

## Full length article

## Steel design based on a large language model



Shaohan Tian <sup>a</sup>, Xue Jiang <sup>a,b,\*</sup> , Weiren Wang <sup>a</sup>, Zhihua Jing <sup>a</sup>, Chi Zhang <sup>a</sup>, Cheng Zhang <sup>a</sup>, Turab Lookman <sup>c,\*</sup> , Yanjing Su <sup>a,\*</sup>

<sup>a</sup> Beijing Advanced Innovation Center for Materials Genome Engineering, Institute for Advanced Materials and Technology, University of Science and Technology Beijing, Beijing 100083, China

<sup>b</sup> Liaoning Academy of Materials, Shenyang, Liaoning 110000, China

<sup>c</sup> AiMaterials Research LLC, Santa Fe, NM 87501, USA

## ARTICLE INFO

**Keywords:**

Property prediction  
Steel design  
Materials language model  
Deep learning  
Artificial intelligence

## ABSTRACT

The success of artificial intelligence (AI) in materials research heavily relies on the integrity of structured data and the construction of precise descriptors. In this study, we present an end-to-end pipeline from materials text to properties for steels based on a large language model. The objective is to enable quantitative predictions of properties with high-accuracy and explore new steels. The pipeline includes a materials language encoder, named SteelBERT, and a multimodal deep learning framework that maps the composition and text sequence of complex fabrication processes to mechanical properties. We demonstrate high accuracy on mechanical properties, including yield strength (YS), ultimate tensile strength (UTS), and elongation (EL) by predicting determination coefficients ( $R^2$ ) reaching 78.17 % ( $\pm 3.40\%$ ), 82.56 % ( $\pm 1.96\%$ ), and 81.44 % ( $\pm 2.98\%$ ) respectively. Further, through an additional fine-tuning strategy for the design of specific steels with small datasets, we show how the performance can be refined. With only 64 experimental samples of 15Cr austenitic stainless steels, we obtain an optimized model with  $R^2$  of 89.85 % ( $\pm 6.17\%$ ), 88.34 % ( $\pm 5.95\%$ ) and 87.24 % ( $\pm 5.15\%$ ) for YS, UTS and EL, that requires the user to input composition and text sequence for processing and which outputs mechanical properties. The model efficiently optimizes the text sequence for the fabrication process by suggesting a secondary round of cold rolling and tempering to yield an exceptional YS of 960 MPa, UTS of 1138 MPa, and EL of 32.5 %, exceeding those of reported 15Cr austenitic stainless steels.

## 1. Introduction

Artificial intelligence (AI) and machine learning (ML) methods have for a number of years been applied in materials science to accelerate materials discovery and optimization [1–7]. However, their success heavily relies on the integrity of high-quality structured data and precise feature engineering, placing strong demands on researchers' expert knowledge. In recent years, large language models (LLMs), such as ChatGPT [8], Falcon [9], and BERT [10], have demonstrated their general "intelligence" capabilities via large-scale data, vast neural networks, self-supervised learning and powerful hardware [11–14]. The Transformer architecture, characterized by the attention mechanism [15], is the fundamental building block of LLMs [16–18] and has been employed to solve many problems in including natural language processing (NLP) [19–24], code generation [25–28] as well as in automation of materials research [29,30]. Zheng et al. [31] presented a framework using prompt engineering to guide ChatGPT to mine

synthesis conditions of metal-organic frameworks from the scientific literature, resulting in a code-free tool for chemical data extraction and analysis. Boiko et al. [32] developed an artificial intelligence system driven by GPT-4 that autonomously designs, plans and performs complex chemical synthesis experiments of organic compounds. These initial successes show the potential of LLMs to transform conventional ML-driven materials research by directly harnessing historical text knowledge. However, the application of LLMs to alloys has never been achieved, specifically, a capability to accurately predict properties and explore new metallic systems utilizing natural language.

Steel, as one of the most important alloys in today's society, has a rich research history that has matured industrially to incorporate highly complex compositional and production processes [33]. The manufacturing of steel involves a series of processes, such as smelting, solidification, multi-pass deformation, and heat treatment protocols [34–37]. The design of steel not only requires a comprehensive understanding of physics, chemistry, and metallurgy, but the extensive body

\* Corresponding authors.

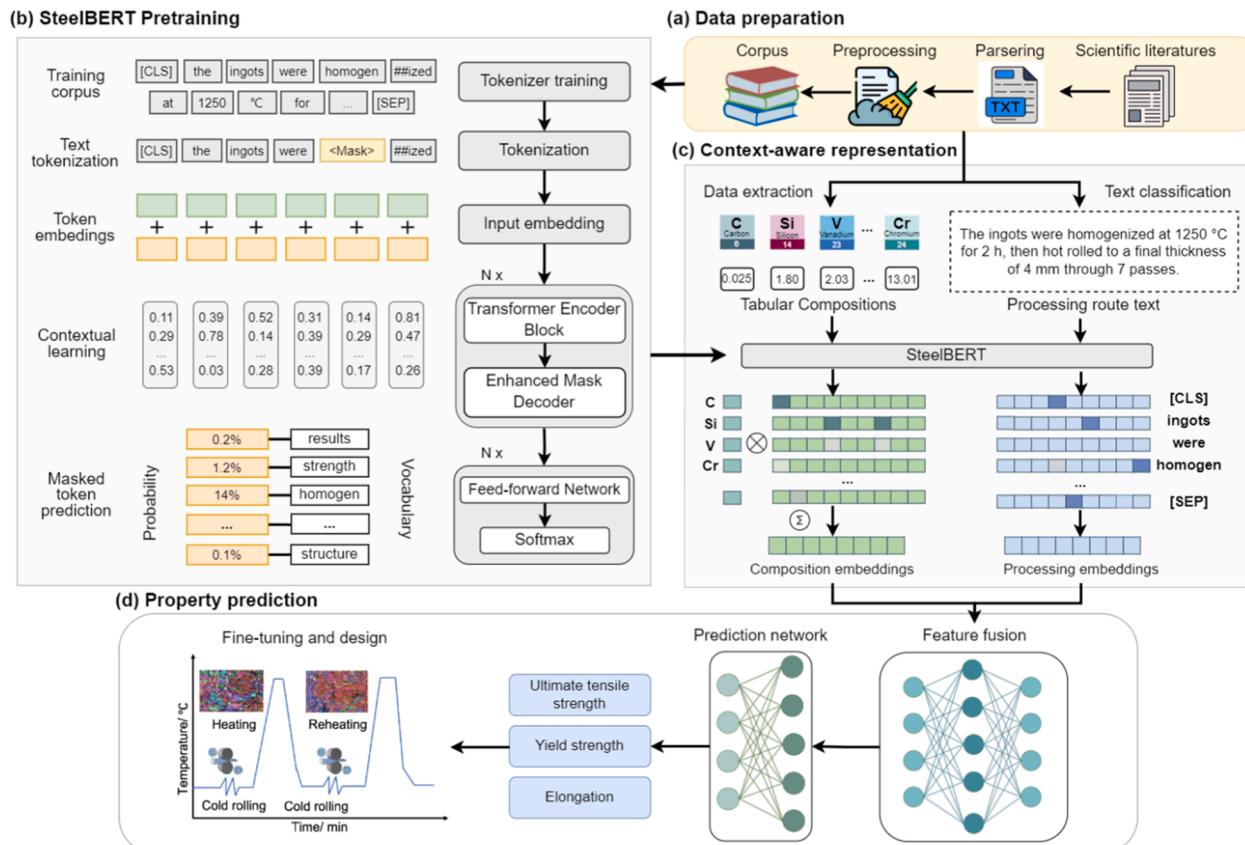
E-mail addresses: [jiangxue@ustb.edu.cn](mailto:jiangxue@ustb.edu.cn) (X. Jiang), [turablookman@gmail.com](mailto:turablookman@gmail.com) (T. Lookman), [yjsu@ustb.edu.cn](mailto:yjsu@ustb.edu.cn) (Y. Su).

of published literature poses challenges for data and knowledge processing. The multidimensional search space of candidates includes at least  $10^{20}$  routes (Fig. S1). For example, quenching and inter-critical tempering (Q&T) or quenching and partitioning (Q&P) heat treatments are usually used to obtain a microstructure of soft austenite phase and hard martensite phase beneficial for controlling the strength and toughness of steels. In addition, the complex phase transformations also ultimately influence mechanical properties and corrosion resistance.

In current ML-driven steel research, composition and processing routes are typically transformed into tabular data before feature selection and model training [38–40]. To develop new processing routes for steels, the use of tabular-type process feature representation becomes intractable due to the generation of a high-dimensional sparse feature matrix. Therefore, the ability to harness natural language to describe diverse sequences of processing routes is increasingly essential, and the integration of steel research with LLMs presents a promising approach to address this challenge.

In this study, we present an end-to-end pipeline from materials text to properties based on an LLM, enabling high-accuracy quantitative predictions of mechanism properties and the means to explore new steels. Firstly, our approach is to build an LLM specifically tailored for the domain of steels, which we refer to as SteelBERT (Fig. 1a). It is pretrained on a comprehensive corpus comprising 4.2 million abstracts related to materials science and 55,000 full-text articles on steels with the assistance of a disentangled attention mechanism, SteelBERT arrives

at an accurate representation of steel knowledge to enable property prediction in the form of natural language as input. Using SteelBERT and multimodal deep learning, we build an end-to-end pipeline (Fig. 1b). This pipeline utilizes the context-aware vector representation of composition and text sequence of complex fabrication processes generated by SteelBERT to reserve the diverse existing information in processing routes. It successfully predicts the mechanical properties of 18 new steels reported in 2022 and 2023, with coefficient of determination ( $R^2$ ) values 78.17% ( $\pm 3.40\%$ ), 82.56% ( $\pm 1.96\%$ ), and 81.44% ( $\pm 2.98\%$ ) for yield strength (YS), ultimate tensile strength (UTS), and elongation (EL), respectively. We further fine-tune the model using a small laboratory dataset to refine the performance on specific steels (Fig. 1c). With only 64 experimental samples of austenitic stainless steels (ASSs), we obtain an optimized model with  $R^2$  of 89.85% ( $\pm 6.17\%$ ), 88.34% ( $\pm 5.95\%$ ) and 87.24% ( $\pm 5.15\%$ ) for YS, UTS and EL. It then efficiently searches the large candidate space for the fabrication processes with highest mechanical properties by suggesting a secondary round of cold rolling and tempering. The resulting steel exhibits outstanding YS of 960 MPa, UTS of 1138 MPa and EL of 32.5%, the best performance that exceeds those of reported 15Cr ASSs.



**Fig. 1.** Quantitative mechanical properties prediction with SteelBERT. (a) Schematic workflow of data preparation. This workflow encompasses several stages of scientific documents download, corpus preprocessing, table and text information extraction. A corpus of scientific articles is scraped and preprocessed for pretraining SteelBERT. The composition and property information are also extracted for further quantitative mechanical properties prediction. (b) SteelBERT is a linguist specialized in the language of steel materials. After tokenization, the training corpus is passed to the DeBERTa model. Each of the 12 Transformer encoders consists of 12 attention heads. The final dense layer with a softmax activation function identifies the masked tokens. (c) Context-aware representation of steel information. We use SteelBERT to generate embeddings with 768 dimensions for textual processing routes and chemical compositions. (d) A schematic of the approach to predict mechanical properties using natural language. We build a predictive model using a deep learning network. It takes embedded composition and text of processing routes as inputs to predict the yield strength, ultimate tensile strength, and total elongation. After fine-tuning on our laboratory dataset, a novel advanced austenitic stainless steel is optimized with outstanding performance.

## 2. Methods

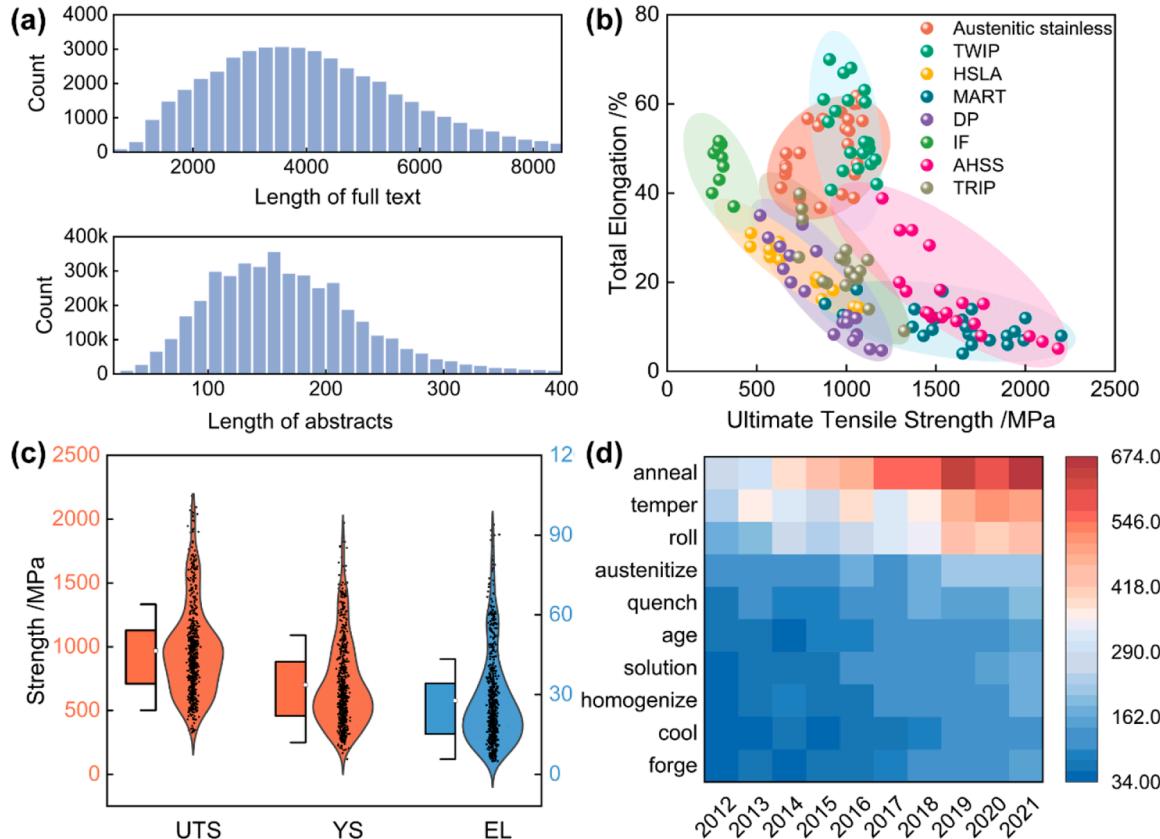
### 2.1. Corpus collection

The corpus used for pre-training comprised 4.2 million abstracts related to materials science and 55,000 full-text articles on steels. Firstly, the full text from journal articles up to 2021 with a focus on “steels” was assembled. To further enhance the robustness of the pre-trained model, we incorporated abstracts from the entire materials field up to 2021, including conference papers, patents, and articles. We collected in total a digital object identifier (DOI) list of approximately 4.2 million materials abstracts and 55,000 full-text steel articles via the CrossRef search application programming interface (API) and Web of Science search engine. The 55,000 full-text was accessed with plain text, extensible markup language (XML), or hypertext markup language (HTML) format via the Elsevier APIs and web scraping for journals published by the Springer, ASME and MDPI. For articles in HTML and XML format, we developed a tailored XML parser with Python scripts for extracting the metadata of articles encompassing title, abstracts, authors, publication year, key words, paragraph text, image URLs, and references information. Table S1 provides details of the number of papers and words for each literature type. The training corpus for SteelBERT in this study encompasses approximately 0.96 billion words, which is 3.5 times larger than MatSciBERT [21] (approximately 285 million words) and represents over 30 % of the original BERT’s [10] word count (3.3 billion words). The average paper length within the corpus is approximately 3000 words, while the abstracts have an average length of around 300 words. As shown in Fig. 2a, the paper length ranges from 1000 to 8000 words, and the abstracts range from 50 to 500 words. Both the abstracts and the full text of steel demonstrate an approximately normal length distribution.

The corpus of properties prediction consists of tabular composition and text sequence of complex fabrication processes. In our previous study, we have developed an automated pipeline tool for alloy data extraction from scientific literature [41–43]. Utilizing this pipeline, we extracted the dataset for steel compositions and mechanical properties mainly from tables. In total 9327 composition tables and 4251 mechanical properties tables were classified. After named entity recognition and relation extraction, we filtered out a dataset with high quality for carbon steels (677 records) with their mechanical properties (UTS, YS and EL), and their corresponding composition (tabular data) and processing information (text data). The processing routes were classified from the full text for each article, containing sentences that describe concrete fabrication procedures, such as solution, deformation, and heat treatment. Fig. 2b depicts an Ashby chart of the automatically extracted UTS and EL labeled by various steels, which is consistent with the expected behavior and provides credibility to the extraction precision. Fig. 2c shows the statistical distributions of the three properties, which is consistent with expected behavior and common sense and proves the extraction precision. Fig. 2d represents the heatmap for the frequency of various actions, such as quenching, aging, forging, solution treatment and cooling, as a function of year from 2012 to 2021. There has been increasing emphasis on quenching, aging, solution treatment, and cooling, all of which significantly impact phase formation and eliminate defects detrimental to steel properties.

### 2.2. Pre-training of steelBERT

SteelBERT was pre-trained based on DeBERTa [44,45] (implemented in Huggingface’s Transformer Python library) using a corpus of approximately 4.2 million materials abstracts and 55,000 full-text steel articles (approximately 0.96 billion words). Moreover, in comparison to



**Fig. 2.** Corpus distributions and visualization of extracted data. (a) Histogram of statistical distributions for abstract and full text lengths. (b) An Ashby chart of the automatically extracted UTS and EL labeled by various steels. (c) Statistical violin plot distributions of the three properties. (d) Heatmap for the frequency of various processing actions.

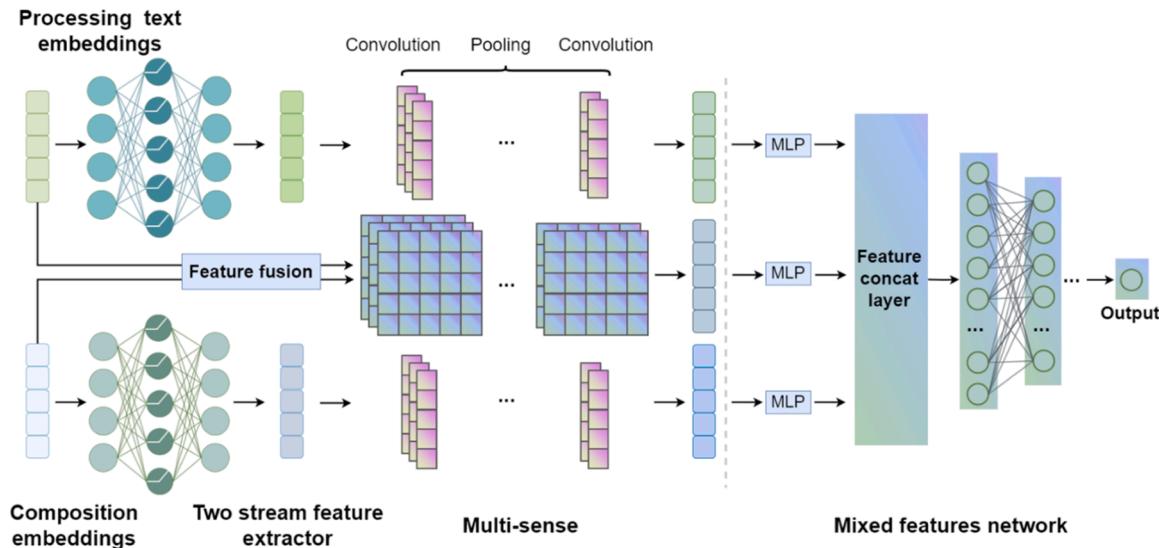
BERT, ALBERT [46], BART [47], and other Transformer-only encoder model architectures, DeBERTa incorporates a disentangled attention mechanism and that can handle long-range dependencies and relationships in language more efficiently. Unlike the typical method of representing each word by a single vector obtained through the summation of content embedding and position embedding, our approach utilizes dual representation with each word being represented by two distinct vectors: one capturing its content and the other encapsulating its relative position. The relative position embeddings are shared across all transformer layers as inputs. This mechanism segregates different aspects of self-attention in the model, allowing it to capture and focus on various parts of the input text more effectively and enables the learning of latent representations of steel knowledge.

We chose the DeBERTa structure to pretrain SteelBERT. Note that there are special symbols used to convey the meaning of processing or mathematical expressions, and at times, these symbols may have multiple Unicode forms. To address this issue, certain Unicode characters lacking significant meaning were eliminated, while others were substituted with a customized character or a sequence of standard characters that closely resemble their previous meaning. While the original DeBERTa model possesses an extensive sub-word vocabulary, this abundance could increase noise during the tokenization of the materials training corpus, leading to variations in word splitting. Consequently, a specialized tokenizer was trained based on our training corpus to construct a vocabulary specific to the steel domain, utilizing the DeBERTa tokenizer (Huggingface's Transformer Tokenizer library). Despite the training corpus comprising only about 6.7 % of the original DeBERTa model, we maintained a consistent vocabulary scale of 128,100 words to ensure the precise capture of latent knowledge. There are 188 million parameters in SteelBERT and the model is constructed using 12 stacked Transformer encoders with each hidden layer incorporating 12 attention heads. SteelBERT masked 15 % of the tokens to create a self-supervised training task, known as pretraining via Masked Language Modeling (MLM), which is a universal and effective pre-training method for various NLP tasks. As shown in Fig. 1b, the initial text inputs were embedded, and information was propagated through multiple transformer layers. Subsequently, the model interfaced with two enhanced masked decoders, introducing absolute position embeddings to enhance the ability to capture more comprehensive information. Specifically, the final hidden vector of the special token "CLS" at the beginning of the sequence was fed into a regressor head, which consisting of one hidden layer with SoftMax as the activation function for prediction.

We set a maximum sentence length of 512 tokens and trained the model until the training loss steps decreasing. The pre-training procedure of SteelBERT used 8 NVIDIA A100 40GB GPUs for 840 h, with a batch size of 576 sequences. We used the AdamW optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$ ,  $\epsilon = 1e^{-6}$ , weight decay =  $1e^{-2}$  and linear decay schedule for learning rate with warmup ratio = 4.8 % and peak learning rate =  $1e^{-4}$ . Pre-training code was written using PyTorch and Transfomers library and available at Hugging Face community <https://huggingface.co/MGE-LLMs/SteelBERT>.

### 2.3. Prediction model architecture

The corpus, with a wide distribution of steel compositions, properties, and processing routes, was then utilized to construct a prediction model of mechanical property using a deep artificial neuron network. The framework is shown in Fig. 3. Composition and processing routes serve as inputs, and the mechanical property is the output. First, there undergoes an embedding representation by SteelBERT. SteelBERT calculates the embedding value for each chemical element and then a steel composition is expressed as the sum of the product of its elemental embeddings with weight percentages. For processing routes, SteelBERT calculates the embedding values of the first special token "CLS" of the processing text for each processing route. In contrast to traditional ML, which involves extracting multiple columns for processing conditions and data alignment, this context-aware representation of composition and processing routes is capable of preserving diversity in processing routes. Processing text and composition embeddings are then fed into the shared-specific feature layers to obtain shared features. Concurrently, we employed a multilayer perceptron (MLP) network to process the respective features. Within the multi sense network, Convolutional neural networks (CNN) were utilized to refine dense informational features. Ultimately, we concatenated features from all components and processed them through a hybrid MLP network layer to generate the output. Simultaneously, a mixed feature matrix with initial composition and processing embeddings propagates the shared and specific features across modalities to compensate for the lack of specific information. Finally, these features are concatenated with the output properties through a MLP network layer. For training, we employed the AdamW optimizer with a linear decay schedule and adopted mean squared error as our loss function. Optuna [48], an open source optimization framework, was then used to automatically search the hyperparameters for initial model architecture in a large space of  $10^7$  candidate architectures. The parameters including the number of layers, number of nodes,



**Fig. 3.** Neural network structure of predictive model.

dropout rates, activation functions, pooling methods and the number of CNN channels were optimized. The optimization process is shown in Fig. S2, each point represents a model architecture for YS, UTS, and EL prediction.

### 3. Results and discussion

#### 3.1. SteelBERT evaluation

During pretraining, SteelBERT was taught to predict masked words representation by adjusting parameters in various network layers. 95 % of the corpus was allocated for training, and 5 % was designated as the validation dataset, maintaining the same ratio as the original DeBERTa. SteelBERT underwent pretraining for approximately 140,000 steps with a minibatch size of 576, while the original BERT model was pretrained for 1000,000 steps with a minibatch size of 256. Given that our training corpus size is approximately 29 % of the size of the BERT training corpus, the calculated pretraining steps for SteelBERT would be  $256 \times 1000,000 \times 29\% / 576$ , resulting in approximately 130,000 steps. However, in order to enhance the model's performance, we extended the training steps to 140,000. Finally, the validation loss reached 1.158 after 840 h of training (Fig. 4a).

We then evaluate the classification precision of processing text based on SteelBERT. We manually labeled 39,852 sentences from 258 articles, and split the dataset into training, validation, and test sets in a 6:2:2 ratio, resulting in 494 positive samples and 39,358 negative samples. Notably, the number of negative samples significantly outweighed that of positive samples, which might lead to an underestimation of the minority class performance. To address this, we retained all positive samples and sampled negative samples at a 1:15 ratio. For each single training epoch, we randomly sampled 1581 sentences. The processing text classifier incorporates a dropout layer and a single MLP layer. We also compared the classification performance based on various published pre-trained models, including BERT, SciBERT [22], MatSciBERT, and SteelBERT (Fig. 4b). SteelBERT achieves the highest performance with the precision, recall, and F1 score of 98.06 %, 94.92 %, and 96.54 %, respectively, based on the random sampling of 52 articles, encompassing approximately 7940 sentences. Additionally, we also tested various negative sampling ratios (1:5, 1:10, and 1:15) to evaluate the impact of imbalanced sampling. The model performed well on unseen negative examples, demonstrating its robustness (see Supplementary Note S1 and Table S2).

#### 3.2. SteelBERT interpretability

Trained on an extensive corpus of materials science text, SteelBERT is equipped with expertise in the linguistic nuances of steels. We first validated SteelBERT's ability on topic clustering of abstracts. Long text

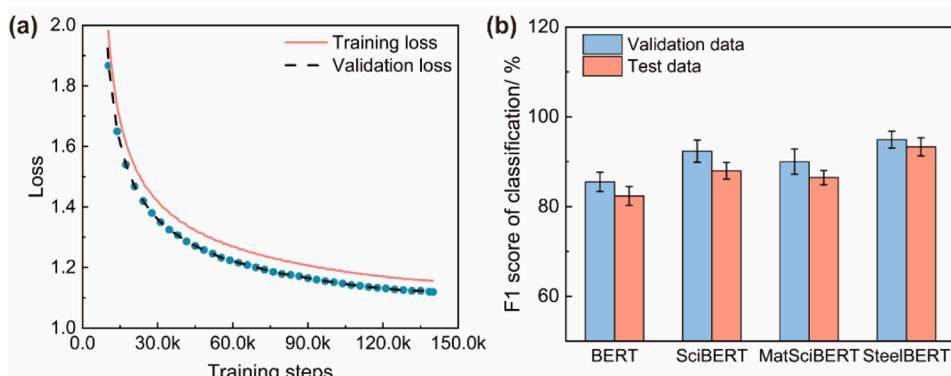
embedding was performed by SteelBERT to convert the 87,497 abstracts from our training corpus text (which specifically focuses on steels) into 768-length vectors. Subsequently, Uniform Manifold Approximation and Projection(UMAP) [49] together with Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) were utilized for dimension reduction and clustering. UMAP can transform high-dimensional data into low-dimension while preserving local structures, and HDBSCAN is capable of identifying clusters of varying shapes and sizes. The integration of UMAP and HDBSCAN significantly enhances the accuracy of cluster identification.

We utilized the "CLS" token to embed the abstracts and generated a matrix of dimensions  $768 \times 87,497$  using SteelBERT. Subsequently, UMAP was employed to reduce the embeddings to a 2-dimensional space, thereby creating a low-dimensional embedding that retains crucial topological features. To maintain a balance between local and global structure, the number of neighbors was set to 700, while the minimum distance was established as 0.0 to regulate the packing density of points in the low-dimensional space. The evaluation of the average "similarity" between and within clusters was conducted using the Davies-Bouldin Index, the Calinski-Harabasz Index, and the Silhouette Coefficient. Following this, the HDBSCAN algorithm was utilized for clustering, with the minimum cluster size set at 30 and the Euclidean distance serving as the metric. Additionally, the Excess of Mass (eom) algorithm, a standard approach in HDBSCAN, was employed as the cluster selection method. We then captured the topic for each cluster by using the class term frequency-inverse document frequency(c-TF-IDF) to generate topic words from their frequencies.

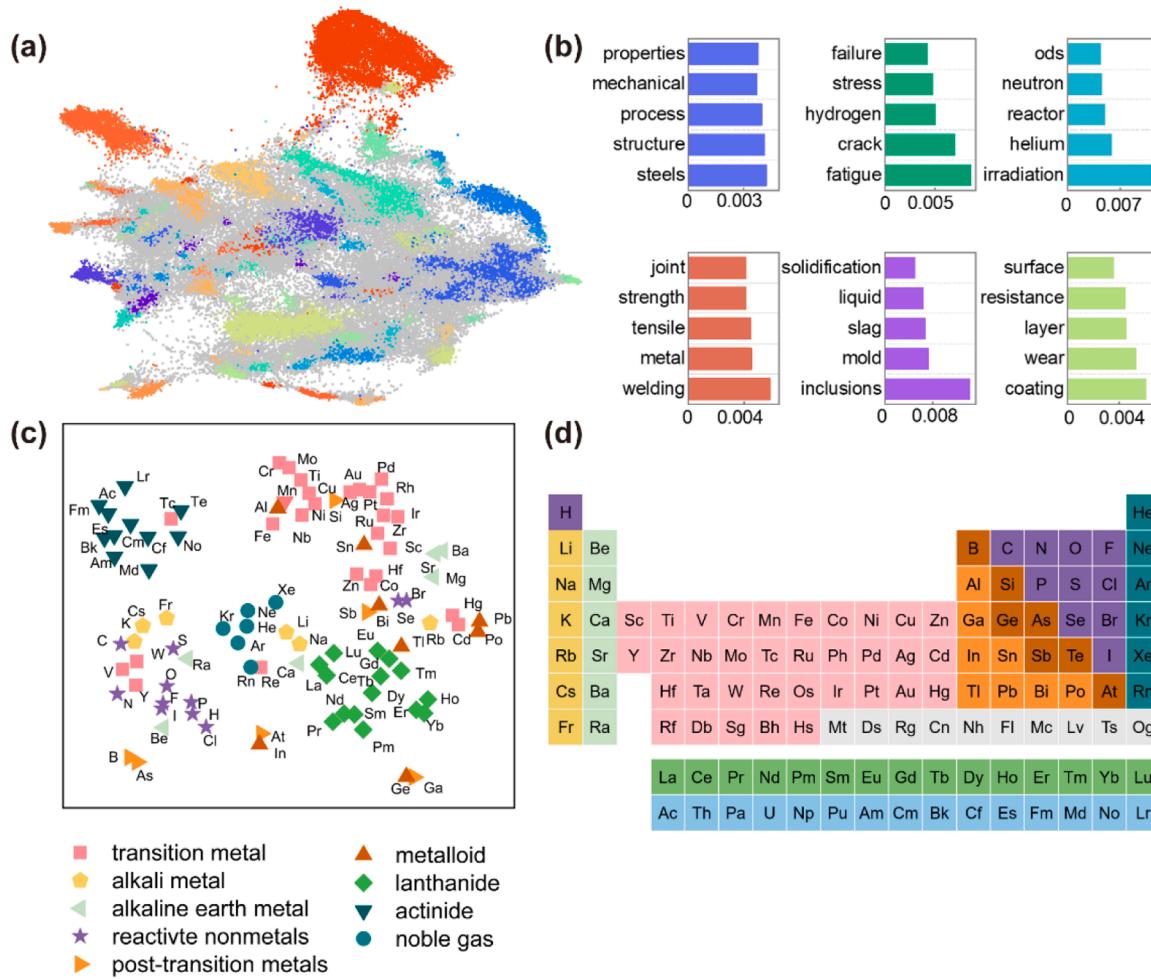
$$c - TF - IDF_i = \frac{t_i}{w_i} \times \log \frac{m}{\sum_j^n t_j} \quad (1)$$

where the frequency of each word  $t$  is extracted for each class  $i$  and divided by the total number of words  $w$ . This action can be seen as a form of regularization of frequent words in the class. The total, unjoined, number of documents  $m$  is divided by the total frequency of word  $t$  across all classes  $n$ .

Fig. 5a illustrates topics distinguished by different colors. The gray points represent topics that surpass the limits of cluster distance. We selected the top 5 words per topic determined by their c-TF-IDF scores to establish a topic representation, as shown in Fig. 5b. The abstracts clustering yields impressive results, clearly highlighting key topics including structure and mechanical properties, fatigue, irradiation, welding, and surface treatment in steel research. We further evaluated the clustering performance quantitatively comparing with BERT, SciBERT, and MatSciBERT. SteelBERT consistently outperformed the other models, demonstrating superior ability to capture domain-specific textual knowledge in steel materials area (see details in Supplementary Note S2 and Table S3).



**Fig. 4.** SteelBERT evaluation. (a) Visualization of the Masked Language Model (MLM) performance on the validation set relative to the SteelBERT training steps. (b) Processing text classification with different LLMs.



**Fig. 5.** SteelBERT Interpretability. (a) Visualization of abstract embedding clustering with the SteelBERT model. The collected abstracts related to the steel topic are embedded into a series of 768-length vectors by the pretrained model. We then use the UMAP approach to reduce high dimensions to a two-dimensional feature space and apply the DBSCAN method to cluster these vectors based on similar topics. (b) Topic generation using c-TF-IDF approach to identify topics within each cluster. (c) A two-dimensional t-SNE projection is utilized to depict word embeddings for 100 chemical elements, which is labeled with its corresponding symbol and grouped by category of transition metal, alkali metal, alkaline earth metal, reactive nonmetals, post-transition metals, metalloid, lanthanide, actinide, and noble gas. Similar elements cluster together, reflecting the periodic table's topology. (d) A periodic table of chemistry using different colors to represent various categories of elements.

We further investigated whether SteelBERT can effectively capture the characteristics of individual chemical elements. SteelBERT was utilized to embed the symbols of the initial 100 chemical elements in the periodic table as 768-length vectors. Subsequently, the t-distributed stochastic neighbor embedding (t-SNE) method was employed to reduce the dimensionality, utilizing cosine distance as the metric for calculating distances between instances and setting the perplexity parameter to 2. We visualized word embeddings for 100 chemical elements labeled with their corresponding symbol and grouped by category, i.e. transition metal, alkali metal, alkaline earth metal, reactive nonmetals, post-transition metals, metalloid, lanthanide, actinide, and noble gas. Because words with similar meanings often appear in similar contexts, the corresponding embeddings will also be similar. Thus, the elements belonging to the same category in the periodic table clustered together in Fig. 5c. The overall distribution faithfully reflects the periodic table's structure, with alkali metals, alkaline earth metals, transition metals, and noble gases arranged from the top left to the bottom right. Additionally, the distribution trend from the top right to the bottom left generally corresponds to increasing atomic numbers. Fig. 5d shows the periodic table colored by category labels. We also compared the element embeddings generated by SteelBERT with those of other models, and SteelBERT outperformed them across multiple metrics, including categorization accuracy and visual interpretability (detailed results in

Table S4).

In predicting the mechanical properties of steel, SteelBERT needs to be able to represent various actions in processing routes. With the help of our previously developed pipeline for extracting alloy synthesis and processing actions [43], we have obtained 515 types of actions from the steel corpus. These actions were further projected and clustered via t-SNE and DBSCAN based on the embeddings given by SteelBERT's embedding, as shown in Fig. 6. The UMAP parameters were set with 700 neighbors and a minimum distance of 0.0, alongside a minimum cluster size of 30. Despite variations in the textual representation of distinct process actions, actions belonging to the same processing category exhibit a consistent distribution, i.e. characterization, mechanical test, welding, electrochemical test, heat treatment, and medium (more details shown in Table S5). These results provide confidence that SteelBERT successfully captures and maintains semantic coherence within the diverse processing actions.

Numerical values in scientific literature are crucial for solving quantitative problems, particularly for predicting mechanical properties. Therefore, it's essential to evaluate the model's ability to capture the semantic significance of numerical values. To assess this, we conducted a number-encoding task using a Euclidean distance matrix between actual numerical values and their corresponding embeddings, covering integers from 0 to 2000 across various models, as shown in

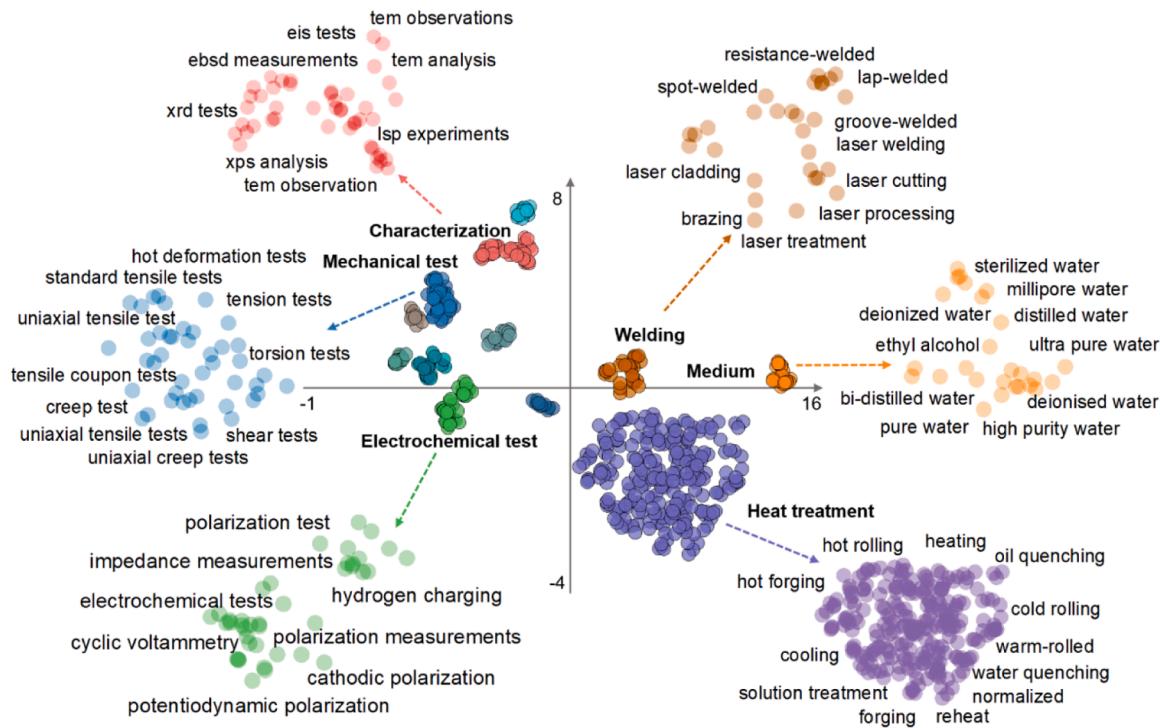


Fig. 6. Clustering of embeddings for processing actions with similar topics.

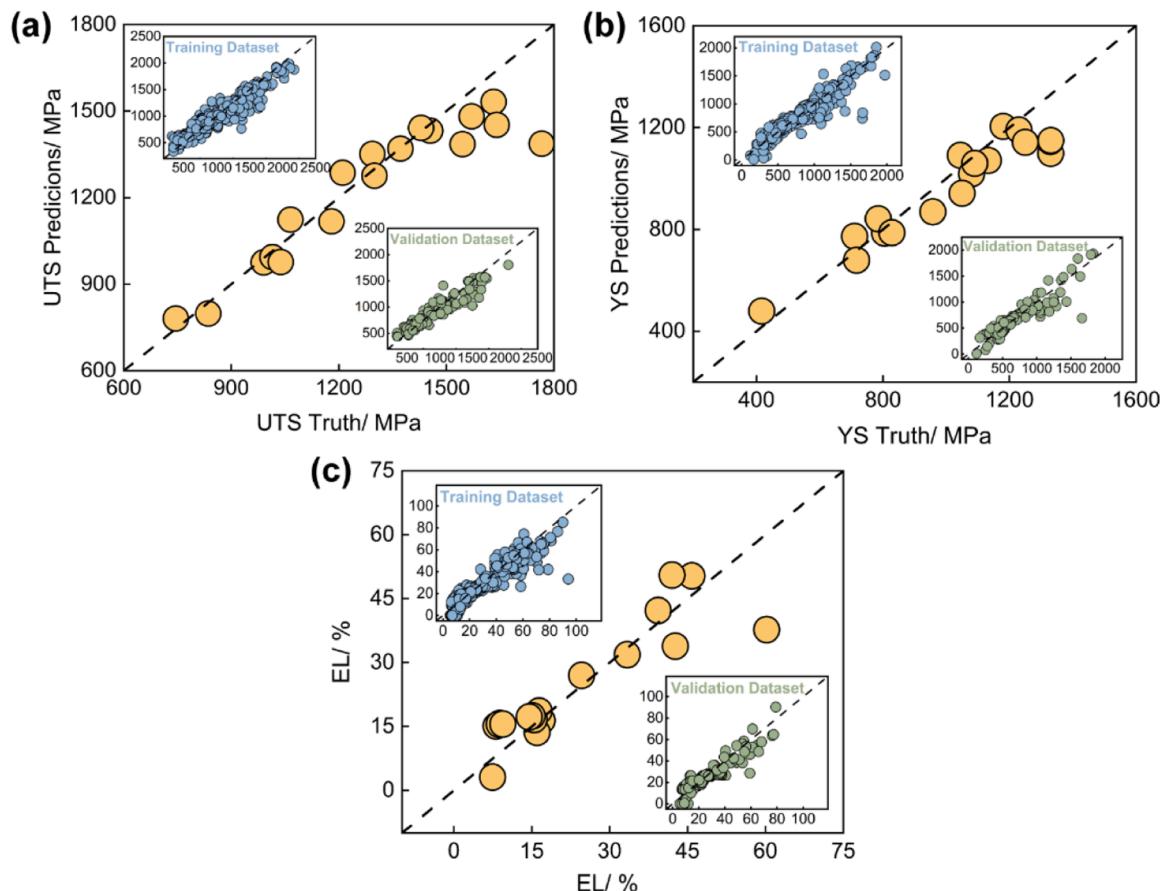


Fig. 7. The training (top-left corner), validation (bottom-right corner), and testing (center) performance of YS, UTS and EL. These points lie closely on the diagonal, indicating high prediction accuracy. The testing set are from the recent literature (2022–2023) that are not part of the training and evaluation datasets.

Fig. S3. SteelBERT performed as expected, with a darker color intensity around  $Y = -X$ , gradually fading towards the edges. This pattern indicates that the model consistently preserves the meaning of the numerical value within the embedding space. As shown in Fig. S4, in our experimental methodology, we systematically trained a probing model to decipher numerical values from their corresponding word embeddings, utilizing a randomly selected 80 % subset of integers ranging from 0 to 2000. For example, the model successfully associated the word embedding of “100” with the numerical value 100.0. We evaluated the predictions across four BERT-based models, with results summarized in Table S6. Notably, SteelBERT showed excellent performance within the training range, the  $R^2$  of the predictive model on the training dataset and validation dataset is 98.35 % ( $\pm 0.15\%$ ) and 98.11 % ( $\pm 0.19\%$ ), respectively, demonstrating accurate decoding and a reliable understanding of numeracy. This can be attributed to its pretraining on a large scientific corpus, coupled with an expansive vocabulary size, which allows it to more effectively capture both terminology and numerical relationships specific to materials science. These results emphasize SteelBERT’s ability to handle numerical data with precision, extending its utility beyond traditional text-based tasks and demonstrating its potential for enhanced performance in fields requiring both numerical and textual comprehension.

### 3.3. Mechanical property prediction

The overall prediction model was trained in an end-to-end manner with 100 randomly generated data splits (8:2 ratio for each split). The  $R^2$  prediction values on the training data for YS, UTS, and EL are 88.61 % ( $\pm 0.99\%$ ), 89.14 % ( $\pm 0.94\%$ ), and 86.29 % ( $\pm 1.52\%$ ), respectively, while on the validation data, these values are 83.48 % ( $\pm 3.62\%$ ), 86.39 % ( $\pm 2.21\%$ ), and 84.85 % ( $\pm 1.99\%$ ), respectively. To further evaluate the performance of the prediction model, we collect a new

dataset (18 recent reported steels) from articles published in 2022 and 2023 as a test dataset for property prediction. Fig. 7 present scatter plots of the predicted values for YS, UTS, and EL, including the training, validation, and testing set in top-left corner, bottom-right corner, and center. We also compared SteelBERT with several open-source LLMs, including BERT, SciBERT, MatSciBERT, Phi-3.5mini-Instruct [50], Mistral-7B-Instruct [51] and LLaMA-3.1 [11] models, using text embeddings and the same predictive framework. Table 1 provides a detailed comparison between SteelBERT and the other models, with performance evaluated using  $R^2$  and MAE on the mechanical property predictions. The results show that SteelBERT consistently outperforms several of the state-of-the-art models, and the prediction  $R^2$  values on testing data for YS, UTS, and EL are 78.17 % ( $\pm 3.40\%$ ), 82.56 % ( $\pm 1.96\%$ ), and 81.44 % ( $\pm 2.98\%$ ), respectively. Notably, the original LLaMA model outperforms its instruct variant, while the LLaMA 70B model underperforms compared to the smaller 8B version. This suggests that larger decoder-only models, though effective for general tasks, may encounter limitations in specialized domains like materials science. The better performance of the LLaMA 8B model is mainly influenced by the distribution and elements of the training corpus. Larger models, such as LLaMA 70B, are typically trained on extensive and diverse datasets to enhance generalizability across a wide range of tasks. However, this broader focus can dilute attention to domain-specific patterns. In contrast, the smaller scale of the LLaMA 8B model may produce embeddings that are less affected by peripheral information from unrelated domains, thereby maintaining a stronger alignment with the core patterns embedded in the training data. Meanwhile, the LLaMA 8B model produces embeddings with a dimensionality of 4098, whereas the LLaMA 70B model generates embeddings of 8192 dimensions, representing a substantial increase in complexity and feature richness. The current downstream network was primarily designed with a focus on balancing the scale of mechanical property dataset and accuracy of

**Table 1**

Performance benchmark of different LLM embeddings for predicting steel mechanical properties with MSEs and  $R^2$  scores.

Model	Dataset	Yield strength		Ultimate tensile strength		Elongation	
		$R^2/\%$	MAE/MPa	$R^2/\%$	MAE/MPa	$R^2/\%$	MAE/%
<b>BERT</b>	Training	62.65 $\pm$ 8.33	150.01 $\pm$ 24.44	85.97 $\pm$ 2.95	92.47 $\pm$ 13.92	76.59 $\pm$ 3.94	5.10 $\pm$ 0.63
	Validation	43.23 $\pm$ 9.63	182.10 $\pm$ 24.08	67.60 $\pm$ 3.48	138.20 $\pm$ 10.74	60.71 $\pm$ 2.69	6.89 $\pm$ 0.50
	Testing	27.00 $\pm$ 9.60	173.22 $\pm$ 12.45	52.55 $\pm$ 5.88	153.64 $\pm$ 14.06	57.57 $\pm$ 2.14	8.03 $\pm$ 0.31
<b>SciBERT</b>	Training	75.14 $\pm$ 0.88	119.20 $\pm$ 3.90	87.30 $\pm$ 1.29	94.21 $\pm$ 6.93	78.12 $\pm$ 1.55	5.03 $\pm$ 0.36
	Validation	58.46 $\pm$ 1.28	149.07 $\pm$ 4.02	67.30 $\pm$ 3.31	150.49 $\pm$ 7.43	70.88 $\pm$ 2.27	6.59 $\pm$ 0.46
	Testing	40.66 $\pm$ 2.76	156.05 $\pm$ 4.27	37.28 $\pm$ 7.03	187.59 $\pm$ 9.04	55.20 $\pm$ 9.76	7.88 $\pm$ 0.72
<b>MatSciBERT</b>	Training	65.00 $\pm$ 4.00	155.71 $\pm$ 9.75	72.88 $\pm$ 5.31	148.50 $\pm$ 18.25	72.20 $\pm$ 9.19	5.61 $\pm$ 1.11
	Validation	47.95 $\pm$ 8.25	179.75 $\pm$ 32.01	51.78 $\pm$ 6.86	188.28 $\pm$ 18.29	55.98 $\pm$ 9.20	6.96 $\pm$ 0.87
	Testing	21.20 $\pm$ 5.90	172.78 $\pm$ 6.85	21.40 $\pm$ 6.51	189.38 $\pm$ 9.64	52.33 $\pm$ 10.18	8.73 $\pm$ 0.96
<b>SteelBERT</b>	Training	88.61 $\pm$ 0.99	67.64 $\pm$ 2.41	89.14 $\pm$ 0.94	78.53 $\pm$ 2.79	86.29 $\pm$ 1.52	4.45 $\pm$ 0.12
	Validation	83.48 $\pm$ 3.62	99.87 $\pm$ 6.41	86.39 $\pm$ 2.21	97.5 $\pm$ 6.65	84.85 $\pm$ 1.99	5.14 $\pm$ 0.31
	Testing	78.17 $\pm$ 3.40	92.63 $\pm$ 7.77	82.56 $\pm$ 1.96	82.29 $\pm$ 4.7	81.44 $\pm$ 2.98	5.28 $\pm$ 0.31
<b>Phi-3.5mini-Instruct</b>	Training	57.98 $\pm$ 6.36	174.38 $\pm$ 18.97	69.04 $\pm$ 4.74	163.80 $\pm$ 12.74	77.88 $\pm$ 0.88	5.07 $\pm$ 0.25
	Validation	47.55 $\pm$ 7.88	198.10 $\pm$ 23.44	52.92 $\pm$ 3.61	194.94 $\pm$ 10.48	73.47 $\pm$ 0.24	6.14 $\pm$ 0.14
	Testing	20.43 $\pm$ 7.75	169.11 $\pm$ 12.22	23.75 $\pm$ 4.71	193.61 $\pm$ 6.23	55.47 $\pm$ 10.30	8.15 $\pm$ 0.60
<b>Mistral-7B-Instruct-V0.2</b>	Training	57.38 $\pm$ 1.50	179.22 $\pm$ 4.59	63.12 $\pm$ 7.07	182.58 $\pm$ 20.71	79.85 $\pm$ 1.20	4.71 $\pm$ 0.34
	Validation	30.35 $\pm$ 5.03	212.67 $\pm$ 1.77	45.42 $\pm$ 4.88	202.36 $\pm$ 17.07	66.77 $\pm$ 2.70	6.31 $\pm$ 0.27
	Testing	19.53 $\pm$ 7.41	195.40 $\pm$ 10.38	19.12 $\pm$ 6.31	202.06 $\pm$ 11.45	61.55 $\pm$ 5.75	8.68 $\pm$ 0.84
<b>Llama-3.1-8B</b>	Training	82.90 $\pm$ 1.41	88.14 $\pm$ 4.30	90.54 $\pm$ 1.06	72.53 $\pm$ 6.21	76.74 $\pm$ 5.66	5.12 $\pm$ 0.77
	Validation	58.83 $\pm$ 2.84	138.84 $\pm$ 5.51	70.19 $\pm$ 2.93	129.13 $\pm$ 8.26	62.62 $\pm$ 5.21	6.64 $\pm$ 0.52
	Testing	57.23 $\pm$ 2.63	136.70 $\pm$ 4.65	58.13 $\pm$ 2.84	146.34 $\pm$ 5.68	64.13 $\pm$ 2.73	7.38 $\pm$ 0.40
<b>Llama-3.1-8B-Instruct</b>	Training	78.70 $\pm$ 3.68	101.99 $\pm$ 10.95	83.94 $\pm$ 3.34	98.55 $\pm$ 12.46	73.34 $\pm$ 3.62	5.55 $\pm$ 0.59
	Validation	58.71 $\pm$ 4.41	151.90 $\pm$ 11.55	65.40 $\pm$ 1.93	141.89 $\pm$ 6.18	61.51 $\pm$ 2.61	7.07 $\pm$ 0.38
	Testing	52.68 $\pm$ 4.49	139.04 $\pm$ 6.91	57.72 $\pm$ 3.35	147.64 $\pm$ 4.97	60.66 $\pm$ 2.82	7.17 $\pm$ 0.23
<b>Llama-3.1-8B-Instruct (ft)</b>	Training	80.75 $\pm$ 1.35	101.14 $\pm$ 0.98	87.96 $\pm$ 0.86	85.44 $\pm$ 3.60	76.48 $\pm$ 2.84	5.44 $\pm$ 0.64
	Validation	62.10 $\pm$ 2.20	151.15 $\pm$ 6.81	69.08 $\pm$ 4.02	131.51 $\pm$ 13.34	71.98 $\pm$ 2.77	6.04 $\pm$ 0.30
	Testing	59.30 $\pm$ 3.21	124.67 $\pm$ 9.84	60.74 $\pm$ 1.78	151.79 $\pm$ 1.78	67.34 $\pm$ 6.36	7.17 $\pm$ 0.73
<b>Llama-3.1-70B</b>	Training	77.45 $\pm$ 2.23	109.00 $\pm$ 5.80	75.58 $\pm$ 5.27	121.67 $\pm$ 14.10	83.17 $\pm$ 1.35	4.34 $\pm$ 0.38
	Validation	52.66 $\pm$ 2.59	148.35 $\pm$ 3.17	59.04 $\pm$ 3.80	159.50 $\pm$ 8.66	67.11 $\pm$ 2.97	6.28 $\pm$ 0.35
	Testing	48.28 $\pm$ 2.31	146.91 $\pm$ 4.13	38.19 $\pm$ 3.86	178.30 $\pm$ 4.88	64.22 $\pm$ 2.46	7.16 $\pm$ 0.22
<b>Llama-3.1-70B-Instruct</b>	Training	66.80 $\pm$ 6.14	149.07 $\pm$ 22.41	76.54 $\pm$ 4.12	131.78 $\pm$ 14.92	71.32 $\pm$ 4.84	6.07 $\pm$ 0.83
	Validation	49.56 $\pm$ 4.56	173.97 $\pm$ 13.07	59.54 $\pm$ 3.20	172.65 $\pm$ 10.70	59.99 $\pm$ 5.39	7.12 $\pm$ 0.47
	Testing	32.41 $\pm$ 4.73	159.72 $\pm$ 5.83	35.86 $\pm$ 5.20	176.06 $\pm$ 8.05	58.20 $\pm$ 5.41	7.27 $\pm$ 0.52

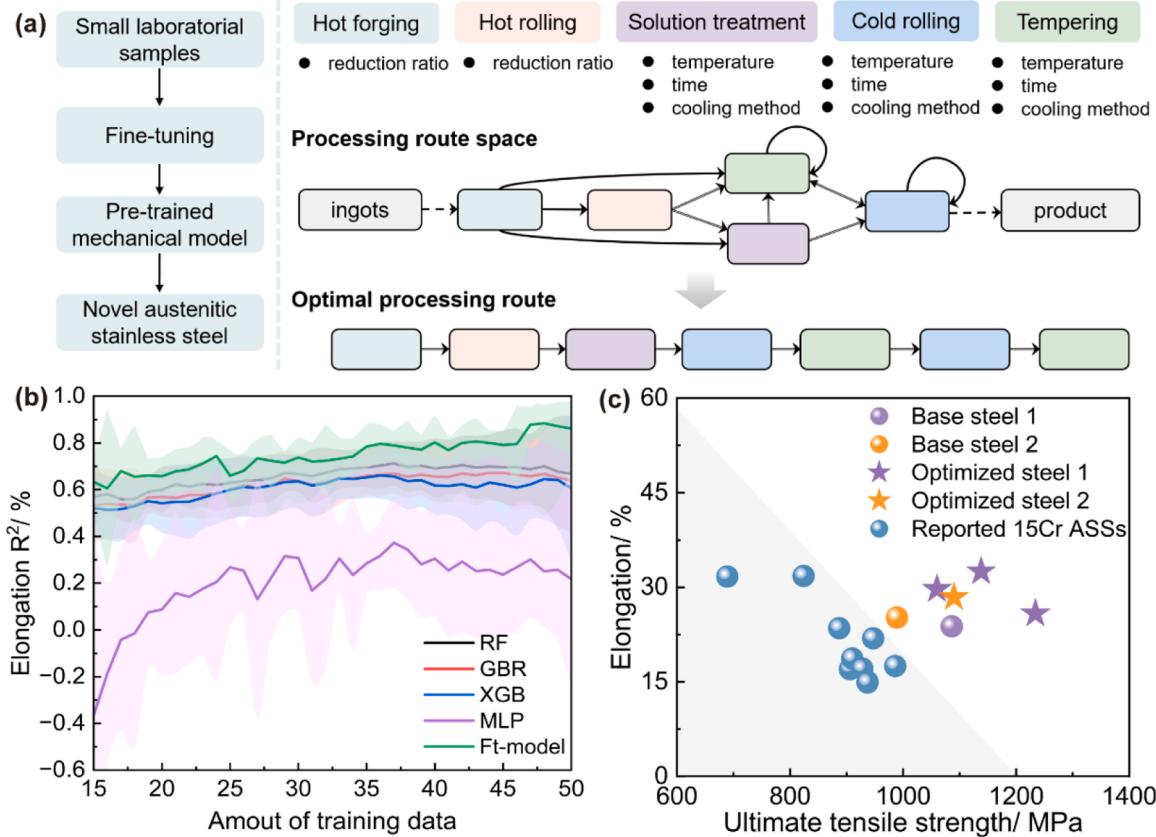
performance prediction. In addition, we also fine-tuned the Llama 8B-instruct model using the same training corpus as SteelBERT, employing a causal language modeling task with LoRA. The embeddings generated by the fine-tuned model were subsequently utilized as input for downstream property prediction tasks. As shown in Table 1, the fine-tuned embeddings demonstrated a slight improvement over the original Llama 8B-Instruct model embeddings, with an average performance increase of approximately 5 %. The reason owes to the limitation of fine-tuning corpus, where constrained dataset sizes may lead the model to memorize specific instances rather than learning broader metallurgical principles. As a result, a fine-tuned model may achieve high performance on the training set but fail to generalize to unseen alloy compositions or novel experimental conditions. Moreover, fine-tuning modifies the model's parameters for task-specific optimization but does not rectify the inherent biases or knowledge deficiencies present in the original pre-trained embeddings. If the pre-trained model does not adequately capture the intricate relationships and specialized vocabulary of steel materials science, fine-tuning fails to embed these connections effectively. We also compared our prediction strategy with GPT [26] by instruct fine-tuning it using the official API with the same dataset (677 records) with train-and-test splits. Prompts for the regression task were tailored. The results of several generative models using parameter-efficient methods to improve adaptation can be found in Supplementary Note S3–4 and Table S7.

SteelBERT demonstrates superior performance in property prediction tasks compared to other models, primarily due to its domain-specific pretraining on an extensive corpus of 0.96 billion words in materials science. This pretraining enables SteelBERT to capture the terminology, semantic relationships, and contextual nuances unique to the field. The model employs a tokenizer based on the Byte Pair

Encoding (BPE) [52] algorithm with a vocabulary size of 128 K, enhancing its ability to represent technical language and steel-specific terms. This feature significantly improves its effectiveness in steel-related tasks. In contrast, MatSciBERT pretrained on broader corpora comprising 0.28 billion words spanning diverse topics such as inorganic glasses, ceramics, and alloys, tends to be less specialized. MatSciBERT employs a tokenizer with a smaller vocabulary size of 31 K using the wordpiece algorithm [53]. These broader datasets and smaller token vocabularies likely dilute the model's focus on steel-specific knowledge. Meanwhile, SteelBERT's lower-dimensional embeddings reduce computational overhead, accelerate convergence, and simplify neural network optimization, further contributing to its efficiency in downstream applications. Moreover, SteelBERT's architecture, built on the DeBERTa framework, incorporates advanced features such as a disentangled attention mechanism and an enhanced mask decoder. These innovations allow SteelBERT to capture token relationships and contextual information with remarkable precision. In summary, SteelBERT's outstanding performance arises from the integration of domain-specific pretraining, lower-dimensional embeddings, and architectural advancements, making it generate accurate embeddings and well-suited for property prediction.

### 3.4. Steel design

Through a fine-tuning strategy, this model continually refines its understanding within specific small datasets. ASSs have attracted growing interest for their high ductility, excellent corrosion and oxidation resistance, whereas their strengths are typically modest. To improve the existing mechanical properties, we then utilized literature-based predictive model to strategically design advanced ASSs. Given that our



**Fig. 8.** Rational design of new steels. (a) Schematic diagram illustrating the fine-tuning process on a small experimental dataset for the exploration of advanced ASSs. (b) Comparison between the fine-tuned model and traditional machine learning models trained with varying amounts of training data ranging from 15 to 50. (c) Scatter plot of reported 15Cr ASSs, base steels and optimized steels.

models were derived from a comprehensive literature dataset, to adapt the model to predict new composition and processing routes, we fine-tuned it using our own set of 64 experimental data on the 15Cr ASSs fabricated via hot forging, solution treatment, cold rolling, and tempering, as depicted in Fig. 8a.

The materials used in this study are ASSs in plate form. The chemical compositions of these experimental ASSs are listed in Table S8. The bullet-shaped 50 kg ingots were hot-forged into a cuboid solid shape with a length of  $80 \times 80 \times 630$  mm. To control the appropriate grain size, these plates were solution-treated at  $1000^{\circ}\text{C}$  and  $1050^{\circ}\text{C}$  for 4 h, followed by water quenching. Subsequently, these specimens were cold rolled with reductions ranging from 25 % to 80 %. Finally, these samples were tempered in the range of 590 to  $780^{\circ}\text{C}$  for 5 to 60 min. This fine-tuning process involves aligning the model with the specific processing routes and textual descriptors unique to our experiments, by only adjusting the parameters of mixed feature layers in Fig. 3.

The data was structured into a tabular format and subsequently divided into a training set and a testing set using an 8:2 ratio. We specifically fixed the shared-specific feature layers when fine-tuning model, which required stabilizing certain weights and biases within the network nodes. Meanwhile, we adjusted the mixed feature layers to enhance flexibility and adaptability using our datasets. We utilized the AdamW optimizer, setting with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e^{-8}$ , weight decay= $1e^{-2}$  followed by a linear decay schedule. We chose the mean squared error as training loss function and setting 1000 approaches for model training. We also compared the fine-tuned model with traditional machine learning models. The gradient boosting regression (GBR), kernel ridge regression (KRR), MLP, support vector regression (SVR), extreme gradient boosting (XGB), random forest (RF) algorithms were employed for model training, incorporating features such as weight percentage of chemical elements, solution treatment temperature and duration, cold-rolling thickness reduction, and both primary and secondary rounds of tempering temperatures and times. Model parameters for all algorithms were optimized by grid search. The  $R^2$  and MAE were used as the metrics, with a more comprehensive assessment presented in Table S9. Furthermore, we compared our model with other open-source LLMs, which were first fine-tuned on the literature dataset and subsequently fine-tuned on the same experimental dataset to ensure consistent comparison (see Table S10 for details). Our model shows superior prediction performance across all mechanical properties, achieving an  $R^2$  of 89.85 % ( $\pm 6.17\%$ ), 88.34 % ( $\pm 5.95\%$ ) and 87.24 % ( $\pm 5.15\%$ ) for YS, UTS and EL, respectively. Fig. 8b compares the fine-tuned model with traditional machine learning models trained using varying amounts of data ranging from 15 to 50 points for the elongation property (Table S11–13). The figure indicates a stronger performance for our

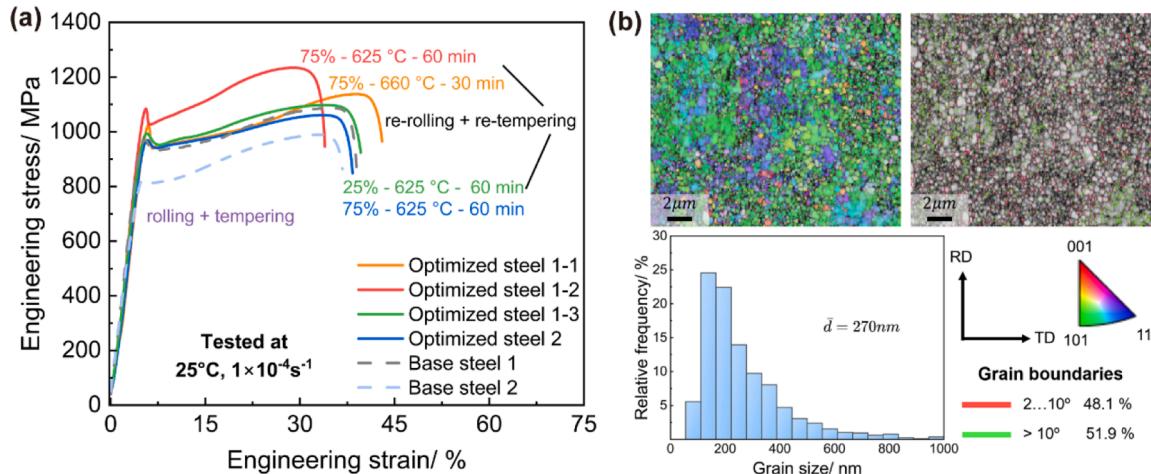
model even with a very limited training data set.

We selected two experiment samples with the same composition of namely 15Cr-7.92Ni-0.83Mn-0.45Si-1.99Mo-0.043Nb-1.49Cu, denoted in purple (base steel 1) and orange (base steel 2) in Fig. 8c, as the base steels for the optimization. These two steels underwent the same hot forging and solution treatment, with differing reduction ratios for a single cold rolling process, as well as variations in temperature and duration for a single tempering process (further details provided in Table S14). We established a search space for processing routes, encompassing various combinations of cold rolling and tempering sequences with different rolling times, reduction ratios, tempering temperatures, and durations. Following a grid search within this extensive space, spanning over 20 h, we identified four candidate processing routes for optimization. The optimized processing routes maintain the same reduction ratio for the initial cold rolling process as their respective base steels but include an additional round of cold rolling and tempering. Three optimized steels are derived from base steel 1, and one is derived from base steel 2. They are also compared with the reported 15Cr ASSs from the literature [54–57] in Fig. 8c and show excellent mechanical properties so far.

Their tensile behaviors are depicted in Fig. 9a, which indicates substantial enhancements in the strength and ductility of the base steels after optimization. During tensile tests, plate-like tensile specimens were extracted along the rolling direction, with dimensions of 65 mm in length, 5 mm in width, and 1 mm in thickness. The slow strain rate tensile tests were conducted using the ForceCreate WDML-1 stress corrosion testing machine at a constant loading rate of 0.12 mm/min. The strength and elongation of each type of steel were determined from the average values of at least three parallel measurements. After a secondary round of cold rolling and tempering, one of the optimized steels (optimized steel 1-1) demonstrated gave a YS of 960 MPa, an UTS of 1138 MPa, and an EL of 32.5 %. This is 50 MPa, 52 MPa and 8.7 % higher than its base steel (comprehensive test results are provided in Table S15). The MAE between true and predicted values for the optimized steels is 34.34 MPa, 42.16 MPa and 1.74 % for YS, UTS and EL, respectively. Fig. 9b shows the Electron Backscatter Diffraction maps of the optimized Steel 1-1, where the austenite is depicted grain boundaries are highlighted using the Grain Boundary map. The refined grains, measuring approximately 270 nm, obtained through optimized processes, may contribute to the observed high strength and ductility.

## 5. Conclusion

We have presented an end-to-end pipeline from materials text to properties with accuracy and ability to further generalize. It is based on



**Fig. 9.** Tensile tests and characterization of steel. (a) Tensile stress strain plots at  $25^{\circ}\text{C}$  and  $1 \times 10^{-4} \text{ s}^{-1}$  of ASSs for base and optimized steels. (b) Electron backscatter diffraction map and grain size distribution of the optimized steel 1-1. (Rolling direction is marked by vertical arrow).

a large language model exposed to historical steel literature data to capture steel knowledge and representation. It “learns” composition-processing-property relationships from a quantitative dataset for steel discovery. We have not only accurately predicted the mechanical properties of 18 steel materials reported in 2022 and 2023, but also improved the prediction accuracy for 64 laboratory datasets by fine-tuning a literature-text-based model. Furthermore, by optimizing the sequences of processes, we have obtained performance that exceeds that of any reported 15Cr ASSs.

The design of steel requires an understanding of physics, chemistry, and metallurgy. Despite the extensive history of steel research, designing high-performance steel still needs considerable computational resources and experimental data. We have explored a pipeline that mainly utilizes textual feature encoding for predicting mechanical properties of steel. This approach is grounded in both literature and experimental data for validation, and it aids in the development of new materials. Compared to traditional ML, our approach has partially resolved the common challenge of extracting structured data for complex steel fabrication processes of steel, processing sequence alignment, and resolving high-dimensional sparse issues, which typically are time consuming. Additionally, the approach demonstrates superior performance in quantitative regression tasks compared to pure Transformer-decoder LLMs. It is relatively easy to implement and requires users to input composition and text sequence for processing and output mechanical properties. There is significant potential in developing ML models with natural language text-based features, which are promising for predicting properties in various materials beyond just steel.

## Data availability

The datasets and codes used in the present work are available at <https://github.com/mgedata/SteelScientist>. Also, the pretrained model SteelBERT in this paper can downloaded at <https://huggingface.co/MGE-LLMs/SteelBERT>. Also, the complete code has been integrated into an application, which can be accessed through our website at <http://smpp.mgedata.cn/>.

## CRediT authorship contribution statement

**Shaohan Tian:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization. **Xue Jiang:** Writing – review & editing, Writing – original draft, Software, Methodology, Funding acquisition, Conceptualization. **Weiren Wang:** Writing – review & editing, Software, Methodology. **Zhihua Jing:** Writing – review & editing, Software, Data curation. **Chi Zhang:** Writing – review & editing, Validation. **Cheng Zhang:** Writing – review & editing, Validation. **Turab Lookman:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Yanjing Su:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work is financially supported by the National Key Research and Development Program of China (2022YFB3707502), National Natural Science Foundation of China (92270001, 52201061, U22A20106), Guangdong Province Key Areas Research and Development Programs (2024B0101080003), Key Areas Research and Development Programs Fundamental Research Funds for the Central Universities (FRF-TP-

22-008A1), USTB MatCom of Beijing Advanced Innovation Center for Materials Genome Engineering.

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.actamat.2024.120663](https://doi.org/10.1016/j.actamat.2024.120663).

## References

- [1] J. Moon, W. Beker, M. Siek, J. Kim, H.S. Lee, T. Hyeon, B.A. Grzybowski, Active learning guides discovery of a champion four-metal perovskite oxide for oxygen evolution electrocatalysis, *Nat. Mater.* 23 (2024) 108–115, <https://doi.org/10.1038/s41563-023-01707-w>.
- [2] P. Raccuglia, K.C. Elbert, P.D. Adler, C. Falk, M.B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, A.J. Norquist, Machine-learning-assisted materials discovery using failed experiments, *Nature* 533 (2016) 73–76, <https://doi.org/10.1038/nature17439>.
- [3] Z. Rao, P.-Y. Tung, R. Xie, Y. Wei, H. Zhang, A. Ferrari, T.P.C. Klaver, F. Körmann, P.T. Sukumar, A. Kwiatkowski Da Silva, Y. Chen, Z. Li, D. Ponge, J. Neugebauer, O. Gutleisch, S. Bauer, D. Raabe, Machine learning–enabled high-entropy alloy discovery, *Science* 372 (2022) 78–85, <https://doi.org/10.1126/science.abe4940>.
- [4] B. Sanchez-Lengeling, A. Aspuru-Guzik, Inverse molecular design using machine learning: generative models for matter engineering, *Science* 361 (2018) 360–365, <https://doi.org/10.1126/science.aat2663>.
- [5] L. Gu, Y. Liu, P. Chen, H. Huang, N. Chen, Y. Li, T. Lookman, Y. Lu, Y. Yu, Bond sensitive graph neural networks for predicting high temperature superconductors, *Mater. Genome Eng. Adv.* 2 (2024) e48, <https://doi.org/10.1002/mgea.48>.
- [6] N. Boehnke, J.P. Straehla, H.C. Safford, M. Kocak, M.G. Rees, M. Ronan, D. Rosenberg, C.H. Adelmann, R.R. Chivukula, N. Nabar, A.G. Berger, N. G. Lamson, J.H. Cheah, H. Li, J.A. Roth, A.N. Koehler, P.T. Hammond, Massively parallel pooled screening reveals genomic determinants of nanoparticle delivery, *Science* 377 (2022) eabm5551, <https://doi.org/10.1126/science.abm5551>.
- [7] J. Xie, Prospects of materials genome engineering frontiers, *Mater. Genome Eng. Adv.* 1 (2023) e17, <https://doi.org/10.1002/mgea.17>.
- [8] Open AI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Alten Schmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschnik, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H.W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eletti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S.P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S.S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, L. Kaiser, A. Kamali, I. Kanitscheider, N.S. Keskar, T. Khan, L. Kilpatrick, J.W. Kim, C. Kim, Y. Kim, H. Kirchner, J. Kirov, M. Knight, D. Kokotajlo, L. Kondracik, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C.M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanj, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S.M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O’Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de A.B. Peres, M. Petrov, H.P.de O. Pinto, Michael, Pokorny, M. Pokrass, V. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotstodt, H. Roussette, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F.P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. Thompson, P. Tillett, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J.F.C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J.J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C.J. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, GPT-4 Technical Report, (2023). <http://arxiv.org/abs/2303.08774>.
- [9] E. Almazrouei, H. Alobeitli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, É. Goffinet, D. Hesslow, J. Launay, Q. Malartic, D. Mazzotta, B. Noune, B. Pannier, G. Penedo, The Falcon Series of Open Language Models, (2023). <http://arxiv.org/abs/2311.1686>.
- [10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proc. 2019 Conf. North, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, <https://doi.org/10.18653/v1/N19-1423>.

- [11] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, LLaMA: open and efficient foundation language models, (2023). [http://doi.org/10.48550/arXiv.2302.13971](https://doi.org/10.48550/arXiv.2302.13971).
- [12] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C.C. Ferrer, M. Chen, G. Cucurull, D. Esioubi, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P.S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poultion, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E.M. Smith, R. Subramanian, X.E. Tan, B. Tang, R. Taylor, A. Williams, J.X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: open Foundation and Fine-Tuned Chat Models, (2023). <http://arxiv.org/abs/2307.09288>.
- [13] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, Palm: scaling language modeling with pathways, *J. Mach. Learn. Res.* 24 (2023) 1–113.
- [14] D. Driss, F. Xia, M.S.M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, P. Florence, PaLM-E: an Embodied Multimodal Language Model, (2023). <http://arxiv.org/abs/2303.03378>.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Lukasz Kaiser, I. Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [16] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A Survey of Large Language Models, (2023). <http://arxiv.org/abs/2303.18223>.
- [17] J. Kaplan, S. McCandlish, T. Henighan, T.B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling Laws for Neural Language Models, (2020). <http://arxiv.org/abs/2001.08361>.
- [18] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, Pre-trained models: past, present and future, *AI Open* 2 (2021) 225–250, <https://doi.org/10.1016/j.aiopen.2021.08.002>.
- [19] S. Huang, J.M. Cole, BatteryBERT: a pretrained language model for battery database enhancement, *J. Chem. Inf. Model.* 62 (2022) 6365–6377, <https://doi.org/10.1021/acs.jcim.2c00035>.
- [20] G. Pan, F. Wang, C. Shang, H. Wu, G. Wu, J. Gao, S. Wang, Z. Gao, X. Zhou, X. Mao, Advances in machine learning- and artificial intelligence-assisted material design of steels, *Int. J. Miner. Metall. Mater.* 30 (2023) 1003–1024, <https://doi.org/10.1007/s12613-022-2595-0>.
- [21] T. Gupta, M. Zaki, N.A. Krishnan, Mausam, MatSciBERT: a materials domain language model for text mining and information extraction, *Npj Comput. Mater.* 8 (2022) 102, <https://doi.org/10.1038/s41524-022-00784-w>.
- [22] I. Beltagy, K. Lo, A. Cohan, SciBERT: a pretrained language model for scientific text, *ArXiv Prepr* (2019). ArXiv190310676.
- [23] C. Kuennen, R. Ramprasad, polyBERT: a chemical language model to enable fully machine-driven ultrafast polymer informatics, *Nat. Commun.* 14 (2023) 4099, <https://doi.org/10.1038/s41467-023-39868-6>.
- [24] M. Yoshitake, F. Sato, H. Kawano, H. Teraoka, MaterialBERT for natural language processing of materials science texts, *Sci. Technol. Adv. Mater. Methods* 2 (2022) 372–380, <https://doi.org/10.1080/27660400.2022.2124831>.
- [25] M. Chen, J. Tworek, H. Jun, Q. Yuan, H.P. de O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillett, F.P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W.H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A.N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, W. Zaremba, Evaluating Large Language Models Trained on Code, (2021). <http://arxiv.org/abs/2107.03374>.
- [26] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, Language models are few-shot learners, *Adv. Neural Inf. Process. Syst.* 33 (2020) 1877–1901.
- [27] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, C. Sutton, Program Synthesis with Large Language Models, (2021). <https://arxiv.org/abs/2108.07732>.
- [28] W.U. Ahmad, S. Chakraborty, B. Ray, K.-W. Chang, Unified Pre-training for Program Understanding and Generation, (2021). <http://arxiv.org/abs/2103.06333>.
- [29] A. Merchant, S. Batzner, S.S. Schoenholz, M. Aykol, G. Cheon, E.D. Cubuk, Scaling deep learning for materials discovery, *Nature* 624 (2023) 80–85, <https://doi.org/10.1038/s41586-023-06735-9>.
- [30] N.J. Szymanski, B. Rendy, Y. Fei, R.E. Kumar, T. He, D. Milsted, M.J. McDermott, M. Gallant, E.D. Cubuk, A. Merchant, H. Kim, A. Jain, C.J. Bartel, K. Persson, Y. Zeng, G. Ceder, An autonomous laboratory for the accelerated synthesis of novel materials, *Nature* 624 (2023) 86–91, <https://doi.org/10.1038/s41586-023-06734-w>.
- [31] Z. Zheng, O. Zhang, C. Borgs, J.T. Chayes, O.M. Yaghi, ChatGPT chemistry assistant for text mining and the prediction of MOF synthesis, *J. Am. Chem. Soc.* (2023), <https://doi.org/10.1021/jacs.3c05819> jacs.3c05819.
- [32] D.A. Boiko, R. MacKnight, B. Kline, G. Gomes, Autonomous chemical research with large language models, *Nature* 624 (2023) 570–578, <https://doi.org/10.1038/s41586-023-06792-0>.
- [33] T. Nanda, V. Singh, V. Singh, A. Chakraborty, S. Sharma, Third generation of advanced high-strength steels: processing routes and properties, *Proc. Inst. Mech. Eng. Part J. Mater. Des. Appl.* 233 (2019) 209–238, <https://doi.org/10.1177/1464420716664198>.
- [34] Y. Li, G. Yuan, L. Li, J. Kang, F. Yan, P. Du, D. Raabe, G. Wang, Ductile 2-GPa steels with hierarchical substructure, *Science* 379 (2023) 168–173, <https://doi.org/10.1126/science.add7857>.
- [35] J. Gao, S. Jiang, H. Zhang, Y. Huang, D. Guan, Y. Xu, S. Guan, L.A. Bendersky, A. V. Davydov, Y. Wu, Facile route to bulk ultrafine-grain steels for high strength and ductility, *Nature* 590 (2021) 262–267, <https://doi.org/10.1038/s41586-021-03246-3>.
- [36] H. Zhi, J. Li, W. Li, M. Elkot, S. Antonov, H. Zhang, M. Lai, Simultaneously enhancing strength-ductility synergy and strain hardenability via Si-alloying in medium-Al FeMnAlC lightweight steels, *Acta Mater.* 245 (2023) 118611, <https://doi.org/10.1016/j.actamat.2022.118611>.
- [37] Y. Zhang, C. Wang, K.M. Reddy, W. Li, X. Wang, Study on the deformation mechanism of a high-nitrogen duplex stainless steel with excellent mechanical properties originated from bimodal grain design, *Acta Mater.* 226 (2022) 117670, <https://doi.org/10.1016/j.actamat.2022.117670>.
- [38] C. Zou, J. Li, W.Y. Wang, Y. Zhang, D. Lin, R. Yuan, X. Wang, B. Tang, J. Wang, X. Gao, Integrating data mining and machine learning to discover high-strength ductile titanium alloys, *Acta Mater.* 202 (2021) 211–221, <https://doi.org/10.1016/j.actamat.2020.10.056>.
- [39] X. Wei, S. van der Zwaag, Z. Jia, C. Wang, W. Xu, On the use of transfer modeling to design new steels with excellent rotating bending fatigue resistance even in the case of very small calibration datasets, *Acta Mater.* 235 (2022) 118103, <https://doi.org/10.1016/j.actamat.2022.118103>.
- [40] H. Zhang, H. Fu, X. He, C. Wang, L. Jiang, L.-Q. Chen, J. Xie, Dramatically enhanced combination of ultimate tensile strength and electric conductivity of alloys via machine learning screening, *Acta Mater.* 200 (2020) 803–810, <https://doi.org/10.1016/j.actamat.2020.09.068>.
- [41] R. Yan, X. Jiang, W. Wang, D. Dang, Y. Su, Materials information extraction via automatically generated corpus, *Sci. Data* 9 (2022) 401, <https://doi.org/10.1038/s41597-022-01492-2>.
- [42] W. Wang, X. Jiang, S. Tian, P. Liu, T. Lookman, Y. Su, J. Xie, Alloy synthesis and processing by semi-supervised text mining, *Npj Comput. Mater.* 9 (2023) 183, <https://doi.org/10.1038/s41524-023-01138-w>.
- [43] W. Wang, X. Jiang, S. Tian, P. Liu, D. Dang, Y. Su, T. Lookman, J. Xie, Automated pipeline for superalloy data by text mining, *Npj Comput. Mater.* 8 (2022) 9, <https://doi.org/10.1038/s41524-021-00687-2>.
- [44] P. He, X. Liu, J. Gao, W. Chen, Deberta: decoding-enhanced bert with disentangled attention, *ArXiv Prepr* (2020). ArXiv200603654.
- [45] P. He, J. Gao, W. Chen, Debertav3: improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, *ArXiv Prepr* (2021). ArXiv211109543.
- [46] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: a Lite BERT for Self-supervised Learning of Language Representations, (2020). <http://arxiv.org/abs/1909.11942>.
- [47] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: denoising sequence-to-sequence pre-training for natural language generation, *Transl. Comprehension* (2019). <http://arxiv.org/abs/1910.13461>.
- [48] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, ACM, Anchorage AK USA, 2019, pp. 2623–2631, <https://doi.org/10.1145/3292500.3330701>.
- [49] L. McInnes, J. Healy, J. Melville, UMAP: uniform Manifold Approximation and Projection for Dimension Reduction, (2020). <http://arxiv.org/abs/1802.03426>.
- [50] M. Abdin, J. Aneja, H. Awadalla, A.A. Awani, N. Bach, A. Bahree, A. Bakhtiarji, I. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. Del Giorno, I. Goswami, S. Gunasekar, E. Haider, J. Hao, R.J. Hewett, W. Hu, J. Huynh, D. Iter, S.A. Jacobs, M. Javarapu, X. Jin, N. Karampatziakis, P. Kauffmann, M. Khademi, D. Kim, Y.J. Kim, L. Kurrienko, J.R. Lee, Y.T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C.C.T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilic, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L.L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, X. Zhou, Phi-3 technical report: a highly capable language model locally on your phone, (2024). <http://arxiv.org/abs/2404.14219>.
- [51] A.Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D.S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L.R. Lavaud, M.-A. Lachaux, P. Stock, T.L. Scao, T. Lavril, T. Wang, T. Lacroix, W.E. Sayed, Mistral 7B, (2023). <http://arxiv.org/abs/2310.06825>.
- [52] Rico Sennrich, Barry Haddow, Alexandra Birch, Neural machine translation of rare words with subword units, in: *Proceedings of the 54th Annual Meeting of the*

- Association for Computational Linguistics 1, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1715–1725. Long Papers).
- [53] M. Schuster, K. Nakajima, Japanese and Korean voice search, in: 2012 IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP, IEEE, 2012, pp. 5149–5152, <https://doi.org/10.1109/ICASSP.2012.6289079>.
- [54] J. Wan, Q. Ran, J. Li, Y. Xu, X. Xiao, H. Yu, L. Jiang, A new resource-saving, low chromium and low nickel duplex stainless steel 15Cr–xAl–2Ni–yMn, Mater. Des. 53 (2014) 43–50, <https://doi.org/10.1016/j.matdes.2013.06.043>.
- [55] J. Wan, H. Ruan, S. Shi, Excellent combination of strength and ductility in 15Cr–2Ni duplex stainless steel based on ultrafine-grained austenite phase, Mater. Sci. Eng. A 690 (2017) 96–103, <https://doi.org/10.1016/j.msea.2017.02.056>.
- [56] J. Wan, H. Ruan, J. Wang, S. Shi, Exploiting the non-equilibrium phase transformation in a 15Cr–2Ni–2Al–11Mn resource-saving duplex stainless steel, Mater. Des. 114 (2017) 433–440, <https://doi.org/10.1016/j.matdes.2016.10.076>.
- [57] Y. Zheng, H. Sun, L. Yan, H. Yang, K. Gao, X. Pang, A.A. Volinsky, Ferrite effects on the hydrogen embrittlement of 17-4PH stainless steel, Anti-Corros. Methods Mater. 69 (2022) 331–338, <https://doi.org/10.1108/ACMM-03-2022-2615>.