A Prompt-Engineered Large Language Model, Deep Learning Workflow for Materials Classification

Siyu Liu, Tongqi Wen,* A. S. L. Subrahmanyam Pattamatta, and David J. Srolovitz†
Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, China
(Dated: February 1, 2024)

With the advent of ChatGPT, large language models (LLMs) have demonstrated considerable progress across a wide array of domains. Owing to the extensive number of parameters and training data in LLMs, these models inherently encompass an expansive and comprehensive materials knowledge database, far exceeding the capabilities of individual researcher. Nonetheless, devising methods to harness the knowledge embedded within LLMs for the design and discovery of novel materials remains a formidable challenge. In this study, we introduce a general approach for addressing materials classification problems, which incorporates LLMs, prompt engineering, and deep learning algorithms. Utilizing a dataset of metallic glasses as a case study, our methodology achieved an improvement of up to 463% in prediction accuracy compared to conventional classification models. These findings underscore the potential of leveraging textual knowledge generated by LLMs for materials especially with sparse datasets, thereby promoting innovation in materials discovery and design.

The concept of "AI for Materials" refers to the utilization of artificial intelligence (AI) techniques, such as machine learning (ML), deep learning (DL), and the increasingly prevalent large language models (LLMs), to design novel materials or investigate the composition-structure-property relationships in different materials [1–4]. Significant strides have been achieved in fields such as biomaterials [5, 6] and organic materials [7, 8], which can be attributed to the relatively uniform representation of organic molecules [9] and the availability of numerous high-quality datasets [10]. However, for inorganic materials, particularly those with large compositional space (element types > 4), the application of AI presents a more complex challenge. This complexity arises due to factors such as the scarcity of experimental data [4], diverse properties of inorganic materials, complicated crystal structures, potential existence of intermediate phases (e.g. intermetallics in alloys), and defects including vacancies, dislocations, and grain boundaries [11, 12]. Contrasting with organic molecules that can be modeled with neural networks, multi-component inorganic materials often involve the amalgamation of various phases and defects or even the formation of new phases, presenting an ongoing challenge in the structural representation [13, 14]. Although large-scale databases such as the Materials Project [15], ICSD [16], and AFLOW [17] exist, data pertinent to specific tasks such as the relationships between composition and mechanical properties in alloys and structural stability of two-dimensional materials are often sparse and dispersed across different datasets, complicating the assembly of substantial databases for training purposes. Furthermore, input features can vary across datasets, sometimes necessitating manual feature construction based on domain knowledge [18]. Collectively, these issues hinder the application of AI in the realm of inorganic materials, subsequently impeding the discovery and design of new materials.

The advancement of LLMs, particularly with the advent of ChatGPT, has sparked a surge of interest in distilling knowledge from these models through prompt engineering [19–21]. Notable examples include constructing scientific question answering knowledge bases using LLMs [22], transferring knowledge from large to small models via in-context learning [23], and training materials-specific LLMs [24, 25]. Given the expanding capabilities of LLMs and the growing volume of trained data, it is conceivable to consider these models as encyclopedia resources for materials science, enabling the extraction of text-based knowledge [26]. As shown in the upper part of Fig. 1, this approach addresses some challenges associated with data collection and feature extraction in conventional processes. By representing all content in textual format, the generated data becomes universally applicable, offering a versatile solution for materials science research.

Therefore, we have established a universal workflow for material text feature-label classification using textual data generated by LLMs, as illustrated in the lower part of Fig. 1. The workflow is structured as follows: (i) define the material classification problem to be addressed; (ii) design prompts via prompt engineering to distill knowledge from LLMs and store it as textual data; (iii) fine-tune a bidirectional encoder representations transformers (BERT) [27] model (commonly employed in natural language processing) to train on the textual data-label pairs; (iv) apply the model to explore new materials or study composition-structure-property relationships. In an example problem involving the classification of 5,577 metallic glasses (MGs) with experimental datasets labeling the MG forming categories (bulk MG, ribbon, or non-ribbon), a BERT model trained with optimized prompts and position embedding methods achieved up to a 48% increase in classification accuracy compared to traditional ML models. The classification models obtained from our workflow achieved an overall accuracy of 97.7%. For the classification of bulk MGs (BMGs) with the smallest sample number ($\sim$11% of the entire dataset), the accuracy improved by up to 463% compared to traditional ML models. These results underscore the superiority of our workflow in addressing material classification problems and this workflow can be extended to various material applications. Additionally, tremendous potential of "AI for Materials" is revealed, grounded in natural language processing and prompt engineering.

* tongqwen@hku.hk
† srol@hku.hk

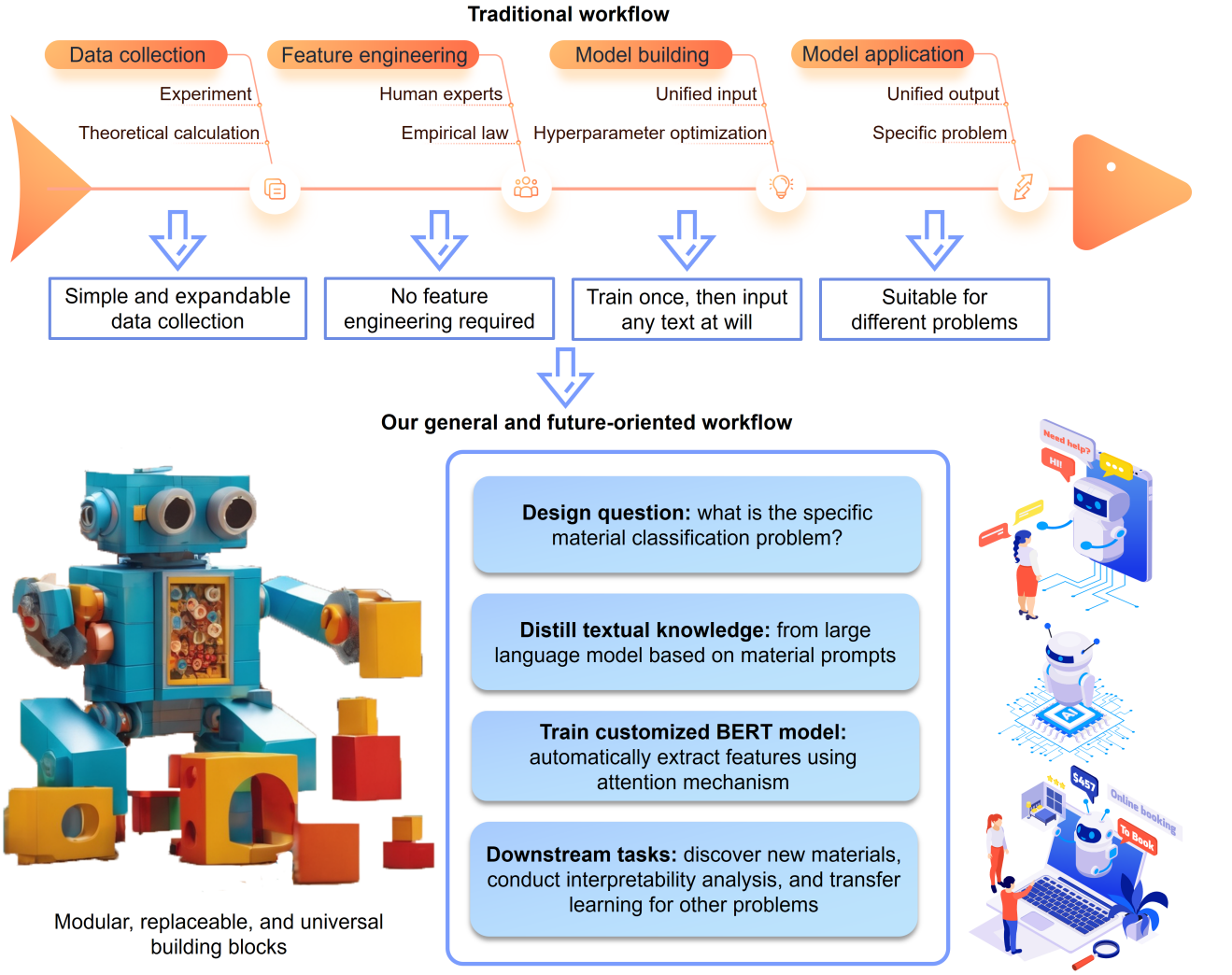

FIG. 1. Comparison between traditional and future-oriented general deep learning workflows. The general workflow consists of generating textual data from prompt-engineered large language models and training deep learning models.

Text-based materials classification workflow

Our material classification workflow comprises the four-tier structure depicted within the blue box in Fig. 1, and is readily adaptable to various material classification problems. Initially, researchers are required to undertake fundamental data collection and delineate the prediction objectives and categories, such as determining whether a material exhibits stability, corrosion resistance, or favorable optoelectronic transmission properties.

Secondly, the inputs for ML models in conventional workflows typically comprise material properties obtained through experiments or theoretical calculations, demanding substantial expertise as well as experimental or simulation efforts. Different researchers may select varying property features, leading to multiple datasets for the same problem [28]. However, training numerical-based ML models on the combination of different datasets often fails due to inconsistencies in input feature vectors. Furthermore, the number of datasets for specific niche tasks in materials science research is quite small ($<$ several hundreds), negligible compared to the millions of samples in classical computer science fields like ImageNet dataset [29]. For example, in the field of alloys, the dataset for fracture and impact toughness of high-entropy alloys contains only 154 data points [30], the fatigue database of complex metallic alloys consists of merely 272 data points [31], and the dataset of mechanical properties of high-entropy alloys has $\sim$370 data points [32]. These limitations hinder the application of ML in alloys. In our workflow, we employ textual inputs generated by LLMs, offering the advantage of eliminating the need for manual feature extraction. Moreover, LLMs can output content relevant to specific problems in materials science with customized prompts. As LLMs possess considerably more parameters than traditional ML models, they can generate a richer and more diversified knowledge base.

Furthermore, within our workflow, the classification BERT model employs the attention mechanism for self-instruct

learning, enabling automatic feature extraction. Different pre-trained models can be utilized for corresponding downstream tasks. Models based on the attention mechanism are not limited to text input but also applicable to areas like image recognition [33] and molecular structure identification [34]. These features allow our workflow to seamlessly support the combination of different datasets and multimodal expansion, addressing the issue of insufficient datasets for specific material problems often encountered in traditional methods.

Finally, the material classification models trained through our workflow can be utilized in numerous downstream tasks, such as discovering new materials across different categories, conducting interpretability analysis to extract composition-structure-property relationships of materials, and even fine-tuning models to achieve excellent results in other classification tasks. Our workflow represents a novel and general approach for material sciences by constructing large-scale databases from prompt-engineered LLMs and training general DL models to expedite materials discovery. We showcase the efficiency and accuracy of the workflow by using the classification problem of MGs as an example.

Application: metallic glasses classification

MGs are amorphous materials typically composed of a combination of metal and other elements [35]. They are renowned for superior engineering performance, including high strength, high thermal stability, and excellent corrosion resistance [36]. Despite ongoing efforts to discover MGs with enhanced comprehensive performance, their formability remains a complex and challenging problem in materials science. Predicting the glass formability is also difficult for atomistic simulations, due to the large size and long timescale (unattainable by first-principles calculations) and the absence of accurate interatomic potentials for molecular dynamics simulations. Moreover, MGs have an extensive compositional space, often comprising three or more elements, rendering the experimental search for new applicable MGs a time-consuming process [37, 38].

Various ML methods [39] have been employed to classify MGs; however, these methods often rely on manually calculated or carefully designed material features. The inconsistency of designed features and datasets across different ML models necessitates researchers to recalculate those features each time when utilizing other models. In the worst-case scenario, certain features rely on experimental calculations, rendering a MG dataset applicable to one model but not another. In this regard, we leveraged the experimental dataset of MGs by [40] and applied it to the proposed workflow here. The dataset compiled by [40] consists of 8,415 alloy compositions with their corresponding MG forming categories. The alloy compositions were classified into three categories based on the critical cooling rate (R_c): BMG ($R_c < 10^3$ K/s), ribbon (10^3 K/s $< R_c <$ 10^6 K/s), and non-ribbon (NR, $R_c > 10^6$ K/s). The glass formability is then BMG $>$ ribbon $>$ NR. From this dataset, 5,577 alloy compositions with only a single glass-forming category were selected as input dataset, with 80% of them being train set while the rest being test set. As shown in Fig. 2a, a customized version of the general workflow was developed for the given input data. The blue arrows indicate the training process, which involves expanding the alloy compositions into textual data using an LLM. Subsequently, a DL model is trained on the textual data from the training set for MG classification. Finally, the analysis of efficiency and accuracy and model interpretation are performed using textual data from the test set. These three main steps are illustrated in Fig. 2b. The green arrows indicate the application stages of the model. For an unexplored alloy composition, the workflow can determine the MG category (BMG, ribbon, or NR) within seconds.

In the first step, we drew inspiration from a previous design of metal-organic framework prompt engineering [25] and designed a prompt called "MetalPrompt" to generate textual descriptions for various alloy compositions. Fig. 3 illustrates the design concept of our "MetalPrompt". The schematic text for the input query of the LLM is on the left and consists of three main components: the prompt section, the one-shot example section, and the input section. The prompt section aims to enable the LLM to focus on generating domain-specific knowledge while minimizing hallucination. The one-shot example section is designed to ensure a consistent output structure and incorporate the information of interest. In the output, we focus on three levels of information: alloy composition (atomic percent of each element), elements (thermodynamic properties of each element), and alloy physical and chemical properties. The input section ensures that the task description is simple and interchangeable, thus preventing the model output from being disrupted by irrelevant information. In terms of overall structure, we have added some emphasis symbols such as "\\" at the beginning and end of the paragraph, as well as "\%composition\%" symbols for compositions, guiding the model to recognize critical information and structural layering in the input [41]. Supplementary Figs. S1-S3 show the output without "MetalPrompt" in different LLMs, and the output text structure is inconsistent and exhibits hallucinations. Supplementary Figs. S4 and S5 compare the prompt baseline and ML models, as well as show the effects of using different prompt methods. In both comparisons, our "MetalPrompt" demonstrates the best performance.

Next, we trained a MGs BERT (MgBERT) classification model that combines self-instruct learning and supervised classification. DL models for natural language processing typically require a vast number of fitting parameters, necessitating the selection of a suitable pre-trained model as the base model. MgBERT is fine-tuned based on MatSciBERT (a pre-trained model based on materials science texts) [42]. Figs. 4a, b display a comparison of different pre-trained BERT models, with MatSciBERT achieving the best results in classification accuracy and model loss on the test set. However, MatSciBERT can only accept inputs of up to 512 tokens, while our text data has a maximum length of 859 when converted to tokens. Consequently, we designed a dynamically resizable embedding layer called "MgBERT Embedding" to meet the requirements of input scalability. This embedding layer can assign weights based on layers and proportions according to

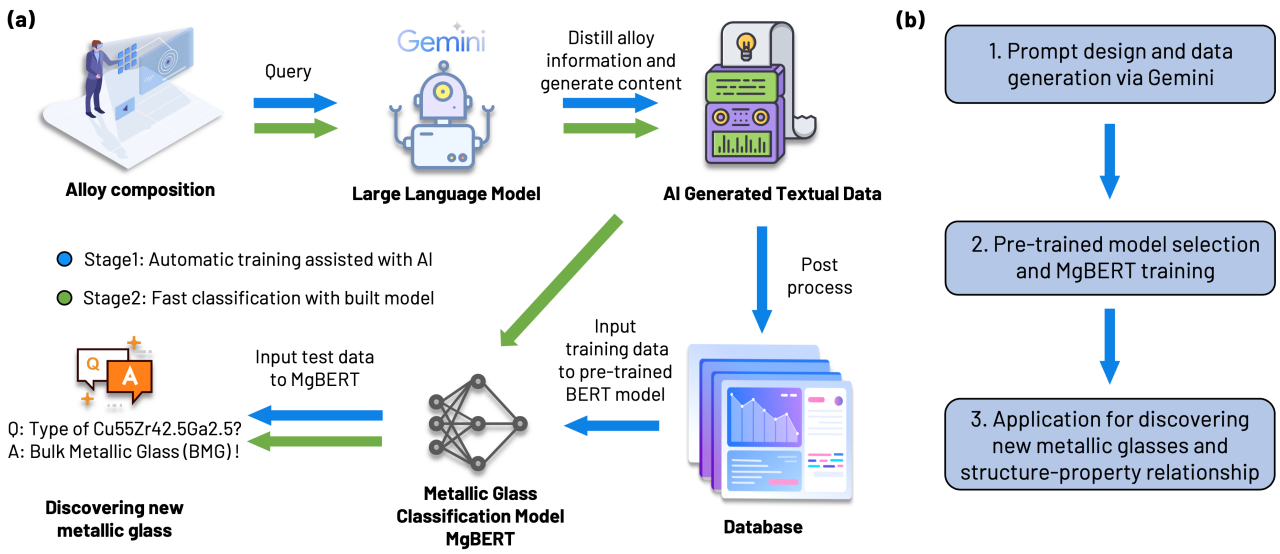

FIG. 2. Application of our workflow for the classification of metallic glasses. (a) Schematic for the customized workflow, from known alloy composition to the discovery of new metallic glasses. (b) Three essential steps in the customized workflow, including the conversion of alloy composition to textual data, training of classification models, and the discovery and interpretation of new metallic glasses.

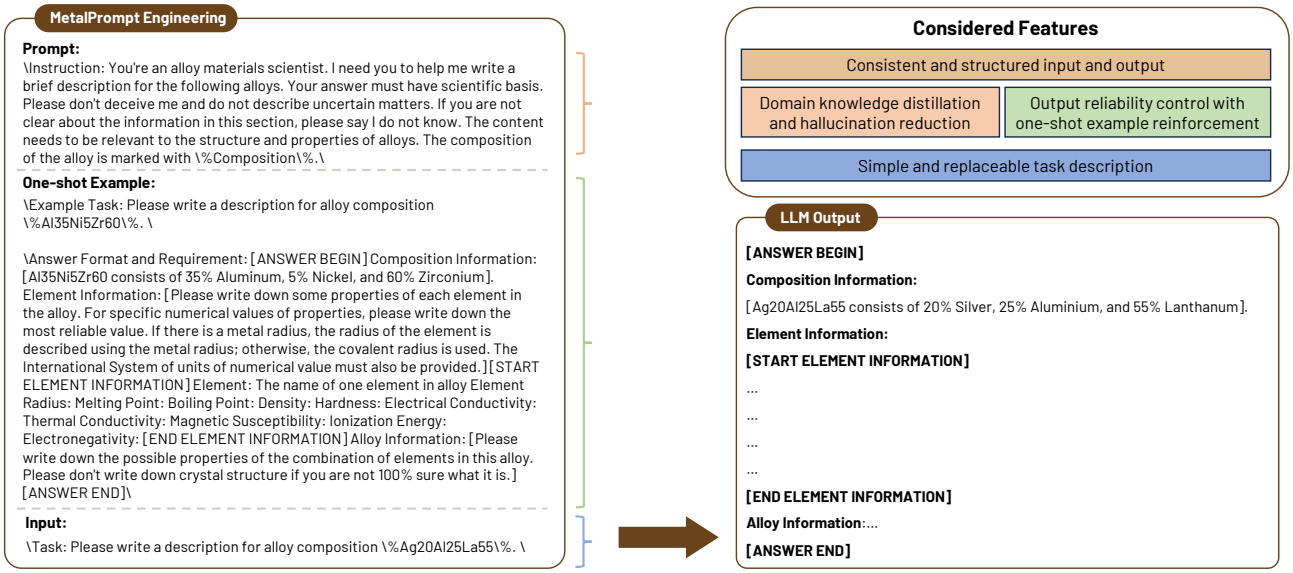

FIG. 3. Schematic of "MetalPrompt" for generating textual data with prompt engineering. The left part depicts the schematic text for the input query of the large language model. The sections enclosed within brackets of different colors (on the right) represent diverse features considered in prompt engineering.

the preset length of the new token, building upon the embedding weights trained by MatSciBERT. This approach generates new position embedding that can support longer text inputs and circumvent time-consuming weight initialization training. This new position embedding layer (in Fig. 4c) is utilized for tokenizing the input text of the model. From the comparison of results in Figs. 4g and h, our MgBERT embedding Method 3 (Fig. 4f) exhibits higher accuracy and lower train set loss, compared to the embedding layer method in Fig. 4d, which uses newly initialized weights, and that in Fig. 4e, which only employs pre-trained weights to cover the first 512 embedded positions. For different pre-trained models trained on the same text data (Figs. 4i, j), our MgBERT has the best performance in training set loss, training set classification accuracy, and test set classification accuracy. Notably, for alloy data in the test set that our model has never encountered before, our model achieved an accuracy of 88.5%, an improvement of 9% compared to Longformer, and an improvement of 4% compared to the best performing MatSciBERT. The detailed classification results can be observed in the confusion matrix in Supplementary Fig. S6. These findings showcase the success of our improvement in embedding

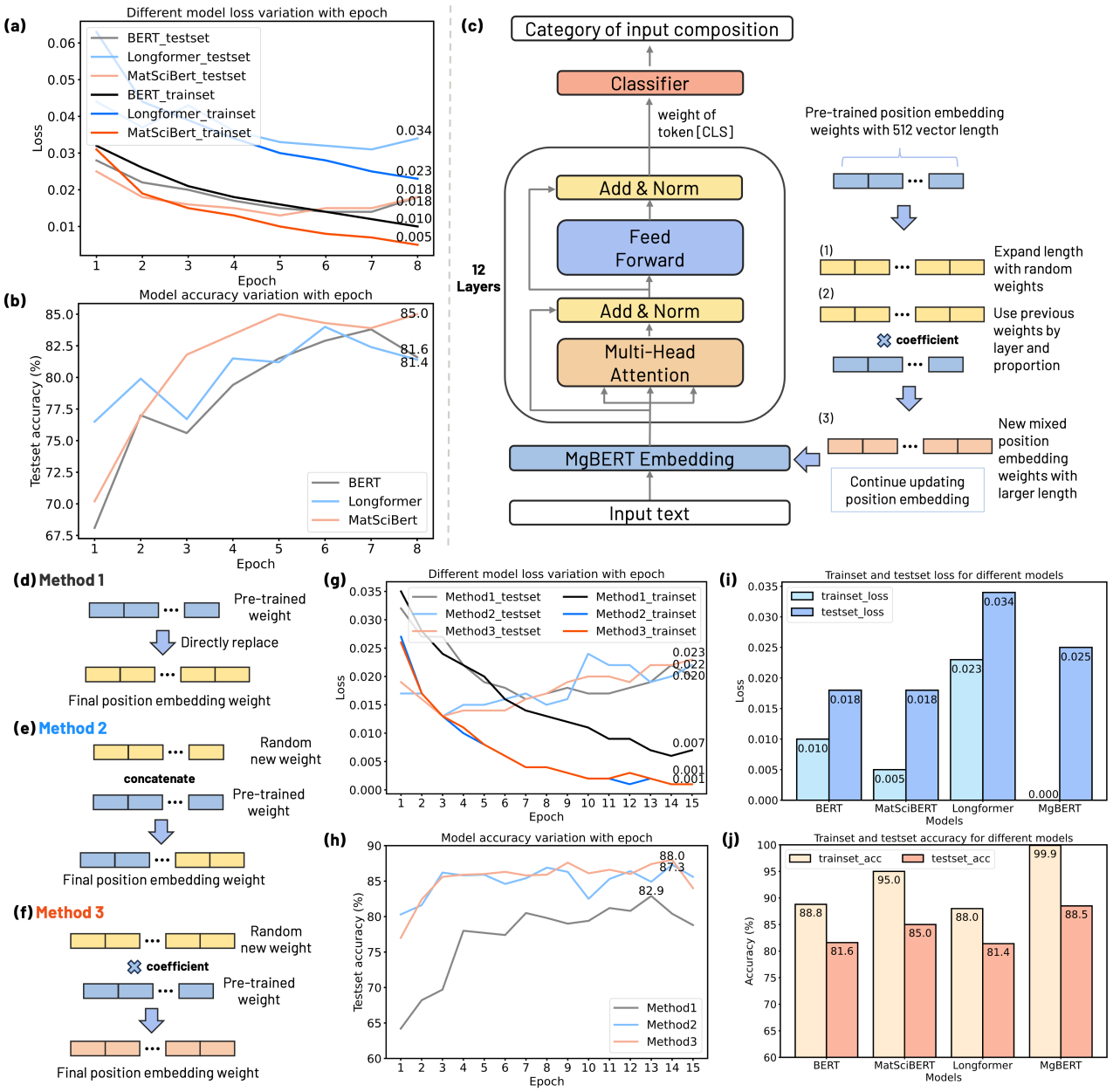

FIG. 4. Performance comparisons and architecture of MgBERT. a, b, performance comparison of various pre-trained models. c, fundamental architecture of the MgBERT model and schematic diagram of the working principle of MgBERT position embedding. d, e, f, three distinct position embedding methods. g, h, performance comparison of different embedding techniques. i, j, performance comparison between MgBERT and other pre-trained models.

structure, as it simultaneously supports longer input text and enhances accuracy. To examine the effectiveness of the entire workflow, we trained a baseline model based on logistic regression (LR) using the training set, as well as a support vector machine (SVM) model and a gradient boosting decision tree (GBDT) model. Then, we conducted tests on the test set and compared the classification performance of different methods (Fig. 5). For the default three-class problem, our workflow achieved a maximum accuracy increase of 48% across the entire dataset and a peak enhancement of 32% on the test set. Even more astonishing is our performance with respect to identifying BMGs, as shown in Fig. 5b. For this task, our accuracy soared by up to 463% on the entire dataset and by as much as 307% on the test set. Considering that BMGs constitute only ∼11% of our dataset, these results highlight the remarkable capability of our text-based material classification workflow in accurately recognizing patterns in data-scarce scenarios, surpassing traditional ML models.

DL models are often considered to lack algorithmic transparency [43]. Language models such as BERT typically have hundreds of millions of parameters [27], making it challenging to provide detailed explanations of the model at the al-

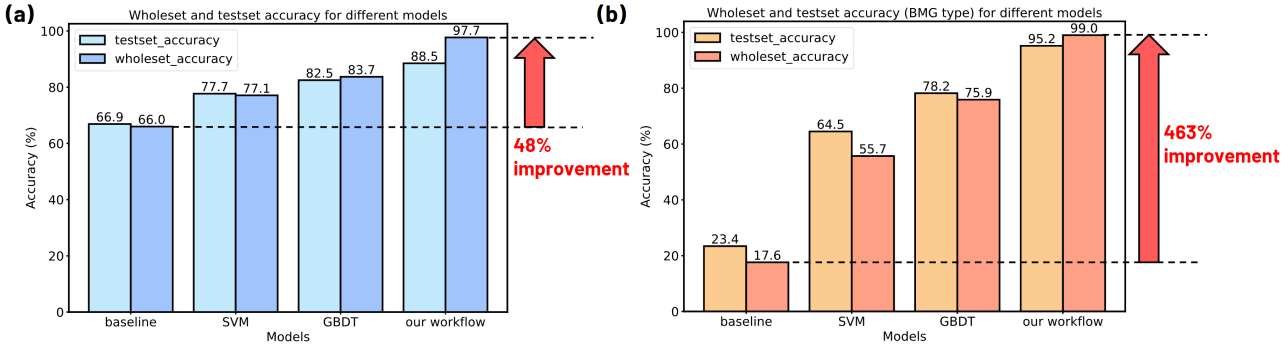

FIG. 5. Comparison of classification accuracy between our workflow and traditional machine learning models. a, comparison in classifying all category tasks. b, comparison in classifying bulk metallic glasses.

gorithmic level. Here, we utilized the local interpretable model agnostic interpretations (LIME) [44] method to analyze from an input-output perspective. We selected text data of alloy compositions $Cu_{55}Zr_{42.5}Ga_{2.5}$ (classified as BMG), $Ag_{20}Al_{25}La_{55}$ (classified as ribbon), and $Al_{40}Mn_{25}Si_{35}$ (classified as NR) as three examples. Fig. 6a and Supplementary Fig. S7 illustrate that alloy composition is the primary contributing factor to these classification results, consistent with the classical Inoue rule [45] and high-throughput experiment results [46]. According to Inoue's rule, components of an alloy introduce varying atomic size differences, altering the internal dense random packing structure. Moreover, the model has extracted some elemental property information. For example, "Melting" refers to the melting point, which is one of the critical criteria for determining the formation of MGs [47]. Additionally, "745" is the ionization energy of copper, which could affect the interatomic forces within the alloy and thereby influence MG formation. Furthermore, "2477" represents the boiling point of gallium, and the polymer-like thermoplastic behavior observed in MGs is thought to be related to boiling points [48], also connected to the formation of functional MGs. These findings indicate that our text classification model can perform self-instruct learning and automatically extract features from a given text. Building on this foundation, in contrast to traditional ML models that rely on manually designed features as input, our text-based workflow enables models to accept raw text as input. This flexibility allows for rapid transfer and recycling of pre-trained model parameters between different materials classification tasks. We also visualized the attention scores within the classification model and the model extracts features based on three distinct patterns (Fig. 6b). This process ensures that after 12 layers of training, the output [CLS] (classify) token possesses sufficient information for classification. Since we use the value of the [CLS] token from the final layer to make judgments in the classification layer, Fig. 6c visualizes the emphasis placed on different sections of the text. The results suggest that the composition and alloy layers have garnered significant attention. A possible explanation is that key tokens within these two layers have captured essential features from the input text. Consequently, the [CLS] token can achieve accurate classification results by focusing on those specific token positions. These findings demonstrate that the classification MgBERT model, generated by our general workflow, has the abilities of dynamic feature extraction, modeling inter-word relationships, hierarchical representation learning, and bidirectional context understanding, which may be potential indications for good classification performance.

Discussion

We emphasize that classification cannot be successfully achieved using direct questioning alone (as seen in Supplementary Figs. S8-S10). Therefore, as part of our workflow, we designed a BERT-based classification model to categorize materials using text generated by the LLM. In practice, employing a classification model is a necessary step at this juncture, as directly querying the LLM yielded a $\sim$0 accuracy rate.

The general workflow here offers a novel approach to address a significant challenge in "AI for material science", where datasets are often small. By utilizing a pre-trained BERT model, such as MatSciBERT optimized for material science, we can fine-tune it (pre-trained model) with textual data from prompt-engineered LLM to achieve high accuracy in classification problem (up to 99.0% for the entire dataset).

We propose a general approach for materials classification and demonstrate its efficiency in an example case of the classification problem in MGs. Our general workflow can be applied to any type of material applications, provided that labeled material samples are available. The findings point towards a future where the integration of advanced language models and domain-specific fine-tuning could revolutionize the material classification process, especially in cases with sparse data. We envision that LLMs may eventually serve as "world models" [49], encapsulating knowledge across diverse materials and domains. Consequently, such models based on text for classification, prediction, or even generating new materials will become potential tools for addressing challenges in material design. A sufficiently robust LLM could potentially provide desired classification outcomes directly through appropriate prompting and significantly streamline the process of materials discovery and design.

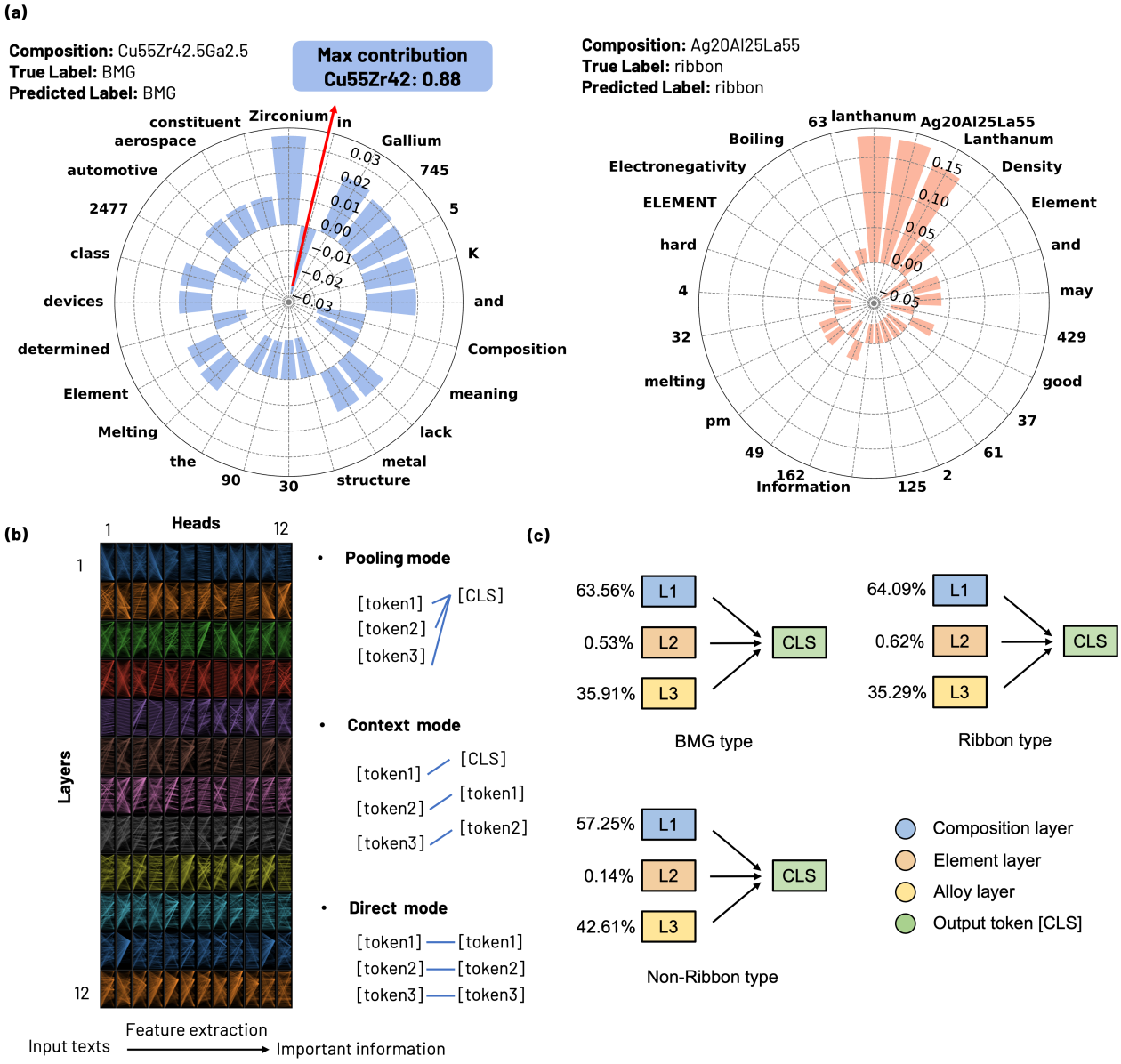

FIG. 6. Interpretability for classification deep learning model. a, contribution of input text to the classification results, with the outward-facing bar on the polar axis representing a positive contribution and the opposite direction signifying a negative contribution. b, c, Attention-level analysis. b, attention score flow diagram across different layers and heads. The model has 12 layers, each with 12 heads, and each head is a neural network. Both input/output for each head consist of 768 tokens. Different colors represent different layers, while lines represent connections between input and output tokens of each head. Line thickness suggests the attention score: a larger attention score implies that the output token focuses more on the input token at a specific position. Three modes on the right represent typical patterns of output token information extraction from input tokens. The pooling mode demonstrates input information aggregation to the same output token, the context mode shows input aggregation to the output token at "position-1" (token 2 to 1), and the direct mode indicates direct input transfer to the output token at the same position. The output from 12 heads in each layer is concatenated and passed to the next layer, enabling the extraction and transfer of crucial input text information to the final classifier. c, attention pooling diagram of the final attention layer of MgBERT. To facilitate differentiation, the input text is divided into three text layers. Blue: composition layer consisting of tokens before "element information" in the input text. Orange: element layer comprising tokens before "alloy information". Yellow: alloy layer containing the remaining tokens. The number on the left indicates the last attention layer's [CLS] (classify) token attention to various text layers.

Methods

Data processing for metallic glasses dataset

In processing the dataset for metallic glasses, we curated our initial compilation of 8,415 entries by excluding alloys comprising multiple categories. This refinement resulted in a dataset encompassing 5,577 single-class alloys, along with their respective classifications. The aim was to construct an unambiguous dataset, thereby facilitating the accuracy of analysis and model training.

To implement prompt engineering, we harnessed the capabilities of the langchain library. We designed prompts with varying templates in advance, establishing a "Composition" variable to dynamically cycle through the list of alloy compositions. This approach enabled us to systematically generate textual data in a large scale, ensuring consistency and efficiency in our data preparation workflow.

For the ingestion of data into the MgBERT model, we divided the 5,577 data points into a training and test set at a ratio of 80:20. This arrangement yielded 4,460 entries for training constituting 80% of the dataset, and 1,117 entries for testing purposes. It was imperative that the MgBERT model was trained comprehensively, indicating that the data allocation is reasonable.

Regarding the evaluation of datasets, while the MgBERT model was assessed using the designated test set, the machine learning models underwent evaluation using a different approach because they can not receive textual data. We employed the Matminer and Pymatgen tools to convert the chemical formulas of alloys into "composition features" as defined within Matminer's framework. However, it is noteworthy that three entries failed to transform correctly. Consequently, the test set for evaluation purposes comprised of 1,116 entries, reducing the overall dataset to 5,574 records. Compared to MgBERT, there were three fewer data sets, which had little impact on accuracy comparison.

Large language model and prompt engineering

In this work, we used the Google Gemini-pro large language model, which is currently free and surpasses human experts in massive multi-task language understanding [50]. As discussed earlier, our research harnesses the automated orchestration capabilities of the langchain library to script the bulk generation of prompts and textual data. This scripting forms an integral part of a generic workflow and offers a high degree of adaptability. The prompt templates we have developed are not rigid constructs; rather, they are designed to be inherently flexible, allowing for arbitrary modifications tailored to the specific demands and objectives of researchers. This level of customization is critical in the context of materials science where the nuances of the subject matter can vary significantly from one study to another. The ability to fine-tune prompts to align with particular research questions or datasets ensures that the language model can be effectively leveraged to generate effective and contextually relevant textual data outputs.

Our investigation sought to demonstrate the effectiveness of prompt engineering by creating a range of alternative prompt templates to act as comparative benchmarks. These templates comprised one that integrated a chain-of-thoughts (CoT) approach with a few-shot learning paradigm (denoted as "few-shot with CoT"), another that depended exclusively on few-shot learning examples (denoted as "few-shot"), a third that was based on zero-shot learning scenarios (denoted as "direct inquiry"), and our custom-designed "MetalPrompt" from our established workflow. By leveraging these diverse templates, we crafted specific prompts that were subsequently employed to interrogate the Gemini-pro model. The underlying rationale for this varied prompt strategy was to gather and juxtapose the accuracy metrics derived from different prompting methodologies.

MgBERT training and evaluation

Here we used MatSciBERT as the basic model and implemented a subsequent classifier function with MgBERT embedding to build MgBERT. Accuracy and cross entropy are used to evaluate the effectiveness of the model. For the classification principle of MgBERT, we used the value of [CLS] token in the last layer of the attention section of MgBERT as the output to aggregate the information of the entire model. Then, through a classifier, we compress the value of the token into a vector of length 3, and label 0 as BMG, 1 as Ribbon, and 2 as Non-ribbon. Then we calculate the maximum position of the output vector value and locate it to the final classification result. The overall expenses associated with developing MgBERT comprise $\sim$\$46 USD for training on Nvidia V100 ($\sim$30 hours and $\sim$\$1.53 USD per hour) and $\sim$\$1.5 USD for $\sim$1.89 million input and $\sim$2.14 million output tokens (according to the official pricing of Gemini-pro on January 29, 2024). Upon applying MgBERT for inference on 1,000 different compositions, the cost is as low as $\sim$\$0.3 USD. This cost-effective nature of training and implementing MgBERT enables researchers to adopt this general approach across various materials.

Interpretability method

To elucidate the relationship between input and output, we used the LIME (local interpretable model agnostic explanations) technique, which is widely used to explain the predictions of machine learning models by constructing the relationship function via interpolation sampling. Here we displayed the top 25 important features and performed 1,000 interpolation samples for function fitting. Although our parameters were initially set to select the top 50 features, due to visualization constraints, we chose the top 25 features for display.

For the explanation at the attention level, we used the bertviz library to complete the attention score flow diagram for 12 layers and 12 attention heads. To attribute the impor-

tance of the [CLS] token in the final attention layer, we first averaged the attention score of the 12 attention heads in the last layer. Then, we divided the input text words into three layers based on their positions, namely, composition, element, and alloy layer. To evaluate the importance of each layer, we calculated the sum of attention scores within that layer, weighted by the token length ratio, and then divided by the total token length.

Data availability

The method section provides the models and algorithms employed in this study, while specific parameter implementations can be found in the supplementary notes in supplementary information. The data used in this work is available in the supplementary data, including the metallic glasses dataset, alloy description files generated by large language model Gemini-pro, MgBERT classification model training logs, and data used for figure plotting.

Code availability

All the codes used in the paper will be made available on GitHub upon the acceptance of manuscript, and these codes can also be acquired upon reasonable requests.

References

References 1-50 are for the main text.

[1] S. G. Louie, Y.-H. Chan, F. H. da Jornada, Z. Li, and D. Y. Qiu, Discovering and understanding materials through computation, Nature Materials **20**, 728 (2021).

[2] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon, and E. D. Cubuk, Scaling deep learning for materials discovery, Nature **624**, 80 (2023).

[3] J. Li, K. Lim, H. Yang, Z. Ren, S. Raghavan, P.-Y. Chen, T. Buonassisi, and X. Wang, Ai applications through the whole life cycle of material discovery, Matter **3**, 393 (2020).

[4] D. Raabe, J. R. Mianroodi, and J. Neugebauer, Accelerating the design of compositionally complex materials via physics-informed artificial intelligence, Nature Computational Science **3**, 873 (2023).

[5] D. F. Nippa, K. Atz, R. Hohler, A. T. Müller, A. Marx, C. Bartelmus, G. Wuitschik, I. Marzuoli, V. Jost, J. Wolfard, *et al.*, Enabling late-stage drug diversification by high-throughput experimentation with geometric deep learning, Nature Chemistry , 1 (2023).

[6] A. Tropsha, O. Isayev, A. Varnek, G. Schneider, and A. Cherkasov, Integrating qsar modelling and deep learning in drug discovery: the emergence of deep qsar, Nature Reviews Drug Discovery , 1 (2023).

[7] T. Weiss, E. Mayo Yanes, S. Chakraborty, L. Cosmo, A. M. Bronstein, and R. Gershoni-Poranne, Guided diffusion for inverse molecular design, Nature Computational Science **3**, 873 (2023).

[8] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, Machine learning for molecular simulation, Annual review of physical chemistry **71**, 361 (2020).

[9] D. S. Wigh, J. M. Goodman, and A. A. Lapkin, A review of molecular representation in the age of machine learning, Wiley Interdisciplinary Reviews: Computational Molecular Science **12**, e1603 (2022).

[10] K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. Billinge, *et al.*, Recent advances and applications of deep learning methods in materials science, npj Computational Materials **8**, 59 (2022).

[11] E. R. Antoniuk, G. Cheon, G. Wang, D. Bernstein, W. Cai, and E. J. Reed, Predicting the synthesizability of crystalline inorganic materials from the data of known material compositions, npj Computational Materials **9**, 155 (2023).

[12] J. Noh, G. H. Gu, S. Kim, and Y. Jung, Machine-enabled inverse design of inorganic solid materials: promises and challenges, Chemical Science **11**, 4871 (2020).

[13] H. Xiao, R. Li, X. Shi, Y. Chen, L. Zhu, X. Chen, and L. Wang, An invertible, invariant crystal representation for inverse design of solid-state materials using generative deep learning, Nature Communications **14**, 7027 (2023).

[14] D. Steinberger, H. Song, and S. Sandfeld, Machine learning-based classification of dislocation microstructures, Frontiers in Materials **6**, 141 (2019).

[15] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, Commentary: The Materials Project: A materials genome approach to accelerating materials innovation, APL Materials **1**, 011002 (2013).

[16] D. Zagorac, H. Müller, S. Ruehl, J. Zagorac, and S. Rehme, Recent developments in the Inorganic Crystal Structure Database: theoretical crystal structure data and related features, Journal of Applied Crystallography **52**, 918 (2019).

[17] S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, and D. Morgan, Aflow: An automatic framework for high-throughput materials discovery, Computational Materials Science **58**, 218 (2012).

[18] T.-S. Vu, M.-Q. Ha, D.-N. Nguyen, V.-C. Nguyen, Y. Abe, T. Tran, H. Tran, H. Kino, T. Miyake, K. Tsuda, *et al.*, Towards understanding structure–property relations in materials with interpretable deep learning, npj Computational Materials **9**, 215 (2023).

[19] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, Large language models in medicine, Nature medicine **29**, 1930 (2023).

[20] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, *et al.*, Chatgpt for good? on opportunities and challenges of large language models for education, Learning and individual differences **103**, 102274 (2023).

[21] B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F. J. Ruiz, J. S. Ellenberg, P. Wang, O. Fawzi, *et al.*, Mathematical discoveries from program search with large language models, Nature **625**, 468 (2023).

[22] J. Pereira, R. Fidalgo, R. Lotufo, and R. Nogueira, Visconde: Multi-document qa with gpt-3 and neural reranking, in *European Conference on Information Retrieval* (Springer, 2023) pp. 534–543.

[23] D. Chen, S. Song, Q. Yu, Z. Li, W. Wang, F. Xiong, and B. Tang, Grimoire is all you need for enhancing large language models (2024), arXiv:2401.03385.

[24] T. Xie, Y. Wa, W. Huang, Y. Zhou, Y. Liu, Q. Linghu, S. Wang, C. Kit, C. Grazian, and B. Hoex, Large language models as master key: Unlocking the secrets of materials science with gpt (2023), arXiv:2304.02213.

[25] Z. Zheng, O. Zhang, C. Borgs, J. T. Chayes, and O. M. Yaghi, Chatgpt chemistry assistant for text mining and the prediction of mof synthesis, Journal of the American Chemical Society **145**, 18048 (2023).

[26] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, Recent advances in natural language processing via large pre-trained language models: A survey, ACM Computing Surveys **56**, 1 (2023).

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding (2018), arXiv:1810.04805.

[28] D. Morgan and R. Jacobs, Opportunities and challenges for machine learning in materials science, Annual Review of Materials Research **50**, 71 (2020).

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in *2009 IEEE conference on computer vision and pattern recognition* (Ieee, 2009) pp. 248–255.

[30] X. Fan, S. Chen, B. Steingrimsson, Q. Xiong, W. Li, and P. K. Liaw, Dataset for fracture and impact toughness of high-entropy alloys, Scientific Data **10**, 37 (2023).

[31] Z. Zhang, H. Tang, and Z. Xu, Fatigue database of complex metallic alloys, Scientific Data **10**, 447 (2023).

[32] S. Gorsse, M. Nguyen, O. N. Senkov, and D. B. Miracle, Database on the mechanical properties of high entropy alloys and complex concentrated alloys, Data in brief **21**, 2664 (2018).

[33] X. Li, X. Hu, X. Chen, J. Fan, Z. Zhao, J. Wu, H. Wang, and Q. Dai, Spatial redundancy transformer for self-supervised fluorescence image denoising, Nature Computational Science **3**, 1067 (2023).

[34] G. Zhou, Z. Gao, Q. Ding, H. Zheng, H. Xu, Z. Wei, L. Zhang, and G. Ke, Uni-mol: a universal 3d molecular representation learning framework (2023).

[35] A. K. Varshneya and J. C. Mauro, Chapter 1 - introduction, in *Fundamentals of Inorganic Glasses (Third Edition)*, edited by A. K. Varshneya and J. C. Mauro (Elsevier, 2019) third edition ed., pp. 1–18.

[36] Q. Halim, N. A. N. Mohamed, M. R. M. Rejab, W. N. W. A. Naim, and Q. Ma, Metallic glass properties, processing method and development perspective: a review, The International Journal of Advanced Manufacturing Technology **112**, 1231 (2021).

[37] G. Liu, S. Sohn, C. S. O'Hern, A. C. Gilbert, and J. Schroers, Effective subgrouping enhances machine learning prediction in complex materials science phenomena: Inoue's subgrouping in discovering bulk metallic glasses, Acta Materialia **265**, 119590 (2023).

[38] Y. Li, S. Zhao, Y. Liu, P. Gong, and J. Schroers, How many bulk metallic glasses are there?, ACS combinatorial science **19**, 687 (2017).

[39] Z. Zhou, Y. Shang, and Y. Yang, A critical review of the machine learning guided design of metallic glasses for superior glass-forming ability, Journal of Materials Informatics **2**, 1 (2022).

[40] L. Ward, S. C. O'Keeffe, J. Stevick, G. R. Jelbert, M. Aykol, and C. Wolverton, A machine learning approach for engineering bulk metallic glass alloys, Acta Materialia **159**, 102 (2018).

[41] A. D. Rodriguez, K. R. Dearstyne, and J. Cleland-Huang, Prompts matter: Insights and strategies for prompt engineering in automated software traceability, in *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)* (IEEE, 2023) pp. 455–464.

[42] T. Gupta, M. Zaki, N. A. Krishnan, and Mausam, Matscibert: A materials domain language model for text mining and information extraction, npj Computational Materials **8**, 102 (2022).

[43] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao, *et al.*, Interpretability of deep learning models: A survey of results, in *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation* (IEEE, 2017) pp. 1–6.

[44] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier (2016), arXiv:1602.04938.

[45] A. Inoue, Stabilization of metallic supercooled liquid and bulk amorphous alloys, Acta materialia **48**, 279 (2000).

[46] F. Ren, L. Ward, T. Williams, K. J. Laws, C. Wolverton, J. Hattrick-Simpers, and A. Mehta, Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments, Science advances **4**, eaaq1566 (2018).

[47] W. Johnson, J. Na, and M. Demetriou, Quantifying the origin of metallic glass formation, Nature communications **7**, 10313 (2016).

[48] W. Wang, Bulk metallic glasses with functional physical properties, Advanced Materials **21**, 4524 (2009).

[49] A. Dawid and Y. LeCun, Introduction to latent variable energy-based models: A path towards autonomous machine intelligence (2023), arXiv:2306.02572.

[50] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, Measuring massive multitask language understanding (2021), arXiv:2009.03300.

[51] I. A. Joiner, Chapter 1 - artificial intelligence: Ai is nearby, in *Emerging Library Technologies*, Chandos Information Professional Series, edited by I. A. Joiner (Chandos Publishing, 2018) pp. 1–22.

[52] Z. Niu, G. Zhong, and H. Yu, A review on the attention mechanism of deep learning, Neurocomputing **452**, 48 (2021).

[53] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, Chain-of-thought prompting elicits reasoning in large language models (2023), arXiv:2201.11903.

[54] P. P. Ray, Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope, Internet of Things and Cyber-Physical Systems **3**, 121 (2023).

[55] J. M. Helm, A. M. Swiergosz, H. S. Haeberle, J. M. Karnuta, J. L. Schaffer, V. E. Krebs, A. I. Spitzer, and P. N. Ramkumar, Machine learning and artificial intelligence: definitions, applications, and future directions, Current reviews in musculoskeletal medicine **13**, 69 (2020).

[56] F. Almeida and G. Xexeo, Word embeddings: A survey (2023), arXiv:1901.09069.

[57] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*, 1st ed. (O'Reilly Media, Inc., 2018).

[58] D. Svozil, V. Kvasnicka, and J. Pospichal, Introduction to multi-layer feed-forward neural networks, Chemometrics and intelligent laboratory systems **39**, 43 (1997).

[59] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, ACM Comput. Surv. **53**, 1 (2020).

[60] A. Natekin and A. Knoll, Gradient boosting machines, a tutorial, Frontiers in neurorobotics **7**, 21 (2013).

[61] L. Yang and A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing **415**, 295 (2020).
[62] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, A survey on in-context learning (2023), arXiv:2301.00234.
[63] J. Gou, B. Yu, S. J. Maybank, and D. Tao, Knowledge distillation: A survey, International Journal of Computer Vision **129**, 1789 (2021).
[64] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, Language models are few-shot learners (2020), arXiv:2005.14165.
[65] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi, Siren's song in the ai ocean: A survey on hallucination in large language models (2023), arXiv:2309.01219.
[66] S. Menard, *Applied logistic regression analysis*, 106 (Sage, 2002).
[67] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, J.-R. Wen, J. Yuan, W. X. Zhao, and J. Zhu, Pre-trained models: Past, present and future (2021), arXiv:2106.07139.
[68] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, Self-instruct: Aligning language models with self-generated instructions (2023), arXiv:2212.10560.
[69] L. Antonelli, M. R. Guarracino, L. Maddalena, and M. Sangiovanni, Integrating imaging and omics data: a review, Biomedical Signal Processing and Control **52**, 264 (2019).
[70] S. Suthaharan, Support vector machine, in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning* (Springer US, Boston, MA, 2016) pp. 207–235.
[71] T. Kudo and J. Richardson, Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing (2018).
[72] S. J. Pan and Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering **22**, 1345 (2009).
[73] J. Tal, Chapter 14 - sample size, in *Strategy and Statistics in Clinical Trials*, edited by J. Tal (Academic Press, Boston, 2011) pp. 229–244.
[74] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Computing Surveys **55**, 1 (2023).
[75] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, React: Synergizing reasoning and acting in language models (2023), arXiv:2210.03629.
[76] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kttler, M. Lewis, W. tau Yih, T. Rocktaschel, S. Riedel, and D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks (2021), arXiv:2005.11401.
[77] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need (2023), arXiv:1706.03762.
[78] I. Beltagy, M. E. Peters, and A. Cohan, Longformer: The long-document transformer (2020), arXiv:2004.05150.
[79] cohere docs, Top-k and top-p, `https://docs.cohere.com/docs/controlling-generation-with-top-k-top-p` (accessed on January 29, 2024).
[80] cohere docs, Temperature, `https://docs.cohere.com/docs/temperature` (accessed on January 29, 2024).

Acknowledgments

This work is supported by Research Grants Council, Hong Kong SAR through the Collaborative Research Fund (C1005-19G) and General Research Fund (17210723). T.W. acknowledges additional support by The University of Hong Kong (HKU) via seed funds (2201100392, 2309100163). S.P. acknowledges additional support by HKU via seed fund (2309100201). We acknowledge helpful discussions with Prof. Yi Ma at HKU and Simon Zhai at UC Berkeley.

Competing interests

The authors declare no competing interests.

Supplementary Information

Supplementary Notes
Supplementary Figures S1-15
Supplementary Tables S1-2
References (51-80)

Supplementary Information for

A Prompt-Engineered Large Language Model, Deep Learning Workflow for Materials Classification

Siyu Liu et al.

1 Supplementary Notes

1.1 Glossary

- **Artificial Intelligence**: artificial intelligence (AI) refers to the development of computer systems capable of performing tasks that typically require human intelligence, including visual perception, decision-making, and language comprehension [51].
- **Attention**: a mechanism enabling models to focus on specific parts of the input sequence during processing, allowing them to weigh the significance of different words when generating output. This attention mechanism assists in capturing long-range dependencies and enhancing performance in various natural language processing (NLP) tasks [52].
- **BERT**: Bidirectional Encoder Representations from Transformers, a widely used pre-trained language model [27].
- **Chain-of-thoughts**: a prompt engineering method that enhances complex reasoning capabilities by adding intermediate reasoning steps to the text input for the model [53].
- **ChatGPT**: Chat Generative Pre-trained Transformer, developed by OpenAI, is a chatbot and a sibling model to InstructGPT, trained to follow instructions in a prompt and deliver detailed responses [54].
- **Cross Entropy Loss**: a performance indicator measuring classification models, also known as log loss, with lower values indicating better predictive performance.
- **Deep Learning**: deep learning (DL) is an AI technology emulating the neural network structure of the human brain, learning and training via multi-level neurons for complex data analysis and processing [55].
- **Embedding**: an NLP technique that maps a high-dimensional space, with a dimensionality equal to the total number of words, to a much lower-dimensional continuous vector space, assigning each word or phrase a vector in the real number domain [56].
- **Feature Engineering**: the process of selecting, transforming, and creating new features from raw data to improve machine leanring (ML) model performance. It involves identification of relevant and informative input variables and their preparation for accurate predictions or classifications [57].
- **Feed Forward**: refers to a deep learning framework, also known as a multi-layer perceptron (MLP), that defines a mapping $y = f(x : \theta)$, and learns the value of parameter θ to achieve the best function approximation [58].
- **Few-shot Learning**: a ML approach where models are trained to make accurate predictions with limited examples per class [59].
- **Gemini**: Google's multimodal large language model (LLM), the first to outperform human experts on MMLU (Massive Multitask Language Understanding), a popular method for testing AI models' knowledge and problem-solving abilities.
- **Gradient Boosting Decision Tree (GBDT)**: also known as Gradient Boosting Machines (GBM), is a popular ensemble learning method combining decision trees with gradient boosting. GBDT continuously fits new models for more accurate estimates of response variables. The principle behind this algorithm is to construct new base learners that maximizes correlation with the negative gradient of the ensemble's loss function [60].
- **Hyperparameters**: external configuration variables input in advance to manage the training of ML models [61].
- **In-context Learning**: refers to enhancing LLM performance using a few examples provided in the input context [62].
- **ImageNet**: an image database organized according to the WordNet hierarchy (currently only nouns), with each hierarchy node represented by hundreds of thousands of images [29].
- **Knowledge Distillation**: refers to extracting knowledge from larger deep neural networks into smaller networks [63].
- **Large Language Model**: large language model (LLM) is a kind of DL model trained on large amounts of text data to learn statistical patterns of natural language [64].
- **LLM Hallucination**: occurs when LLMs generate content deviating from user input, contradicting previously generated context, or inconsistent with established world knowledge [65].

- **Logistic Regression**: a supervised learning algorithm utilizing logistic functions to estimate label probabilities [66].
- **Machine Learning**: machine learning (ML) demonstrates the experiential 'learning' associated with human intelligence, along with the ability to enhance its analyses through using computational algorithms [55].
- **Multi-head Attention**: represents multiple attention modules within a single attention layer of the model, allowing for different focus on various parts of a sequence.
- **Pre-trained Model**: a model trained on a large corpus of data and can be fine-tuned to solve various tasks [67].
- **Prompt Engineering**: a method to generate textual data from LLMs by embedding task descriptions in the input, effectively conveying specific parameters to the model as part of a problem statement [54].
- **Self-instruct Learning**: a method for improving the instruction-following capabilities of pre-trained language models by bootstrapping off their own generations [68].
- **Supervised Classification**: a ML task where the goal is to categorize input data into predefined classes or categories based on labeled training examples [69].
- **Support Vector Machine**: a supervised learning algorithm for classification and regression tasks that identifies the optimal boundary (hyperplane) separating data points of different classes [70].
- **Tokenization**: the process of converting text into smaller structural markers, called tokens. The tool used to handle this process is known as a tokenizer [71].
- **Transfer Learning**: a ML technique employing a pre-trained model as the starting point for a new related task, instead of training a model from scratch. It allows for different domains, tasks, and distributions in training and testing [72].

1.2 Workflow details

1.2.1 Data processing of metallic glasses dataset

As mentioned in the "Methods" section, our initial dataset consists of 5,577 samples. To train different BERT models, it is necessary to divide it into a training and a test set. We employed a stratified sampling method for this purpose, ensuring a representative and balanced representation of the different subgroups [73], which enhances the quality and reliability of sample. Based on the overall 80:20 split ratio for the training and test sets, we have performed non-repetitive stratified sampling on three alloy data categories. In the end, we obtained a training set with 4,460 samples and a test set with 1,117 samples, maintaining the 80:20 ratio for each category.

It should be noted that this section aims to establish the relationship between alloy composition and labels. The textual data generated through Gemini-pro is used to train different BERT classification models. The data obtained in this section is stored in the "original data" folder, with the data format shown in Table S1.

1.2.2 Prompt design and textual data generation

Upon acquiring the alloy composition data, we proceeded with prompt design. Prompt engineering is a prominent direction in the field of LLMs and is considered a method that significantly improve data generation effectiveness for LLMs [74]. Techniques such as few-shot learning [53], CoT [53], synergizing reasoning and acting (ReAct) [75], and retrieval-augmented generation (RAG) [76] are deemed effective prompt engineering methods. Due to the additional knowledge sources required for the latter two methods, we tested direct inquiry, few-shot, and CoT, and compared them with our MetalPrompt. Figs. S8, S9, and S10 are templates for three benchmark methods, while Fig. S4 shows the comparison results. Our "MetalPrompt" achieved the best performance.

For the prompt inquiry setting with the LLM, we used Top-K = 1 and Temperature = 0 as default parameters. As shown in Fig. S11 and Fig. S12, a smaller Top-K and lower temperature indicate more reliable model output. Top-K = 1 implies that the model will perform greedy decoding and select the most probable value. Unless specifically explained, the parameters of the language model mentioned below are consistent with these settings.

Using the above parameter settings and "MetalPrompt" as the input template, we generated textual data for 5,577 alloy compositions. The replacement of alloy elements in the template was assisted by the langchain library. Simultaneously, based on the alloy composition category relationship of the original training and test dataset, we labeled the generated textual data with

the same category. This dataset is used for training the classification model below, with the partitioning of training and testing sets consistent with the original data.

1.2.3 MgBERT training and evaluation

BERT model and attention mechanism: BERT, introduced by Google researchers in 2018 [27], is a pre-training language representation method that significantly impacted the field of NLP. Traditional language models analyzed text data unidirectionally, either from left to right or right to left, limiting their understanding of language context. BERT, employing the Transformer architecture, processes each word concerning all other words in a sentence, rather than sequentially. Owing to its state-of-the-art results in numerous tasks, BERT was selected as the foundation for our classification model.

Fig. 4c in the main text shows the basic BERT architecture, comprising data pre-processing, input encoding, model training, post-processing, and output results. During data processing, tokenizers divide sentences or words into individual tokens, such as marking "for example" as "for" and "example". Then, as demonstrated in Fig. S13, the encoding method transforms pre-processed text data into the input representation of the model. For instance, "for" is converted to a vector $[00..1..00]$. Model training is crucial to BERT, as it uses multi-head attention to extract essential features from input embeddings. Attention mechanisms enhance model performance in NLP and other sequence data processing tasks by focusing on the most relevant parts of the input sequence. Models can learn to assign various attention weights to input information from different positions, concentrating on crucial aspects when processing input sequences. This mechanism allows the model to capture long-distance dependencies and important patterns in sequences more effectively, thus improving its performance in processing sequence data [77]. Fig. S14 shows a schematic diagram of the operation of the attention mechanism. Multi-head attention, compared to a single module, employs multiple attention modules in the same calculation, extracting different feature information. The calculation formula for attention score is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{S1}$$

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{S2}$$

Where $head_i = Attention(QWi^Q, KWi^K, VWi^V)$.

In Equation S1, $\sqrt{d_k}$ refers to the queries and dimension keys. Q, K, and V refer to the input query, key, and value, respectively. $softmax$ refers to the softmax function.

Pre-trained models selection: We compared three pre-trained models based on BERT, including the basic BERT model [27], Longformer supporting longer input text lengths [78], and MatSciBERT trained on materials science texts [42].

All models employ default parameter settings; however, to expedite calculations, the maximum input length for Longformer is limited to 1200 tokens, while other models have a default maximum length of 512. To ensure reproducibility, the random seed is fixed at 42. All models are trained using cross-entropy loss and optimized with the Adam optimizer for parameters. All training was conducted on a 32G Nvidia V100. Table S2 presents the basic hyperparameters for model training. Due to its larger embedding layer requiring more graphics memory, Longformer has a slightly smaller batch size.

MgBERT embedding and training: Upon the training of MgBERT, we found that the file with the longest tokens is 'Fe68.3C6.9Si2.5B6.7P8.8Cr2.2Mo2.5Al2.1.txt' with 859 tokens, and 300 files have more than 512 tokens. Our basic model, MatSciBERT, only supports inputs with a maximum token length of 512. Considering the potential inconsistency in text input length for different materials and scalability requirements, we designed a variable-length MgBERT position embedding layer that does not necessitate complete retraining. The following Algorithm 1 is a pseudocode representation of its principle.

For the comparison of three embedding methods, we ran 16 epochs with a learning rate of 3e-5 and a batch size of 26. In Figs. 4g, h of the main text, only the first 15 epochs are displayed due to the loss output retaining the first three decimal places. In the 16th epoch, some models exhibit training set loss < 0.001. The complete training information can be found in the "train log" folder.

For the final MgBERT model, we continued training with the highest classification accuracy model weight of Method 3 at a learning rate of 2e-5 and compared it to the case with 1e-5. The highest accuracy of 88.5% was achieved when the learning rate was 2e-5 and the epoch was 4. We utilized the model weights trained under these parameters as the final MgBERT model.

Model evaluation: During the model evaluation stage, we compared some ML models, with their concepts outlined in the "Glossary" section. Model implementation is based on the "scikit-learn" software and employs its default hyperparameters for training. The random seed is fixed at 42 for reproducibility. The training and test set rely on stratified sampling, as described in the "Methods" section of the main text.

Algorithm 1 MgBERT Position Embedding

Input: New maximum position embedding n, Current position embeddings E
Output: Resized position embeddings E_{new}
Set $E_{new} = \text{Embedding}(n, E.size(1))$
Set $l = E.shape[0]$
Set $L = \lceil n/l \rceil$
if $l > n$ **then**
    Set $E_{new}.weight = E[: n].detach()$
else
    for $i \in [0, L-1]$ **do**
        if $i == 0$ **then**
            Set $E_{new}.weight[: l] = E.detach()$
        else if $i > 0$ and $i < L-1$ **then**
            Set $E_{new}.weight[i * l : (i+1) * l] = E.detach()$
        else
            Set $E_{new}.weight[i * l :] = E.detach()[: (n - i * l)]$
        end if
    end for
end if
return E_{new}

1.2.4 Model interpretability

We used the LIME model [44] for model interpretation. LIME utilizes a trained local surrogate model to explain individual predictions. For a black-box model of interest, LIME samples around the instance of interest and collects perturbed instances along with their black-box model predictions. A local interpretable model (e.g. linear regression, decision tree) is then trained on this new dataset to approximate the behavior of the black-box model locally. Fig. S15 from [44] shows its working principle, wherein the dashed line is a trained linear model used to explain the black-box model with complex boundaries in the red and blue part of the graph. The hyperparameters we set here is to perform 1,000 samples and filter the top 50 contributing features. Considering the maximum token length does not exceed 900, this choice is reasonable under the limitation of computational rate, as the training time increases exponentially with the increase of sampling times.

Additionally, we visualized the attention score of the [CLS] (classify) token in the last layer. Different attention scores represent the degree of importance that a specific token places on other tokens, with larger attention scores indicating higher importance. The classifier processes the value of the last layer's [CLS] token, compressing this 768-dimensional vector into three dimensions. Each dimension represents the probability of different categories. Therefore, calculating the focus on the previous layer assists in interpreting the model.

2 Supplementary figures

Output without MetalPrompt

Gemini-pro

Ag20Al25La55 is a ternary alloy composed of silver (Ag), aluminum (Al), and lanthanum (La) with a nominal composition of 20 atomic percent silver, 25 atomic percent aluminum, and 55 atomic percent lanthanum. This alloy exhibits a unique combination of properties, including high strength, good ductility, and excellent corrosion resistance.
The high strength of Ag20Al25La55 is attributed to the formation of a stable intermetallic compound, Ag2Al, which acts as a reinforcing phase within the alloy. This intermetallic compound is characterized by a high melting point and a strong atomic bonding, resulting in a material with exceptional strength and hardness. Additionally, the presence of lanthanum contributes to the alloy's strength by forming a solid solution with silver and aluminum, further enhancing the material's mechanical properties.
The good ductility of Ag20Al25La55 is a result of the relatively low melting point of the alloy, which allows for easy deformation and shaping. The presence of lanthanum also contributes to the alloy's ductility by promoting grain refinement and preventing the formation of brittle intermetallic phases. This combination of strength and ductility makes Ag20Al25La55 a suitable material for applications requiring both high mechanical strength and the ability to withstand deformation without fracture.
The excellent corrosion resistance of Ag20Al25La55 is attributed to the formation of a protective oxide layer on the surface of the alloy. This oxide layer is composed primarily of aluminum oxide (Al2O3), which is highly stable and resistant to corrosion. Additionally, the presence of silver and lanthanum further enhances the alloy's corrosion resistance by forming a passive layer that inhibits the formation of corrosion products. This combination of factors makes Ag20Al25La55 a suitable material for applications in harsh environments where corrosion resistance is critical.
Overall, Ag20Al25La55 is a versatile alloy with a unique combination of properties, including high strength, good ductility, and excellent corrosion resistance. These properties make it a suitable material for a wide range of applications, including aerospace, automotive, and medical industries.

FIG. S1. Gemini-pro outputs without using “MetalPrompt”.

Output without MetalPrompt

GPT4-turbo

The alloy composition denoted by Ag20Al25La55 is a ternary alloy, meaning it is composed of three different elements. The composition is given in atomic percentages, which represent the proportion of each element in the alloy in terms of the number of atoms, not by weight.

•Ag (Silver): 20% of the atoms in the alloy are silver. Silver is a precious metal known for its high electrical and thermal conductivity, as well as its reflective properties and resistance to corrosion.

•Al (Aluminum): 25% of the atoms in the alloy are aluminum. Aluminum is a lightweight metal with good corrosion resistance, high thermal and electrical conductivity, and is known for its ability to form hard, durable oxides.

•La (Lanthanum): 55% of the atoms in the alloy are lanthanum. Lanthanum is a rare earth element that is soft and malleable. It has a relatively high melting point and is used to improve the malleability of other metals and to create specialized alloys.

The specific properties of the Ag20Al25La55 alloy would depend on the interaction between these three elements when combined. The high percentage of lanthanum suggests that the alloy may have characteristics that are significantly influenced by the properties of rare earth metals, such as improved strength at high temperatures, resistance to oxidation, or unique magnetic properties.

This type of alloy could be of interest for specialized applications where the combined properties of silver, aluminum, and lanthanum are beneficial, such as in high-performance materials for aerospace, electronics, or advanced manufacturing technologies. However, the actual performance characteristics would need to be determined through experimental testing and analysis, as the behavior of alloys can be complex and not always predictable from the properties of the constituent elements alone.

FIG. S2. GPT4-turbo outputs without using “MetalPrompt”.

Output without MetalPrompt

llama2-70b-steerlm-chat

Ag20Al25La55 is an alloy composition consisting of silver (Ag), aluminum (Al), and lanthanum (La). The composition is expressed in weight percentages, with Ag20 representing 20% silver, Al25 representing 25% aluminum, and La55 representing 55% lanthanum.

This alloy composition is known for its high thermal conductivity, low coefficient of thermal expansion, and good mechanical properties. It is often used in applications that require a material with good thermal management properties, such as in electronic devices, heat exchangers, and solar cells.

The addition of lanthanum to the alloy composition can improve its mechanical properties, including its strength and ductility. It can also reduce the alloy's coefficient of thermal expansion, making it more dimensionally stable at high temperatures.

Aluminum is a common additive to alloy compositions due to its lightweight and corrosion-resistant properties. It can also improve the alloy's mechanical properties, including its strength and ductility.

Silver is known for its high electrical and thermal conductivity, as well as its corrosion-resistant properties. It is often used in electronic applications, as well as in jewelry and other decorative items.

Overall, Ag20Al25La55 is a highly specialized alloy composition that is designed for use in applications that require a material with good thermal management properties and mechanical strength.

FIG. S3. llama2 outputs without using “MetalPrompt”.

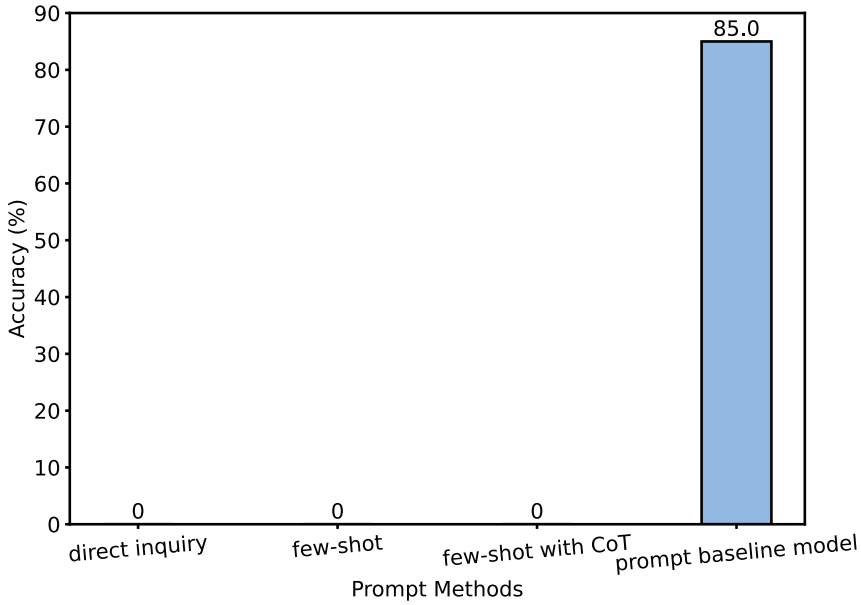

FIG. S4. Comparison of accuracy of different prompt methods. 'direct inquiry' refers to directly requesting LLM metallic glass classification results without using a prompt. 'few-shot' involves using alloy composition and its classification results as input samples in a prompt for an LLM and inquiring about the classification results for other specific alloy compositions. 'few-shot with CoT' not only adds examples of alloy composition and categories, but also integrates some thought processes into the prompt.

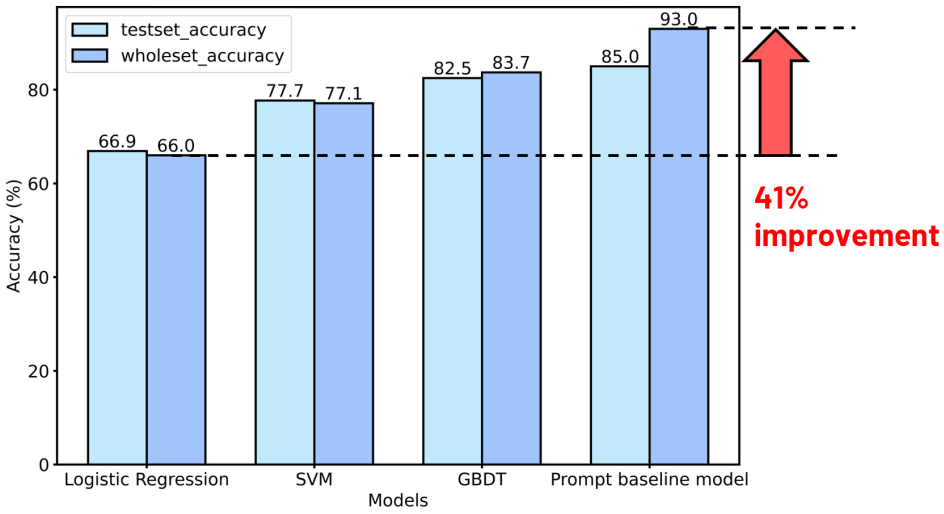

FIG. S5. Accuracy comparison between prompt baseline model and machine learning models. The logistic regression serves as our baseline model, SVM denotes the support vector machine model, and GBDT refers to the gradient boosting decision tree model.

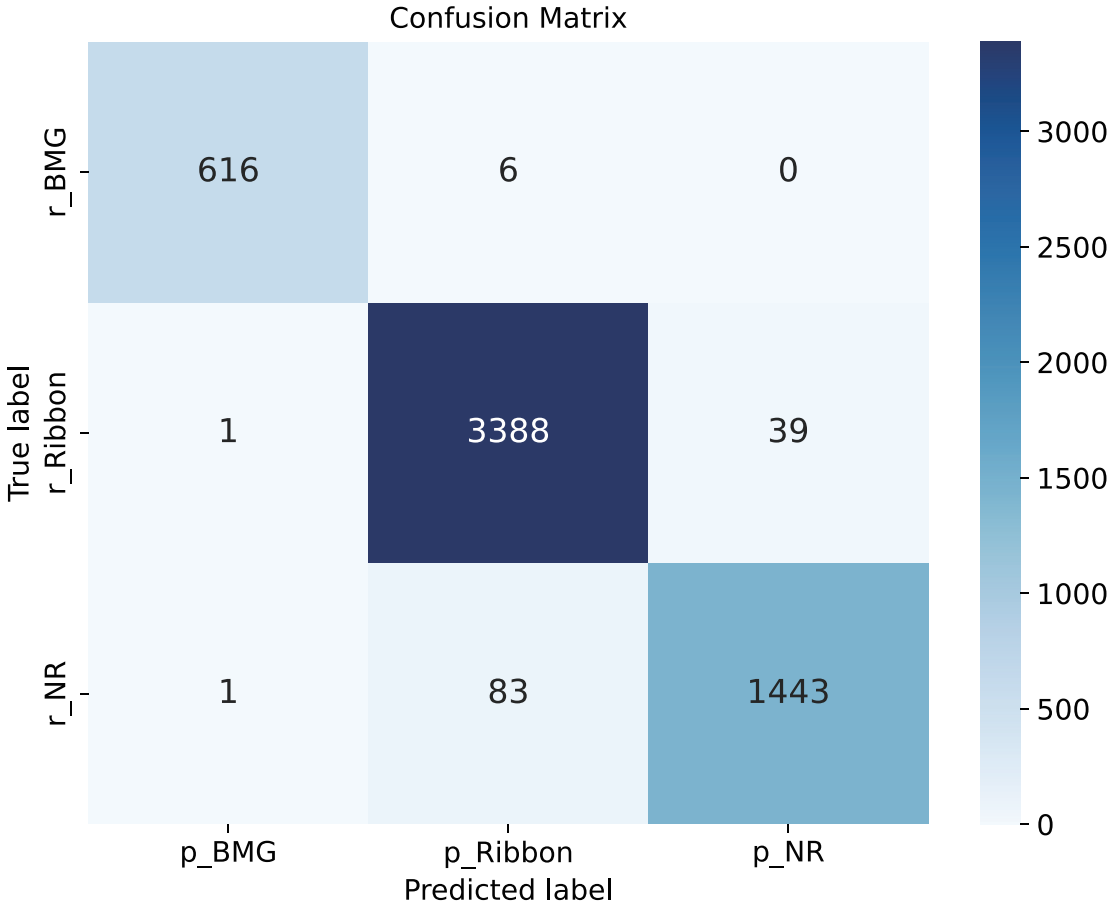

FIG. S6. Confusion matrix for MgBERT classification results. The x-axis refers the model prediction of the alloy composition belonging to a particular metallic glass category, while the y-axis indicates the actual category of the alloy composition. For example, the '616' in the upper left corner signifies that there are 616 components classified as metallic glasses, which MgBERT also predicts to be metallic glasses.

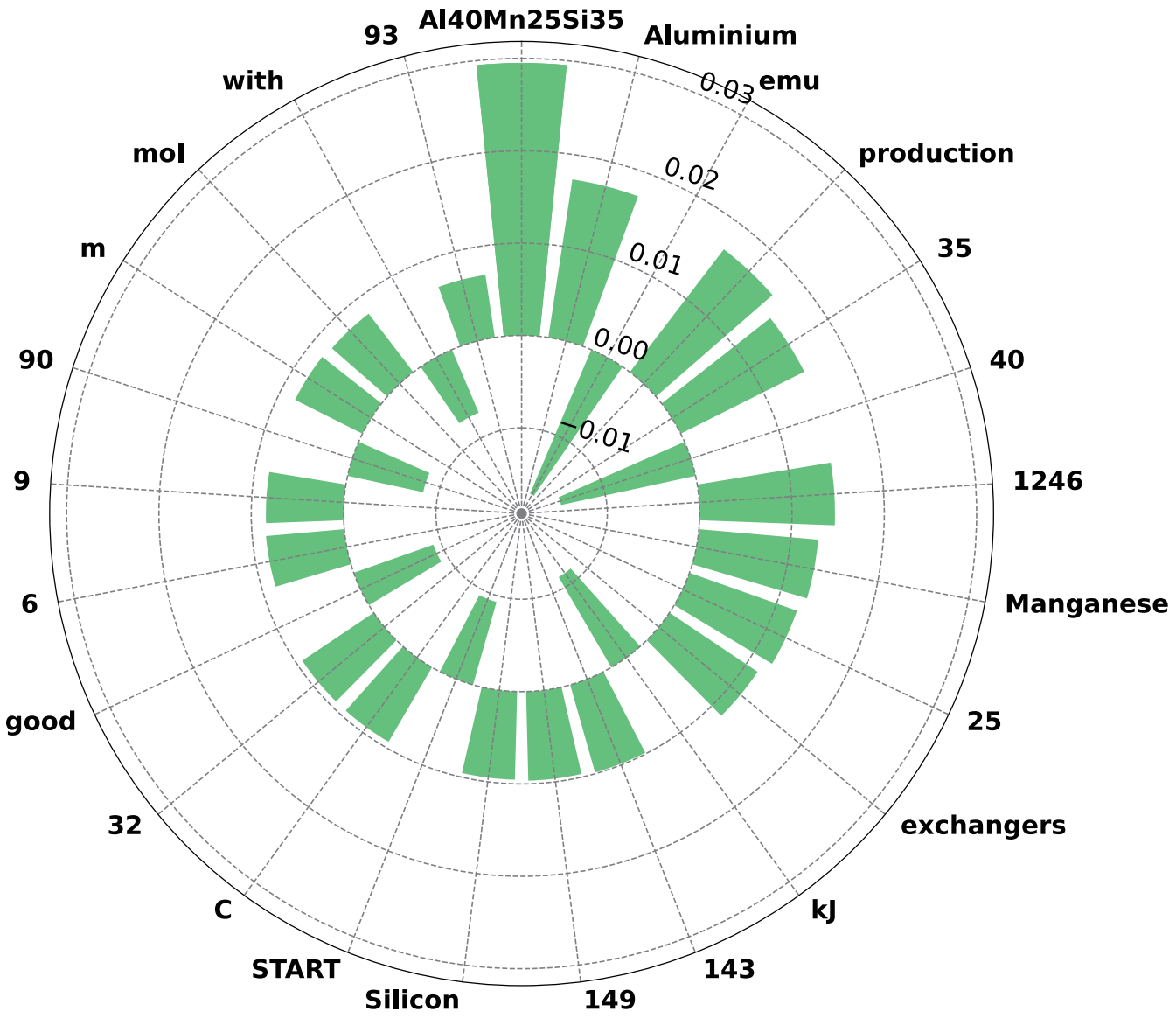

FIG. S7. Contribution of input text for non-ribbon type alloy $Al_{40}Mn_{25}Si_{35}$ to output classification results. A positive value means that the word has a positive contribution to the model's prediction that the component is non-ribbon, and vice versa.

Template1

direct inquiry

Instruction: "You are a materials scientist majoring in metallic glass. Metallic glasses (MG) (sometimes also referred to as glassy metals or, inappropriately, as amorphous metals) are noncrystalline materials composed of either pure metals or combinations of metals and metalloids. Rc is the critical cooling rate. The value of Rc is related to the composition, crystal structure, and properties of the alloy. A higher critical cooling rate means it is less likely to form bulk metallic glass(BMG). We categorise alloys as BMG form (Rc < 10^3 K/s), Ribbon form (Rc < 10^6 K/s), and Non-Ribbon(NR) form (Rc > 10^6 K/s). I want to know if an alloy is BMG type, Ribbon type, or non-ribbon type. Your answer can only be derived from ['BMG', 'Ribbon', 'NR']."

Question: What's the metallic glass formation category of alloy composition Al84Fe4Ni12?

You only need to give me the category from ['BMG', 'Ribbon', 'NR'] without answering the reasoning process. Please think with materials knowledge. If you don't know, you just say "I don't know".

FIG. S8. Prompt template for direct inquiry.

Template2

few-shot

Instruction: "You are a materials scientist majoring in metallic glass. Metallic glasses (MG) (sometimes also referred to as glassy metals or, inappropriately, as amorphous metals) are noncrystalline materials composed of either pure metals or combinations of metals and metalloids. Rc is the critical cooling rate. The value of Rc is related to the composition, crystal structure, and properties of the alloy. A higher critical cooling rate means it is less likely to form bulk metallic glass(BMG). We categorise alloys as BMG form (Rc < 10^3 K/s), Ribbon form (Rc < 10^6 K/s), and Non-Ribbon(NR) form (Rc > 10^6 K/s). I want to know if an alloy is BMG type, Ribbon type, or non-ribbon type. Your answer can only be derived from ['BMG', 'Ribbon', 'NR']."
Examples: "Here are some examples of classification of alloy composition and their metallic glass formation type.
Question: What's the metallic glass formation category of alloy composition Al86Co14Ni0?
Answer: NR
Question: What's the metallic glass formation category of alloy composition Co5Ni77P18?
Answer: Ribbon
Question: What's the metallic glass formation category of alloy composition Mg65Cu25Tb10?
Answer: BMG
Question: What's the metallic glass formation category of alloy composition Al84Fe4Ni12?

You only need to give me the category from ['BMG', 'Ribbon', 'NR'] without answering the reasoning process. Please think with materials knowledge. If you don't know, you just say "I don't know".

FIG. S9. Prompt template for few-shot method.

Template3

few-shot with CoT

Instruction: "You are a materials scientist majoring in metallic glass. Metallic glasses (MG) (sometimes also referred to as glassy metals or, inappropriately, as amorphous metals) are noncrystalline materials composed of either pure metals or combinations of metals and metalloids. Rc is the critical cooling rate. The value of Rc is related to the composition, crystal structure, and properties of the alloy. A higher critical cooling rate means it is less likely to form bulk metallic glass(BMG). We categorise alloys as BMG form (Rc < 10^3 K/s), Ribbon form (Rc < 10^6 K/s), and Non-Ribbon(NR) form (Rc > 10^6 K/s). I want to know if an alloy is BMG type, Ribbon type, or non-ribbon type. Alloys that comply with the Inoue rule are more likely to form BMG. The more inconsistent it is, the more likely it is to be of non-ribbon type. The Inoue rules that characterise an alloy to be a BMG former are: a) a composition close to a deep eutectic, b) atomic size difference of larger than 12%, c) a large negative heat of mixing among at least two constituent elements. Your answer can only be derived from ['BMG', 'Ribbon', 'NR']."
Examples: "Here are some examples of classification of alloy composition and their metallic glass formation type.
Question: What's the metallic glass formation category of alloy composition Al86Co14Ni0?
Answer: Al86Co14Ni0 contains 86% aluminum and 14% cobalt. Aluminium's radius is 118 pm and cobalt's radius is 116 pm. Based on these, we should calculate the liquid temperature of the alloy, liquid temperature reduction, Atomic size difference, Atomic size ratio, Atomic size range, Maximum heat of mixing, and Mean heat of mixing. Based on the above characteristic features, we can infer that Al86Co14Ni0 is an NR-type alloy.
Question: What's the metallic glass formation category of alloy composition Co5Ni77P18?
Answer: Co5Ni77P18 contains 5% cobalt, 77% nickel, and 18% phosphorus. Cobalt's radius is 116 pm, nickle's radius is 115 pm and phosphorus's radius is 110 pm. Based on these, we should calculate the liquid temperature of the alloy, liquid temperature reduction, Atomic size difference, Atomic size ratio, Atomic size range, Maximum heat of mixing, and Mean heat of mixing. Based on the above characteristic features, we can infer that Co5Ni77P18 is a Ribbon type alloy.
Question: What's the metallic glass formation category of alloy composition Mg65Cu25Tb10?
Answer: Mg65Cu25Tb10 contains 65% magnesium, 25% copper and 10% Terbium. Magnesium's radius is 136 pm, copper's radius is 117 pm and Terbium's radius is 178.2 pm. Based on these, we should calculate the liquid temperature of the alloy, liquid temperature reduction, Atomic size difference, Atomic size ratio, Atomic size range, Maximum heat of mixing, and Mean heat of mixing. Based on the above characteristic features, we can infer that Mg65Cu25Tb10 is a BMG-type alloy.
Question: What's the metallic glass formation category of alloy composition Al84Fe4Ni12?
You only need to give me the category from ['BMG', 'Ribbon', 'NR'] without answering the reasoning process. Please think with materials knowledge. If you don't know, you just say "I don't know".

FIG. S10. Prompt template for few-shot with CoT method.

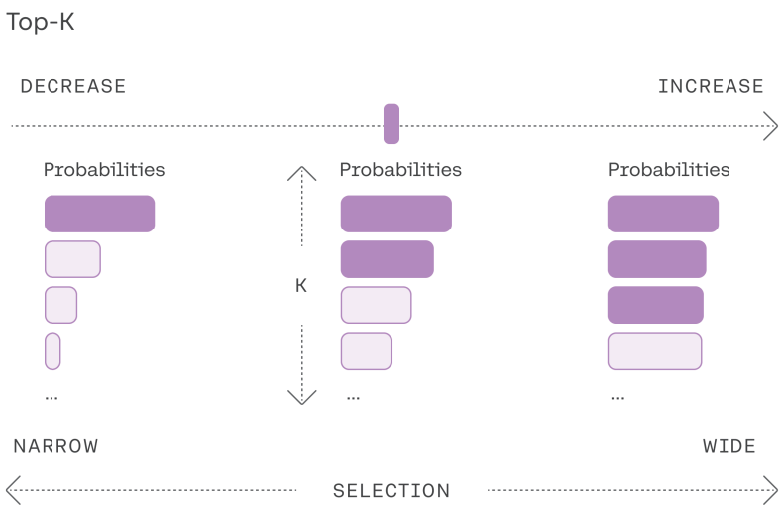

FIG. S11. Tuning the Top-K setting [79]. The value of K represents the model's selection of the first k words with the highest likelihood of being the output. Decreasing the hyperparameter "Top-K" results in a narrower model selection, focusing on candidate words with higher probabilities. This figure is from [79].

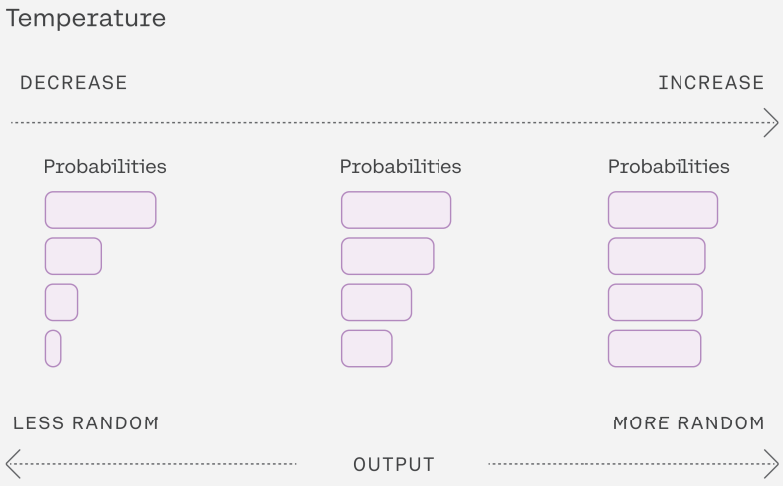

FIG. S12. Tuning the temperature setting [80]. Similar to the "Top-K", a lower value of the hyperparameter temperature implies that the model will favor words with higher probabilities as output. This figure is from [80].

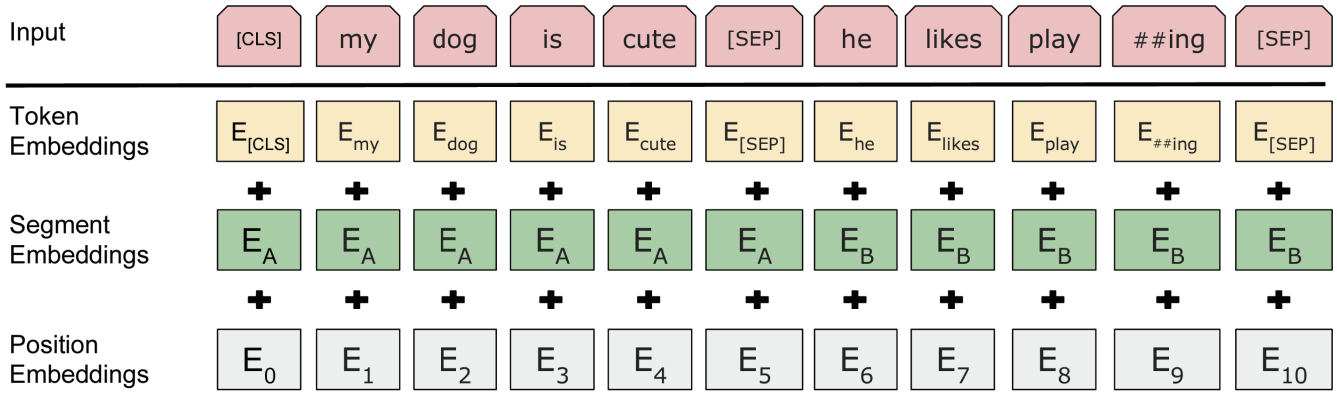

FIG. S13. BERT input representation. The input embeddings are the sum of the token embeddings, the segment embeddings and the position embeddings [27]. This figure is from [27].

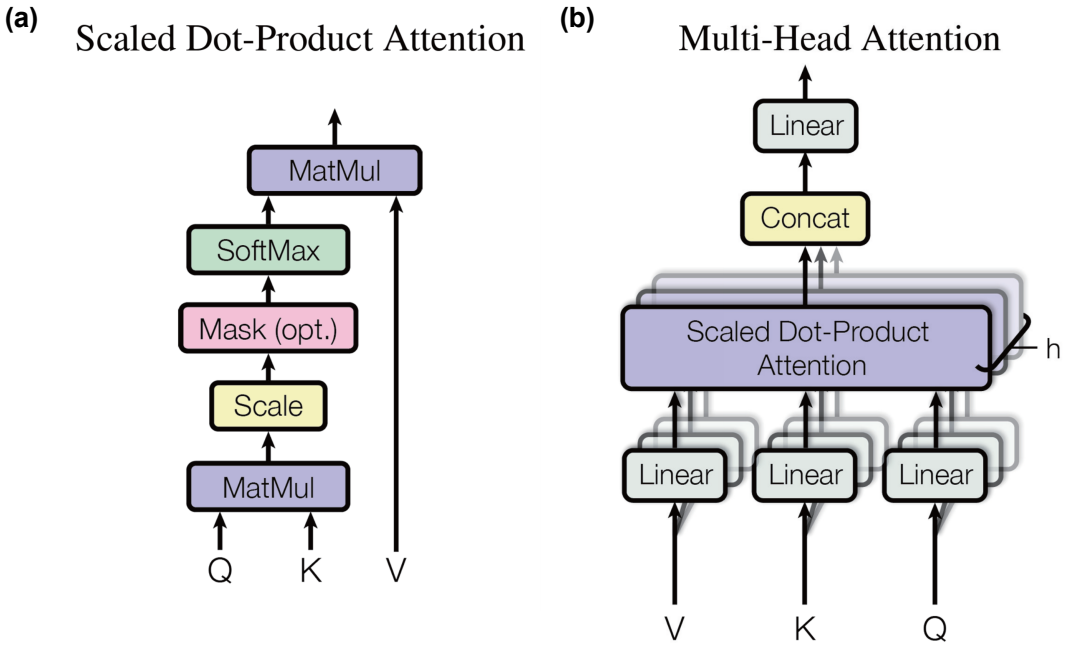

FIG. S14. Schematic diagram of attention mechanism. a, Scaled dot-product attention. b, Multi-head attention consists of multiple attention layers running in parallel [77]. This figure is from [77].

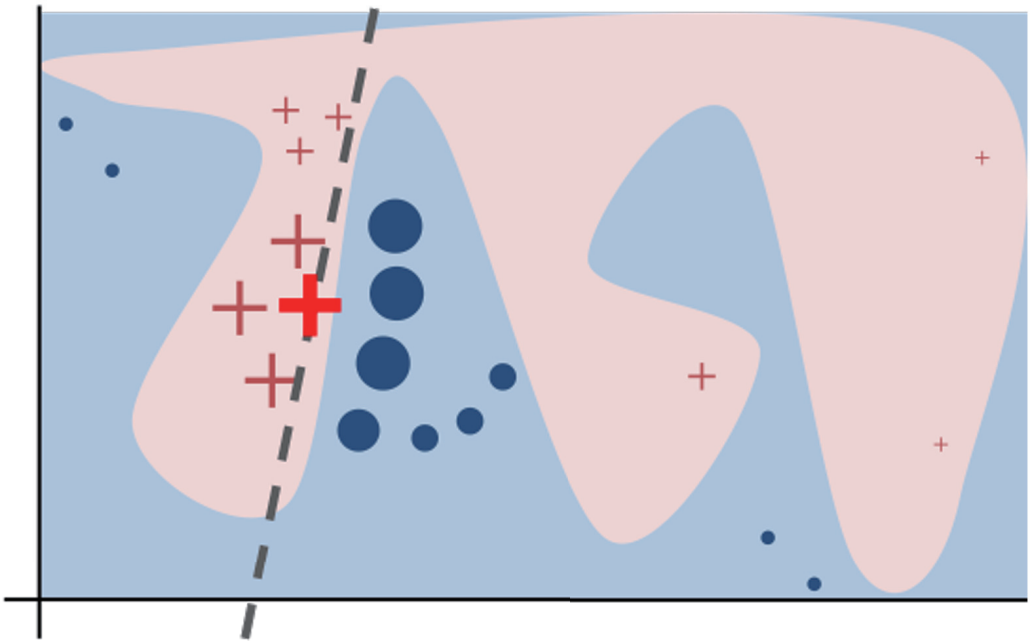

FIG. S15. A simple example of LIME [44]. The gray dashed line is a linear interpreter identified by LIME, used to distinguish the positive and negative contributions of sample features to the output of the black box model (i.e. our model). The symbols in the figure correspond to data points generated by LIME's perturbation sampling, while the red and blue areas denote the complex decision areas of the black box model. This figure is from [44].

3 Supplementary Tables

TABLE S1. Format of raw data.

glass forming category	composition
Ribbon	Ag20Al25La55
BMG	Cu55Zr42.5Ga2.5
NR	Ag6Ce8Cu86
...	...

TABLE S2. Hyperparameters of model training.

model name	hyperparameters
BERT	Epoch: 8; Learning rate: 3e-5; Batch size: 26
longformer	Epoch: 8; Learning rate: 3e-5; Batch size: 12
MatSciBERT	Epoch: 8; Learning rate: 3e-5; Batch size: 26