Supplemental Information for

Leveraging Feature Gradient for Efficient Acquisition Function Maximization in Material Composition Design

Yuehui Xian, Yunfan Wang, Pengfei Dang, Xinquan Wan, Yumei Zhou*, Xiangdong Ding*, Jun Sun, Dezhen Xue*

State Key Laboratory for Mechanical Behavior of Materials, Xi'an Jiaotong University, Xi'an 710049, China

This PDF file includes 8 sections with Figures S1 to S5 and Tables S1 to S8.

Section 1. Clarification of the compositional design space

In multi-component alloy composition design, discrete step sizes (e.g., 1%, 0.2%, 0.1%) are typically used to adjust the compositional values of different elements, subject to the constraint that the sum of all components equals 100%. All possible composition points satisfying both "non-negativity" and "normalization" (sum to 100%) constraints form a simplex space. For an n-component system, the composition space constitutes an (n-1)-dimensional simplex. When additional constraints are applied to this simplex space based on domain knowledge, such as requiring certain elements to have compositions greater than 10%, the feasible solutions form a subset within a simplex space, with all valid alloy compositions lying either within, or on the vertices/edges of this polytope.

To illustrate the rapid growth of the number of possible compositions with respect to the number of components, let us consider a simplified scenario where: a) Each element's composition uses a step size of 1%; b) Compositions range within [0, 100%]. Under these conditions, each component can take discrete values of 0%, 1%, 2%, ..., 100%. The problem of finding all possible discrete composition points can be transformed into a combinatorial mathematics problem: finding the number of non-negative integer solutions to the equation:

 $x_1 + x_2 + \ldots + x_n = 100\%$,

where x_n represents the composition value of the n-th element. This can be solved using the "stars and bars" theorem, yielding:

$$C_{100+n-1}^{n-1} = \frac{(n+99)!}{100! \cdot (n-1)!},$$

$$\frac{(n+99)!}{(n-1)!} = n \cdot (n+1) \cdot \dots \cdot (n+99) \approx n^{100}, \text{ when } n \to +\infty.$$

In practical multi-component alloy design, since elements typically have non-zero lower bounds (e.g., based on domain knowledge) for their compositions, the feasible compositions form a convex polytope subset. If with smaller compositional step sizes in optimization, this will lead to a rapid polynomial growth with respect to the number of possible compositions. Therefore, exhaustively enumerating all possible compositions (essentially a brute force approach) becomes computationally prohibitive for high-dimensional discrete compositional spaces. While various optimization techniques exist for continuous problems, the discrete nature and constraints of alloy composition design present unique challenges. For these specific challenges, implementing gradient-based methods for compositions. This rapid polynomial growth demonstrates why efficient optimization strategies are crucial for exploring high-dimensional compositional spaces.

It is worth noting that the choice of 0.1% as the compositional step size aligns with practices in materials design literatures [1-4]. This precision is particularly justified in cases like shape memory alloys (SMAs), where minor variations (0.1%) in the composition of key elements (e.g., Ti, Ni) can significantly affect phase transformation behaviors [5]. In fact, we do not always have a priori knowledge on how sensitive material properties are to compositional changes across different concentration ranges, making this fine-grained exploration necessary, although such precision may not be required for all cases.

Section 2. Formal formulation of the inner loop

Within the BO framework, the experimental optimization of materials properties involves solving the following inner optimization [6] problem: $argmax_{c \in C} \alpha(c)$,

where $\alpha(\cdot)$ represents the acquisition function, which here is calculated based on the mean and uncertainty predictions from the surrogate model, but could be purely focused on exploitation or exploration as well. The composition vector *c* needs to satisfy the following constraints:

$$C = \left\{ \langle x_1, x_2, \dots, x_n \rangle \mid x_i \in D_i, \sum_{1}^{n} x_i = 100\% \right\}.$$

Here, $\langle x_1, x_2, ..., x_n \rangle$ represents the alloy composition vector, where D_i denotes the set of possible

discrete compositional values for the i-th element. These formulations capture both the optimization objective and the constraints in alloy design: each element's composition must fall within its allowed range $x_i \in D_i$, and the sum of all compositional values must equal 100% (normalization constraint).

The acquisition function $\alpha(\cdot)$ plays a crucial role in balancing exploration and exploitation during the optimization process. It utilizes both the predicted mean and the associated uncertainty from the surrogate model to guide the search for optimal compositions. This inner loop optimization problem, while mathematically well-defined, presents significant computational challenges due to the high-dimensional compositional space and the multi-modal landscape of the acquisition function, which is particularly the case we consider in this work.

Section 3. Formulations of GPR, $\alpha(\cdot)$, feature transformations and the numerical estimation of AF gradients with respect to *c*

Gaussian Process Regression (GPR)

Given a set of training data X with corresponding target values y under noise-free conditions, GPR enables us to predict both the mean (μ) and standard deviation (σ) at any test point x. We denote the GPR as $g(\cdot)$. The mathematical formulation is as follows:

$$\mu = K(X, x)^{T} K(X, X)^{-1} y,$$

$$\sigma^{2} = K(x, x) - K(X, x)^{T} K(X, X)^{-1} K(X, x),$$

where K(X, x) represents the kernel matrix between training data X and the test point x, K(X, X) represents the kernel matrix among training points and K(x, x) represents the self-covariance at the test point x.

For measuring the similarity between compositional points, we employ the Matérn 5/2 kernel function:

$$k(x, x') = \sigma_f^{2} \left(1 + \frac{\sqrt{5r}}{l} + \frac{5r^2}{3l^2} \right) \exp\left(-\frac{\sqrt{5r}}{l}\right),$$

where r = ||x - x'|| is the Euclidean distance between input points x and x', σ_f^2 is the signal variance

and l is the length scale parameter. The training process of GPR involves optimizing the kernel hyperparameters (e.g., σ_f and l) by maximizing the marginal likelihood, ensuring the model best explains the observed data. This kernel choice ensures differentiability with respect to the input x except at the point where x equals x', which is crucial for our gradient-based optimization approach.

Acquisition Functions

For the acquisition functions, we consider three commonly used ones: Expected Improvement (EI), Probability of Improvement (POI), and Upper Confidence Bound (UCB), which are defined as follows:

$$Z = \frac{(\mu - \mu^{+})}{\sigma},$$

$$\alpha_{EI} = \begin{cases} (\mu - \mu^{+}) \Phi(Z) + \sigma \phi(Z), & if \sigma > 0 \\ 0, & if \sigma = 0 \end{cases}$$

$$\alpha_{POI} = \begin{cases} \Phi(Z), & if \sigma > 0 \\ 0, & if \sigma = 0 \end{cases},$$

$$\alpha_{UCB} = \mu + \kappa \sigma,$$

where μ^+ is the best experimental observation within *X*, $\Phi(\cdot)$ is the normal cumulative distribution function, $\phi(\cdot)$ is the normal probability density function and κ is a parameter for tuning the degree of exploration.

These considered acquisition functions maintain differentiability with respect to input, exhibiting piecewise differentiability except at points where $\sigma(\cdot) = 0$. However, this piecewise differentiability does not pose significant challenges for current state numerical optimization methods.

Feature Transformations

Table S1 Mathematical formulas for transforming elemental properties into material features

$$E = \sum_{i=1}^{n} c_i \cdot E_i$$

$$\delta E = \sqrt{\sum_{i=1}^{n} c_i \cdot (1 - E_i/E)}$$

$$MAX.E = \max(E_i), \text{ for } c_i \neq 0$$

$$MIN.E = \min(E_i), \text{ for } c_i \neq 0$$

$$Range.E = MAX.E - MIN.E$$

$$MAXC.E = c_j \cdot E_j, \text{ where } c_j = \arg\max_{c_{i'}, c_i \neq 0} E_i$$

$$MINC.E = c_j \cdot E_j, \text{ where } c_j = \arg\min_{c_{i'}, c_i \neq 0} E_i$$

 $RangeC.E = c_j \cdot E_j - c_k \cdot E_k, where \ c_j = argmax_{c_i, c_i \neq 0}E_i \ and \ c_k = argmin_{c_i, c_i \neq 0}E_i$

Table S1 presents the mathematical formulas used to transform elemental properties into material features [7]. Here, c_i represents the molar fraction of element *i*, E_i denotes the corresponding elemental property value, and *n* is the total number of elements in the composition. These transformations enable the conversion of discrete elemental properties into continuous material features that can effectively capture composition-property relationships. To enable gradient-based optimization, we implemented differentiable versions of the mathematical formulas shown in Table S1. A key challenge was to make the *argmax/argmin* operations differentiable. The traditional argmax operation returns an index of the maximum value, which is non-differentiable. We addressed this by implementing a "soft" version that produces a probability distribution over indices using the *softmax* function:

$w = softmax(\tau \cdot x)$,

where τ is a temperature parameter (set to 1×10^6 in our implementation) that controls the sharpness of the approximation, and *w* represents the weight distribution across all positions. As τ approaches infinity, the weights *w* approach a one-hot vector with 1 at the position of the maximum value, effectively approximating the behavior of *argmax* while maintaining differentiability. For numerical stability, we also incorporated a small position-dependent epsilon sequence (decreasing from 1×10^{-4} to 0) to break potential ties in the input values.

The soft *argmin* operation is implemented analogously using *softmax*(-*x*). These differentiable operations are particularly crucial for *MAXC.E*, *MINC.E*, and *RangeC.E* calculations, where the gradient information needs to be properly propagated through the selection of maximum and minimum values. Additionally, we applied a *sigmoid* function with the same temperature parameter to smooth the compositional masks used for filtering zero-concentration elements:

 $mask = sigmoid(\tau \cdot (c - \varepsilon))$,

where *c* represents the element-wise compositions and ε is a small threshold (1×10⁻⁵). This comprehensive differentiable implementation ensures stable gradient flow through the entire optimization process while closely approximating the original mathematical formulations.

Numerical Gradient Estimation of Acquisition Functions

We only provide a glimpse here. The foundation of this numerical gradient estimation approach lies in the fundamental properties of GPR. GPR can be understood as a weighted sum of basis functions in an infinite-dimensional function space, which is guaranteed by Mercer's theorem. To numerically estimate the gradient value of acquisition functions, we first utilize the following formulation [8]:

$$\alpha(x;\Theta,D) = E[u(g(x),\Theta)|D] \cong \hat{\alpha}_{N_f}(x;\Theta,D) = \frac{1}{N_f} \sum_{i=1}^{N_f} u(\xi_D^i(x),\Theta),$$

where $\xi_D^i(x)$ represents function samples drawn from the posterior distribution $g_D(\cdot)$, *D* represents the training dataset (*i.e.*, the experimented materials), Θ are parameters independent of *x* (*e.g.*, κ in UCB) and $u(\cdot)$ is a utility function that defines the acquisition function $\alpha(\cdot)$. In the case of EI, the $u(\cdot)$ refers to $max(g(x) - \mu^+, 0)$. The gradient estimation process follows these steps:

a) Starting with the basic form of gradient estimation:

$$\nabla_{x} \hat{\alpha}_{N_{f}}(x;\Theta,D) = \nabla_{x} (\frac{1}{N_{f}} \sum_{i=1}^{N_{f}} u(\xi_{D}^{i}(x),\Theta))$$

b) Applying the chain rule yields:

$$\nabla_{x} \hat{\alpha}_{N_{f}}(x;\Theta,D) = \frac{1}{N_{f}} \sum_{i=1}^{N_{f}} \nabla_{\xi} u(\xi_{D}^{i}(x),\Theta) \cdot \nabla_{x} \xi_{D}^{i}(x)$$

c) By introducing $h_D(x, \epsilon)$ as the auxiliary variable related to the posterior distribution (*i.e.*, the reparameterization trick), we can express:

$$\nabla_{x}u(\xi_{D}^{i}(x),\Theta) = \nabla_{\xi}u(\xi_{D}^{i}(x),\Theta) \cdot \nabla_{x}h_{D}(x,\epsilon)$$

This ensures that $\xi \sim g_D(\cdot)$ is transformed into a differentiable deterministic function $\xi = h_D(x, \epsilon)$, and stochasticity is only related with ϵ , not x.

d) The complete gradient expression becomes:

$$\nabla_{x} \hat{\alpha}_{N_{f}}(x;\Theta,D) = \frac{1}{N_{f}} \sum_{i=1}^{N_{f}} \nabla_{\xi} u(\xi_{D}^{i}(x),\Theta) \cdot \nabla_{x} h_{D}(x,\epsilon)$$

This derivation demonstrates how we can compute numerical gradients through the reparameterization of posterior samples, which is particularly effective when implemented within GPR using (function) samples drawn from a posterior distribution.

For subsequent use in gradient ascent/descent, we can leverage this gradient information to obtain

$$x^*$$
, specifically conditioned on *D*, facilitated by *N* drawn $\xi_D^{\iota}(x)$ samples:
 $x_N^* = argmax_{x=X} \hat{\alpha}_N(x;\Theta, D)$.

For the computational pipeline from materials features to GPR predictions and then to AF values, we utilized BoTorch [8], which is built on GPyTorch [9] and leveraged PyTorch [10] for forward computational graph and backward gradient propagation. For detailed implementation of this part, interested readers may refer to Balandat *et. al.* [8].

For the transformation from composition to materials features, we self-implemented this part and bridged it with the aforementioned framework, also leveraging PyTorch, to achieve stable gradient flow from AF back to compositions, i.e., the numerical gradient vector with respect to each composition variable. When utilizing this gradient vector in SLSQP optimization [11], special consideration must be given to constraints. In fact, our choice of SLSQP was partially motivated by its ability to handle linear constraints on inputs effectively.

We now summarize the whole computational pipeline as Table S2:

Table S2. Overview of the computational pipeline and its gradient computation. The pipeline is decomposed into three main transformations: from compositions to material features, surrogate predictions and acquisition function values. For each transformation, we specify the input-output relationship, function notation, based on which we implemented, and the corresponding gradient/Jacobian computation.

Function Name	Input \rightarrow Output	Function Notation	Implementation	Gradient Vector / Jacobian Matrix [†]
Feature Transformation Function	composition → material features	$\mathcal{E}(\cdot)$	self-implemented	$\nabla_c \varepsilon$
Gaussian Process Regression	material features \rightarrow surrogate predictions	$g(\cdot)$	BoTorch	$ abla_arepsilon g$
Acquisition Function	surrogate predictions \rightarrow acquisition function value	α(·)		$ abla_g lpha$

Note: [†]We use ∇ for both gradient vector and Jacobian matrix as it is a common notation abuse in machine learning. Although we use symbolic notation, we actually use numerical gradient values.

Section 4. Enumeration-based Bayesian optimization in compositional design



Figure S1. Illustration of the standard workflow of enumeration-based Bayesian Optimization for materials design.

 Table S3 Pseudo code of enumeration-based Bayesian Optimization for materials composition design

[1]	Given composition-property data D_{comp} and design space D_{space} . A composition c is
	transformed into materials features by combining elemental properties and mole fractions
	through a mathematical formula $\varepsilon(c)$, a surrogate model $g(\varepsilon(c))$ is built, and acquisition
	function $\alpha(g(\varepsilon(c)))$ values are calculated.
[2]	for $t \leftarrow 1,, T$, do // For all outer experimental iterations
[3]	Calculate materials features for D_{comp} using $\varepsilon(c)$.
[4]	Train a GP model using calculated materials features as input and alloy property
	values as output.
[5]	Build an acquisition model $f_{acq}(\cdot)$ using the trained GP. Input into the GP should be
	calculated materials features.
[6]	$D_{space} \leftarrow D_{space}$ - $c_{experimented}$
[7]	$c^* \leftarrow argmax_{c \in Dspace} f_{acq}(f_{elem}(c))$
[8]	Do experiment on c^* and add experimental results into D_{comp}
[9]	Return maximum property value in D_{comp} as best so far property.

Figure S1 presents a flowchart of the enumeration-based Bayesian Optimization (BO) framework for materials design. The process iteratively refines a surrogate model through statistical inference of experimental data, enabling predictions with uncertainty estimates across the search space. The acquisition function typically balances exploration-exploitation trade-offs by ranking candidates based on both predicted performance and prediction uncertainty. Table S3 details the algorithmic implementation, where a GPR model serves as the surrogate, mapping materials features $\varepsilon(c)$ into model predictions. The algorithm exhaustively explores the design space by selecting compositions that maximize the AF value, with experimental results continuously updating the surrogate model to drive this iterative design process.

It is emphasized that the enumeration-based compositional design process still dominates in many research studies on multicomponent alloy design.



Figure S2. Pair plots showing the distribution of element compositions for SMA alloys explored during optimization. Gray points represent training data, red points show compositions selected by the enumeration method, and blue points indicate compositions chosen by the feature gradient method. Diagonal plots display the kernel density estimation of compositional distributions for each element. This visualization demonstrates that the gradient optimization method (blue) explores a broader range of compositional space with more uniform distributions (evidenced by the flatter blue density curves), while the enumeration method (red) tends to produce more peaked distributions focused on limited compositional regions, consistent with the entropy analysis in Figure 5.

To further validate our observations regarding state entropy and compositional diversity, we examined the pairwise distributions of elemental compositions in the SMA system with 4 components (Figure S2). The pair plots reveal distinct exploration patterns between the two optimization strategies. The enumeration method (red points) exhibits more concentrated sampling in specific regions of the compositional space, with narrower distributions visible in the diagonal density plots. In contrast, the feature gradient method (blue points) demonstrates wider and more uniform exploration across the available compositional diversity directly corresponds to the higher state entropy values observed for the gradient optimization method in Figure 5.

Section 5. Examining materials features for compositional design

Table S4 Pseudo code for evaluating the effectiveness of materials features

versus raw compositions in predicting material properties. The algorithm compares the predictive performance of both input types through 10-fold cross-validation experiments and T-test analysis

[1]	Given composition-property data D_{comp} , mathematical formulas to calculate materials			
	features.			
[2]] Prepare metric buffers, $B_{composition}$ and $B_{feature}$.			
[3]	3] for $t \leftarrow 1,, T$, do // For one comparison experiment.			
[4]	Randomly choose 100 composition-property data (D_{temp}) from D_{comp} .			
[5]	Conduct 10-fold CV using compositions as input and alloy property values as output.			
	The resulting MAPE value is m_c .			
[6]	Push m_c into $B_{composition}$.			
[7]	Find mathematical formula(s) $\varepsilon(c)$ that minimizes 10-fold CV MAPE, using			
	calculated materials feature as input. The resulting MAPE value is m_{f} .			
[8]	Push m_f into $B_{feature}$.			
[9]	Conduct T-test on $B_{composition}$ and $B_{feature}$. // Statistical analysis.			

Table S4 presents the algorithmic implementation for verifying the effectiveness of materials features versus raw compositions as model input in predicting material properties. The comparison procedure consists of T independent experiments, each processing 100 randomly sampled composition-property pairs. For each experiment, two parallel evaluations are conducted: one using raw compositions as input (Lines 4-6) and another using calculated materials features (Lines 4, 7-8). Both evaluations employ 10-fold cross-validation with standard normalization to ensure fair comparison, yielding mean absolute percentage error (MAPE) values as performance metrics. These MAPE values are collected in separate buffers ($B_{composition}$ and $B_{feature}$) for subsequent statistical analysis using paired t-tests (Line 9). This evaluation procedure enables quantitative assessment of the predictive advantages offered by materials features over raw compositions. The statistical significance of performance differences (p < 0.01, indicating strong statistical evidence against the null hypothesis and suggesting that the observed differences are highly unlikely under the assumption of no true effect) across multiple independent trials provides evidence for using materials features as surrogate model inputs, despite their additional computational cost in the Bayesian Optimization process.



Figure S3. Comparative analysis of predictive performance between models using raw compositions (blue) and material features (red) as inputs for predicting martensitic transformation temperature (SMAs), Vickers hardness (Ti alloys), and yield strength (HEAs). Box plots show the distribution of mean absolute error (MAE) values from 10-fold cross-validation across 64 comparative experiments for each dataset. Statistical significance of the performance differences was confirmed through paired t-tests (pt-test < 0.01, meaning that there is less than 1% probability that the differences were due to chance).

We also evaluated model performance using mean absolute error (MAE) as shown in Figure S3. The calculation procedure followed the same experimental design described in Table S4, with the only modification being lines 5 and 7, where MAPE was replaced with MAE. As shown in Figure S3, models using material features consistently achieved lower MAE values compared to those using raw compositions across all three material systems (SMAs, Ti alloys, and HEAs). The statistical significance of these performance differences was confirmed through paired t-tests (p < 0.01), reinforcing the conclusion that materials features provide better predictive capabilities. It is important to note that when interpreting Figure S3, each subplot should be considered independently, and direct comparisons should not be made across the three subplots due to the different scales and units of the properties being measured.

Abbreviation	Description
AN	Atomic Number
QN	Quantum Number
С	Column in the Periodic Table
RAM	Relative Atomic Mass
VA	Atom Volume
R	Atom Radius
AEN	Atomic Environment Number
VEC	Valence Electron Concentration
XMB	Electronegativity (Martynov&Batsanov)
ХР	Electronegativity Mismatch (Pauling)
XA	Electronegativity Mismatch (Allen)
G	Shear Modulus
Е	Elastic Modulus
FIE	First Ionization Energy
SIE	Second Ionization Energy
TIE	Third Ionization Energy
СРМ	Chemical Potential (Miedema)
WF	Work Function
NCE	Effective Nuclear Charge (Slater)
CNE	Effective Charge Nuclear (Clementi)
MT	Melting Point
TB	Boiling Temperature
EV	Vaporization Enthalpy
EM	Melting Enthalpy
EA	Atomization Enthalpy
CE	Cohesive Energy
MC	Compression Modulus
DVE	Valence Electron Distance (Schubert)
DCE	Core Electron Distance (Schubert)
D	Density

Table S5 Elemental properties utilized for material

features calculation

The considered features include simple metrics such as atomic number and quantum number, physical properties like atomic radius and volume, electronic properties such as valence electron concentration and electronegativity, and thermodynamic properties including various enthalpy measures and phase transition temperatures [7, 12]. These elemental properties serve as the foundation for calculating material features that potentially capture informative relationships of multi-component systems. Table S5 lists the comprehensive set of elemental properties used in our feature engineering process. These properties span from fundamental atomic characteristics to complex physicochemical attributes, providing a diverse basis for materials feature calculation. These properties are selected based on their potential relevance to material behavior and performance prediction.

The feature transformation based on elemental properties and alloy compositions can be understood as a manual feature extraction process that enhances information sharing among elements.



Section 6. The SMA, Ti alloys and HEA test environments.

Figure S4. Box plots of figure of merit (FOM) values across the three material systems in our datasets.

The datasets used in this study comprise experimental composition-property data for three distinct material systems: shape memory alloys (SMAs), titanium alloys (Ti Alloys), and high-entropy alloys (HEAs). These datasets, which are available within our GitHub repository (<u>https://github.com/wsxyh107165243/FeatureGradientBO</u>), present distinct FOM distributions reflecting the different optimization objectives for each material system, as shown in Figure S4.



Figure S5. Pearson correlation matrices showing relationships between material composition, properties, and FOM for each test environment: (a) SMAs, showing correlations between Δ H, Mp, and Ap, individual elements, and FOM; (b) Titanium alloys, displaying correlations between yield strength (YS), ultimate tensile strength (UTS), Vickers hardness (VH), elemental compositions, and FOM; (c) HEAs, illustrating correlations between YS, UTS, elongation (El), elemental compositions, and FOM. Red indicates positive correlation, blue indicates negative correlation, with color intensity representing correlation strength.

To investigate the underlying relationships between elemental compositions and material properties, we conducted Pearson correlation analysis on the training datasets used for our three test environments (Figure S5). These correlation matrices reveal composition-property relationships within each material system. For SMAs, Hf shows strong positive correlation (0.62) with FOM, while Ti exhibits strong negative correlation (-0.66). In Ti alloys, Mo demonstrates the strongest correlation with FOM (0.36). For HEAs, Co appears as the most influential element (0.41). It is important to note that these Pearson correlation coefficients only capture linear relationships between variables and may not fully represent more complex, non-linear interactions present in the materials systems.

Our neural networks consist of a convolutional section followed by fully connected layers. The input layer processes alloy compositions (N_{elem} elements) and their corresponding elemental features (30 features per element). The convolutional section comprises two Conv2D layers with kernel size 1×31, each followed by batch normalization and ELU activation ($\alpha = 0.2$). A residual connection is implemented between the input and the output of the convolutional section to facilitate gradient flow. The flattened output is concatenated with three processing condition parameters before

entering the fully connected section, which consists of a hidden layer with 128 neurons and an output layer. Dropout (p=0.5) is applied after the hidden layer to prevent overfitting.

These networks were trained using the Adam optimizer with a learning rate of 5×10^{-4} and batch size of 16. The mean squared error (MSE) was used as the loss function. All input features were standardized using the standard scaler from scikit-learn [11]. The dataset was randomly split into training (70%), validation (15%), and test (15%) sets. The model was trained for 1000 epochs with early stopping monitored on the validation set to prevent overfitting. Model performance was evaluated using the coefficient of determination (R²) score.

	1			
Element	Lower limit (at. ratio)	Upper limit (at. ratio)	Step (at. ratio)	
Ti	0.2	0.55		
Ni	0.2	0.55		
Cu	0	0.2		
Hf	0	0.2		
Со	0	0.1		
Zr	0	0.2	0.001	
Fe	0	0.1		
Pd	0	0.33		
Та	0	0.05		
Nb	0	0.2		
V	0	0.1		

 Table S6 Composition constraints for SMAs

 Table S7 Composition constraints for Ti Alloys

Element	Lower limit (at. ratio)	Upper limit (at. ratio)	Step (at. ratio)
Ti	0.6	0.8	
Al	0	0.1	
Zr	0	0.2	
Мо	0	0.2	
V	0	0.15	
Ta	0	0.2	0.001
Nb	0	0.2	
Cr	0	0.1	
Mn	0	0.1	
Fe	0	0.1	
Sn	0	0.1	

Table S8 Composition constraints for HEAs

Element	Lower limit (at. ratio)	Upper limit (at. ratio)	Step (at. ratio)
С	0	0.06	
Al	0	0.16	
V	0	0.33	
Cr	0	0.4	
Mn	0	0.5	0.001
Fe	0	0.6	0.001
Co	0	0.5	
Ni	0	0.6	
Cu	0	0.36	
Мо	0	0.1	

Table S6-S8 present the considered elements and corresponding compositional constraints for SMAs, Ti Alloys HEAs test environments. For determining the number of all possible compositions in the design spaces, we implemented a dynamic programming approach, which is available in our

open-source code repository.

Section 7. Brief discussion of gradient ascent as inner loop argmax

We note that for EI, one common associated issue is that the AF value gradually approximates 0 as the experimental iterations proceed. This is challenging for gradient-based optimization methods (including SLSQP), as the "gradient values" will almost vanish in these cases. However, researchers have already come up with using logEI [13] instead of the original EI, as EI approaches 0, logEI \rightarrow - ∞ , mitigating the vanishing EI value problem. For our experiments here, we only use the original EI function values for the comparison results provided in the main text. This will cause an issue for gradient-based EI maximization and we expect better results if we implement logEI for our methods.

We examined two current, still actively maintained BO implementations, BayesianOptimization [14] and BoTorch. While BayesianOptimization relies on gradient values from finite difference calculations, BoTorch defaults to using automatic differentiation based on PyTorch for gradient computation. However, neither provides gradient calculations from GPR inputs back to composition inputs, which motivated our work.

Section 8. Brief clarification of design choice considerations

The number of initial compositional guesses, $20 \times N_{elem}$, was empirically determined from our large scale simulated experiments, showing good performance across different systems while maintaining reasonable resource consumption. It is worth noting that in both BoTorch and BayesianOptimization packages, this parameter is typically set as a fixed value (e.g., around 10) and does not vary with *x*/composition dimensionality. We believe our choice of this parameter setting is sufficiently conservative to achieve (at least) comparable inner loop results with BoTorch (considering multiple parallel runs).

As for random sampling strategies, while this approach has been validated by both BoTorch and BayesianOptimization packages, there remains potential for improvement. Specifically, strategies for generating more diverse initial samples as start points to initiate gradient ascent would be interesting. However, this exploration remains beyond the current scope of this work.

Citations

[1] K.E. Knipling, R.A. Karnesky, C.P. Lee, D.C. Dunand, D.N. Seidman, Precipitation evolution in Al–0.1 Sc, Al–0.1 Zr and Al–0.1 Sc–0.1 Zr (at.%) alloys during isochronal aging, Acta Materialia 58(15) (2010) 5184-5195.

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

[3] P. Dang, J. Hu, Y. Xian, C. Li, Y. Zhou, X. Ding, J. Sun, D. Xue, Elastocaloric Thermal Battery: Ultrahigh Heat-Storage Capacity Based on Generative Learning-Designed Phase-Change Alloys, Advanced Materials (2025) 2412198.

[4] C. Li, Q. Liang, Y. Zhou, D. Xue, A knowledge-based materials descriptor for compositional dependence of phase transformation in NiTi shape memory alloys, Materials Genome Engineering Advances (2024) e72.

[5] J. Frenzel, A. Wieczorek, I. Opahle, B. Maaß, R. Drautz, G. Eggeler, On the effect of alloy composition on martensite start temperatures and latent heats in Ni–Ti-based shape memory alloys, Acta Materialia 90 (2015) 213-231.

[6] J. Wilson, F. Hutter, M. Deisenroth, Maximizing acquisition functions for Bayesian optimization, Advances in neural information processing systems 31 (2018).

[7] Y. Zhang, C. Wen, P. Dang, T. Lookman, D. Xue, Y. Su, Toward ultra-high strength high entropy alloys via feature engineering, Journal of Materials Science & Technology 200 (2024) 243-252.

[8] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A.G. Wilson, E. Bakshy, BoTorch: A framework for efficient Monte-Carlo Bayesian optimization, Advances in neural information processing systems 33 (2020) 21524-21538.

[9] J. Gardner, G. Pleiss, K.Q. Weinberger, D. Bindel, A.G. Wilson, Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration, Advances in neural information processing systems 31 (2018).

[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825-2830.

[12] C. Wen, Y. Zhang, C. Wang, H. Huang, Y. Wu, T. Lookman, Y. Su, Machine-Learning-Assisted Compositional Design of Refractory High-Entropy Alloys with Optimal Strength and Ductility, Engineering (2024).

[13] S. Ament, S. Daulton, D. Eriksson, M. Balandat, E. Bakshy, Unexpected improvements to expected improvement for Bayesian optimization, Advances in Neural Information Processing Systems 36 (2023) 20577-20612.

[14] F. Nogueira, Bayesian Optimization: Open source constrained global optimization tool for Python, 2014. <u>https://github.com/bayesian-optimization/BayesianOptimization</u>. 2025).