Element2Vec: Learning Chemical Element Representation from Text for Property Prediction

Yuanhao Li[a,*], Keyuan Lai[a,b,*], Tianqi Wang[c,a], Qihao Liu[d,a], Jiawei Ma[e], Yuan-Chao Hu[a,**]

[a]*Songshan Lake Materials Laboratory, Dongguan, Guangdong, China*
[b]*International Academic Center of Complex Systems, Beijing Normal University, Zhuhai, Guangdong, China*
[c]*Department of Applied Physics, The Hong Kong Polytechnic University, Hong Kong, China*
[d]*Department of Computer Science, City University of Hong Kong (Dongguan), Dongguan, Guangdong, China*
[e]*Department of Computer Science, City University of Hong Kong, Hong Kong, China*

Abstract

Accurate property data for chemical elements is crucial for materials design and manufacturing, but many of them are difficult to measure directly due to equipment constraints. While traditional methods use the properties of other elements or related properties for prediction via numerical analyses, they often fail to model complex relationships. After all, not all characteristics can be represented as scalars. Recent efforts have been made to explore advanced AI tools such as language models for property estimation, but they still suffer from hallucinations and a lack of interpretability. In this paper, we investigate Element2Vecto effectively represent chemical elements from natural languages to support research in the natural sciences. Given the text parsed from Wikipedia pages, we use language models to generate both a single general-purpose embedding (*Global*) and a set of attribute-highlighted vectors (*Local*). Despite the complicated relationship across elements, the computational challenges also exist because of 1) the discrepancy in text distribution between common descriptions and specialized scientific texts, and 2) the extremely limited data, *i.e.*, with only 118 known elements, data for specific properties is often highly sparse and incomplete. Thus, we also design a test-time training method based on self-attention to mitigate the prediction error caused by Vanilla regression clearly. We hope this work could pave the way for advancing AI-driven discovery in materials science.

Keywords: Large language model, Material science, Element encoding, Deep learning, Text mining, Data science

1. Introduction

Accurate property data of chemical elements, *e.g.*, *atomic radius and conductivity*, are of paramount importance to materials design and manufacturing, *e.g.*, *barrier material design*. Nevertheless, the direct measurement of specific properties for certain elements presents inherent challenges, *e.g.*, *"Lithium-Ion diffusion barrier"* where measuring ionic conductivity of Lithium requires complicated synthesis for dense ceramic pellets and thin films while the results are highly sensitive to materials defects. As an alternative, numerical analysis that leverages other properties of the element, *e.g.*, *anion size* or analogous properties of related elements, *e.g.*, *Ti, Ge, Zr*, is employed, which may still require profound expert knowledge and fail to model intricate element-wise relationship comprehensively, resulting in limited effectiveness.

With the recent advances of deep learning, the graph neural networks have emerged as powerful tools to implicitly learn element correlation for property prediction [1, 2, 3, 4, 5, 6, 7], but may always require the data of properties used in input to be complete. In contrast, the data in materials science is of inherent and severe sparsity. Then, the large language models (LLMs) are finetuned for automatic chemical knowledge extraction from literature, followed by encoding in pre-defined formats [8, 9, 10], and direct material property prediction from chemical composition [11, 12]. Nevertheless, both approaches still rely on one-hot or handcrafted descriptors for elements, hindering the generalizability. Furthermore, issues like hallucination [13] and data distribution shift can compromise the reliability and utility of LLM further.

In this work, we propose to employ LLM as an embedding extractor and construct a vector representation of chemical elements from free-form textual descriptions. For one thing, the materials are multifaceted entities with the description requiring a synthesis of physical, chemical, and structural attributes [16]. Then, the many properties, *e.g.*, *phase behavior such as gas and liquid*, as well as the material design process, cannot be intuitively represented as mathematical descriptors. For another thing, the strong representation encoded with multifaceted characteristics may enable application of a simple algorithm for reliable and explainable prediction [17, 18]. Thus, for each element, by integrating diverse information into a unified, context-rich representation [19], we may enable broader downstream applications.

Specifically, we propose Element2Vecthat leverages the LLM to extract embeddings from text parsed from a Wikipedia webpage, and these embeddings can be directly compared in the same representation space. As illustrated in Fig. 1 (Left), motivated by the image encoding techniques that produce both a single embedding to describe entire image (globally) and a set of localized embeddings focusing on specific sub-regions, for

*Y.L. and K.L. contributed equally to this work.
**Corresponding author: yuanchao.hu@sslab.org.cn [Y.-C.H.]

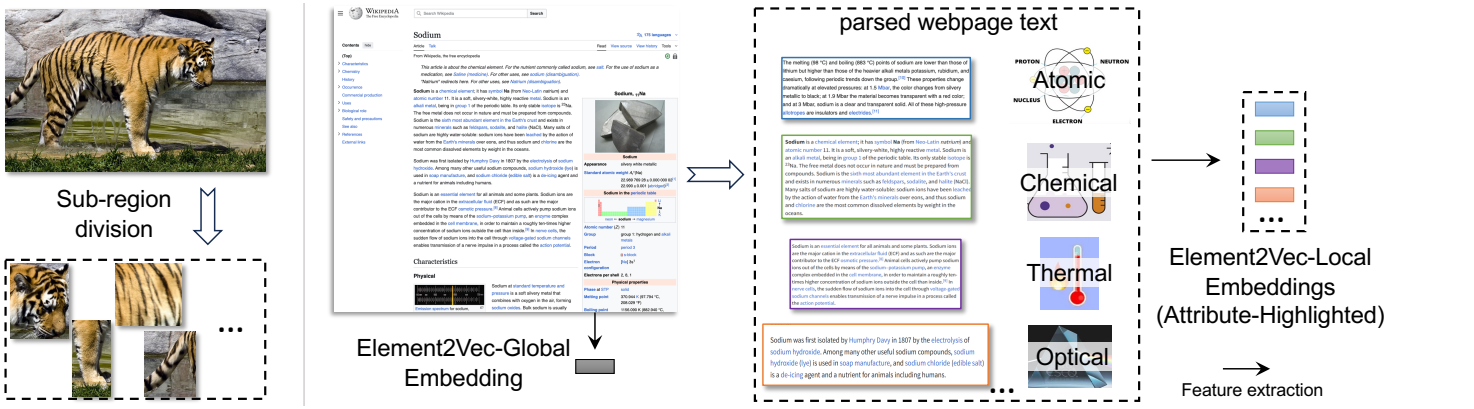

Figure 1: **Encode Element Representation from Text based on inspiration from Image Representation.** (Left) Conventional image representation can be either a global embedding extracted from the whole image, or a set of local embeddings extracted from a sub-region that focuses on several key components [14], such as tail and head. (Right) Analogously, given the tremendous amount of text in element description and its diverse application and aspects, we can also extract a single *Global* embedding from all text or following the definition in [15] to extract *Local* embeddings highlighting different element attributes like atomic, chemical, thermal, and optical.

each element, in addition to a *Global* embedding from the entire Wikipedia page to capture holistic information, we also design the streamline to extract a set of *Local* embeddings, where each is tailored to a specific attribute category, *e.g.*, chemical and atomic, following the taxonomy defined in [15]. Without loss of generality, we assess properties framed as both classification and regression tasks. In particular, for regression under high data sparsity, we propose a test-time training method based on self-attention that recasts prediction as an imputation problem to enhance precision. With the contribution summarized as follows, we hope that our efforts could open new avenues for accelerating materials design [20]:

- We investigate the representations of chemical elements from text by leveraging the capabilities of LLM. Moving beyond numerical data, we believe that integrating diverse information formats into a unified representation is essential for improving downstream tasks like property estimation.
- We investigate Element2Vec, a training-free framework that generates and compares a *Global* embedding distilled from the entire text corpus, *i.e.*, Wikipedia, and *Local* embeddings that highlights specific attributes.
- Property estimation involves both problems in the form of classification and regression. For regression under high data sparsity, we propose a test-time training method based on self-attention and formulate the problem as imputation.

2. Related work

AI for materials discovery. Deep learning graph networks such as CGCNN [1] and MEGNet [3] have been applied for property prediction [2], while bond angles are further incorporated to improve predictions [21, 22]. Unsupervised natural language processing techniques have also uncovered chemical knowledge [23]. With the recent success of language models, MatSciBERT [24] is designed and tailored to materials text for information extraction, and LLMs like GPT-3 [25] are fine-tuned on chemistry tasks to predict molecular and materials properties competitively in low-data regimes [8, 9, 10].

Chemical elements embedding. Early work used physical properties or compositional data to project elements into vector spaces that reflect periodic trends. For instance, methods based on self-organizing maps [26] and Magpie features [27] successfully grouped elements by similarity, while unsupervised approaches like Atom2Vec [28] and SkipAtom [29] learned embeddings from compound structures that mirrored the periodicity of properties in an unsupervised manner. However, the dimensionality-reduction techniques [28, 30] primarily rediscover known relationships from existing data, lacking a mechanism for extrapolation beyond established knowledge. Nevertheless, all of the methods mentioned above depend on fixed or task-specific elemental encodings.

Despite prior explorations with word embeddings [18] and materials knowledge graphs showing that language captures periodic trends and structure–property relations [23, 31, 32, 33], the graph model still privileges geometric features but ignores language-informed semantics [34, 22, 35]. Following the great success of large language models, recent efforts emphasize text mining, KG construction, or text-guided generation [20, 36]. Very recent "universal atomic embeddings" are trained from crystal data rather than textual descriptions [19]. Still, the gap in constructing element-level embeddings from text still exists, which motivate our Element2Vec approach that learns broadly applicable element embeddings from text and evaluates them in material property prediction.

Large Language Models (LLM) have revolutionized text processing, enabling the flexible extraction of structured knowledge from unstructured text [37, 25]. Their capabilities in document segmentation, section identification, and intent classification allow for the precise capture of property-bearing statements from scientific literature [38, 39, 40]. Research in scientific NLP has developed domain-specialized LLMs to extract structured knowledge from text [41, 42, 43].

This has spurred the development of domain-specialized

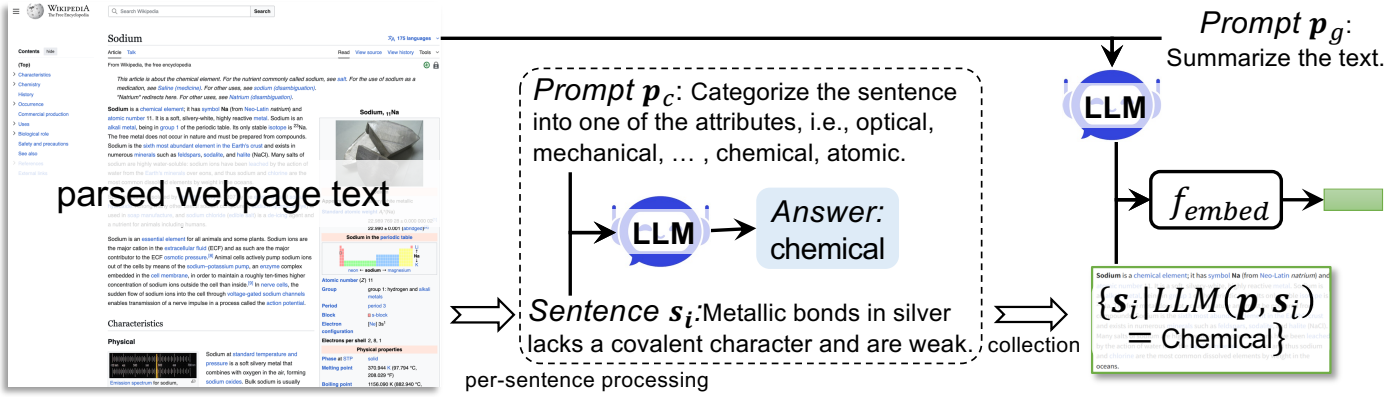

Figure 2: **Element2Vec-Locals Embeddings Extraction.** For each element, given the text parsed from the Wikipedia webpage, we use LLM to label each sentence about the attribute categories [15] independently and summarize the full text to shorten the words via the prompts $\mathbf{p}_c$ and $\mathbf{p}_g$ respectively. For each subset, it is concatenated with the summary and fed into f_{emb} for extraction.

LLMs, like SciBERT [44, 39], which demonstrate superior performance in scientific information extraction. Techniques such as prompting further enable complex tasks like classification and rationale generation without task-specific model fine-tuning. In our pipeline, we leverage these advancements to process Wikipedia text: an LLM segments it into attribute-specific units and filters noisy sentences, thus generating clean, structured input for subsequent embedding and prediction tasks.

Text-embedding models convert textual descriptions into fixed-size vectors that preserve semantic relationships [45]. Early word embeddings like word2vec utilize word co-occurrence to discern word meanings and identify patterns [46]. Nevertheless, they encounter difficulties in comprehending extended sentences and acquiring detailed knowledge about specific entities. **(author?)** [47] proposed SIF as a simple yet strong baseline for sentence embeddings, and **(author?)** [48] introduced QuickThoughts with a contrastive context-prediction objective. However, many text-only pipelines can be trained upon a sufficient amount of data and do not necessarily utilize structured knowledge, *e.g.*, Wikipedia2Vec [49], ERNIE [50], KEPLER [51]. Recent large-scale embedding models leverage LLMs for data generation or as backbones [52, 53]. In this work, we use Gemini Embedding, which is trained atop the Gemini LLM and explicitly designed to retain knowledge within fixed-size vectors [54].

3. Element2Vec: Chemical Elements Representation from Text

For each chemical element, we utilize the text parsed from its Wikipedia webpage to construct the representation and consider both *Global* and *Local* embeddings.

Element2Vec-Global. As an intuitive baseline, given the text corpus $\{\mathbf{s}_i\}_{i=1}^{N_e}$ of element e that consists of N_e sentence $\mathbf{s}_i$. The *Global* embedding is defined as a holistic representation of the entire corpus, which can be obtained as $f_{emb}(\{\mathbf{s}_i\}_{i=1}^{N_e})$ where f_{emb} is implemented as Gemini embedding [54].

Despite the *Global* embedding's aim for a comprehensive representation, a single embedding may struggle to explicitly represent all distinct attributes [15] covered in the content. Meanwhile, the order at the attribute level depends on the sentence order, as the original sequential order of sentences preserved in the webpages also varies, the order-sensitive nature of f_{emb} architecture can then introduce noise into the embeddings.

Element2Vec-Locals is thus proposed to highlight attributes in the extracted embedding without losing the global context. For efficiency purposes and without LLM retraining, we extract from a combination of a global summary and an attribute-specific subset, *i.e.*, a filtered collection of sentences, as illustrated in Fig. 2.

In detail, following the concepts in [15], we define eight attribute categories, *i.e.*, "Mechanical (Physical)", "Optical", "Electrical and Magnetic", "Thermal", "Chemical", "Atomic and Radiational", "Applications", and "Abundance". Then, we classify each sentence $\mathbf{s}_i$ using a prompt $\mathbf{p}_c$ that instructs the LLM to assign it to an attribute, *i.e.*, LLM($concat(\mathbf{p}_c, \mathbf{s}_i)$). Sentences tagged with the same attribute are automatically grouped into a distinct subset. Finally, by concatenating with a concise global context, LLM($concat(\mathbf{p}_s, \{\mathbf{s}_i\}_{i=1}^{N_e})$), summarized via LLM by the prompt $\mathbf{p}$, we extract the corresponding Gemini embedding [54]. The justification for including both components is provided by the ablation studies in Sec. 4.1.

4. Property Prediction by Element2Vec

We first evaluate the characteristics of the *Global* and *Local* embeddings via element classification Sec. 4.1 and Sec. 4.2. Subsequently, we apply these embeddings to property estimation, framing it as a regression problem and introducing and demonstrating the efficacy of the test-time training approach Sec. 4.3 and Sec. 4.4 even under conditions of high data sparsity.

4.1. Classification of chemical elements

In this paper, we define family as a group of elements in the periodic table that share similar chemical properties, electronic configurations, and reactivity patterns. Specifically, families are defined as the vertical columns (element groups), e.g., Alkali

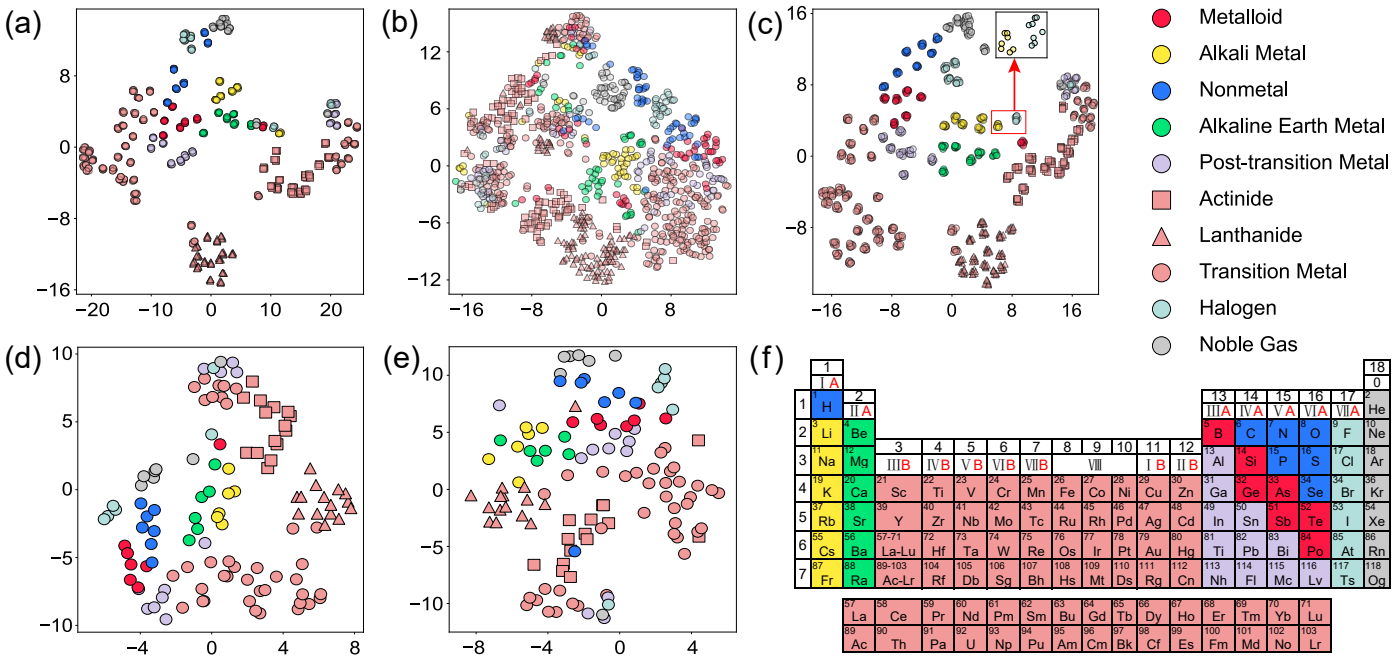

Figure 3: **t-SNE visualization of `element2vec` representations**. (a) Embeddings obtained by sending the entire Wikipedia page with a prompt. (b) Local embeddings obtained from 8 attributes without summary. (c) Local embedding, one point per element obtained from the attribute plus its summary. (d) Aggregated embeddings from (c), 8 attribute-level embeddings are combined into a single point per element. (e) Global embedding: one point per element obtained directly. (f) Periodic table with the same color legend as in (a–e).

metals (Group 1: Li, Na, K, . . .). Family classification is the process of categorizing elements into these families.

We use t-SNE to compare Element2Vec-Global and Element2Vec-Locals representations under identical projection hyperparameters and a fixed random seed (Fig. 3). Fig. 3(a) feeds the entire Wikipedia page to the Gemini embedding model with an added instruction prefix ("prompt") intended to induce attribute-aware encoding. Fig. 3(b) uses an LLM to segment the page into eight predefined attributes and embeds each attribute text without any summary. Fig. 3(c) applies our Element2Vec-Locals method, in which each attribute text is concatenated with the element-level summary before embedding; the inset figure in Fig. 3(c) highlights two elements to show that the eight attribute points of the same element form a compact local cluster rather than collapsing to a single point. Fig. 3(d) aggregates the eight attribute embeddings from Fig. 3(c) into one vector per element by concatenation (details in Sec . 3), whereas Fig. 3(e) shows the Element2Vec-Global pipeline where one vector is obtained directly from the full page. Fig. 3(f) provides the periodic table with the same color legend; colors denote chemical families, and markers distinguish lanthanides (triangles) and actinides (squares).

Three observations follow. First, in Fig. 3(a), the attribute-level points of a given element nearly coincide, indicating that adding a prompt to the full-page input does not induce meaningful attribute selectivity; the representations behave almost as if the same text were embedded repeatedly, which means that our prompt did not work in Gemini embedding. Second, in Fig. 3(b), the eight attributes per element scatter broadly and do not form identifiable per-element clusters, suggesting that removing the shared global context makes attribute snippets too heterogeneous to anchor to their source element. Third, Fig. 3(c) yields a markedly more coherent geometry: for nearly all elements, the eight attribute points cluster together while retaining small intra-cluster spread, consistent with the design goal of preserving both shared element identity (via the summary) and attribute-specific variation (via the segmented text).

At the element level, Fig. 3(d) (Element2Vec-Locals) aligns more closely with the periodic families than Fig. 3(e) (Element2Vec-Global). Families such as metalloids, alkali metals, nonmetals, and alkaline earth metals form tighter, cleaner groups in Fig. 3(d), and transition metals—including lanthanides and actinides marked by different symbols—also exhibit improved compactness. Because all panels share identical t-SNE settings, these differences reflect the representations themselves. In summary, summary-augmented Element2Vec-Locals embeddings produce attribute-level micro-clusters anchored on their parent elements and yield Element2Vec-Locals that respect periodic families more faithfully than Element2Vec-Global.

4.2. Entropy analysis with summary-augmented Element2Vec-Locals embeddings

Based on Fig. 3(d) and Fig. 3(e), where Element2Vec-Locals yielded tighter intra-element clusters and clearer family-level separation than Ement2Vec-Global, we now move from qualitative visualization to a quantitative assessment of prediction decisiveness.

Encoding an entire element page into a single vector can dilute attribute-specific signals, whereas encoding only attribute snippets may lose the global context that relates an element's properties. We therefore adopt an Ement2Vec-Locals scheme

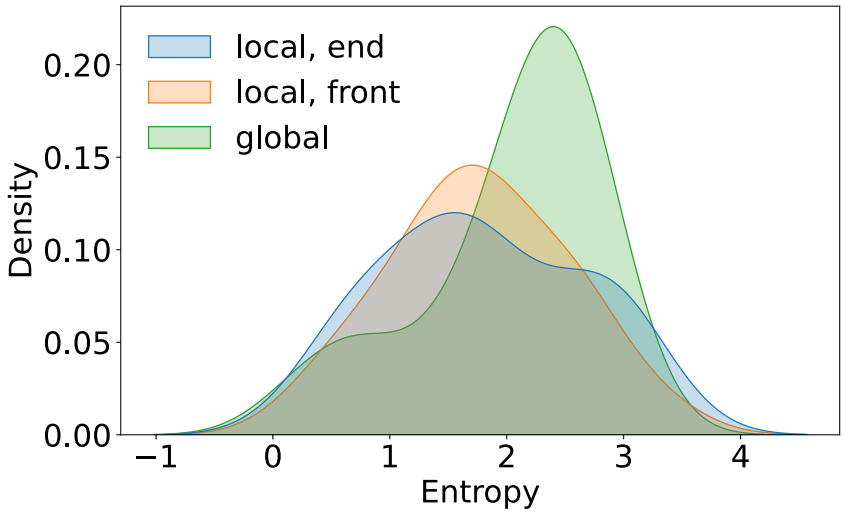

Figure 4: **Entropy distributions for family classification.** Kernel density estimation (KDE) of prediction entropy for the 10-way periodic-family classifier under three embeddings.

augmented with an element-level summary: each Wikipedia page is segmented into eight predefined attributes; a concise summary of the full page is produced. For each attribute, the attribute's text is concatenated with the summary (front or end), encoded with the Gemini embedding model, and used in downstream tasks. We evaluate three datasets: Element2Vec-Global (global), Ement2Vec-Locals with summary prepended (local, front), and Ement2Vec-Locals with summary appended (local, end).

To quantify predictive uncertainty for the 10-way family classifier, we compute the Shannon entropy of the softmax posterior for each test element x,

$$H(x) = -\sum_{k=1}^{K} p_k(x)\,\log_2(p_k(x) + 10^{-9}), \tag{1}$$

where $K = 10$ and $p_k(x)$ is the predicted probability of class k; lower $H(x)$ indicate sharper, less ambiguous posteriors. The constant 10^{-9} is to ensure $\log_2 0$ is never evaluated when $p_k(x) = 0$.

The resulting distributions (Fig. 4) show that both `local, front` and `local, end` shift entropy toward lower values compared with `global`, indicating sharper posteriors and reduced ambiguity. Between the two summary placements, `local, end` consistently exhibits a slightly larger mass at lower entropies than `local, front`. A possible interpretation is that appending the summary preserves attribute-specific leading context while still injecting a shared global prior, yielding marginally more decisive predictions. Overall, augmenting attribute segments with a brief summary reduces uncertainty relative to a single global embedding, and placing the summary at the end offers a modest additional gain.

4.3. Test-Time Training for Robust Property Value Imputation

Different from the family classification problem formulated above where the labels of all elements are given, a key objective for property prediction is to estimate the property values for certain elements which are unknown or hard to measure.

As an intuitive baseline and shown in Fig. 5 (b), the inductive training pipeline can be considered, where a predictor f_p, with typical architecture as linear layer or MLP, is first trained on elements $\mathcal{E}_k$ with known property value and directly tested on the other elements $\mathcal{E}_{uk}$ whose property values are missed, where $\mathcal{E}_k$ and $\mathcal{E}_{uk}$ stands for the set of embeddings for the corresponding element set, *i.e.*, $|\mathcal{E}_k \cup \mathcal{E}_{uk}| = 118$ and $\mathcal{E}_k \cap \mathcal{E}_{uk} = \emptyset$. Then, the core assumption is that the model will generalize from the training elements to unseen test elements, which could be nevertheless sub-optimal when the training data is limited. After all, for certain properties, the property data is highly sparse where the values are available for less-than-20% of the total 118 elements comprising the periodic table.

Motivated by the fact that the properties of an element can be influenced by those of others, it is crucial to model inter-element correlations for accurate prediction. Consequently, we propose to employ a network f_{sa} based on self-attention architecture [55] and design a corresponding test-time training pipeline [56] as shown in Fig. 5(a) as the training of network parameters and the property prediction of elements in $\mathcal{E}_{uk}$ can be done simultaneously.

In detail, we formulate the $\mathcal{E}_k \cup \mathcal{E}_{uk}$ as input sequence and fed into the self-attention layer, where the corresponding output of each input embedding is projected via a shared linear layer and used as the predicted property value. Then, for the network supervision, the objective is to only minimize the loss, *e.g.*, mean squared error [1] of elements in $\mathcal{E}_k$ and no supervision will be applied for elements in $\mathcal{E}_{uk}$.

In this way, different from the conventional inductive training pipeline, where the gradients used for predictor parameter update is only derived from elements $\mathcal{E}_k$ with known property values, the gradients to update the self-attention parameters are also calculated from elements $\mathcal{E}_{uk}$ with unknown property values, which essentially training the network parameters on $\mathcal{E}_{uk}$ in an unsupervised manner and convert the problem of prediction on

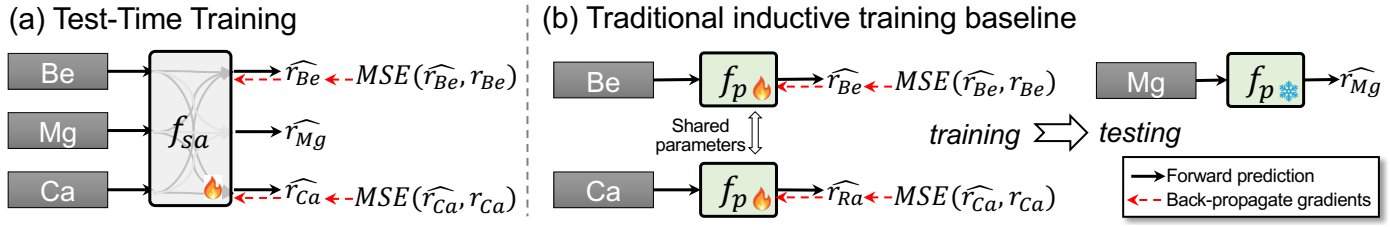

Figure 5: Pipeline comparison between (a) test-time training and (b) standard inductive training.

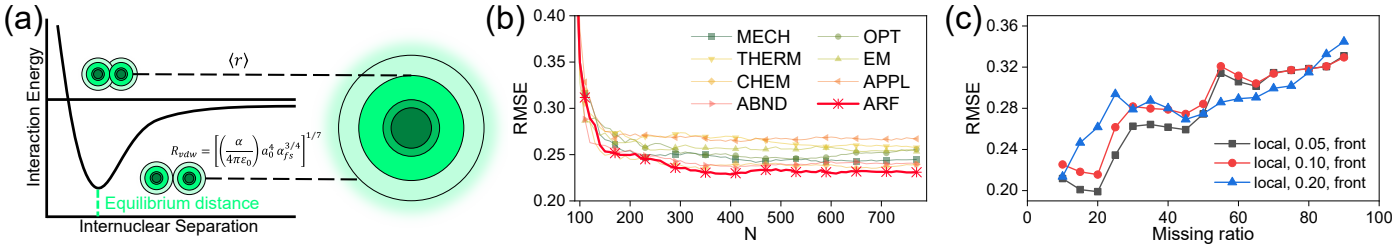

Figure 6: **Evaluation of van der Waals radius prediction via local embeddings.** (a) Schematic of interatomic interaction energy as a function of internuclear separation, showing the definition of the van der Waals radius (R_{vdW}). (b) Root mean square error (RMSE) as a function of embedding dimensions N for different local embeddings of elemental attributes, with atomic and radiational features (ARF) highlighted in red. (c) RMSE versus data missing ratio under different ways of constructing the local embeddings.

$\mathcal{E}_{uk}$ to value imputation. Consequently, the model can be directly trained from scratch and then use the model output at the last step as the final prediction, mitigating the performance drop due to scarce data and the distribution shift between $\mathcal{E}_k$ and $\mathcal{E}_{uk}$.

During implementation, as the true property values of elements in the set $\mathcal{E}_{uk}$ are naturally unknown and cannot be used to quantitatively evaluate the model performance, we instead simulate a scenario by helding out certain elements via random selection as a validation set from $\mathcal{E}_k$. Specifically, we partition $\mathcal{E}_k$ into a training set $\mathcal{E}_{tr}$ and a test set $\mathcal{E}_{te}$, such that $\mathcal{E}_k = \mathcal{E}_{tr} \cup \mathcal{E}_{te}$ and $\mathcal{E}_{tr} \cap \mathcal{E}_{te} = \emptyset$. The ratio $|\mathcal{E}_{te}|\,|\mathcal{E}_k|$ is defined as missing rate, reflecting the difficulty of the property estimation.

For material research, since the atomic number, *i.e.*, the indexes of the elements in the periodic table, is a critical factor in property prediction, the position encoding [55] should have been incorporated into our algorithm as inputs. However, this is unnecessary from our observation and we think the reason is because that the embedding extracted from the source text already encapsulate the corresponding information, thus rendering explicit encoding redundant.

4.4. prediction of atomic properties

Following Section 4.2, to avoid confounding representation changes, all subsequent regression experiments therefore use the summary-prepended setting (`local, front`). Fig. 6 turns to a representative scalar property—the van der Waals (vdW) radius—both to motivate the task and to test whether text-derived embeddings carry element-level size information.

Fig. 6(a) shows the interaction–distance curve and the R_{vdW} relation used here, underscoring why R_{vdW} matter: they enter noncovalent contact a steric accessibility, packing, and force-field parameterization across chemistry, materials, and biomolecular modeling [57, 58, 59, 60]. At the same time, R_{vdW} is difficult to determine and not uniquely defined; estimates depend on environment and methodology, often inferred from contact distributions, dispersion models, or polarizability trends rather than measured directly [61, 62, 63]. This importance–difficulty tradeoff motivates evaluating whether our embeddings can predict element-level R_{vdW}.

In Fig. 6(b), we conduct an experiment to validate whether a specific property aligns with a specific attribute (R_{vdW}-ARF). We first remove elements with any missing attribute text or missing target value, yielding 96 elements. For each of the eight `local, front` attribute embeddings, we run 10-fold cross-validation and report RMSE on held-out folds. To study feature sufficiency, in every fold, a linear regressor is trained after ranking dimensions on the training split and selecting the top N features ($N \in [100, 768]$; selection is performed within the training fold to avoid leakage), and a full-feature linear baseline is also fitted on the same split. As N increases, RMSE decreases and stabilizes; among attributes, Atomic and radiational features attain the lowest error (red curve), consistent with the expectation that R_{vdW} correlates with atomic size and electronic extent captured by this attribute.

In Fig. 6(c), we vary the summary-to-page ratio in `local, front` (0.05, 0.10, 0.20) and probe robustness under scarce data by sweeping the missing ratio (test fraction) up to 80%. For each level, we randomize train/test splits five times and average results. Because RMSE can fluctuate with feature count, we aggregate performance using the best-tail average RMSE: identify N_{best} with the minimum RMSE and average RMSE over $N \in [N_{\mathrm{best}}, N_{\max}]$. As shown in Fig. 6(c), when the missing ratio is below 50%, the 0.05 summary consistently performs best, suggesting that a shorter summary preserves attribute-specific signal while supplying a light global prior. Between 50% and 80%, the 0.20 summary becomes superior, indicating that stronger shared context helps stabilize regression when training evidence is limited and attribute coverage is uneven. Overall, summary-

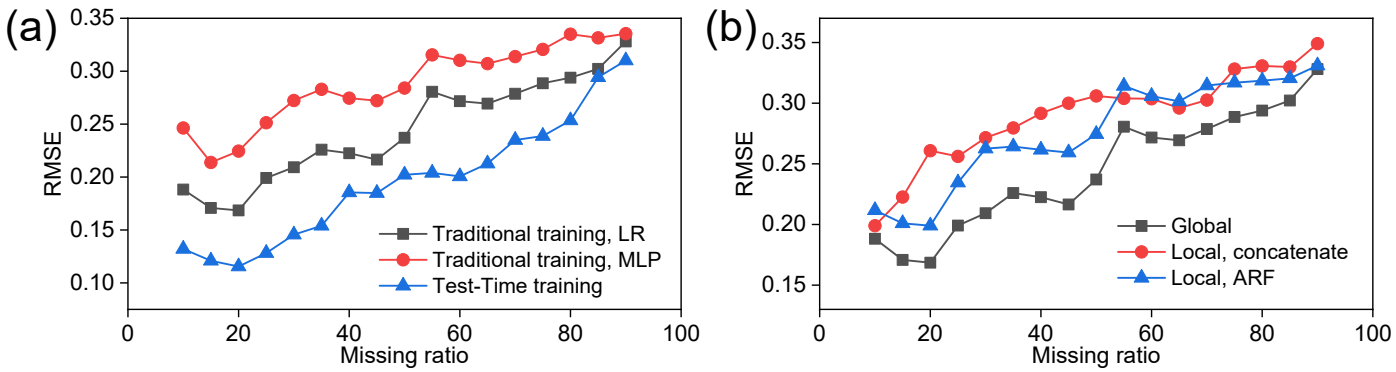

Figure 7: **Model performance comparison against an increasingly incomplete dataset.** (a) For global embedding, the RMSE vs. missing ratio obtained from traditional training and test-time training. (b) RMSE vs. missing ratio predicted via linear regression for global, local-concatenated, and local-ARF embeddings (with 0.05 summary ratio put at the front).

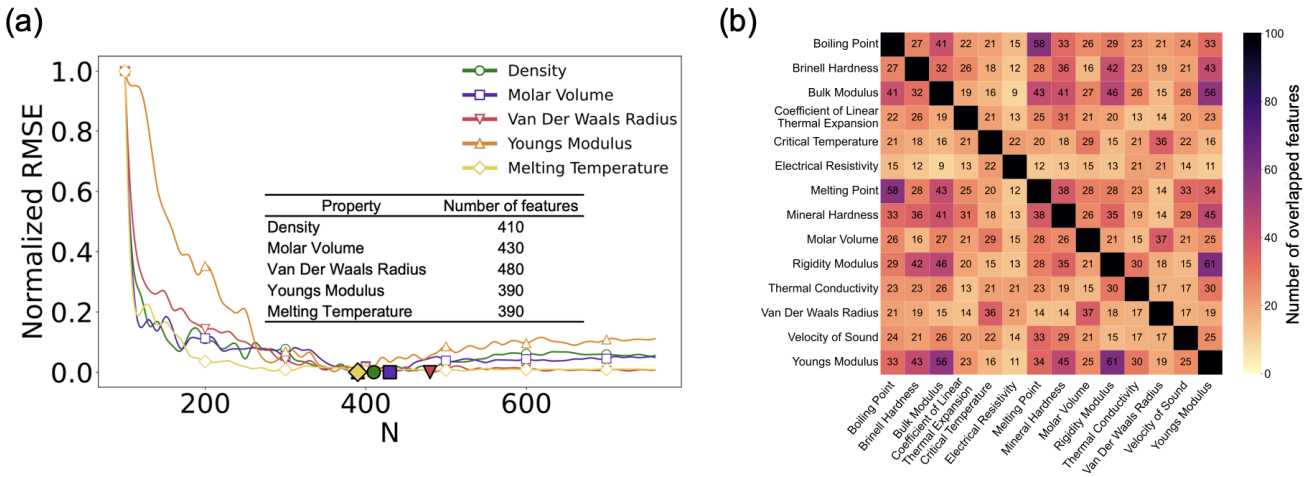

Figure 8: **The shared feature dependencies across different material properties.** (a) Normalized RMSE curves for selected materials properties, along with the corresponding saturation dimensions required to reach convergence. (b) Clustered heat map of feature-overlap similarity among the 14 selected materials properties, which is measured by the size of the intersection between their Top-k selected dimensions.

augmented Element2Vec-Locals embeddings support R_{vdW} prediction, with shorter summaries favored in data-richer regimes and longer summaries acting as a useful prior as data become scarce.

Followingly, given the limited dataset available for inferring atomic-level properties, we keep using the global embedding and then compare the performance of different predictive models in Fig. 7.

First of all, we observe that test-time training consistently performs better than the traditional training approaches (both linear regression(LR) and MLP), particularly as the missing ratio increases. When comparing LR and MLP, when the labeled data is pretty limited, aggressively increasing the size of the network will actually make the prediction even worse. In contrast, our test-time training algorithm can adapt the model effectively to mitigate the impact of incomplete inputs, leading to substantially lower RMSE values across all missing ratios, especially at the higher ones.

In Fig. 7(b), we show the trends of the influence of different embedding strategies. Fig. 6(c) has already shown that 0.05 summary putting at the front brings the lowest error among the Element2Vec-Locals. Therefore, we decide to show these Element2Vec-Locals results here for comparison against the Element2Vec-Global. Interestingly, the global embedding generally exhibits the lowest error across most missing ratios, indicating that Element2Vec-Global are inherently more accurate than localized representations for property predictions. These insights point to the combined potential of Element2Vec-Global with test-time training as a promising direction for building reliable models under real-world conditions where data incompleteness is unavoidable. The selected representative elements' predictions of R_{vdW} via Element2Vec-Global with 95% confidence interval are presented in Table A.3 in the Appendix.

Finally, to establish ranking among different global embedding dimensions for a more generalized material property prediction, we perform linear regression with a feature-budget analysis. For a given target material property p, we first fit the linear model to all 768 dimensions in the global embedding and rank

dimensions by the absolute value of standardized coefficients $w_j = |\beta_{p,j}|$. For each budget $k \in \{100, 101, \ldots, 768\}$, the linear model is retrained from scratch using only the top-k ranked dimensions $\{r_p(1), \ldots, r_p(k)\}$ (Fig. 8(a)). Generalization is assessed via 10-fold cross-validation as described above.

In Fig. 8(b), we select 14 representative material properties and visualize the overlap of their Top-k dimensions using a clustered heatmap. Interestingly, boiling point and melting point share the highest degree of overlap, reflecting their close thermodynamic relationship, whereas their intersection with other thermal properties (e.g., thermal conductivity) is minimal. This observation suggests that while some properties naturally cluster due to shared underlying physical principles, others remain more distinct. More broadly, this finding highlights how the learned embeddings capture both intrinsic scientific connections and the way such relationships are expressed in human language, thereby bridging the gap between physical attributes and linguistic representation.

5. Conclusion

In this paper, we have proposed `Element2Vec`, an LLM-based framework for embedding chemical elements that turns human-experienced knowledge into structured materials information. By leveraging the Wikipedia page as a resource, we construct both *Global* and *Local* attribute-specific embeddings that capture the correlations among elements. Our experiments demonstrate that these embeddings can recover periodic trends and improve element classification. We also showcase the stability imputation of missing values through test-time training via an attention network. Beyond performance gains, the framework integrates large language models with domain-specific structure to uncover latent knowledge and provide transferable, interpretable representations. Future directions will be addressed to build foundation model representations for more generalizable tasks in materials science, such as the prediction of material composition and the synthesis process. The work paves the way for new materials discovery by turning unstructured scientific knowledge into physics-informed representations.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (52471178).

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[2] X. Shi, L. Zhou, Y. Huang, Y. Wu, Z. Hong, A review on the applications of graph neural networks in materials science at the atomic scale, Materials Genome Engineering Advances 2 (2) (2024) e50.

[3] S. S. Omee, S.-Y. Louis, N. Fu, L. Wei, S. Dey, R. Dong, Q. Li, J. Hu, Scalable deeper graph neural networks for high-performance materials property prediction, Patterns 3 (5) (2022).

[4] E. O. Pyzer-Knapp, J. W. Pitera, P. W. Staar, S. Takeda, T. Laino, D. P. Sanders, J. Sexton, J. R. Smith, A. Curioni, Accelerating materials discovery using artificial intelligence, high performance computing and robotics, npj Computational Materials 8 (1) (2022) 84.

[5] A. D. Handoko, R. I. Made, Artificial intelligence and generative models for materials discovery–a review, arXiv preprint arXiv:2508.03278 (2025).

[6] Y.-C. Hu, J. Tian, Data-driven prediction of the glass-forming ability of modeled alloys by supervised machine learning, Journal of Materials Informatics 3 (1) (2023) N–A.

[7] Y. Li, How machine learning predicts fluid densities under nanoconfinement, arXiv preprint arXiv:2508.17732 (2025).

[8] J. Van Herck, M. V. Gil, K. M. Jablonka, A. Abrudan, A. S. Anker, M. Asgari, B. Blaiszik, A. Buffo, L. Choudhury, C. Corminboeuf, et al., Assessment of fine-tuned large language models for real-world chemistry and material science applications, Chemical science 16 (2) (2025) 670–684.

[9] Z. Xie, X. Evangelopoulos, Ö. H. Omar, A. Troisi, A. I. Cooper, L. Chen, Fine-tuning gpt-3 for machine learning electronic and functional properties of organic molecules, Chemical science 15 (2) (2024) 500–510.

[10] R. Jacobs, M. P. Polak, L. E. Schultz, H. Mahdavi, V. Honavar, D. Morgan, Regression with large language models for materials and molecular property prediction, arXiv preprint arXiv:2409.06080 (2024).

[11] B. Madika, A. Saha, C. Kang, B. Buyantogtokh, J. Agar, C. M. Wolverton, P. Voorhees, P. Littlewood, S. Kalinin, S. Hong, Artificial intelligence for materials discovery, development, and optimization, ACS nano (2025).

[12] 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 (1) (2022) 59.

[13] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al., A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, ACM Transactions on Information Systems 43 (2) (2025) 1–55.

[14] J. Ma, P.-Y. Huang, S. Xie, S.-W. Li, L. Zettlemoyer, S.-F. Chang, W.-T. Yih, H. Xu, Mode: Clip data experts via clustering, in: Proceedings of the IEEE/CVF conference on

computer vision and pattern recognition, 2024, pp. 26354–26363.

[15] F. Cardarelli, Materials handbook: a concise desktop reference, Springer, 2008.

[16] M. J. Willatt, F. Musil, M. Ceriotti, Atom-density representations for machine learning, The Journal of chemical physics 150 (15) (2019).

[17] M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[18] Y. Yang, J. Ma, S. Huang, L. Chen, X. Lin, G. Han, S.-F. Chang, Tempclr: Temporal alignment representation with contrastive learning, in: 11th International Conference on Learning Representations (ICLR 2023), International Conference on Learning Representations, ICLR, 2023.

[19] L. Jin, Z. Du, L. Shu, Y. Cen, Y. Xu, Y. Mei, H. Zhang, Transformer-generated atomic embeddings to enhance prediction accuracy of crystal properties with machine learning, Nature Communications 16 (1) (2025) 1210.

[20] X. Jiang, W. Wang, S. Tian, H. Wang, T. Lookman, Y. Su, Applications of natural language processing and large language models in materials discovery, npj Computational Materials 11 (1) (2025) 79.

[21] P. R. Kaundinya, K. Choudhary, S. R. Kalidindi, Prediction of the electron density of states for crystalline compounds with atomistic line graph neural networks (alignn), JOM 74 (4) (2022) 1395–1405.

[22] K. Choudhary, B. DeCost, Atomistic line graph neural network for improved materials property predictions, npj Computational Materials 7 (1) (2021) 185.

[23] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, A. Jain, Unsupervised word embeddings capture latent knowledge from materials science literature, Nature 571 (7763) (2019) 95–98.

[24] T. Gupta, M. Zaki, N. A. Krishnan, Mausam, Matscibert: A materials domain language model for text mining and information extraction, npj Computational Materials 8 (1) (2022) 102.

[25] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[26] L. Ward, A. Dunn, A. Faghaninia, N. E. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, et al., Matminer: An open source toolkit for materials data mining, Computational Materials Science 152 (2018) 60–69.

[27] M. R. Lemes, A. Dal Pino, Periodic table of the elements in the perspective of artificial neural networks, Journal of Chemical Education 88 (11) (2011) 1511–1514.

[28] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.

[29] L. M. Antunes, R. Grau-Crespo, K. T. Butler, Distributed representations of atoms and materials for machine learning, npj Computational Materials 8 (1) (2022) 44.

[30] I. Jolliffe, Principal component analysis, in: International encyclopedia of statistical science, Springer, 2011, pp. 1094–1096.

[31] V. Venugopal, E. Olivetti, Matkg: An autonomously generated knowledge graph in material science, Scientific Data 11 (1) (2024) 217.

[32] Y. Ye, J. Ren, S. Wang, Y. Wan, I. Razzak, B. Hoex, H. Wang, T. Xie, W. Zhang, Construction and application of materials knowledge graph in multidisciplinary materials science via large language model, in: A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems, Vol. 37, Curran Associates, Inc., 2024, pp. 56878–56897.

[33] J. Ma, H. Xie, G. Han, S.-F. Chang, A. Galstyan, W. Abd-Almageed, Partner-assisted learning for few-shot image classification, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10573–10582.

[34] K. Yan, C. Fu, X. Qian, X. Qian, S. Ji, Complete and efficient graph transformers for crystal material property prediction, in: The Twelfth International Conference on Learning Representations, 2024.

[35] P. Reiser, et al., Graph neural networks for materials science and chemistry, Nature Reviews Materials (2022).

[36] K. Das, S. Khastagir, P. Goyal, S.-C. Lee, S. Bhattacharjee, N. Ganguly, Periodic materials generation using text-guided joint diffusion model, arXiv preprint arXiv:2503.00522 (2025).

[37] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM computing surveys 55 (9) (2023) 1–35.

[38] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, E. Chen, Large language models for generative information extraction: A survey, Frontiers of Computer Science 18 (6) (2024) 186357.

[39] J. Dagdelen, A. Dunn, S. Lee, N. Walker, A. S. Rosen, G. Ceder, K. A. Persson, A. Jain, Structured information extraction from scientific text with large language models, Nature Communications 15 (1) (2024) 1418.

[40] Q. Chen, Y. Hu, X. Peng, Q. Xie, Q. Jin, A. Gilson, M. B. Singer, X. Ai, P.-T. Lai, Z. Wang, et al., Benchmarking large language models for biomedical natural language processing applications and recommendations, Nature communications 16 (1) (2025) 3280.

[41] C. Ling, X. Zhao, J. Lu, C. Deng, C. Zheng, J. Wang, T. Chowdhury, Y. Li, H. Cui, X. Zhang, et al., Domain specialization as the key to make large language models disruptive: A comprehensive survey, ACM Computing Surveys (2023).

[42] Y. Zhang, L. Chen, S. Li, N. Cao, Y. Shi, J. Ding, Z. Qu, P. Zhou, Y. Bai, Way to specialist: Closing loop between specialized llm and evolving domain knowledge graph (2024). arXiv:2411.19064.

[43] X. Gong, M. Liu, X. Chen, Large language models with knowledge domain partitioning for specialized domain knowledge concentration (Jun 2024). doi:10.31219/osf.io/srgpx.

[44] S. Gupta, A. Mahmood, P. Shetty, A. Adeboye, R. Ramprasad, Data extraction from polymer literature using large language models, Communications materials 5 (1) (2024) 269.

[45] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Improving text embeddings with large language models (2024). arXiv:2401.00368.

[46] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems 26 (2013).

[47] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in: International conference on learning representations, 2017.

[48] L. Logeswaran, H. Lee, An efficient framework for learning sentence representations, arXiv preprint arXiv:1803.02893 (2018).

[49] I. Yamada, A. Asai, J. Sakuma, H. Shindo, H. Takeda, Y. Takefuji, Y. Matsumoto, Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia, arXiv preprint arXiv:1812.06280 (2018).

[50] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, H. Wu, Ernie: Enhanced representation through knowledge integration, arXiv preprint arXiv:1904.09223 (2019).

[51] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang, Kepler: A unified model for knowledge embedding and pre-trained language representation, Transactions of the Association for Computational Linguistics 9 (2021) 176–194.

[52] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Improving text embeddings with large language models, arXiv preprint arXiv:2401.00368 (2023).

[53] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, M. Zhang, Towards general text embeddings with multi-stage contrastive learning, arXiv preprint arXiv:2308.03281 (2023).

[54] J. Lee, F. Chen, S. Dua, D. Cer, M. Shanbhogue, I. Naim, G. H. Ábrego, Z. Li, K. Chen, H. S. Vera, X. Ren, S. Zhang, D. Salz, M. Boratko, J. Han, B. Chen, S. Huang, V. Rao, P. Suganthan, F. Han, A. Doumanoglou, N. Gupta, F. Moiseev, C. Yip, A. Jain, S. Baumgartner, S. Shahi, F. P. Gomez, S. Mariserla, M. Choi, P. Shah, S. Goenka, K. Chen, Y. Xia, K. Chen, S. M. K. Duddu, Y. Chen, T. Walker, W. Zhou, R. Ghiya, Z. Gleicher, K. Gill, Z. Dong, M. Seyedhosseini, Y. Sung, R. Hoffmann, T. Duerig, Gemini embedding: Generalizable embeddings from gemini (2025). arXiv:2503.07891.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[56] J. Ma, Z. Shou, A. Zareian, H. Mansour, A. Vetro, S.-F. Chang, Cdsa: cross-dimensional self-attention for multivariate, geo-tagged time series imputation, arXiv preprint arXiv:1905.09904 (2019).

[57] Y. V. Zefirov, P. Zorkii, Van der waals radii and their application in chemistry, Russian Chemical Reviews 58 (5) (1989) 421.

[58] S. Vemparala, B. B. Karki, R. K. Kalia, A. Nakano, P. Vashishta, Large-scale molecular dynamics simulations of alkanethiol self-assembled monolayers, The Journal of chemical physics 121 (9) (2004) 4323–4330.

[59] A. G. Street, S. L. Mayo, Pairwise calculation of protein solvent-accessible surface areas, Folding and Design 3 (4) (1998) 253–258.

[60] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, et al., Charmm general force field: A force field for drug-like molecules compatible with the charmm all-atom additive biological force fields, Journal of Computational Chemistry 31 (4) (2010) 671–690.

[61] M. Mantina, A. C. Chamberlin, R. Valero, C. J. Cramer, D. G. Truhlar, Consistent van der waals radii for the whole main group, The Journal of Physical Chemistry A 113 (19) (2009) 5806–5812.

[62] A. Tkatchenko, M. Scheffler, Accurate molecular van der Waals interactions from ground-state electron density and free-atom reference data, Physical Review Letters 102 (7) (2009) 073005.

[63] M. Rahm, R. Hoffmann, N. Ashcroft, Atomic and ionic radii of elements 1–96, Chemistry–A European Journal 22 (41) (2016) 14625–14632.

Appendix A. Learning elemental properties by multi-layer perceptron

The multilayer perceptron (MLP) used in this study has a three-layer architecture consisting of an input layer, a single hidden layer with 100 neurons using ReLU activation, and an output layer. It is configured for efficient training using the Adam optimizer with very weak L2 regularization (alpha=0.0001). The model employs a maximum of 500 iterations and has early stopping enabled to prevent overfitting by halting training if the validation score stops improving. Given that the abbreviations of attribute names are summarized in Table A.1, we summarize the statistics of word count for each local attributes in the Table A.2. The detailed performance of all local embedding datasets are summarized in Fig. A.1.

Table A.1: Attribute names and abbreviations

Full name	Abbr.
Mechanical properties (Attr.1)	MECH
Optical properties (Attr.2)	OPT
Electrical & Magnetic properties (Attr.3)	EM
Thermal properties (Attr.4)	THERM
Chemical properties (Attr.5)	CHEM
Atomic & radiational features (Attr.6)	ARF
Applications (Attr.7)	APPL
Abundance (Attr.8)	ABND

Table A.3: R_{vdW}**: ground truth and predicted values with 95% confidence interval.** The values in the table are in Å; 1 Å = 10^{-10} m.

Element	True Value	Predicted Value
He	1.40	1.42 ± 0.11
C	1.70	1.65 ± 0.12
Ca	2.31	2.27 ± 0.06
Ar	1.88	1.78 ± 0.10
Br	1.85	1.88 ± 0.08
Au	2.14	2.11 ± 0.06
Nb	2.18	2.31 ± 0.04
Sm	2.36	2.38 ± 0.06

Table A.2: **Word count of Element2Vec-Locals dataset with different summary ratio(S/R)**

S/R	Elem.	Attr.1	Attr.2	Attr.3	Attr.4	Attr.5	Attr.6	Attr.7	Attr.8	Sum.
0.01	He	1234	212	540	2202	2081	731	2741	3112	442
	C	1463	666	766	1535	7619	5652	4168	5677	291
	Ca	414	37	185	363	3314	1837	1524	1569	244
	Ar	182	656	987	579	4155	1823	3537	1508	162
	Br	260	520	801	1475	9146	843	4411	1858	345
	Au	1622	1217	1642	172	4782	1009	6346	6019	567
	Nb	1294	617	2254	711	3856	582	2764	1086	256
	Sm	613	1764	1960	2565	4750	3067	3468	1583	216
0.05	He	2519	901	758	1950	2644	2551	3043	5736	1119
	C	1790	675	721	1452	6642	3761	4302	5691	1821
	Ca	616	34	652	257	6012	3498	4215	709	1408
	Ar	59	629	464	404	2655	1357	1560	644	646
	Br	114	730	381	1664	9318	585	3561	1330	1730
	Au	1425	1239	1897	273	5050	1885	6014	5381	2835
	Nb	1245	1016	1957	702	3589	1107	5424	718	822
	Sm	672	1853	2768	1299	4883	3682	3482	1927	1325
0.1	He	1996	800	1065	3663	3196	5414	5150	5838	2027
	C	2658	746	714	1365	7653	6621	2518	5445	2620
	Ca	793	34	620	478	5828	2784	3622	1795	2817
	Ar	59	568	424	432	2674	1585	1467	486	1143
	Br	244	647	703	3237	10074	957	4476	1525	3336
	Au	1333	1061	1582	49	5752	1090	10126	6468	4486
	Nb	1209	527	2225	529	4204	449	4761	888	1504
	Sm	613	1545	2066	1554	4624	4690	3826	2366	2650
0.2	He	1608	470	802	3083	2249	1971	1872	3565	3383
	C	1801	580	624	1182	7590	5409	2836	6681	2444
	Ca	553	87	510	342	7695	3992	3952	1738	4369
	Ar	251	471	403	534	2136	1735	2499	633	1380
	Br	629	420	642	2417	11770	761	3657	1281	5104
	Au	1950	1238	2060	278	4604	1060	5453	5570	4110
	Nb	1051	814	2273	629	3732	583	3908	1063	3040
	Sm	487	1451	2527	1053	5516	4218	3775	2107	2306

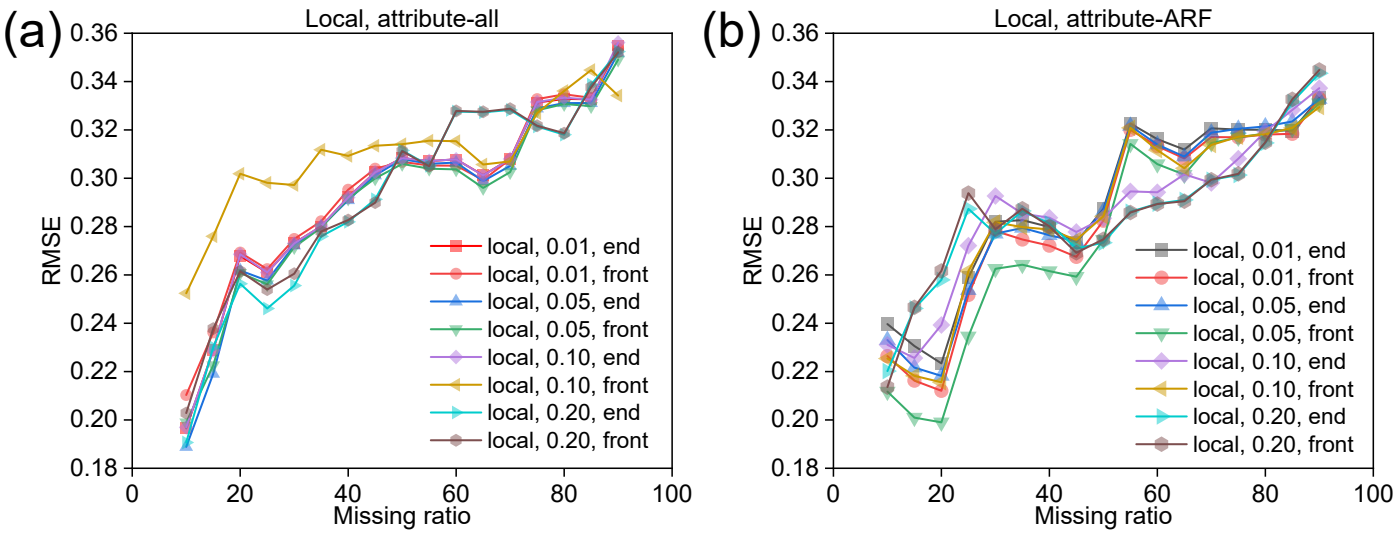

Fig. A.1: (a) The LR result of all Element2Vec-Locals embedding datasets with full attributes vector. (B) The LR result of all Element2Vec-Locals embedding datasets with only ARF attribute vector.