_{fr}$	Energy required during finish rolling
$RF^{FR}$	Roll force during finish rolling	$GS_{fr}$	Austenite grain size after finish rolling
$h_f$	Thickness of the slab after deformation	$v_{fr}$	Strip exit velocity after finish rolling
$\sigma_{st}$	Surface tension	$b_f$	Width of the slab after deformation
$V$	Volume of the deformation zone	$v$	Slab velocity at the start of deformation
$\sigma_T$	Mean flow stress	$\tau$	Duration of the metal in the deformation zone
$\varepsilon$	Strain	$R_{roll}$	Roll radius
$GS_{rot}$	Ferrite grain size after ROT	$X_{fer}$	Volume fraction of ferrite
$ILS$	Pearlite interlamellar spacing	$T_{coil}$	Coiling temperature
$T_f$	Finish temperature for phase transformation	$T_s$	Start temperature for phase transformation
$n, K$	Exponent and pre-factor for the phase transformation calculation	$X_{pear_{max}}$	Equilibrium pearlite volume fraction
$d_a$	Ferrite grain size	$K_{gs}, n_{gs}, m_{gs}$	Pre-factor, cooling rate exponent, and grain size exponent for the ferrite grain size calculation
$C_o, C_i$	Fitting parameters for pearlite interlamellar spacing calculation	$Ar_i$	Austenite to ferrite transformation end temperature during cooling
$Ae_i$	Equilibrium austenite to ferrite transformation end temperature	$D_{alloy}$	Diffusivity of carbon in the austenite in the alloy
$D_{Fe-C}$	Diffusivity of carbon in austenite in the pure Fe-C system	$YS, \sigma_y$	Yield strength
$UTS$	Ultimate tensile strength	$elon$	Uniform elongation
$K_{Ic}$	Fracture toughness	$E'$	Effective young's modules
$\varepsilon_f$	Strain to failure	$n$	Strain hardening exponent

Abbreviation: ROT: Run-out table

Objectives:

- Minimize:
  - Energy consumption as a function of elemental extraction, reheating, and rolling deformation
  - Carbon footprint
  - Overall cost
- Maximize:
  - Weldability
  - Corrosion resistance

### 3.3. DRL agent architecture

The agent architecture adopted for the automated design of hot-rolled steel sheet design is described in Figure 9. It consists of six DRL agents, which are assigned to composition selection, casting, reheating, rough rolling, finish rolling, and ROT and coiling processes, respectively. The rough rolling DRL agent is used for conventional casting route only. The casting route is provided as an additional input to train and predict solutions corresponding to a given processing route. The state, actions, and rewards set up for each of the six decision problems are described in Table 2 (refer to section 2.3 for guidelines). Simulation models from the ICME framework are used to create DRL environments for each processing step.

### 3.4. Reward functions for DRL agents

In the reward functions, a fuzzy penalty approach is adopted to provide smooth gradients near constraint boundaries and to reduce sensitivity to small violations. The local reward function for each of the processing steps and the global reward function are detailed below:

Composition selection:

$$LR_{compn} = wt_{eee} \cdot f_{norm}(EEE, EEE_{max}) + wt_{cf} \cdot f_{norm}(CF, CF_{max}) + wt_{cost} \cdot f_{norm}(Cost, Cost_{max}) + wt_{ce} \cdot f_{norm}(CE, CE_{max}) + wt_{crls} \cdot f_{norm}(CRloss, CRloss_{max}) \quad (I)$$

Casting:

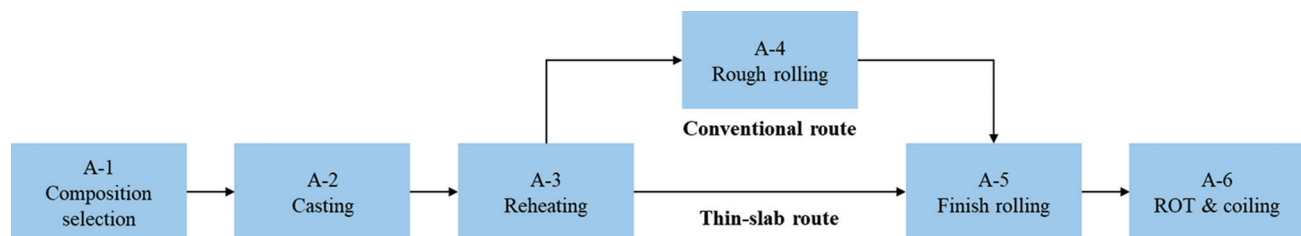
$$LR_{casting} = wt_{castgs} \cdot f_{ul}(GS_{cast}, GS_{cast_{ul}}, \alpha_f) + wt_{castdas} \cdot f_{ul}(SDAS_{cast}, SDAS_{cast_{ul}}, \alpha_f) \quad (II)$$

Reheating:

**Table 2. DRL problem setup for various processing steps in hot-rolled steel sheet design**

Problem	State	Action
Composition selection	<ul style="list-style-type: none"> <li>LL of YS, UTS, elongation</li> <li>EEE, CFP, cost, weldability, corrosion loss</li> </ul>	%C, %Mn, %Si, %Cr, %Ni
Casting	<ul style="list-style-type: none"> <li>Composition</li> <li>LL of YS, UTS, elongation</li> <li>Grain size, SDAS</li> </ul>	Cooling rate
Reheating	<ul style="list-style-type: none"> <li>Composition</li> <li>LL of YS, UTS, elongation</li> <li>Casting grain size</li> <li>Processing route</li> <li>Grain size</li> </ul>	Temperature, time
Rough rolling	<ul style="list-style-type: none"> <li>Composition</li> <li>LL of YS, UTS, elongation</li> <li>Reheating grain size, temperature</li> <li>Exit temperature, roll force</li> </ul>	No. of passes, slab velocity, % reduction
Finish rolling	<ul style="list-style-type: none"> <li>Composition</li> <li>LL of YS, UTS, elongation</li> <li>Initial grain size, temperature, recrystallization, and strain</li> <li>Sheet thickness</li> <li>Exit temperature, strip velocity, roll force</li> </ul>	No. of passes, slab velocity
ROT & coiling	<ul style="list-style-type: none"> <li>Composition</li> <li>LL of YS, UTS, elongation</li> <li>FRT, strip velocity, grain size</li> <li>Ferrite grain size, volume fraction, pearlite spacing, and coiling temperature</li> </ul>	ROT length, cooling rate

Abbreviations: CFP: Carbon footprint; EEE: Elemental extraction energy; FRT: Finish rolling temperature; LL: Lower limit; ROT: Run-out table; SDAS: Secondary dendrite arm spacing; UTS: Ultimate tensile strength; YS: Yield strength.



**Figure 9.** Deep reinforcement learning agent architecture adopted for hot-rolled steel sheet design

$$LR_{reheating} = wt_{reheatgs} \cdot f_{ul}(GS_{rh}, GS_{rh_{ul}}, \alpha_f) + wt_{reheatE} \cdot f_{norm}(Q_{rh}, Q_{rh_{max}}) \quad (III)$$

Rough rolling:

$$LR_{RR} = wt_{RRgs} \cdot f_{ul}(GS_{rr}, GS_{rr_{ul}}, \alpha_f) + wt_{RRrrt} \cdot f_{ll}(RRT, RRT_{ll}, \alpha_f) + wt_{RRE} \cdot f_{norm}(Q_{rr}, Q_{rr_{max}}) + \sum_{i=1}^n wt_{RRrf} \cdot f_{ul}(RF_i^{RR}, RF_{ul}^{RR}) \quad (IV)$$

Finish rolling:

$$LR_{FR} = wt_{FRgs} \cdot f_{ul}(GS_{fr}, GS_{fr_{ul}}, \alpha_f) + wt_{FRv} \cdot f_{ul}(v_{fr}, v_{ul}, \alpha_f) + wt_{FRrt} \cdot f_{ll}(FRT, FRT_{ll}, \alpha_f) + wt_{FRE} \cdot f_{norm}(Q_{fr}, Q_{fr_{max}}) + \sum_{i=1}^n wt_{FRrf} \cdot f_{ul}(RF_i^{FR}, RF_{max}^{FR}) \quad (V)$$

ROT and coiling:

$$LR_{ROT} = wt_{ROTgs} \cdot f_{ul}(GS_{rot}, GS_{rot_{ul}}, \alpha_f) + wt_{ROTILS} \cdot f_{ul}(ILS, ILS_{ul}, \alpha_f) + wt_{ROTvf} \cdot f_{ll}(X_{fer}, X_{fer_{ll}}, \alpha_f) + wt_{ROTT} \cdot f_{ll}(T_{coil}, T_{coil_{ll}}, \alpha_f) \quad (VI)$$

Global reward:

$$GR = wt_{ys} \cdot f_{ll}(YS, YS_{ll}, \alpha_f) + wt_{uts} \cdot f_{ll}(UTS, UTS_{ll}, \alpha_f) + wt_{elon} \cdot f_{ll}(elon, elon_{ll}, \alpha_f) + wt_{klc} \cdot f_{norm}(K_{lc}, K_{lc_{max}}) \quad (VII)$$

Each agent receives a reward consisting of its own local reward and the shared global reward, combined through fixed mixing/stiffness multipliers:

$$TR_i = wt_{LR}^i \cdot LR_i + wt_{GR} \cdot GR \quad (VIII)$$

The above reward functions utilize upper limit and lower limit fuzzy function to evaluate the fuzzy rewards depending upon the constraint type and a normalization function to normalize the rewards within the range of  $[-1, 0]$ . The functions are given below:

Upper limit fuzzy function:

$$f_{ul}(x, x_{ul}, \alpha_f) = \begin{cases} 0 & x < (1 - \alpha_f)x_{ul} \\ -1.0 & x > (1 + \alpha_f)x_{ul} \\ -1 \left( \frac{x - (1 - \alpha_f)x_{ul}}{2\alpha_f} \right) & else \end{cases} \quad (IX)$$

Lower limit fuzzy function:

$$f_{ll}(x, x_{ll}, \alpha_f) = \begin{cases} 0 & x > (1 - \alpha_f)x_{ll} \\ -1.0 & x < (1 + \alpha_f)x_{ll} \\ \left( \frac{x - (1 - \alpha_f)x_{ll}}{2\alpha_f} \right) - 1 & else \end{cases} \quad (X)$$

Normalization function:

$$f_{norm}(x, x_{max}) = -1 \left( \frac{x}{x_{max}} \right) \quad (XI)$$

In the above fuzzy functions,  $\alpha_f$  defines the fuzzy range with respect to the constraint limit. In the present scenario  $\alpha_f$  is taken as 0.1 for all the cases. Different types of weights used and the specific value of the weights selected by the domain experts are provided in the Appendix.

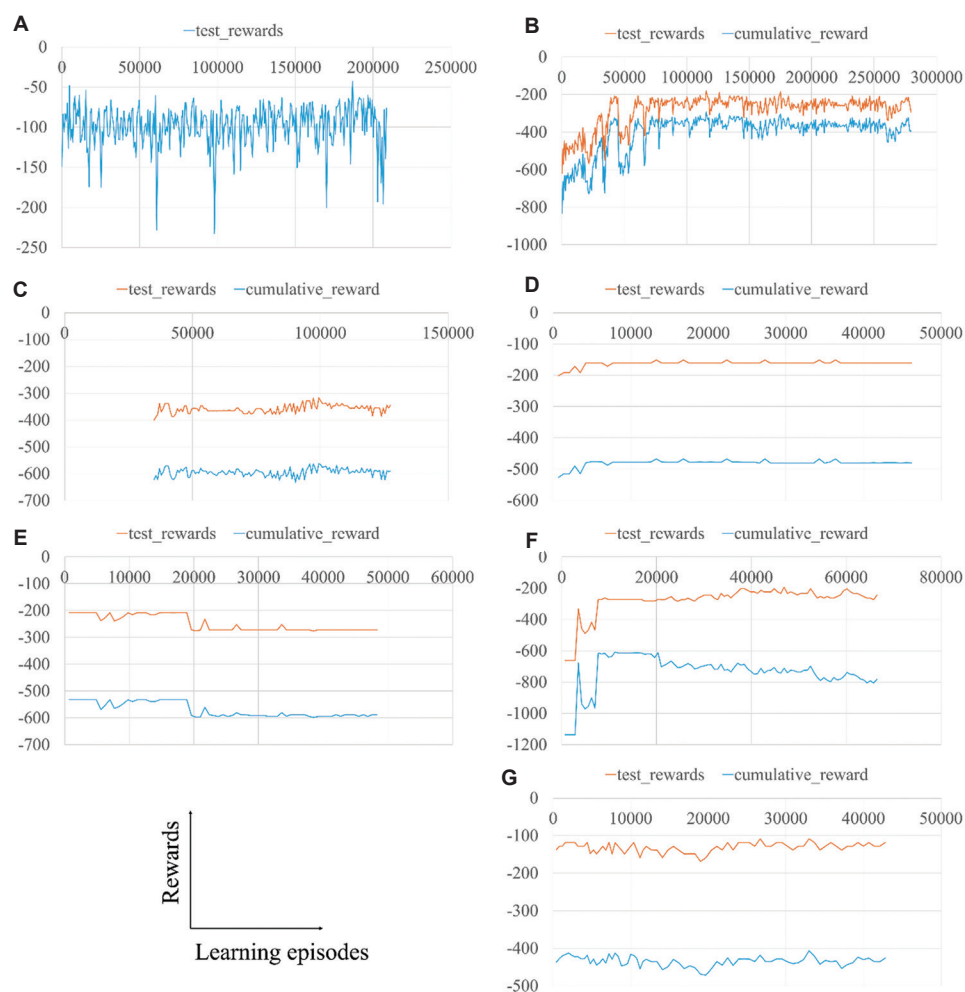
### 3.5. Agent training details

The hot-rolled steel sheet design process involves mostly continuous variables. To handle this, an off-policy DRL algorithm: Deep deterministic policy gradient (DDPG),<sup>55</sup> along with epsilon-greedy approach<sup>56</sup> for exploration during training is selected. DDPG is suitable for continuous action spaces and employs an actor-critic architecture. The actor network generates deterministic actions, while the critic network estimates their Q-values. Both networks are implemented as deep neural networks. Although DDPG is used in this study, other DRL algorithms can be substituted if needed. The DRL agents are implemented in Python using PyTorch.<sup>57</sup>

A synthetic design dataset is generated using Design of Experiments (DOE) methods, including Latin Hypercube Sampling and Box-Behnken designs, and is converted into training data as described in section 2.4. The ICME framework takes approximately 35 CPU-seconds for a full simulation. For each agent, around 1.8 million learning experiences are generated and stored in individual replay buffers.

Training is performed in reverse order of material evolution (see section 2.4). For example, while training the rough rolling agent, the trained agents for finish rolling and ROT & coiling are used to predict downstream designs. This allows the rough rolling agent to account for the performance of subsequent agents. All agents are trained through online RL, using both simulation data and synthetic design data stored in the replay buffers.





**Figure 10.** Training performance of reinforcement learning agents across multiple stages of the hot-rolled steel sheet design. (A) Rotating and coiling; (B) Finish rolling; (C) Rough rolling; (D) Reheating; (E) Casting; (F) Composition selection for the conventional casting route; (G) Composition selection for the thin-slab casting route.

### 3.5.1. Trained DRL agents

The learning curves of the DRL agents are presented in Figure 10. From the finish rolling agent onwards, a cumulative reward is plotted, representing the sum of rewards obtained by the agent being trained along with the subsequent agents. For example, during rough rolling agent training, the cumulative reward includes contributions from rough rolling, finish rolling, and ROT and coiling agents. The ROT and coiling and finish rolling agents are trained first. These agents underwent extended training durations to study convergence behavior and learning patterns. For the remaining agents, specific performance targets are defined, and training continued until these targets are met. During training, the casting agent initially preferred thin-slab casting but later switches to the conventional casting route. Instances of both casting agents are saved to train separate composition generation

**Table 3.** Testing performance of individual deep reinforcement learning agents for hot-rolled steel sheet design

DRL agent	Satisfaction (%)
ROT and coiling	82.00
Finish rolling	67.80
Rough rolling	53.50
Reheating	67.80
Casting (conventional)	58.90
Casting (thin-slab)	60.71
Composition (conventional)	64.28
Composition (thin-slab)	78.57

agents for the two routes. However, a single DRL agent can handle composition selection for both routes by incorporating the casting method as a variable into the

state.

Table 3 summarizes the testing performance of individual DRL agents for hot-rolled steel sheet design. Performance is measured by the percentage of cases where all target properties are satisfied in the first prediction. Since the testing datasets vary across agents, direct comparison is not possible. However, the performance of downstream agents influences the effectiveness of upstream agents.

The total training time for six agents is 188.5 CPU-days on an Intel Xeon processor (base clock 2.25 GHz). Once trained, deployment primarily relies on fast policy inference, with ICME evaluations used only for candidate verification and objective/constraint assessment, resulting in substantially lower computational cost per generated design. The upfront training cost can be reduced by: (i) Accelerating environment evaluations via parallel execution and GPU acceleration where model components support it, (ii) pre-loading replay buffers with additional experimental/simulation logs to reduce exploration overhead and improve sample efficiency, (iii) adopting multi-fidelity training in which agents are pre-trained using fast ML surrogate models and subsequently refined using higher-fidelity ICME evaluations (future work), and (iv) leveraging transfer learning/model reuse when extending to new manufacturing routes by reusing trained agents for overlapping steps (including composition selection) and fine-tuning only route-specific modules.

## 4. Results and discussion

The DRL agents are evaluated using an unseen dataset of 56 target cases, each specifying final property requirements and sheet thickness. The prediction behavior and solution diversity of each agent are analyzed. To benchmark the framework, materials engineers also attempt to design solutions for three target cases, and their results are compared with the DRL-based framework. Details of these analyses are discussed below.

### 4.1. Agent performance evaluation

An unseen test dataset with 56 diverse target cases is used, covering YS (100–400 MPa), UTS (400–1500 MPa),

**Table 4. Performance of the trained multi-agent deep reinforcement learning framework on the diverse test data**

Prediction type	Satisfaction (%)
First prediction	76.7
Learning through multiple episodes	94.6
Prediction with Gaussian noise	100.0

elongation (5–25%), and sheet thickness (2–25 mm). The framework initially achieves a 76.7% success rate in meeting all property targets on the first prediction (Table 4).

Two generative design strategies are then applied:

- (i) Learning through multiple episodes: Five solutions per route are generated. Together, the conventional and thin-slab agents succeed in 53 out of 56 cases, showing complementary strengths.
- (ii) Prediction with Gaussian noise: Another five solutions per route are generated with noise injection, improving diversity and achieving 100% success across all test cases.

Overall, the conventional route achieves 85.7%, and the thin-slab route achieves 98.2% success. Figure 11 illustrates various alloy and processing designs generated for a target requirement, highlighting trade-offs in weldability, corrosion resistance, energy use, cost, and carbon footprint.

### 4.2. Validation using experimental data from literature

To benchmark the multi-agent DRL framework, two experimental literature cases are used. Their compositions are input into the ICME framework, while processing parameters are determined through DoE and manual trial-and-error to match the mechanical properties. This step ensures evaluation consistency, as the ICME framework is also used to train the multi-agent framework.

- (i) Case 1: Experimental low-alloy hot-rolled steel<sup>58</sup>
  - Literature: YS = 257 MPa, UTS = 421 MPa
  - ICME framework output: YS = 256 MPa, UTS = 718 MPa, Elongation = 20%
- (ii) Case 2: Industrial-grade SM 400 B steel from JIS G 3106-2020<sup>59</sup>
  - Literature: YS = 245 MPa, UTS = 400–510 MPa, Elongation = 18%
  - ICME framework output: YS = 224 MPa, UTS = 453 MPa, Elongation = 18%

The minor deviations in properties are attributed to the performance of the ICME framework's underlying models and are accepted for validation.

Using the ICME-derived properties as input, the multi-agent DRL framework generated new alloy and processing designs for both conventional and thin-slab casting routes (refer to Table 5). These designs meet the mechanical property targets and achieve higher fracture toughness ( $K_{Ic}$ ), as the framework is trained to optimize this metric.

The compositions generated differ from the literature cases. These new compositions resulted in lower carbon footprint, extraction energy, and elemental cost. As shown in Table 6, the sustainability indices for the generated

Requirements						YS	245 MPa	UTS	450 MPa	Elongation	17%	Sheet Thickness				7 mm								
RS	Composition (wt%)					Casting		Reheating		Rough Rolling			Finish Rolling		ROT		Properties		Sustainability					
	C	Mn	Si	Cr	Ni	Route	CR	t	T	N	v	r	N	v	L	CR	W	Cor	E	Cost	CF			
	Y	0.15	0.5	0.04	0.08	0	CC	5	1800	1100	4	1	90	5	0.59	83.9	10.9	H	L	L	L	L		
	Y	0.2	0.5	0.04	0.45	0	CC	5	1800	1100	4	1.26	90	5	0.8	150	7.9	M	H	H	M	H		
	Y	0.2	0.55	0.04	0.5	0	TSC	10	600	1000	-	-	-	6	0.25	115	4.5	M	H	H	M	H		
Y	0.2	1.5	0.04	0.5	0	TSC	10	1216	1000	-	-	-	8	0.31	70.2	8.9	L	H	H	H	H			
RS	Requirement Satisfied					CR	Cooling Rate (C/s)					v	Slab Velocity (m/s)				H	High						
Y	Yes					t	Time (s)					r	% Reduction				M	Medium						
CC	Conventional Casting					T	Temperature (C)					L	Table Length (m)				L	Low						
TSC	Thin-slab Casting					N	Number of Roll Passes					W	Weldability				E	Energy						
YS	Yield Strength					UTS	Ultimate Tensile Strength					CF	Carbon Footprint											

Figure 11. Alloy and processing designs generated for a set of requirements

Table 5. Comparison of the design parameters of the literature cases obtained using the ICME framework with the gen alloy designer prediction

Stages	Parameters	Case 1			Case 2		
		Literature case	Gen alloy designer		Literature case	Gen alloy designer	
			TSC	CC		TSC	CC
Composition	C	0.06	0.070	0.149	0.2	0.09	0.2
	Mn	0.827	0.663	0.525	1.5	0.5	0.5
	Si	0.077	0.051	0.061	0.36	0.097	0.04
	Cr	0	0	0	0	0	0.017
	Ni	0.057	0	0.055	0	0	0
Casting	Cooling rate	1.0	9.4	5.0	4.2	9.7	5.0
Reheating	Temperature	1118	1013	1107	1115	1021	1100
	Time	8584	609	5546	16504	862	1800
Rough rolling	Roll passes	8	NA	4	6	NA	4
	Incoming slab velocity	1.21	NA	1.0	1.1	NA	1.0
	Reduction	87	NA	90	88	NA	90
Finish rolling	Roll passes	6	6	4	6	5	4
	Incoming slab velocity	0.36	0.35	0.44	0.31	0.25	0.32
ROT	Table length	143	103	62	132	50	51
	Cooling rate	6.0	12.5	11.9	1.0	16.5	4.7
Achieved properties	YS	256	253	261	224	224	248
	UTS	718	988	735	453	823	769
	Elongation	20.0	21.5	20.9	18	22	22
	K <sub>IC</sub>	1425	1652	1533	1222	1626	1658

Abbreviations: CC: Conventional casting; ICME: Integrated Computational Material Engineering; K<sub>IC</sub>: Fracture toughness; ROT: Run-out table; TSC: Thin-slab casting; UTS: Ultimate tensile strength; YS: Yield strength.

designs consistently outperform the literature benchmarks. This behavior reflects the framework's training objective to minimize sustainability metrics while meeting mechanical property targets. The casting cooling rates selected by the framework are at the higher end of the allowable range, while reheating temperatures are at the lower end. These choices indicate a strategy to reduce grain size and

secondary dendrite arm spacing (SDAS) during casting and suppress grain growth during reheating, thereby satisfying microstructural constraints. Process parameters across rough rolling, finish rolling, and ROT are also adjusted to meet step-specific constraints, showing the framework's ability to manage process interdependencies and optimize each stage to support downstream property goals. The

**Table 6. Comparison of the sustainability indices for the literature cases obtained using ICME framework with the gen alloy design predictions**

Parameter	Case 1			Case 2		
	Literature case	Gen alloy designer		Literature case	Gen alloy designer	
		TSC	CC		TSC	CC
Carbon footprint	231	230	230	232	230	230
Extraction energy	2483	2466	2468	2522	2464	2465
Cost	44.4	43.4	43.9	44.9	43.2	43.2

Abbreviations: CC: Conventional casting; ICME: Integrated Computational Material Engineering; TSC: Thin-slab casting.

**Table 7. Comparison of expert and DRL agents' predictions for various target requirements**

Category	Parameter	Case 1 (245, 400, 22%, 3 mm)	Case 2 (440, 590, 13%, 9 mm)	Case 3 (665, 780, 14%, 2.25 mm)
Expert predictions	Best score	−8.51	−10.60	−18.48
	Target achieved	5/44	4/36	0/33
	Average score	−27.19	−26.82	−32.99
Multi-agent DRL framework predictions	Best score	−8.27	−7.14	−11.58
	Target achieved	7/10	9/10	2/10
	Average score	−15.98	−9.57	−21.30

Abbreviation: DRL: Deep reinforcement learning.

literature validation confirms that the multi-agent DRL framework can successfully bias its sampling to meet mechanical property requirements while improving sustainability and fracture toughness, demonstrating the framework's feasibility and biasing capability under the adopted ICME models. However, experimental confirmation of the generated alloy and process designs is beyond the scope of this study and is planned for future work.

### 4.3. Benchmark study with experts

To validate the multi-agent DRL framework's performance, a benchmark study is conducted involving four experienced materials engineers in hot-rolled steel design. Both the experts and the multi-agent DRL framework are given three sets of target requirements and are allowed up to 10 iterations to identify optimal alloy compositions and process parameters. The ICME framework is used to evaluate final properties and process outcomes. The multi-agent DRL framework generated five design solutions for each of the two processing routes per case, totaling ten solutions per target, using the iterative learning approach. Each design, whether from the experts or the multi-agent DRL framework, is scored based on satisfaction of process constraints, design objectives (such as cost, sustainability, and manufacturability), and final property targets (refer to Table 7). The score is calculated as follows:

$$Score = \sum_{i=comp}^{ROT} wt_{LR}^i \cdot LR_i + wt_{GR} \cdot GR \quad (XII)$$

The multi-agent DRL framework consistently outperformed the experts in terms of best score, number of successful predictions, and average score. The complexity of the use cases is reflected in the performances of the experts and the framework alike as observed in Case 3.

## 5. Conclusion

This study introduces a multi-agent DRL framework for the *in silico* development of alloys and their corresponding processing routes, which is then demonstrated through the design of hot-rolled steel sheets considering two processing routes with multiple manufacturing processes. The framework effectively generates multiple alloy and processing designs across diverse target requirements, showcasing its adaptability and robustness.

The key contributions of the work include:

- Developing a modular, multi-agent architecture for decision-making in alloy and processing design that can integrate composition selection and multiple manufacturing processing steps effectively;
- A two-level reward formulation and agent training strategy that addresses credit assignment and ensures effective communication and collaboration between agents;



- A generative design step to produce multiple viable design alternatives for a given target, utilizing trained DRL agents.

The multi-agent framework successfully generated alloy and processing designs on two experimental cases from literature, which satisfies the property targets while improving sustainability and cost. In benchmark comparisons with experienced materials engineers, the framework consistently outperformed human experts across three use cases, achieving higher design scores, better target satisfaction rates, and improved average performance. The approach shows strong potential to reduce alloy development time and cost while enabling more sustainable design choices. Moreover, the framework is readily extendable to other material systems. However, the DRL-based setup requires significant computational resources for training and environment initialization. The selection and performance of simulation models directly influence the quality and efficiency of the trained agents.

Future work will focus on enhancing the framework through iterative training across all agents, expanding its application to other alloy systems, and exploring the integration of multi-fidelity simulation models.

## Acknowledgments

The authors sincerely thank Harshad Khadilkar for his guidance and insightful comments, which helped in the development and implementation of the DRL agent strategy. They also extend their gratitude to Dr. Gerald Tennyson for his valuable guidance and comments to refine the alloy design use-case problem. Furthermore, the authors thank Srimannarayana Pusuluri, Surya Ardhama, Harisankar K.R., and Sandeep Pusuluri for their active participation in the benchmark study. Finally, the authors would like to thank TCS Research for supporting this work.

## Funding

None.

## Conflict of interest

The authors declare that they have a pending patent titled 'Methods and systems for automated design of materials and its manufacturing process for desired properties' assigned to Tata Consultancy Services Ltd.

## Author contributions

*Conceptualization:* Bilal Muhammed, Akash Bhattacharjee, B. P. Gautham

*Data curation:* Akash Bhattacharjee

*Formal analysis:* Bilal Muhammed, Akash Bhattacharjee

*Investigation:* Bilal Muhammed, B. P. Gautham

*Methodology:* All authors

*Software:* Bilal Muhammed

*Supervision:* B. P. Gautham

*Validation:* Akash Bhattacharjee, B. P. Gautham

*Visualization:* Bilal Muhammed, Akash Bhattacharjee

*Writing – original draft:* Bilal Muhammed

*Writing – review & editing:* Akash Bhattacharjee, B. P. Gautham, Amol Joshi

## Ethics approval and consent to participate

Not applicable.

## Consent for publication

Not applicable.

## Availability of data

The data used in this study are proprietary and subject to organizational confidentiality agreements, and therefore cannot be shared.

## Further disclosure

Part of the findings have been presented by Akash Bhattacharjee in the AI/ML & Multiscale Modeling for Materials Discovery Symposium at IIT Delhi, New Delhi. The title of the presentation was "Framework for *in-silico* generative alloy design."

## References

1. Pollock TM, Van der Ven A. The evolving landscape for alloy design. *MRS Bull.* 2019;44(4):238-246.  
doi: 10.1557/mrs.2019.69
2. Ishida K. Alloy design and development of advanced materials based on phase diagrams and microstructural control. *Mater Trans.* 2020;65(5):807-819.  
doi: 10.2320/matertrans.mt-m2019362
3. Gorsse S, Tancrét F. Current and emerging practices of CALPHAD toward the development of high entropy alloys and complex concentrated alloys. *J Mater Res.* 2018;33(19):2899-2923.  
doi: 10.1557/jmr.2018.152
4. Wu M, Wang S, Huang H, Shu D, Sun B. CALPHAD aided eutectic high-entropy alloy design. *Mater Lett.* 2020;262:127175.  
doi: 10.1016/j.matlet.2019.127175
5. Carvalho SR, Ong TH, Guimarães G. A mathematical and computational model of furnaces for continuous steel strip

- processing. *J Mater Process Technol.* 2006;178(1):379-387.  
doi: 10.1016/j.jmatprotec.2006.04.083
6. Albertin E, Beneduce F, Matsumoto M, Teixeira I. Optimizing heat treatment and wear resistance of high chromium cast irons using computational thermodynamics. *Wear.* 2011;271(9-10):1813-1818.  
doi: 10.1016/j.wear.2011.01.079
7. Frydrych K, Karimi K, Pecelerowicz M, et al. Materials informatics for mechanical deformation: A review of applications and challenges. *Materials (Basel).* 2021;14(19):5764.  
doi: 10.3390/ma14195764
8. Zou C, Li J, Wang WY, et al. Integrating data mining and machine learning to discover high-strength ductile titanium alloys. *Acta Mater.* 2021;202:211-221.  
doi: 10.1016/j.actamat.2020.10.056
9. Hart GLW, Mueller T, Toher C, Curtarolo S. Machine learning for alloys. *Nat Rev Mater.* 2021;6(8):730-755.  
doi: 10.1038/s41578-021-00340-w
10. Gao X, Wang H, Tan H, Xing L, Hu Z. Data-driven machine learning for alloy research: Recent applications and prospects. *Mater Today Commun.* 2023;36:106697.  
doi: 10.1016/j.mtcomm.2023.106697
11. Golmohammadi M, Aryanpour M. Analysis and evaluation of machine learning applications in materials design and discovery. *Mater Today Commun.* 2023;35:105494.  
doi: 10.1016/j.mtcomm.2023.105494
12. Vanpoucke DEP, Van Knippenberg OSJ, Hermans K, Bernaerts KV, Mehrkanon S. Small data materials design with machine learning: When the average model knows best. *J Appl Phys.* 2020;128(5):054901.  
doi: 10.1063/5.0012285
13. Noh J, Gu GH, Kim S, Jung Y. Machine-enabled inverse design of inorganic solid materials: Promises and challenges. *Chem Sci.* 2020;11(19):4871-4881.  
doi: 10.1039/d0sc00594k
14. Debnath A, Krajewski AM, Sun H, et al. *Generative Deep Learning as a Tool for Inverse Design of High-Entropy Refractory Alloys.* [arXiv Preprint]; 2021.  
doi: 10.48550/arXiv.2108.12019
15. Nguyen P, Tran T, Gupta S, Rana S, Venkatesh S. *Hybrid Generative-Discriminative Models for Inverse Materials Design.* [arXiv Preprint]; 2018.  
doi: 10.48550/arXiv.1811.06060
16. Chen L, Zhang W, Nie Z, Li S, Pan F. Generative models for inverse design of inorganic solid materials. *J Mater Inform.* 2021;1:4.  
doi: 10.20517/jmi.2021.07
17. Sousa T, Correia J, Pereira V, Rocha M. Generative deep learning for targeted compound design. *J Chem Inf Model.* 2021;61(10):5343-5361.  
doi: 10.1021/acs.jcim.0c01496
18. Rui X, Siriwardane EMD, Song Y, et al. Active-learning-based generative design for the discovery of wide-band-gap materials. *J Phys Chem C.* 2021;125(29):16118-16128.  
doi: 10.1021/acs.jpcc.1c02438
19. Witman M, Ek G, Ling S, et al. Data-driven discovery and synthesis of high-entropy alloy hydrides with targeted thermodynamic stability. *Chem Mater.* 2021;33(11):4067-4076.  
doi: 10.1021/acs.chemmater.1c00647
20. Sheikh S, Vela B, Honarmandi P, et al. High-throughput alloy and process design for metal additive manufacturing. *NPJ Comput Mater.* 2025;11:179.  
doi: 10.1038/s41524-025-01670-x
21. Lee JW, Park WB, Lee D, Kim S, Goo NH, Sohn KS. Dirty engineering data-driven inverse prediction machine learning model. *Sci Rep.* 2020;10:20443.  
doi: 10.1038/s41598-020-77575-0
22. Couperthwaite R, Molkeri A, Khatamsaz D, Srivastava A, Allaire D, Arróyave R. Materials design through batch Bayesian optimization with multisource information fusion. *JOM.* 2020;72(10):4431-4443.  
doi: 10.1007/s11837-020-04396-x
23. Honarmandi P, Attari V, Arróyave R. Accelerated materials design using batch Bayesian optimization: A case study for solving the inverse problem from materials microstructure to process specification. *Comput Mater Sci.* 2022;210:111417.  
doi: 10.1016/j.commatsci.2022.111417
24. Khatamsaz D, Vela B, Singh P, Johnson DD, Allaire D, Arróyave R. Bayesian optimization with active learning of design constraints using an entropy-based approach. *NPJ Comput Mater.* 2023;9:74.  
doi: 10.1038/s41524-023-01006-7
25. Sardeshmukh A, Reddy S, Gautham BP. Bayesian framework for inverse inference in manufacturing process chains. *Integr Mater Manuf Innov.* 2019;8(2):95-106.  
doi: 10.1007/s40192-019-00140-9
26. Rao Z, Tung PY, Xie R, et al. Machine learning-enabled high-entropy alloy discovery. *Science.* 2022;378(6615):78-85.  
doi: 10.1126/science.abo4940
27. Wen C, Zhang Y, Wang C, et al. Machine learning assisted design of high entropy alloys with desired property. *Acta Mater.* 2019;170:109-117.  
doi: 10.1016/j.actamat.2019.03.010

28. Coto AG, Precker CE, Andersson T, *et al.* The use of generative models to speed up the discovery of materials. *Comput Methods Mater Sci.* 2023;23(1):13-26.  
doi: 10.7494/cmms.2023.1.0802
29. Li Z, Nash WT, O'Brien SP, Qiu Y, Gupta RK, Birbilis N. cardiGAN: A generative adversarial network model for multi-principal element alloys. *J Mater Sci Technol.* 2022;125:81-96.  
doi: 10.1016/j.jmst.2022.03.008
30. Dan Y, Zhao Y, Li X, Li S, Hu M, Hu J. Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials. *NPJ Comput Mater.* 2020;6:1-7.  
doi: 10.1038/s41524-020-00352-0
31. Iyer A, Dey B, Dasgupta A, Chen W, Chakraborty A. *Conditional Generative Model for Predicting Material Microstructures.* [arXiv Preprint]; 2019.  
doi: 10.48550/arXiv.1910.02133
32. Zhou Z, Shang Y, Liu X, Yang Y. A generative deep learning framework for inverse design of compositionally complex bulk metallic glasses. *NPJ Comput Mater.* 2023;9:15.  
doi: 10.1038/s41524-023-00968-y
33. Sardeshmukh A, Reddy S, Gautham BP, Bhattacharyya P. *Material Microstructure Design using VAE-Regression with Multimodal Prior.* [arXiv Preprint]; 2024.  
doi: 10.48550/arxiv.2402.17806
34. Menon D, Ranganathan R. A generative approach to materials discovery, design, and optimization. *ACS Omega.* 2022;7(30):25958-25973.  
doi: 10.1021/acsomega.2c03264
35. Chen CT, Gu GX. Generative deep neural networks for inverse materials design using backpropagation and active learning. *Adv Sci (Weinh).* 2020;7(5):1902607.  
doi: 10.1002/advs.201902607
36. Pei Z, Rozman KA, Do ÖN, *et al.* Machine-learning microstructure for inverse material design. *Adv Sci (Weinh).* 2021;8(23):2101207.  
doi: 10.1002/advs.202101207
37. Popova M, Isayev O, Tropsha A. Deep reinforcement learning for *de novo* drug design. *Sci Adv.* 2018;4(7):eaap7885.  
doi: 10.1126/sciadv.aap7885
38. Turk H, Landini E, Kunkel C, Margraf JT, Reuter K. Assessing deep generative models in chemical composition space. *Chem Mater.* 2022;34(21):9455-9467.  
doi: 10.1021/acs.chemmater.2c01860
39. Karpovich C, Pan E, Olivetti EA. Deep reinforcement learning for inverse inorganic materials design. *NPJ Comput Mater.* 2024;10:287.  
doi: 10.1038/s41524-024-01474-5
40. Volk AA, Epps RW, Yonemoto DT, *et al.* AlphaFlow: Autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning. *Nat Commun.* 2023;14:1403.  
doi: 10.1038/s41467-023-37139-y
41. Xian Y, Dang P, Tian Y, *et al.* Compositional design of multicomponent alloys using reinforcement learning. *Acta Mater.* 2024;274:120017.  
doi: 10.1016/j.actamat.2024.120017
42. Sui F, Guo R, Zhang Z, Gu GX, Lin L. Deep reinforcement learning for digital materials design. *ACS Mater Lett.* 2021;3(8):1433-1439.  
doi: 10.1021/acsmaterialslett.1c00390
43. Yang J, Tian B, Chen L, *et al.* Deep reinforcement learning for multiphase microstructure design. *Comput Mater Contin.* 2021;68(1):1285-1302.  
doi: 10.32604/cmc.2021.016829
44. Rajak P, Chen ASC, Kim JY, *et al.* Autonomous reinforcement learning agent for kirigami design of 2D materials. *NPJ Comput Mater.* 2021;7:72.  
doi: 10.1038/s41524-021-00572-y
45. Pandit P, Abdusalamov R, Itskov M, Rege A. Deep reinforcement learning for microstructural optimisation of silica aerogels. *Sci Rep.* 2024;14:1511.  
doi: 10.1038/s41598-024-51341-y
46. Dornheim J, Morand L, Zeitvogel S, Iraki T, Link N, Helm D. Deep reinforcement learning methods for structure-guided processing path optimization. *J Intell Manuf.* 2022;33:333-352.  
doi: 10.1007/s10845-021-01805-z
47. Mianroodi JR, Siboni NH, Raabe D. *Computational Discovery of Energy-Efficient Heat Treatment for Microstructure Design using Deep Reinforcement Learning.* [arXiv Preprint]; 2022.  
doi: 10.48550/arXiv.2209.11259
48. Ghafarollahi A, Buehler MJ. Automating alloy design and discovery with physics-aware multimodal multiagent AI. *Proc Natl Acad Sci USA.* 2025;122:e2414074122.  
doi: 10.1073/pnas.2414074122
49. Hu Z, Huang C, Xie L, Hua L, Yuan Y, Zhang LC. Machine learning assisted quality control in metal additive manufacturing: A review. *Adv Powder Mater.* 2025;4(6):100342.  
doi: 10.1016/j.apmate.2025.100342
50. Li Y. *Deep Reinforcement Learning: An Overview.* [arXiv Preprint]; 2017.  
doi: 10.48550/arXiv.1701.07274

51. Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. *A Brief Survey of Deep Reinforcement Learning*. [arXiv Preprint]; 2017.  
doi: 10.48550/arXiv.1708.05866
52. Hernandez-Leal P, Kartal B, Taylor ME. A survey and critique of multiagent deep reinforcement learning. *Auton Agents Multi-Agent Syst*. 2019;33(6):750-797.  
doi: 10.1007/s10458-019-09421-1
53. Canese L, Cardarilli GC, Di Nunzio L, *et al*. Multi-agent reinforcement learning: A review of challenges and applications. *Appl Sci (Basel)*. 2021;11:4948.  
doi: 10.3390/app11114948
54. Mountstephens J, Teo J. Progress and challenges in generative product design. *Computers*. 2020;9(4):80.  
doi: 10.3390/computers9040080
55. Lillicrap TP, Hunt JJ, Pritzel A, *et al*. *Continuous Control with Deep Reinforcement Learning*. [arXiv Preprint]; 2015.  
doi: 10.48550/arXiv.1509.02971
56. Sutton RS, Barto AG. Reinforcement learning: An introduction. In: *IEEE Transactions on Neural Networks*. 1<sup>st</sup> ed., vol. 9. New York: IEEE; 1998.  
doi: 10.1109/tnn.1998.712192
57. Paszke A, Gross S, Massa S, *et al*. PyTorch: An Imperative style, High-Performance Deep Learning Library. In: *33<sup>rd</sup> Annual Conference on Neural Information Processing Systems*. Vol. 32; 2019.  
doi: 10.48550/arXiv.1912.01703
58. Zambrano PC, Guerrero MP, Colas R, Leduc LA. Microstructural analysis of hot-rolled, low-carbon steel strips. *Mater Charact*. 2001;47(3-4):275-282.  
doi: 10.1016/S1044-5803(01)00188-7
59. JFE Steel Corp. *Hot Rolled Steel Sheet Catalogue*. JFE Steel Corp. Available from: <https://www.jfe-steel.co.jp/en/products/sheets/catalog/b1e-001.pdf> [Last accessed on 2025 Nov 05].
60. Jarfors AEW, Du A, Yu G, Zheng J, Wang K. On the sustainable choice of alloying elements for strength of aluminum-based alloys. *Sustainability*. 2020;12:1059.  
doi: 10.3390/su12031059
61. Ginzburg VB, Ballas R. *Flat Rolling Fundamentals*. Boca Raton, FL: CRC Press; 2000.  
doi: 10.1201/9781482277357
62. Townsend H. Effects of alloying elements on corrosion of steel in industrial atmospheres. *Corrosion*. 2001;57:497-501.  
doi: 10.5006/1.3290374
63. Miettinen J, Louhenkilpi S, Kytönen H, Laine J. IDS: Thermodynamic-kinetic-empirical tool for modelling of solidification, microstructure and material properties. *Math Comput Simul*. 2010;80:1536-1550.  
doi: 10.1016/j.matcom.2009.11.002
64. Lee SJ. Predictive model for austenite grain growth during reheating of alloy steels. *ISIJ Int*. 2013;53:1902-1904.  
doi: 10.2355/isijinternational.53.1902
65. Sims RB. Calculation of roll force and torque in hot rolling mills. *Proc Inst Mech Eng*. 1954;168:191-200.  
doi: 10.1243/pime\_proc\_1954\_168\_023\_02
66. Zhang J, Cui Z. Simulation of multi-pass hot rolling by a mixed analytical-numerical method. *Int J Appl Mech*. 2011;3:469-489.  
doi: 10.1142/S1758825111001081
67. Medina SE, Quispe A. Improved model for static recrystallization kinetics of hot-deformed austenite in low alloy and Nb/V microalloyed steels. *ISIJ Int*. 2001;41:774-781.  
doi: 10.2355/isijinternational.41.774
68. Chubenko V, Khinotskaya A, Yarosh T, Saithareiev L. Sustainable development of the steel plate hot rolling technology due to energy-power process parameters justification. *E3S Web Conf*. 2020;166:06009.  
doi: 10.1051/e3sconf/202016606009
69. Singh SB, Krishnan K, Sahay SS. Modeling non-isothermal austenite to ferrite transformation in low carbon steels. *Mater Sci Eng A*. 2007;445-446:310-315.  
doi: 10.1016/j.msea.2006.09.044
70. Umemoto M, Guo ZH, Tamura I. Effect of cooling rate on grain size of ferrite in carbon steel. *Mater Sci Technol*. 1987;3:249-255.  
doi: 10.1179/mst.1987.3.4.249
71. Wang L, Tang D, Song Y. Prediction of mechanical behavior of ferrite-pearlite steel. *J Iron Steel Res Int*. 2017;24:321-327.  
doi: 10.1016/S1006-706X(17)30046-8
72. Hahn GT, Rosenfield AR. Sources of fracture toughness: The relation between  $K_{Ic}$  and the ordinary tensile properties of metals. In: Conrad H, Jaffee RI, Kessler HP, Minkler WW, editors. *Applications Related Phenomena in Titanium Alloys*. United States: ASTM International; 1968. p. 5-32.  
doi: 10.1520/STP33617S
73. JMatPro, Sente Software Ltd. *Modelling the Plane Strain Fracture Toughness of Titanium and Aluminium Alloys*. Sente Software Ltd. Available from: <https://www.sentesoftware.co.uk/site-media/fracture-toughness-ti-al> [Last accessed on 2024 Mar 04].



## Appendix

### Appendix I

The details of various empirical models used in the ICME framework are provided in the tables below.

The sustainability model, as mentioned in Table A1, consists of empirical calculators for extraction energy, carbon footprint, and cost, all driven by alloy composition. It also includes weldability and corrosion resistance estimators, which use composition-based correlations to assess joining feasibility and durability.

The casting model, as mentioned in Table A2, uses a grain size estimator and a secondary dendrite arm spacing (SDAS) estimator to predict the as-cast microstructure based on cooling rate and alloy composition. Grain size and SDAS are calculated from empirical relationships that link faster cooling rates to finer microstructure and composition-dependent segregation effects.

The reheating model, as mentioned in Table A3, combines a grain growth model and an energy model to capture the effect of time–temperature history and alloy composition on the microstructure. Grain growth is driven by thermal activation and solute drag effects from composition, which determine the starting grain size for subsequent deformation.

The hot rolling model, as mentioned in Table A4, integrates a deformation model, microstructure evolution model, and energy model to capture the interplay between process parameters and material response. Rolling temperature and reduction ratio govern strain accumulation and stored energy, driving recrystallization and grain refinement. For each rolling pass, deformation-induced strain and temperature history are fed into the microstructure model to update grain size and recrystallization state, while the energy model tracks stored energy evolution. This coupling ensures that microstructural evolution and energy prediction remain physically interpretable and process-sensitive.

The run-out table (ROT) and coiling model, as mentioned in Table A5, incorporate ferrite and pearlite transformation kinetics to predict microstructural evolution during cooling after rolling. Using alloy composition and cooling rate, the model estimates ferrite grain size, ferrite volume fraction, and pearlite interlamellar spacing, which strongly influence final mechanical properties.

The mechanical testing model, as mentioned in Table A6, employs the rule of mixtures and iso-work

principles for strain partitioning to estimate flow curves and mechanical properties based on microstructure and alloy composition. Phase fractions and their individual strengths are combined to predict overall stress–strain behavior, enabling accurate property estimation for multi-phase microstructures.

### Appendix II

The relative importance of reward components is controlled through explicit weight assignment. The purpose of this scheme is to (i) reflect the relative importance of competing objectives, (ii) enforce feasibility constraints at each processing step, and (iii) balance step-level performance with overall property satisfaction through a two-level reward formulation.

A consistent weight-setting workflow is applied. First, the working range (lower and upper bounds) of each reward component is identified and each component is normalized to the interval [0,1]. Working ranges can be obtained by (a) performing a DoE analysis over the feasible design space of composition and process variables, or (b) using historical experimental/simulation datasets, domain knowledge, and allowable variable ranges. Normalization is carried out assuming uniform distribution within the identified working range.

Four classes of weights are then specified:

1. Intrastep component weights within each composition selection (CS)/processing step (PS) agent, capturing the relative importance of local sustainability indices, cost, and intermediate constraints.
2. Final property weights for overall requirements (e.g., YS, UTS, and elongation), enabling prioritization when some target properties are more critical than others.
3. Local–global mixing weights that control the tradeoff between CS/PS-specific local rewards and the global reward associated with final property satisfaction.
4. Constraint penalty (stiffness) multipliers that increase the enforcement strength of critical constraints relative to less critical constraints.

All weights used in the hot-rolled steel case study are specified in Table A7 to ensure reproducibility.

The local global mixing weights used in the study is  $wt_{LR}^i = 0.25$  for all the local rewards and  $wt_{GR} = 1.0$  for the global reward. The constraint penalty multiplier for the YS, UTS, and elongation is taken as 10.0, for FRT, roll forces, and coiling temperature is taken as 7.5., and for all other constraints is taken as 4.0

Table A1. Alloy composition-based properties

Phenomenon	Details	References
Extraction energy	Estimated based on extraction energy of alloying elements	60
Carbon footprint	Estimated based on carbon footprint of alloying elements	60
Cost	Approximated using alloying element cost	60
Weldability	Carbon equivalent: $CE = C_C + \frac{C_{Mn} + C_{Si}}{6} + \frac{C_{Cr}}{5} + \frac{C_{Ni}}{15}$	61
Corrosion	Material loss: $C_{loss} = At^B$	62

Table A2. Casting simulation

Phenomenon	Details	References
SDAS	$\lambda = 123T^{-0.33} \exp(-0.281C_{C+} - 0.175C_{Mn} - 0.063C_{Cr} - 0.091C_{Ni})$	63
Grain Size	$d_\gamma = 1841 - 1836 \frac{\exp(\dot{T})}{1 + \exp(\dot{T})} + 3.44 \times 10^{-5} \exp\left(\frac{T^\gamma}{80}\right)$	63

Abbreviation: SDAS: Secondary dendrite arm spacing.

Table A3. Reheating simulation

Phenomenon	Details	References
Grain growth	$\frac{\partial d_\gamma}{\partial t} = 0.107 \exp(10.313 + 0.93C_C(1 - 0.469C_{Cr}) + \sum p_i C_i) \exp\left(-\frac{77978 + 3116C_C + \sum q_i C_i}{RT}\right) t^{-0.893}$	64
Energy	$Q_{rh} = mc_p (T_{rh} - T_l)$	64

Table A4. Hot rolling simulation

Phenomenon	Details	References
Deformation	Multi-pass rolling: strain, strain rate, loads, temperature	65,66
Microstructure	Recrystallization and grain growth during interpass	67
Energy	$Q_{hr} = \left(0.5\rho b_i h_i v^3\right) + \left(2b_i h_i v \sigma_{st} + \frac{V \sigma_T}{\tau}\right) + \frac{\sigma_T \varepsilon b_i h_i R_{roll}}{\tau}$	68

Table A5. ROT and coiling simulation

Phenomenon	Details	References
Ferrite transformation	$X_{fer} = \int_{T_s}^{T_f} nK^n t^{n-1} \exp(-(Kt)^n) (1 - X_{pear_{max}}) dT$ ; Grain size: $d_\alpha = K_{gs} (d_\gamma)^{m_{gs}} (\dot{T})^{n_{gs}}$	69,70
Pearlite transformation	ILS: $\lambda = \left( C_0 + \frac{C_1}{\dot{T}} \left( \frac{D_{alloy}}{D_{Fe-C}} \right) \left( \frac{Ar_1}{Ae_1} \right) \left( \frac{D_{alloy}}{D_{Fe-C}} \right) \right)$	69,70

Table A6. Mechanical testing

Phenomenon	Details	References
Flow curve	Rule of mixture for ferrite-pearlite; iso-work for strain partitioning	71
Fracture toughness	$K_{Ic} = n\sqrt{0.05E'\sigma_y\epsilon_f}$	72,73

Table A7. Details of the weights, normalization, and constraint parameters used in the hot-rolled steel case study

Parameter	Value	Parameter	Value
$wt_{eee}$	0.4	$EEE_{max}$	2862.6
$wt_{cf}$	0.4	$CF_{max}$	246.3
$wt_{cost}$	1.0	$Cost_{max}$	56.1
$wt_{ce}$	0.6	$CE_{max}$	0.625
$wt_{closs}$	0.6	$CRloss_{max}$	175.4
$wt_{castgs}$	0.5	$GS_{cast_{ul}}$	350
$wt_{castsdas}$	0.5	$SDAS_{cast_{ul}}$	100
$wt_{reheatgs}$	1.0	$GS_{rh_{ul}}$	450
$wt_{reheatE}$	0.5	$Q_{rh_{max}}$	1000
$wt_{RRgs}$	0.8	$GS_{rr_{ul}}$	120
$wt_{RRrrt}$	0.5	$RRT_{ll}$	1050
$wt_{RRE}$	0.5	$Q_{rr_{max}}$	11.0
$wt_{RRrf}$	1.0	$RF_{ul}^{RR}$	40000
$wt_{FRgs}$	0.8	$GS_{fr_{ul}}$	30
$wt_{FRv}$	0.5	$v_{ul}$	15
$wt_{FRfrt}$	1.0	$FRT_{ll}$	850
$wt_{FRE}$	0.5	$Q_{fr_{max}}$	11.0
$wt_{FRrf}$	1.0	$RF_{ul}^{FR}$	40000
$wt_{ROTgs}$	0.8	$GS_{rot_{ul}}$	25
$wt_{ROTILS}$	0.8	$ILS$	150
$wt_{ROTVf}$	0.8	$X_{fer_{ll}}$	0.5
$wt_{ROTT}$	1.0	$T_{coil_{ll}}$	600
$wt_{ys}$	0.8	$YS_{ll}$	Required YS
$wt_{uts}$	0.8	$UTS_{ll}$	Required UTS
$wt_{clon}$	0.8	$elon_{ll}$	Required elongation
$wt_{eee}$	1.0	$K_{Ic_{max}}$	2000