ACCELERATING MULTI-PROPERTY MOLECULAR DESIGN VIA ENTROPIC-RISK-BASED COUNTERFACTUAL EXPLANATIONS

Pasan Dissanayake*
Dept. of Electrical and Computer Engineering
University of Maryland
College Park, MD, USA
pasand@umd.edu

Po-Yen Chen
Dept. of Chemical and Biomolecular Engineering
University of Maryland
College Park, MD, USA
checp@umd.edu

Sanghamitra Dutta
Dept. of Electrical and Computer Engineering
University of Maryland
College Park, MD, USA
sanghamd@umd.edu

ABSTRACT

Inverse design of composite molecular structures is computationally expensive because of vast search spaces. The challenge is further exacerbated if the desired molecule is expected to simultaneously satisfy multiple properties. For instance, biopolymer nanocomposites offer significant promise as sustainable plastic alternatives, but the candidates are required to meet multiple performance criteria simultaneously, such as mechanical strength, biodegradability, optical transparency, etc. Existing techniques often focus on only one property at a time and rely on computationally expensive genetic algorithms or perturbation-based optimization techniques. In this paper, we propose **FINDER** (**F**ast **IN**verse **D**esign via **E**ntropic-**R**isk-Based Counterfactual Explanations), a unified framework for composite molecular design that can cater to multiple target properties efficiently using a novel iterative counterfactual generation mechanism. Counterfactual explanations, a term from explainable AI, typically refer to the smallest possible changes to an input that can lead a machine learning model to give a different desired output. FINDER brings in several new innovations: (i) proposing a new constrained optimization for finding counterfactual explanations that satisfy multiple target properties; (ii) introducing a flexible tuning knob via entropic risk that balances different properties rather than a worst-case multi-property optimization (min-max); (iii) incorporating iterative projected gradient descent that is much faster; and (iv) integrating with strings like SMILES to modify functional groups and ultimately arrive at realistic and synthetically-achievable molecules. Our experiments on the high-entropy alloys benchmark successfully find candidates with minimal composition changes that satisfy multiple properties simultaneously. We further validate FINDER on a real-world lab-generated dataset of biopolymer nanocomposites, finding entirely new composite molecules with not just adjusted ratios but modified functional groups altogether.

1 INTRODUCTION

Artificial Intelligence (AI) has generated significant interest in accelerating material discovery across several applications (Han et al., 2025; Horton et al., 2025), such as batteries, solar cells, electronics, biomaterials, drug discovery, alloys, manufacturing catalysts, coatings, and advanced ceramics. A vast majority of AI-based techniques focus on data-driven forward modeling for accurately predicting molecular properties, allowing one to screen thousands of molecular candidates before they are synthesized in a laboratory. The alternative paradigm in AI-based molecular discovery is functionality-driven discovery (inverse design), i.e., declaring the desired functionality or property first (e.g., transparency above 90%) and then seeking to find candidates that have it. Inverse design is challenging because

*Corresponding author

the design space of composite molecules and the ratios of their constituents can be quite vast, rendering brute-force search too slow and computationally expensive.

Recently, counterfactual explanations (Wachter et al., 2017; Verma et al., 2024) have emerged as a potential computationally efficient alternative to brute-force exploration in molecular discovery (Wellawatte et al., 2022; Teufel et al., 2025). Counterfactual explanations, also referred to as counterfactuals, are a post-hoc explanation technique widely used in explainable machine learning applications, such as hiring, healthcare, and finance. These explanations provide insights into how a model's prediction can be altered by identifying the smallest possible changes to the input that result in a different desired output. Molecular counterfactual explanations aim to identify the minimum perturbations to a given molecule that can lead to a different property. However, prior work (Wellawatte et al., 2022) on molecular counterfactual explanations have largely focused on only a single property at a time and relied upon genetic algorithms or perturbation-based optimization techniques which are computationally expensive (could take **hours** to find a single molecular counterfactual when varying both compositions of constituents as well as functional groups).

For a vast majority of applications, we are interested in finding molecular candidates that satisfy multiple properties simultaneously. A notable example is new alloy synthesis (Rao et al., 2022; Hastings et al., 2025) by finding optimal composition of constituent elements, leading to a desired stiffness and strength. Data-driven approaches have advanced atomistic simulation of complex alloys, improving property prediction capabilities Wu & Li (2024). Another significant use case is identifying new biopolymer nanocomposites (Hauschild & Bjørn, 2023; Otoni et al., 2021; Zhang et al., 2022) to serve as sustainable plastic alternatives that satisfy multiple performance criteria, such as mechanical strength, biodegradability, optimal transparency, etc. *Finding a single molecular counterfactual that simultaneously satisfies all target properties is nontrivial.* Each individual property might lead to a different counterfactual, whereas trying to find a counterfactual that satisfies all the properties may increase cost or may not converge at all. Furthermore, simply adjusting the ratios of available components may not suffice in all scenarios; modifications to functional groups, leading to entirely new molecules or derivatives may be required to satisfy the properties simultaneously.

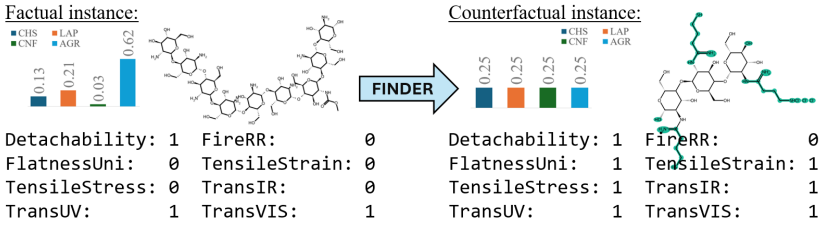


Figure 1: FINDER alters both compositions and functional groups in biopolymer nanocomposites, finding new derivatives of Chitosan within **seconds** that satisfy multiple desirable properties of sustainable plastics simultaneously.

Towards addressing these challenges, we introduce **FINDER** (**F**ast **IN**verse **D**esign via **E**ntropic-**R**isk-Based Counterfactual Explanations), a unified framework for molecular design that can cater to multiple target properties quickly and efficiently using a novel counterfactual generation mechanism (see Fig. 1). Our main contributions are as follows:

- **Entropic-Risk-Based Counterfactual Explanations Balancing Multiple Properties.** Rethinking an overly conservative worst-case approach for satisfying all properties (requires min-max optimization), we introduce a flexible tuning knob that elegantly balances different properties via entropic risk (Föllmer & Schied, 2002; Noorani et al., 2025), a measure with roots in mathematical finance. This leads us to propose a new constrained optimization for finding counterfactuals for composite molecules that we can solve quickly via iterative projected gradient descent (Bubeck, 2015). In contrast to Noorani et al. (2025) which studies the entropic-risk aggregation of models predicting the same property, FINDER aggregates the risk over heterogeneous properties, which introduces an inter-property trade-off instead of model consensus. We also derive a theoretical guarantee for the fraction of properties satisfied in Theorem 2.2.
- **Unified Input Representation Integrating Both Molecular Fingerprints and Ratios.** FINDER relies on an integrated input representation that contains provisions for: (i) molecular fingerprints (often derived from SMILES strings (Weininger, 1988)) of individual components; and (ii) their ratios. This enables us to find actionable molecular counterfactuals with not just new ratios, but discover entirely new molecules with modified functional groups.
- **Experimental Validation on Real-World Datasets.** We first apply FINDER on the high-entropy alloys (Machaka, 2021) benchmark and successfully find compositions which satisfy multiple target properties simultaneously. Next, we validate FINDER on a real-world lab-generated dataset of naturally-occurring biopolymer nanocomposites to identify sustainable plastic alternatives that satisfy multiple performance criteria simultaneously, e.g., optical and tensile properties. The input consists of 23 biopolymer fingerprints and their ratios. Here, FINDER focuses on

chitosan (Qian et al., 2023; Glentham Life Sciences, 2022; Jiménez-Gómez & Cecilia, 2020), a versatile natural fiber with superior petrochemical properties preferred in biopolymer applications. FINDER not only alters ratios but identifies new chitosan derivatives that satisfy multiple target properties simultaneously (recall Fig. 1).

2 OUR PROPOSED FRAMEWORK: FINDER

Theoretical Foundations: Counterfactuals are strongly tied to inverse design. Let m be a machine learning model that maps an input vector $x \in \mathcal{X}$ (e.g. a single molecular fingerprint or ratios of different components) to a coarse-grained target property e.g., mechanical strength 'high' or 'low', transparency $> 90\%$ or not, etc. Now, given an input $x \in \mathcal{X}$ with $m(x)$ being undesirable as is (e.g., $m(x)$ = 'low' for mechanical strength), a counterfactual is another nearby instance $x' \in \mathcal{X}$ for which the property is in a desirable range, e.g., $m(x')$ = 'high'. Formally,

$$x' = \arg\min_{u \in \mathcal{X}} c(x, u) \quad \text{s.t.} \quad \ell(m(u)) < \tau. \tag{P1}$$

Here $c(x, u)$ is the cost, e.g., molecular distance or similarity between x and u, and $\ell(.)$ is a loss which ensures $m(x')$ is in the desired class (satisfies properties). We define ℓ to be $\ell(m(u)) = 1 - m(u)$ where $m(u) \in [0, 1]$, so that the model predictions become as close to 1 as possible. Additional considerations for counterfactuals include sparsity, chemical feasibility, etc. Few works in material science used genetic algorithms (Chakraborti, 2004) or perturbation-based optimization (Nigam et al., 2021; Wellawatte et al., 2022) to find counterfactuals x' which are model agnostic but computationally expensive. In explainability literature, gradient-based solvers have been used for counterfactual search in finance and healthcare (Wachter et al., 2017; Verma et al., 2024) when the model m is differentiable.

Interestingly, despite advances in counterfactual explanations, the problem of finding a single counterfactual that simultaneously satisfies multiple target properties is nontrivial. Formally, if $\mathcal{M}$ is the set of all the models predicting different properties, P1 can be extended to the worst-case optimization given by:

$$x' = \arg\min_{u \in \mathcal{X}} c(x, u) \quad \text{s.t.} \quad \max_{m \in \mathcal{M}} \ell(m(u)) < \tau. \tag{P2}$$

The worst-case loss $\max_{m \in \mathcal{M}} \ell(m(u))$ hedges against the worst possible (most challenging) property and can be overly conservative. Sometimes, it may not even identify any counterfactual if there is no feasible region where all the desired properties overlap. To mitigate this issue, we propose a new flexible constrained optimization using entropic risk measure (Föllmer & Schied, 2002), a tool from mathematical finance. Entropic risk allows a tuning knob to balance different properties, while trading off computational cost. A formal definition is given below:

Definition 2.1 (Empirical Entropic Risk)**.** *Let $m_n (n = 1, \ldots, N)$ be the set of property-predicting models. The empirical entropic risk $\hat{\rho}_\theta^{ent}(x)$ for the input $x \in \mathcal{X}$ and a risk aversion parameter $\theta \in \mathbb{R}^+$ is defined as:*

$$\hat{\rho}_\theta^{ent}(x) = \frac{1}{\theta} \log \left(\frac{1}{N} \sum_{n=1}^{N} e^{\theta \ell(m_n(x))} \right). \tag{1}$$

Definition 2.1 helps relax the worst-case constraint in P2 and propose a constrained optimization objective as follows:

$$x' = \arg\min_{u \in \mathcal{X}} c(x, u) \quad \text{s.t.} \quad \hat{\rho}_\theta^{\text{ent}}(u) < \tau. \tag{P3}$$

As $\theta \to \infty$, the optimization converges to the worst-case constraint (most challenging). The parameter θ is a tuning knob that can be adjusted to control how many properties we can satisfy as we show in Theorem 2.2 (proof in Appendix A). Our proposed optimization P3 elegantly merges all the desirable properties into a single constraint along with a flexible tuning knob θ. Choosing a high value of θ helps satisfy more properties but requires more computational time. τ acts as a convergence tolerance. A higher τ relaxes the feasibility criterion, allowing the algorithm to terminate earlier at the cost of achieving fewer target properties. Theorem 2.2 captures the interplay between these parameters.

Theorem 2.2. *[Guarantee on Properties Satisfied] Let x' be a counterfactual with $\hat{\rho}_\theta^{ent}(x') < \tau$ (a feasible solution for P3). Also, let $\mathbb{A}_\alpha(x')$ be the set of property-predicting models m_n which satisfy $m_n(x') > \alpha$, i.e., $\mathbb{A}_\alpha(x') = \{m_n | m_n(x') > \alpha, n = 1, \ldots, N\}$. Then, the fraction of properties satisfied, i.e., $\frac{|\mathbb{A}_\alpha(x')|}{N}$ is at least $\frac{1 - e^{\theta(\tau + \alpha - 1)}}{1 - e^{-\theta(1-\alpha)}}$.*

Proposed Algorithm: FINDER efficiently solves our proposed optimization P3 to find molecular counterfactuals that simultaneously satisfy multiple properties. The key innovations in FINDER include:

Unified Input Representation: To allow provision for altering both molecular structures and ratios, we first propose a representation based on molecular fingerprints (Rogers & Hahn, 2010) of the constituents computed from their SMILES strings. A composite molecule $x \in \mathcal{X}$ of K constituents is represented by a $K \times (d+1)$-dimensional vector, comprising of K blocks of length $(d+1)$, with each block representing a constituent of the mixture, i.e., $x = [x_1, x_2, \ldots, x_K]$ where $x \in [0,1]^{K\times(d+1)}$ and $x_i \in [0,1]^{d+1}$ for $i = 1, \ldots, K$. A constituent block consists of its molecular fingerprint of length d and its relative ratio in the mixture. Our unified input representation has three main benefits: (i) It is a denser, fixed-length representation of the mixture compared to naively specifying ratios of all possible ingredients such that only K values are non-zero; (ii) It is amenable to counterfactual modifications made to the constituents themselves in addition to the ratios (by altering the fingerprint); and (iii) with a careful encoder design, the input satisfies *permutation invariance* with respect to the blocks, i.e., the order of the constituents will not alter the prediction. We train neural network models which use this input representation to predict multiple properties.

Fast Gradient-Based Solver: Once we have the property-predicting models and a given composite molecule $x \in \mathcal{X}$, our goal is to solve P3 to obtain the desired molecular counterfactuals that satisfy the desired properties. *We propose a novel two step process based on projected gradient descent.* First, an ordinary counterfactual x' (which may not satisfy all properties) is generated using any existing counterfactual generating method ($\ell_2$norm closest counterfactual in our case). Then, this partially valid counterfactual is updated until the entropic risk constraint $\rho_\theta^{\text{ent}}(x') < \tau$ is satisfied. This is done through a projected gradient descent process $x' \leftarrow x' - \eta \nabla_{x'} \rho_\theta^{\text{ent}}(x')$. If the counterfactual exceeds the valid region, it is projected back, e.g., ratios/fingerprints lie in $[0,1]$ (see Algorithm 1).

Mapping Back to Realistic and Synthetically-Achievable Molecules: The initial counterfactual will be a real-valued vector with values in the range $[0,1]$ from the projected gradient descent, but the molecular fingerprints are binary vectors. To obtain realistic fingerprints, we first round off the values to the nearest integer (0 or 1). Next, we find the nearest neighbor from a list of synthesizable modifications (e.g., the derivatives of Chitosan) based on the Tanimoto similarity (Tanimoto, 1958) of the two fingerprints, which becomes our final suggested molecular counterfactual.

Algorithm 1 lists the main steps of iteratively optimizing P3 using projected gradient descent (Bubeck, 2015). We use a multi-objective version of the minimum cost counterfactual generation (Wachter et al., 2017) as the initial ordinary counterfactual generation method $\mathcal{C}_p(.)$ to get a starting point (which may or may not satisfy all the properties), after which we integrate our proposed FINDER to iteratively refine this counterfactual to satisfy multiple properties. See details on Scalarized MO baseline in Appendix B for more details on $\mathcal{C}_p(.)$

Algorithm 1: FINDER: **F**ast **IN**verse **D**esign via **E**ntropic-**R**isk-Based Counterfactual Explanations

Input: Input instance x, Model ensemble $\mathcal{M} = \{m_1, \ldots, m_n\}$, $\theta > 0, \tau > 0$, Gradient descent step size η, max_iter $\in \mathbb{Z}^+$, Set of features allowed to be modified $\mathcal{F}$, Permitted ranges $\mathcal{P}$
Output: A counterfactual x' which satisfies multiple target properties simultaneously.

$x' \leftarrow \mathcal{C}_p(x, \mathcal{M})$; `// Generate ordinary counterfactual`
while $\rho_\theta^{ent}(x') \geq \tau$ ***and*** $i <$ *max_iter* **do**
  for $f \in \mathcal{F}$ **do**
    $x'[f] \leftarrow x'[f] - \eta \nabla_{x'} \rho_\theta^{\text{ent}}(x')[f]$; `// Update only the features in` $\mathcal{F}$
  end
  $x' \leftarrow$ `Project`$(x', \mathcal{P})$; `// Project features to the permitted ranges`
  $i \leftarrow i + 1$;
end
if $\rho_\theta^{ent}(x') < \tau$ **then**
  return x' ; `// Return the valid counterfactual`
else
  return *Error* ; `// Return error if a valid counterfactual is not found`
end

3 EXPERIMENTS

High-Entropy Alloys (HEAs). HEAs are a class of materials made from five or more principal elements, unlike traditional alloys that have fewer base elements. The High-Entropy Alloys dataset (Machaka, 2021) consists of 1,460 microstructural observations from 418 peer-reviewed studies. They cover a vast compositional space, along with 36

metallurgy-specific predictor features (multi-class classification). Our inverse design objective is to identify minimal changes in alloy compositions that can satisfy seven properties: Density_Calc, dSmix, dGmix, Tm, Phases, VEC, and Elect_Diff (see results in Table 1). We compare the average counterfactual generation time of FINDER with four baselines including one alternative method and three established techniques: (i) A scalarized multi-objective loss based on binary cross entropy (*Scalarized MO*), which is also used as the base counterfactual generating method for FINDER; (ii) A basic genetic algorithm (Fortin et al., 2012) (*Genetic*); (iii) Non-dominated Sorting Genetic Algorithm-II (*NSGA-II*) (Deb et al., 2002); and (iv) Bayesian optimization (Nogueira, 2014) (*Bayesian*). Table 2 presents the results. The hyperparameters for Genetic and Bayesian methods were tuned using `optuna` within computationally feasible ranges with the objective of maximizing the average number of target properties. The results show that FINDER offers a better trade-off between the generation time and the number of target properties achieved, clearly improving the base generating method. Table 6 presents the per-target achievements rates averaged over the generated counterfactuals. It can be observed that Density_Calc remains to be the most challenging target property to achieve.

Additional details and results including the model architecture (Fig. 2), proportion of counterfactuals achieving each property (Table 6), and details on the baselines and the dataset are in Appendix B.

Table 1: FINDER on HEA dataset: Avg. time taken to find a counterfactual (in seconds), Avg. distance to the counterfactual from the original instance (ℓ_2 norm), and Avg. number of target properties satisfied by the counterfactual.

τ	$\theta = 0.1$			$\theta = 1$			$\theta = 10$		
	Time	**Distance**	**# Properties**	**Time**	**Distance**	**# Properties**	**Time**	**Distance**	**# Properties**
0.3	0.9681	4.3701	6.4454	1.0879	4.3717	6.4720	2.1763	4.3623	6.9735
0.4	0.9601	4.3686	6.4454	0.9470	4.3690	6.4690	1.0954	4.3735	6.7198

Table 2: Average counterfactual generation time of FINDER and the baselines. In order to solely focus on the number of target properties achieved, we set the corresponding weight λ_{dist} to zero in each optimization. The average number of target properties achieved by the original instances is given for reference.

Method	**Hyperparameters**	**Time**	**# Properties**
FINDER	$\tau = 0.3$, $\theta = 10$, Scalarized MO as the base generating method	0.7656	6.9941
Scalarized MO	$\lambda_{\text{dist}} = 0$	0.6244	6.6165
Genetic	population size = 43, number of generations = 28, cross-over prob. = 0.662, mutation prob. = 0.148, $\lambda_{\text{dist}} = 0$	1.2046	6.0472
NSGA-II	population size = 10, number of generations = 10, cross-over prob. = 0.5, mutation prob. = 0.2, $\lambda_{\text{dist}} = 0$	4.1387	6.1386
Bayesian	number of initial points = 4, maximum number of attempts = 10, $\lambda_{\text{dist}} = 0$	6.2940	4.0000
Original instances	not applicable	n/a	3.8614

Biopolymer Nanocomposites as Plastic Alternatives. Biopolymer nanocomposites can potentially match or exceed the performance of petrochemical plastics in mechanical resilience, transparency, dielectric strength, and antimicrobial properties (Chen et al., 2024; Hauschild & Bjørn, 2023). However, the space of naturally occurring biopolymers, combined with diverse chemical functionalization routes, generates countless potential candidates for inverse design. We use a real-world lab-generated dataset of 1472 composites derived from a library of 23 naturally occurring and Generally Recognized As Safe (GRAS) components, such as cellulose, chitosan, gelatin, starch, etc. (see Appendix C). The dataset consists of biopolymer nanocomposites formed by combining these 23 components (we consider mixtures of $N = 4$ components) and their optical and mechanical properties. Of particular interest is chitosan, a prominent biopolymer derived naturally from chitin, with significant potential in sustainable material design. Given a biopolymer containing chitosan, our inverse design objective is to identify new chitosan derivatives that satisfy desirable properties. Here, FINDER is able to successfully modify the functional group leveraging Morgan fingerprints computed based on the SMILES string (going beyond simply adjusting ratios) and arrive at nearby chitosan derivatives within seconds (see Algorithm 2 in Appendix C). FINDER allows a trade-off between the computational time and the number of satisfied target properties (see Table 3).

Table 4 shows the average number of target properties achieved by counterfactuals after going through the steps of heuristic rounding-off and most similar chitosan derivative substitution. The results show that the effect is not consistent over the steps causing both increments and reductions. Table 6 presents the per-targe achievements rates averaged over the generated counterfactuals. While FireRR and TransUV remain to be the most challenging target properties to achieve, it can be observed that increasing θ systematically improves the proportion of counterfactuals

which achieve a given target property. Additional details and results including model architecture (Fig. 6) and the unified input representation (Fig. 5) are in Appendix C.

Table 3: FINDER on Biopolymer Dataset. Avg. time taken to compute a counterfactual (in seconds, **Time**), Avg. Tanimoto similarity (**Sim.**) of the counterfactual Chitosan derivative Morgan fingerprint, and Avg. number of target properties satisfied (out of 8, **# Prop.**).

τ	$\theta = 0.1$			$\theta = 0.3$			$\theta = 0.5$			$\theta = 1$			$\theta = 10$		
	Time	**Sim.**	**# Prop.**	**Time**	**Sim.**	**# Prop.**	**Time**	**Sim.**	**# Prop.**	**Time**	**Sim.**	**Prop.**	**Time**	**Sim.**	**# Prop.**
0.4	9.61	0.17	6.056	9.95	0.17	6.176	10.20	0.17	6.125	10.54	0.17	6.333	13.27	0.17	6.857
0.42	8.38	0.19	5.905	8.95	0.19	5.900	8.75	0.19	5.900	9.22	0.19	5.842	11.74	0.18	6.000
0.45	7.35	0.20	5.750	7.74	0.20	5.750	7.96	0.20	5.783	7.95	0.20	5.696	10.58	0.19	5.643
0.48	6.52	0.20	5.704	6.80	0.20	5.731	7.20	0.20	5.654	6.85	0.20	5.577	9.13	0.20	5.263

Table 4: Ablation study: Average number of target properties achieved by a counterfactual at each step of generation (out of 8 properties in total).

STEP	$\theta = 0.1$		$\theta = 1$		$\theta = 10$	
	$\tau = 0.42$	$\tau = 0.48$	$\tau = 0.42$	$\tau = 0.48$	$\tau = 0.42$	$\tau = 0.48$
STEP 1: Raw counterfactuals	5.905	5.704	5.842	5.577	6.000	5.263
STEP 2: Rounded off	6.095	5.926	5.947	5.808	6.250	5.632
STEP 3: Substituted	6.048	5.889	5.895	5.769	5.917	5.579

Limitations and Future Work: FINDER requires the predictive models to be differentiable and faithful to the actual chemical attributes of the materials. Chemically-aware generation of counterfactuals and real-world validation of the generated counterfactuals has not been studied in this work, and are interesting directions for future research. While the biopolymer nanocomposite dataset provides a real-world benchmark for FINDER, some of the sub-datasets are smaller in size. Evaluating FINDER on multiple larger real datasets remains an important future work.

Conclusion: This work pushes the boundary of inverse design by addressing the inherent challenges of multi-property optimization, where a perfect solution satisfying all criteria often does not exist. By providing a flexible approach based on entropic risk, FINDER allows for a controlled trade-off between computational search time and the number of properties satisfied. Furthermore, noting that simply adjusting ratios may fail to find viable candidates, FINDER significantly expands the search space by also exploring new functional groups, enabling the discovery of a broader range of realistic and synthetically achievable molecules.

REFERENCES

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015. doi: 10.1561/2200000050.

N Chakraborti. Genetic algorithms in materials design and processing. *International Materials Reviews*, 49(3-4): 246–260, 2004.

Tianle Chen, Zhenqian Pang, Shuaiming He, Yang Li, Snehi Shrestha, Joshua M Little, Haochen Yang, Tsai-Chun Chung, Jiayue Sun, Hayden Christopher Whitley, et al. Machine intelligence-accelerated discovery of all-natural plastic substitutes. *Nature Nanotechnology*, 19(6):782–791, 2024.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197, April 2002.

Hans Föllmer and Alexander Schied. Convex Measures of Risk and Trading Constraints. *Finance and stochastics*, 6 (4):429–447, 2002.

Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

Glentham Life Sciences. Sustainability at glentham life sciences. `https://www.glentham.com/en/news/article/63/`, 2022. Accessed: 2026-02-01.

Xiao-Qi Han, Xin-De Wang, Meng-Yuan Xu, Zhen Feng, Bo-Wen Yao, Peng-Jie Guo, Ze-Feng Gao, and Zhong-Yi Lu. AI-driven inverse design of materials: Past, present, and future. *Chinese Physics Letters*, 42(2):027403, 2025.

Trevor Hastings, Mrinalini Mulukutla, Danial Khatamsaz, Daniel Salas, Wenle Xu, Daniel Lewis, Nicole Person, Matthew Skokan, Braden Miller, James Paramore, et al. Accelerated multi-objective alloy discovery through efficient bayesian methods: application to the fcc high entropy alloy space. *Acta Materialia*, pp. 121173, 2025.

Michael Z. Hauschild and Anders Bjørn. Pathways to sustainable plastics. *Nature Sustainability*, 6(5):487, 2023.

Matthew K Horton, Patrick Huck, Ruo Xi Yang, Jason M Munro, Shyam Dwaraknath, Alex M Ganose, Ryan S Kingsbury, Mingjian Wen, Jimmy X Shen, Tyler S Mathis, et al. Accelerated data-driven materials science with the materials project. *Nature Materials*, pp. 1–11, 2025.

Carmen P Jiménez-Gómez and Juan Antonio Cecilia. Chitosan: a natural biopolymer with a wide and varied range of applications. *Molecules*, 25(17):3981, 2020.

Ronald Machaka. Machine learning-based prediction of phases in high-entropy alloys. *Computational Materials Science*, 188:110244, 2021.

AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. *Chemical science*, 12(20):7079–7090, 2021.

Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014. URL `https://github.com/bayesian-optimization/BayesianOptimization`.

Erfaun Noorani, Pasan Dissanayake, Faisal Hamman, and Sanghamitra Dutta. Counterfactual explanations for model ensembles using entropic risk measures. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1566–1575, 2025.

C. G. Otoni, H. M. Azeredo, B. D. Mattos, M. Beaumont, D. S. Correa, and O. Rojas. The food–materials nexus: Next generation bioplastics and advanced materials from agri-food residues. *Advanced Materials*, 33(43), 2021.

J. Qian, Q. Dong, K. Chun, D. Zhu, X. Zhang, Y. Mao, J. N. Culver, S. Tai, J. R. German, D. P. Dean, J. T. Miller, L. Wang, T. Wu, T. Li, A. H. Brozena, R. M. Briber, D. K. Milton, W. E. Bentley, and L. Hu. Highly stable, antiviral, antibacterial cotton textiles via molecular engineering. *Nature Nanotechnology*, 18(2):168, 2023.

Z. Rao, P.-Y. Tung, R. Xie, Y. Wei, H. Zhang, A. Ferrari, T. Klaver, F. Körmann, P. T. Sukumar, A. Kwiatkowski da Silva, et al. Machine learning–enabled high-entropy alloy discovery. *Science*, 378(6615):78–85, 2022.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. ISSN 1549-9596. doi: 10.1021/ci100050t.

Taffee T. Tanimoto. An elementary mathematical theory of classification and prediction. Technical report, International Business Machines Corporation (IBM), New York, NY, 1958. Internal Report.

Jonas Teufel, Annika Leinweber, and Pascal Friederich. Improving counterfactual truthfulness for molecular property prediction through uncertainty quantification. In *World Conference on Explainable Artificial Intelligence*, pp. 317–339. Springer, 2025.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*, 56(12):1–42, 2024.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005. URL `https://doi.org/10.1021/ci00057a005`.

Geemi P Wellawatte, Aditi Seshadri, and Andrew D White. Model agnostic generation of counterfactual explanations for molecules. *Chemical science*, 13(13):3697–3705, 2022.

Lianping Wu and Teng Li. A machine learning interatomic potential for high entropy alloys. *Journal of the Mechanics and Physics of Solids*, 187:105639, 2024.

M. Zhang, G. M. Biesold, W. Choi, J. Yu, Y. Deng, C. Silvestre, and Z. Lin. Recent advances in polymers and polymer composites for food packaging. *Materials Today*, 53, 2022.

SUPPLEMENTARY MATERIALS

A ADDITIONAL DETAILS ON OUR THEORETICAL RESULTS

Entropic risk has its roots in mathematical finance (Föllmer & Schied, 2002; Noorani et al., 2025).

To make sure counterfactual explanations are useful and actionable to the users, we not only need them to be close but also require them to stay valid under a reasonable fraction of the models within an ensemble. In general, it might even be impossible to guarantee the existence of a counterfactual that stays valid for all possible models in an ensemble. However, one might be able to ensure acceptance for a subset of models. This generates a need for an adjustable knob to obtain counterfactuals that accommodate varying fraction of properties.

Theorem 2.2. *[Guarantee on Properties Satisfied] Let x' be a counterfactual with $\hat{\rho}_\theta^{ent}(x') < \tau$ (a feasible solution for P3). Also, let $\mathbb{A}_\alpha(x')$ be the set of property-predicting models m_n which satisfy $m_n(x') > \alpha$, i.e., $\mathbb{A}_\alpha(x') = \{m_n | m_n(x') > \alpha, n = 1, \ldots, N\}$. Then, the fraction of properties satisfied, i.e., $\frac{|\mathbb{A}_\alpha(x')|}{N}$ is at least $\frac{1-e^{\theta(\tau+\alpha-1)}}{1-e^{-\theta(1-\alpha)}}$.*

Proof of Theorem 2.2. For brevity, we let $\eta_\alpha(x')$ denote the fraction of properties satisfied, i.e., $\frac{|\mathbb{A}_\alpha(x')|}{N}$.

From the upper bound on the entropic risk,

$$\hat{\rho}_\theta^{\text{ent}}(x') < \tau \implies \frac{1}{\theta} \log \left(\frac{1}{N} \sum_{n=1}^{N} e^{\theta\left(1-m_n(x')\right)} \right) < \tau \tag{2}$$

$$\implies \sum_{n=1}^{N} e^{\theta(1-m_n(x'))} < N e^{\theta\tau}. \tag{3}$$

Note that if $m_n(x') \leq \alpha$ then $e^{\theta(1-m_n(x'))} \geq e^{\theta(1-\alpha)}$ and therefore,

$$\hat{\rho}_\theta^{\text{ent}}(x') < \tau \implies \sum_{m_n \in \mathbb{A}_\alpha(x')} \underbrace{e^{\theta(1-m_n(x'))}}_{\geq e^{\theta(1-1)}=1} + \sum_{m_n \notin \mathbb{A}_\alpha(x')} \underbrace{e^{\theta(1-m_n(x'))}}_{\geq e^{\theta(1-\alpha)}} < N e^{\theta\tau} \tag{4}$$

$$\implies \sum_{m_n \in \mathbb{A}_\alpha(x')} 1 + \sum_{m_n \notin \mathbb{A}_\alpha(x')} e^{\theta(1-\alpha)} < N e^{\theta\tau} \tag{5}$$

$$\implies N\eta_\alpha(x') + N(1-\eta_\alpha(x'))e^{\theta(1-\alpha)} < N e^{\theta\tau} \tag{6}$$

$$\implies \eta_\alpha(x') > \frac{1-e^{\theta(\tau+\alpha-1)}}{1-e^{-\theta(1-\alpha)}}.$$

□

Significance. The significance of our theoretical guarantee is that it demonstrates that any feasible solution to our proposed optimization will always satisfy a fraction of properties. This enables us to adjust the hyperparameters τ and θ based on our requirements, so that we can tailor the algorithm to satisfy the desired number of properties.

B ADDITIONAL DETAILS ON EXPERIMENTAL SETUP 1: HEA DATASET

Dataset: The High-Entropy Alloys (HEA) dataset (Machaka, 2021) consists of 1,460 microstructural observations from 418 peer-reviewed studies to aid in the design of future High-Entropy Alloys (HEAs). They cover a vast compositional space and a broad range of microstructures. The dataset comprises features including the proportions of 25 elements in a metal alloy, along with their metallurgy-specific predictor features. Out of these, we consider the set of proportions of each element in the alloy as the input to a machine learning model. Seven properties (Density_calc, dSmix, dGmix, Tm, Phases, VEC, and Elect_Diff.) selected out of the remaining features are considered as the targets. These seven targets were selected in order to achieve a minimum number of missing values. In order to convert the numerical properties into classification labels, we threshold them at their median value. Multi-class labels were converted to binary by grouping the classes into two groups. Rows with missing values are removed. Consequently,

the final dataset comprises 1355 instances with 25 input features and 7 targets, which is split into train, validation and test sets with sizes 50%, 25%, and 25%, respectively.

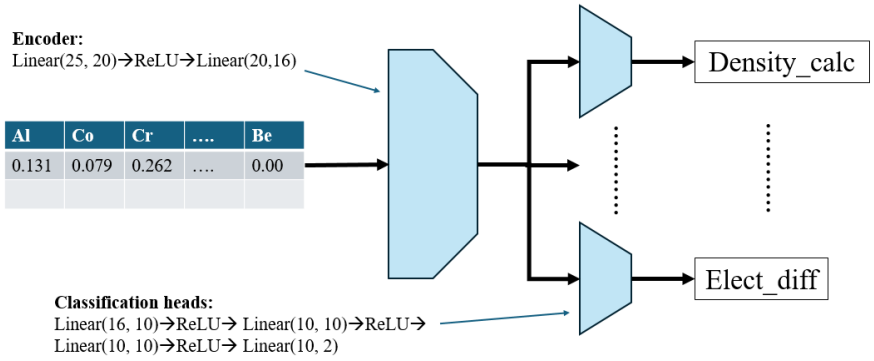


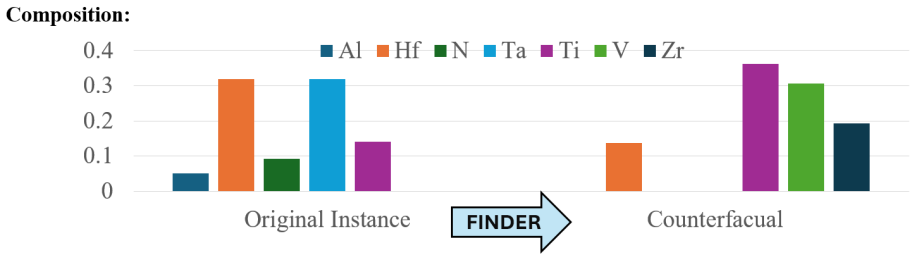
Figure 2: Model architecture used with the HEA dataset.

Model: A neural network consisting of a shared encoder and seven binary classification heads (one per each target) is used for prediction. The input is normalized so that the effect of proportions will depend only on the relative magnitudes. Fig. 2 illustrates the model architecture. The loss for the multi-target classification was computed as

$$\mathcal{L}(\eta) = \sum_{i=1}^{7} \mathcal{L}_{\text{CE}}\left(\hat{y}_i(\eta), y_i\right) \tag{7}$$

where η denotes the parameters of the model, $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, $\hat{y}_i(\eta)$ is the predicted class probability, and y_i is the true class for the i^{th} target property. Adam optimizer was used for training with a learning rate of 0.001 and a batch size of 32. The model was trained for 100 epochs and the model with the best validation loss was selected.

Counterfactual generation using FINDER: The inverse design objective is to identify minimal changes in alloy compositions that can satisfy the seven target properties (i.e., achieve a prediction of 1 for all the seven properties). We iteratively solve P3 using projected gradient descent to ensure all the proportions stay within the range $[0, 1]$.



Prediction:

Density_calc	dSmix	dGmix	Tm	Phases	VEC	Elect_Diff
0	0	1	1	0	0	1

Density_calc	dSmix	dGmix	Tm	Phases	VEC	Elect_Diff
0	1	1	1	1	1	1

Figure 3: An example counterfactual generation for the HEA dataset.

Baselines: We compare the average counterfactual generating time of FINDER with four baselines: Scalarized MO, Genetic, NSGA-II, and Bayesian. Below, we explain each method in detail.

1. Scalarized MO is the base counterfactual generating method $\mathcal{C}_p(x, \mathcal{M})$ used in FINDER (see Algorithm 1). The method involves using projected gradient descent to minimize the following loss function:

$$\mathcal{L}(x') = \lambda_{\text{prop}} \sum_{i=1}^{n} \mathcal{L}_{\text{CE}}(m_i(x'), 1) + \lambda_{\text{dist}} \|x - x'\|_p. \tag{8}$$

Here, $\mathcal{L}_{\text{CE}}(\cdot, 1)$ is the ordinary cross entropy loss with the true label set to 1 in order to achieve the target properties. $\|\cdot\|_p$ is the vector $p-$norm, and x denotes the original instance. $\lambda_{\text{prop}}, \lambda_{\text{dist}} \in \mathbb{R}_0^+$ balance the trade-off between the distance and the number of target properties achieved. We use gradient descent to minimize $\mathcal{L}(x')$ over x'.

2. Genetic method uses the basic genetic algorithm `eaSimple` offered by the `DEAP` library (Fortin et al., 2012), which maximizes the following objective:

$$\mathcal{J}(x') = \lambda_{\text{prop}} \sum_{i=1}^{n} \mathbb{1}[y_i(x') = 1] - \lambda_{\text{dist}} \|x - x'\|_p \tag{9}$$

where $y_i(\cdot)$ is the predicted label from model m_i and $\mathbb{1}[\cdot]$ is the indicator function.

3. NSGA-II baseline uses the Non-dominated Sorting Genetic Algorithm-II (Deb et al., 2002) to maximize the same objective function equation 9 as the Genetic method.

4. Bayesian method uses Bayesian optimization (Nogueira, 2014) to maximize the same objective function as NSGA-II and Genetic methods.

Hyperparameters of the Genetic and the Bayesian methods were optimized using the `optuna` library, with the hyperparameter ranges given in Table 5. The objective was the number of target properties achieved, computed for a subset of the original instances in the test set.

Table 5: Hyperparameter optimization details of the baselines.

Method	**Hyperparameter range**	**Trials**
Genetic	Population size $\in [5, 50]$ Number of generations $\in [5, 50]$ Cross-over probability $\in [0.1, 0.7]$ Mutation probability $\in [0.1, 0.3]$ $\lambda_{\text{prop}} \in [5.0, 20.0]$ $\lambda_{\text{dist}} \in [0.5, 2.0]$	20 iterations with 140 instances
Bayesian	Number of initial points $\in [3, 20]$ Maximum number of attempts $\in [5, 10]$ $\lambda_{\text{prop}} \in [1.0, 30.0]$ $\lambda_{\text{dist}} \in [0.1, 5.0]$	20 iterations with 100 instances

Additional experimental results: Fig. 3 shows an example instance of counterfactual generation using the FINDER algorithm. The algorithm suggests a new composition which satisfies a significantly higher number of target properties when compared to the original instance. Table 6 shows the proportion of counterfactuals which achieves each target property. The proportions increase with increasing θ and decreasing τ.

Table 6: Proportion of counterfactuals that satisfy each target property (HEA dataset). The proportion of counterfactuals satisfying each property increases with higher θ and lower τ.

Target property	$\boldsymbol{\theta = 0.1}$		$\boldsymbol{\theta = 1}$		$\boldsymbol{\theta = 10}$	
	$\boldsymbol{\tau = 0.3}$	$\boldsymbol{\tau = 0.4}$	$\boldsymbol{\tau = 0.3}$	$\boldsymbol{\tau = 0.4}$	$\boldsymbol{\tau = 0.3}$	$\boldsymbol{\tau = 0.4}$
Density_calc	0.457	0.457	0.481	0.481	0.991	0.737
dSmix	0.991	0.991	0.991	0.991	0.991	0.991
dGmix	1.000	1.000	1.000	1.000	1.000	1.000
Tm	1.000	1.000	1.000	1.000	0.991	0.991
Phases	1.000	1.000	1.000	1.000	1.000	1.000
VEC	0.997	0.997	1.000	0.997	1.000	1.000
Elect_Diff.	1.000	1.000	1.000	1.000	1.000	1.000

C Additional Details on Experimental Setup 2: Biopolymers Dataset

Dataset: The dataset consists of 4 independently collected sub-datasets covering 8 target properties namely FlatnessUni, Detachability, TransVis, TransIR, TransUV, TensileStress, TensileStrain, and FireRR. The input features are the ratios of 23 components (see Fig. 4 and Table 8) in a nanocomposite mixture. For this particular experiment, we filter out the mixtures with only $N = 4$ components at a time. All the properties except FlatnessUni and Detachability are originally numerical. In order to convert them to binary labels, we threshold the values at the median. Accordingly, our goal is to find a nanocomposite which satisfies all the target properties, i.e., achieves a prediction of class 1 for all the targets. For the counterfactual generation, we consider nanocomposites with chitosan as a component for initial instances, leaving us with 27 initial candidates for counterfactual generation. The set of features allowed to be modified during the generation includes the ratios of all the components and the fingerprint of the chitosan. This way, the resultant counterfactual will always comprise a chitosan derivative and the remaining original component, with suitably modified ratios. See Fig. 5 for more details on the input format. Table 7 gives additional information about the sub-datasets. All datasets were split into train, validation and test sets with sizes 60%, 20% and 20%, respectively.

Table 7: Details of the sub-datasets used in the experimental setup 2.

Sub-dataset	Properties	Number of samples
grade	FlatnessUni, Detachability	990
optical	TransVis, TransIR, TransUV	171
tensile	TensileStress, TensileStrain	207
fire	FireRR	104

Low-Dimension Materials	Biopolymers			Additives	
Laponite (LAP)	Silk (SLK)	Sodium Alginate (ALG)	Furfural (FFA)	Glycerol (GLY)	Lactic Acid (LAC)
Montmorillonite (MMT)	Gelatin (GEL)	Pullulan (PUL)	Pectin (PEC)	Xylitol (XYL)	Levulinic Acid (LEV)
Sodium Carboxymethyl Cellulose (CMC)	Chitosan (CHS)	Carrageenan (CAR)	Zein (ZIN)	Sorbitol (SRB)	Succinic Acid (SUA)
Cellulose Nanofiber (CNF)	Agarose (AGR)	Starch (STA)	Gluten (GLU)	Phytic Acid (PHA)	

Figure 4: List of 23 naturally-occurring GRAS materials

Input Representation: To allow provision for altering both molecular structures and ratios, we first propose a representation based on molecular fingerprints of the constituents computed based on their SMILES strings (see Fig. 5). Accordingly, each nanocomposite mixture $x \in \mathcal{X}$ is represented by an $N \times (d+1)$-dimensional vector comprising of $N = 4$ blocks of length $(d+1)$, with each block representing a constituent of the mixture, i.e., $x = [x_1, \ldots x_4]$. The fingerprint is computed using `rdkit` Morgan generator, with a length of $d = 512$ and a radius of 2. A given block consists of its molecular fingerprint and the ratio in the mixture, i.e., $x_i = [p_{i,1}, \ldots p_{i,d}, r_i]$ where $p_{i,j}, j = 1, \ldots, d$ are the elements of the fingerprint and r_i is the ratio of the i^{th} component. When generating counterfactuals, component 1 is always fixed to be chitosan.

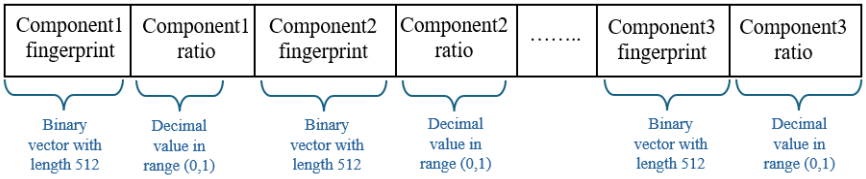


Figure 5: Input representation. Input vector consists of $N = 4$ blocks, each with length $d + 1$ where $d = 512$. Each block consists of the molecular fingerprint of the corresponding component and its ratio in the nanocomposite. When generating counterfactuals, component 1 is always fixed to be chitosan.

Model: The model comprises a shared encoder and multiple classification heads, one per each target property. Both components are trained end-to-end on all the 4 datasets, with sample mixing. As the first step, the encoder generates

Table 8: Library of 23 components used for preparing the lab-generated dataset of biopolymer nanocomposites, along with their SMILES representations.

Material	Abbr.	SMILES string
Agarose	AGR	C1C2C(C(O1) C(C(O2)O)O)OC3C(C(C(C(O3)CO)O) OC4C(C5C(C(O4)CO5)OC6C(C(C(C(O6)CO) O)O)O)O)O
Sodium Alginate	ALG	C1(C(C(OC(C1O)O)C(=O)[O-])O)O.[Na+]
Carboxymethyl Cellulose	CMC	CC(=O)[O-].C(C(C(C(C(C=O)O)O)O)O)O.[Na+]
Carrageenan	CAR	CCC(=O)C1=C(C(=C(C=C1)F)C2CC2NC(=O)NC3=NC=C(C=C3)C#N)O.CC(=O)[O-].CC(=O)[O-].[Zn+2]
Cellulose Nanofiber	CNF	C(C1[C@H](C(C(C(O1)O)O)O)O[C@H]2C(C(C(O2)CO)O)O)O)O
Chitosan	CHS	COC(=O)N[C@@H]1[C@H]([C@@H]([C@H](O[C@H]1O[C@@H]2[C@H](O[C@H]([C@@H] ([C@H]2O)N)O[C@@H]3[C@H](O[C@H]([C@@H] ([C@H]3O)N)O)CO)CO)CO)O[C@H]4[C@@H]([C@H] ([C@@H]([C@H](O4)CO)O[C@H]5[C@@H]([C@H] ([C@@H]([C@H](O5)CO)O[C@H]6[C@@H]([C@H]([C@@H] ([C@H](O6)CO)O[C@H]7[C@@H] ([C@H]([C@@H]([C@H](O7)CO)O[C@H]8[C@@H] ([C@H]([C@@H]([C@H](O8)CO)O[C@H]9[C@@H] ([C@H]([C@@H]([C@H](O9)CO)O)O)N)O)N)O)N)O)N)O)N)O)N)O
Furfural	FFA	C1=COC(=C1)C=O
Gelatin	GEL	[H][C@@]1(C[C@@](C)(OC)[C@@H](O) [C@H](C)O1)O[C@H]1[C@H](C)[C@@H](O[C@]2([H])O[C@H](C)C[C@@H] ([C@H]2O)N(C)C)[C@](C)(O)C[C@@H](C)C(=O)[C@H](C) [C@@H](O)[C@](C)(O)[C@@H](CC)OC(=O)[C@@H]1C
Gluten	GLU	CC(C)CC(C(=O)O)NC(=O)C(CC1=CNC2=CC=CC=C21)NC(=O)CNC(=O)CNC(=O)C(CC3=CC=C(C=C3)O)N
Glycerol	GLY	C(C(CO)O)O
Lactic Acid	LAC	CC(C(=O)O)O
Laponite	LAP	[Li+].[Li+].[OH-].[OH-].[OH-].[OH-].[OH-].[OH-].[OH-].[OH-].[OH-].[OH-].[OH-].[OH-].[O-][Si]12O[Si]3(O[Si](O1)(O[Si](O2)(O3)[O-])[O-])[O-].[O-][Si]12O[Si]3(O[Si](O1)(O[Si](O2)(O3)[O-])[O-])[O-].[O-][Si]12O[Si]3(O[Si](O1)(O[Si](O2)(O3)[O-])[O-])[O-].[O-][Si]12O[Si]3(O[Si](O1)(O[Si](O2)(O3)[O-])[O-])[O-].[O-][Si]12O[Si]3(O[Si](O1)(O[Si](O2)(O3)[O-])[O-])[O-]. [Na+].[Na+].[Mg+2].[Mg+2].[Mg+2].[Mg+2]. [Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2].[Mg+2]
Levulinic Acid	LEV	CC(=O)CCC(=O)O
Montmorillonite	MMT	O.[O-2].[O-2].[O-2].O=[Si]=O.O=[Si]=O.O=[Si]=O.O=[Si]=O.[Al+3].[Al+3]
Pectin	PEC	CC1CCOC(C1O)(C2CC3C(O2) C=CC (=CC(CC4(CCC(O4)C56CCC(O5) (CC(O6)C7C(=O)CC(O7)(C(C8CCC9(O8) CCCC(O9)C(C(=O)O3) C)O)C)C(=O)O)C)C)C)O
Phytic Acid	PHA	C1(C(C(C(C1OP(=O)(O)O)OP(=O)(O)O)OP(=O)(O)O)OP(=O)(O)O)OP(=O)(O)O)OP(=O)(O)O
Pullulan	PUL	[OC[C@H]1O[C@H](O[C@H]2O[C@H] (CO)[C@@H](O)[C@H](O)[C@H]2O [C@H]3O[C@H](CO)[C@@H](O) [C@H](O)[C@H]3O)[C@@H](O)[C@H](O)[C@H]1O]
Silk	SLK	[NC(=O)[C@H](CO)NC(=O)[C@H](C)NC(=O)CNC(=O)[C@H](C)N]
Sorbitol	SRB	C([C@H]([C@H]([C@@H]([C@H](CO)O)O)O)O)O
Starch	STA	C([C@H]1[C@H]([C@@H]([C@H](O1)O[C@H]2[C@H]([C@@H]([C@H](O2)O)O)O)O)O)O)*
Succinic Acid	SUA	C(CC(=O)O)C(=O)O
Xylitol	XYL	OC[C@@H](O)[C@H](O)[C@@H](O)CO
Zein	ZIN	[NH]C(=O)[C@H]C

a latent representation for each component of the nanocomposite by acting upon each block of the input vector. Next, these latent representations are aggregated by summing them together, resulting in a permutation-invariant (in terms of the order of the components in the mixture) intermediate representation of the nanocomposite. Finally, the classification head generates the binary prediction based on this latent representation. See Fig. 6 for more details on the architecture. The model was trained for 100 epochs with a learning rate of 0.001, a weight decay of 0.0001, a dropout probability of 0.5, and a batch size of 32. Best epoch in terms of average AUC-ROC over the validation set was used as the final trained model.

Counterfactual generation using FINDER: A realistic counterfactual corresponding to a given initial instance is found in three steps. First, an initial raw counterfactual is found using the entropic-risk-based counterfactual generating algorithm. This is done by allowing the chitosan fingerprint and the ratios of all the components to vary. The ranges for the values are restricted to $[0, 1]$ as it suits both the elements of the fingerprint and the ratios. While the ratios are guaranteed to be realistic, the resulting counterfactual fingerprint can now contain decimal-valued elements. In order to convert them to binary values, we next round off the elements to the nearest integer (0 or 1 since the values are within [0, 1]), giving a rounded-off counterfactual. Finally, to ensure the fingerprint corresponds to a realistic and a synthesizable chitosan derivative, it is replaced with the fingerprint of the most similar chitosan derivative in terms of the Tanimoto similarity (Tanimoto, 1958) selected from a library of chitosan derivatives. Algorithm 2 summarizes the above steps.

Additional experimental results: Table 6 presents the per-target achievement rates averaged over the generated counterfactuals. While FireRR remains to be the most challenging target property to achieve, it can be observed that increasing θ systematically improves the proportions.

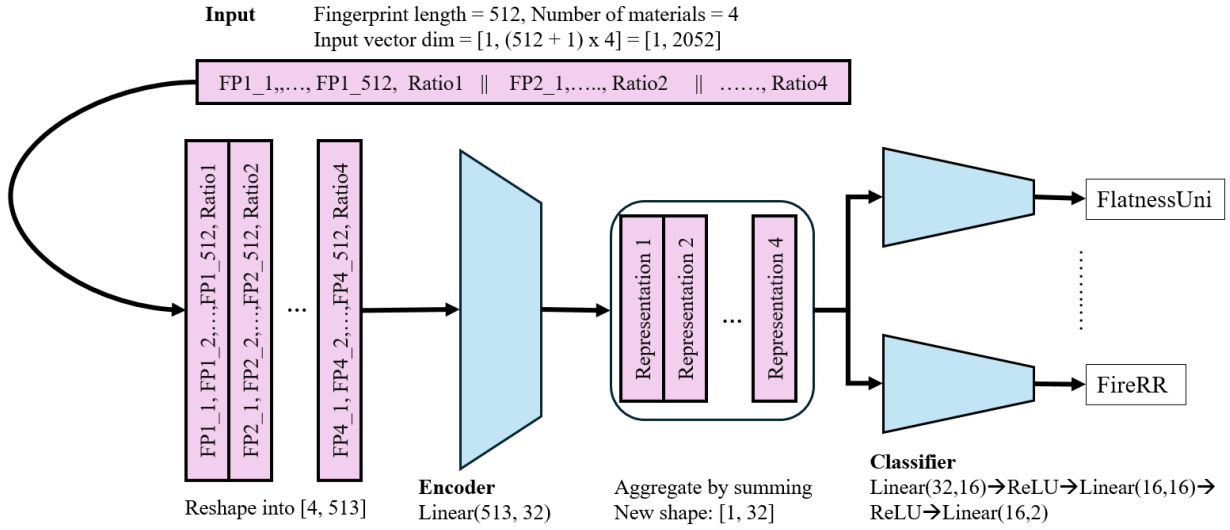


Figure 6: Model architecture for predicting properties of biopolymer nanocomposites. A shared encoder is used with multiple classification heads, one per each target property.

Algorithm 2: Utilizing FINDER for novel chitosan-based biopolymer nanocomposites design

Input: Initial nanocomposite mixture x, machine learning model m, hyperparameters θ and τ, chitosan derivative library $\mathcal{C}$, fingerprint generator g
Output: A realistic, synthesizable entropic-risk-based counterfactual x'
STEP 1: Find a raw counterfactual
$x'_{\text{raw}} \leftarrow \text{FINDER}\big(x, m, \theta, \tau, \text{features to vary} = \{p_{1,1}, \ldots, p_{1,d}, r_1, \ldots, r_N\} \text{ within range } [0, 1]\big)$
STEP 2: Round off fingerprint elements
$x'_{\text{round}} \leftarrow x'_{\text{raw}}$
for $i \leftarrow 1$ **to** d **do**
  $x'_{\text{round}}[i] \leftarrow \texttt{round}(x'_{\text{round}})[i]$
end
STEP 3: Substitute nearest chitosan derivative
max_similarity $\leftarrow 0$
$x' \leftarrow x'_{\text{round}}$
for *chitosan derivative* $c \in \mathcal{C}$ **do**
  similarity $\leftarrow \texttt{Tanimoto}\,(x'_{\text{round}}[1:d], g(c))$
  if *similarity* $>$ *max_similarity* **then**
    max_similarity $\leftarrow$ similarity
    $x'[1:d] \leftarrow g(c)$
  end
end
return x' ;

Table 9: Proportion of counterfactuals that satisfy each target property (Biopolymers dataset). The proportion of counterfactuals satisfying each property increases with higher θ and lower τ.

Target property	$\boldsymbol{\theta = 0.1}$		$\boldsymbol{\theta = 0.3}$		$\boldsymbol{\theta = 0.5}$		$\boldsymbol{\theta = 1.0}$		$\boldsymbol{\theta = 10}$	
	$\boldsymbol{\tau = 0.48}$	$\boldsymbol{\tau = 0.4}$	$\boldsymbol{\tau = 0.48}$	$\boldsymbol{\tau = 0.4}$	$\boldsymbol{\tau = 0.48}$	$\boldsymbol{\tau = 0.4}$	$\boldsymbol{\tau = 0.48}$	$\boldsymbol{\tau = 0.4}$	$\boldsymbol{\tau = 0.48}$	$\boldsymbol{\tau = 0.4}$
Detachability	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
FireRR	0.074	0.167	0.077	0.118	0.077	0.125	0.115	0.133	0.158	0.286
FlatnessUni	0.741	0.944	0.769	0.941	0.769	0.938	0.769	1.000	1.000	1.000
TensileStrain	0.963	0.944	0.962	1.000	0.962	1.000	0.923	1.000	0.895	1.000
TensileStress	0.519	0.833	0.538	0.824	0.538	0.812	0.538	0.867	0.842	1.000
TransIR	0.926	0.889	0.923	0.941	0.885	0.938	0.885	0.933	0.737	1.000
TransUV	0.593	0.444	0.577	0.471	0.538	0.438	0.538	0.467	0.211	0.571
TransVis	0.889	0.833	0.885	0.882	0.885	0.875	0.808	0.933	0.421	1.000