Hierarchical Network Fusion for Multi-Modal Electron Micrograph Representation Learning with Foundational Large Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Characterizing materials with electron micrographs is a crucial task in fields such as semiconductors and quantum materials. The complex hierarchical structure of micrographs often poses challenges for traditional classification methods. In this study, we propose an innovative backbone architecture for analyzing electron micrographs. We create multi-modal representations of the micrographs by tokenizing them into patch sequences and, additionally, representing them as vision graphs, commonly referred to as patch attributed graphs. We introduce the Hierarchical Network Fusion (HNF), a multi-layered network structure architecture that facilitates information exchange between the multi-modal representations and knowledge integration across different patch resolutions. Furthermore, we leverage large language models (LLMs) to generate detailed technical descriptions of nanomaterials as auxiliary information to assist in the downstream task. We utilize a cross-modal attention mechanism for knowledge fusion across cross-domain representations(both image-based and linguistic insights) to predict the nanomaterial category. This multi-faceted approach promises a more comprehensive and accurate representation and classification of micrographs for nanomaterial identification. Our framework outperforms traditional methods, overcoming challenges posed by distributional shifts, and facilitating high-throughput screening.

1 Introduction

Semiconductors are the foundation of modern electronics, driving advancements in computing, communication systems, transportation systems, and space exploration. The precise design, development, and testing of semiconductor devices is essential for ensuring the reliability, durability, and performance of high-tech chips. Advanced imaging and analysis techniques[55] are key to fabricating and integrating nanoscale components and enabling advanced inspection, which is essential for the development of next-generation miniaturized semiconductor devices[13], with sizes now reaching as small as 7 nm or even smaller. However, the increased complexity of producing chips under 7 nanometers introduces greater potential for error, jeopardizing the consistency of high-quality chip production and magnifying variability in chip performance. The semiconductor industry utilizes various electron beam tools, including scanning and transmission electron microscopy, to create high-resolution images of these devices. These images, known as electron micrographs, reveal the complex microstructures of materials, which are crucial for the accurate design and evaluation of semiconductor devices. The fabrication of nanoscale components is a challenging task that requires precise control over the manufacturing process. Furthermore, these images facilitate monitoring of the process and defect detection, enabling subsequent process optimization or design adjustments to mitigate defects. The autolabeling of electron micrographs for nanomaterial identification, while advantageous, remains a significant challenge. Figure 1 shows the challenges in nanomaterial identification tasks. This is largely attributed to distributional shifts such as manufacturing variations or material property changes, exacerbated by high intra-class dissimilarity within nanomaterials, high inter-class similarity between different nanomaterials, and the existence of visual patterns at

Submitted to the Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023. Do not distribute.

multiple scales or spatial heterogeneity. To overcome the challenges in this work, we propose an end-to-end framework for automatic nanomaterial identification based on hierarchical network fusion for multi-modal electron micrograph representation learning with large language models (referred to as “MultiFusion-LLM" for shorthand notation). We hypothesize that electron micrographs exhibit hierarchical dependencies among patches (segmented portions of an electron micrograph). These dependencies can be captured using multiple patch sequences and vision graph structures at different spatial resolutions of the patches. To explore this, we tokenize the electron micrographs into grid-like patches to obtain a patch sequence. Additionally, we represent the micrograph as a vision graph, where patches are connected by undirected edges that represent pairwise visual similarity. Figure 6 shows the modalities (patch sequence, graph) that offer unique insights and assist in capturing complex patterns. We introduce a $<cls>$ token to the patch sequence and a virtual node to the vision graph. This special token/virtual node encapsulates the entire patch sequence and captures global graph information in their respective contexts. We aim to capture fine- and coarse-grained hierarchical dependencies by treating the micrographs as sequence structures and vision graphs at multiple scales of patch size. The main contributions of this work can be summarized:

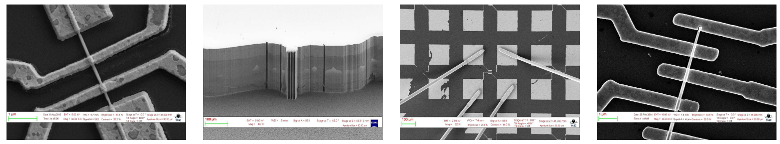
- ✓ We have developed the Hierarchical Network Fusion (HNF), a cascading network architecture that enhances the classification accuracy by analyzing and integrating two complementary representations of electron micrographs: patch sequences and vision graphs, which are created at various patch sizes. Vision graphs, constructed using a nearest-neighbor graph technique, identify local patch relationships and capture graph-structured priors. Meanwhile, patch sequences help in capturing spatial dependencies between various patches in a micrograph, going beyond the limitations of sparse graph structure priors. The HNF is a multi-layered network featuring an inverted pyramid architecture that generates a multi-scale representation of an electron micrograph by creating a series of patch sequences and vision graphs at different scales of patch size. This inverted pyramid is constructed by progressively increasing the patch size at each layer. Each layer of the pyramid represents the original micrograph-based patch sequence and vision graph at a distinct scale, offering increasingly higher resolutions. By considering information at multiple scales, the HNF facilitates a more comprehensive representation of the electron micrograph, capturing both fine- and coarse-grained details. At each layer, the patch embeddings are iteratively refined using bidirectional Neural Ordinary Differential Equations (Neural ODEs) [20], while the Graph Chebyshev Convolution (GCC) Networks [51, 28] encode the vision graphs in a layer-wise manner to compute graph-level embeddings. A mixture-of-experts (MOE) technique with a gating mechanism optimally combines predictions from both modalities at each layer by calculating a weighted sum of classification token and virtual node embedding to improve classification accuracy. This facilitates an intermodal mutual information exchange, fostering interaction and knowledge integration between the two modalities. This innovative approach enables the seamless integration of causal information from patch sequences to refine the vision graph embeddings, and structural and semantic information from visual graphs to ground the patch embeddings, fostering enhanced interaction and knowledge fusion within the architecture. Our framework constructs a multi-scale representation of a micrograph with the aim of optimally preserving both the high-level features and structural information embedded in the graphs, as well as the causal relations embedded in the patch sequences, thereby enabling a more comprehensive representation of the micrograph.
- ✓ Our approach utilizes Zero-shot Chain-of-Thought (Zero-Shot CoT) prompting with large language models (LLMs)[10, 25, 93] to generate technical descriptions of nanomaterials, including synthesis methods, properties, and applications. We pre-train smaller language models (LMs) [30, 52] through self-supervised masked language modeling (MLM)[5, 30] on these generated textual descriptions, enabling domain-specific customization for improved language understanding. Subsequently, we fine-tune the pre-trained LMs for task-adaptation to compute contextualized token embeddings for nanomaterial identification tasks. We employ a weighted sum-pooling attention mechanism to compute text-level embeddings from token embeddings, encapsulating the vast domain-specific knowledge present in the text data. Our approach leverages LLM-based technical descriptions on nanomaterials to identify characteristic features that distinguish them from other nanomaterial categories, incorporating domain-specific knowledge as auxiliary information for downstream training.

2 Problem Statment

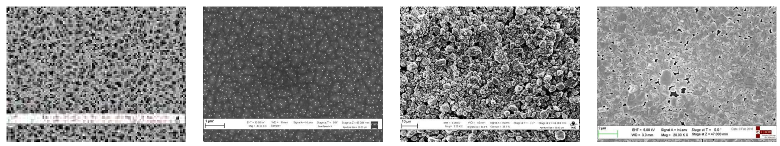
In this study, the focus is on the electron micrograph classification task, a type of inductive learning task where the objective is to assign labels to new, unseen micrographs utilizing a labeled dataset denoted as $\mathcal{D}_L = (\mathcal{I}_L, \mathcal{Y}_L)$. A multi-modal encoder, formulated as the non-linear function $f_\gamma : \mathcal{I} \rightarrow \mathcal{Y}$ is trained on labeled dataset to predict labels ($\mathcal{Y}_U$) of unlabeled micrographs ($\mathcal{I}_U$). Here, γ denotes the trainable parameters. The objective is to minimize the loss function $\mathcal{L}_\mathcal{I}$, which is articulated as

$$\min_{\gamma} \mathcal{L}_\mathcal{I} \left(\mathcal{I}_i, \gamma\right) = \sum_{(\mathcal{I}_i, y_i) \in \mathcal{D}_L} \ell\big(f_\gamma(\mathcal{I}_i), y_i\big) \tag{1}$$

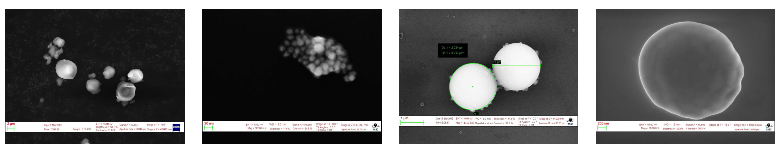
where $y_i^{\text{pred}} = f_\gamma(\mathcal{I}_i)$ denote the multi-modal encoder predictions and $\ell(\cdot, \cdot)$ denotes the cross-entropy loss.



(a) High intra-class dissimilarity: The electron micrographs of the same nanomaterial (*MEMS* device) can exhibit a high degree of heterogeneity.



(b) High inter-class similarity: Electron micrographs across different nanomaterial categories (*listed from left to right as porous sponges, particles, powders, and films*) exhibit a noteworthy degree of similarity.



(c) Multi-spatial scales of patterns: The spatial heterogeneity of visual patterns in electron micrographs of *nanoparticles* is evident.

Figure 1: The figure provides a visual representation of the challenges of classifying electron micrographs in the SEM dataset([4]).

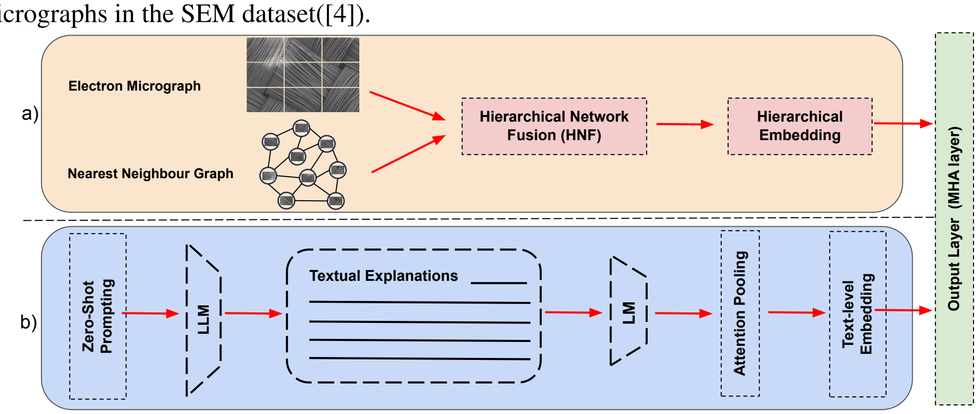


Figure 2: Our framework includes three methods: (a) Hierarchical Network Fusion (HNF), (b) Zero-shot Chain-of-Thought (Zero-Shot CoT) prompting with large language models (LLMs), and (c) an output layer modeled with the multi-head attention (MHA) mechanism [95] for integrating cross-domain embeddings and facilitating label prediction. LLMs take a prompt, not an electron micrograph, as input.

3 Proposed Method

3.1 Formalism

Let's consider an input electron micrograph denoted by $\mathbf{I}'''$, which has dimensions of $h \times w \times c$, where h, w, and c represent the height, width, and number of channels of the micrograph, respectively. We divide the micrograph into a grid of patches, each having dimensions of $p \times p \times c$, with p representing the patch size. The number of patches along each spatial dimension is given by $n = hw/p^2$. Subsequently, we reshape the 3D micrograph into a 2D patch tensor, denoted as $\mathbf{I}'' \in \mathbb{R}^{n \times (p^2 c)}$. These patches are linearly transformed to create a new tensor, $\mathbf{I}' \in \mathbb{R}^{n \times d}$, where d is the patch embedding dimension. To account for the position of each patch within the micrograph, we introduce position embeddings represented by a matrix $\mathbf{E}_{pos} \in \mathbb{R}^{n \times d_{pos}}$, where d_{pos} denotes the position embedding dimension. We then add the position embedding matrix to the transformed patch tensor $\mathbf{I}'$, resulting in the final tensor $\mathbf{I} \in \mathbb{R}^{n \times d}$. In general, $d_{pos} = d$. Finally, we construct a k-nearest neighbors graph to analyze the pairwise relationships between micrograph patches. This vision graph, denoted as $\mathcal{G}$, is undirected and represents the connectivity of patches based on their pairwise proximity. The graph structure is described by a binary adjacency matrix, $A \in \mathbb{R}^{n \times n}$. If patch j is one of the k-nearest neighbors of patch i, then $A_{ij} = 1$; otherwise, $A_{ij} = 0$.

3.2 Hierarchical Network Fusion(HNF)

We tokenize electron micrographs by dividing them into grid-like patches. This approach yields two complementary representations of micrographs: (a) We represent an electron micrograph as a vision graph, where patches are connected by edges that represent pairwise visual similarity constructed using a nearest-neighbor graph technique. The vision graph captures local patch relationships and utilizes graph-structural priors to analyze pairwise spatial dependencies within the micrograph. (b) Additionally, we represent electron micrographs as a patch sequence, capturing pairwise spatial dependencies beyond the original sparse graph structure between different patches within a micrograph. Representing electron micrographs as both patch sequences and vision graphs serves distinct purposes in their respective contexts. We append a classification token (<*cls*>) to a patch sequence to obtain an embedding of the entire patch sequence that captures global information. We augment each vision graph by introducing a virtual node that is bidirectionally connected to all the other nodes in the graph through virtual edges. These virtual edges represent the pairwise relations between each real node and the virtual node. The virtual node embedding captures the long-range dependencies between nodes by considering the global information of the vision graph. We hypothesize that electron micrographs exhibit hierarchical dependencies among patches, which can be captured using multiple patch sequences or vision graph structures at different spatial resolutions of the patches. We present Hierarchical network fusion (HNF), a cascading network architecture that constructs a multi-scale representation of an electron micrograph by creating a series of patch sequences and vision graphs at multiple scales of patch sizes with increasing resolutions.

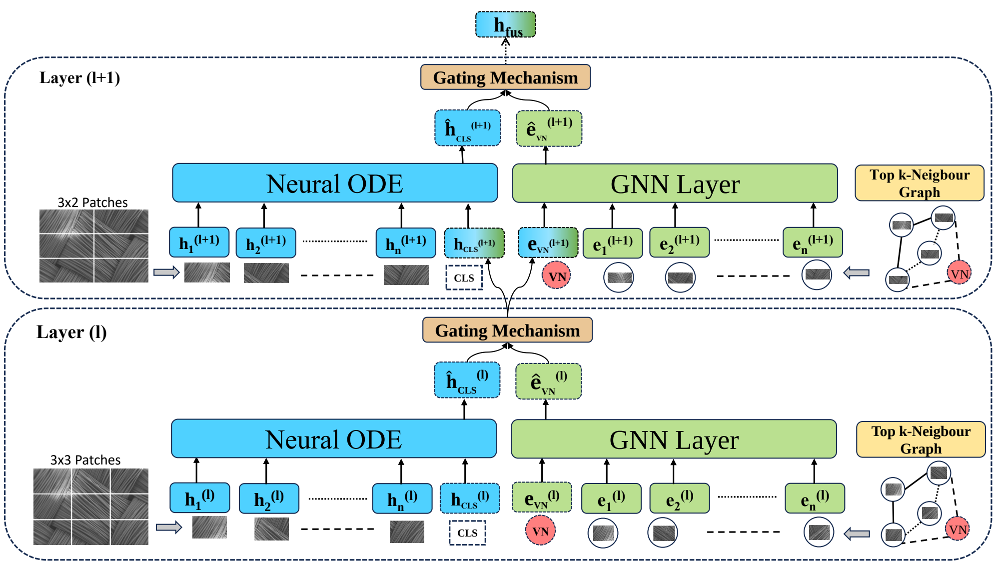


Figure 3: Overview of the HNF module. The HNF module utilizes a multi-layered network with increasing patch sizes to represent the electron micrograph-based patch sequence and vision graph at various scales, facilitating computation of hierarchical embeddings that encapsulate the global context. The cascaded structure incorporates multiple stacked layers; each layer involves bidirectional Neural ODEs and Graph Chebyshev convolution to compute patch sequence and vision graph embeddings, respectively. A gating mechanism integrates these cross-domain embeddings, generating unified hierarchical embeddings that offer a comprehensive view of the electron micrographs. Overall, the HNF module, facilitates seamless information fusion at multiple scales, producing a cohesive representation of the micrographs. <*cls*> is the cls token and VN is the virtual node. h_i^l and e_i^l denotes the patch and node representation at layer l of patch or node i, respectively.

The HNF architecture synergistically combines patch sequences and vision graphs representations at different scales, enhancing electron micrograph analysis by seamlessly integrating global insights through a multi-layered network structure. The layers are constructed by progressively increasing the patch size. Each layer of the network represents the original micrograph-based patch sequence and vision graph at different scales, with increasing resolutions. By considering information at multiple scales, the network offers a more comprehensive representation of the micrograph, capturing both fine-grained details and the global context. Figure 3 illustrates the Hierarchical Network Fusion (HNF) method. Each layer uses a bidirectional Neural ODE (refer to the appendix) to iteratively refine patch embeddings, facilitating the smooth, causal evolution of the patch embedding and capturing global inter-patch relationships and dependencies. It also incorporates a Graph Chebyshev Convolution Network (refer to the appendix) that maps the high-dimensional discrete vision graph information to low-dimensional node-level embeddings while optimally preserving the high-level visual features and structural information embedded in the graphs. Additionally, at each layer, the mixture-of-experts (MOE) technique employs a gating mechanism to combine predictions from the bidirectional Neural ODEs and the Graph Chebyshev Convolution methods. These predictions are integrated through a weighted sum of their $<cls>$ token and virtual node embeddings. The training objectives include optimizing the weight distribution of the gating function for accurate classification of nanomaterial categories in electron micrographs and training the methods using the weights determined by the gating function. Overall, our framework aims to improve classification accuracy by leveraging the strengths of multiple learning methods and optimizing the weights of the gating mechanism, which serves as the bottleneck through which the two modalities interact to obtain the fused representation. In the subsequent layers, the fused information is combined with the individual modalities at higher patch resolutions. Our framework incorporates bidirectional Neural ODEs and Graph Chebyshev networks to facilitate the exchange of mutual information between patch sequences and visual graphs across multiple scales of patch size through the gating mechanism. This approach allows the patch embeddings to be grounded with structural and semantic information from the vision graph while enabling causal relations within the patch sequence to transform the graph embeddings. Overall, the framework fosters interactive knowledge integration between modalities within its architecture.

3.3 Beyond Conventional Analysis: Leveraging LLMs for Nanomaterial Characterization

The advent of large pre-trained language models (LLMs), such as OpenAI's ChatGPT [10], Google's PaLM [25], and Meta's LLaMA [93], has significantly revolutionized performance in various natural language processing tasks, achieving state-of-the-art results across a wide range of applications. In contrast, small-scale language models (LMs), such as BERT [30] and DeBERTa [52], lack the strong logical reasoning capabilities of LLMs and are limited in their ability to generate coherent and contextually relevant responses compared to larger models. However, small-scale LMs are computationally affordable for fine-tuning using labeled data for specialized task adaptation. In addition, they allow access to logits or token embeddings for downstream applications of smaller LMs across various tasks, aiding in explainability. Owing to their substantial model complexity and scale, general-purpose LLMs require significant computational resources for repurposing through fine-tuning for task-specific customization. Additionally, they do not provide access to latent token embeddings and logits, this black-box nature can limit the interpretability of LLMs. To overcome the challenges, the Language Modeling as a Service (LMaaS [83]) platform provides access to LLMs via text-based API interaction through cloud-based services. However, the integration of LLMs with vision graphs remains an underexplored area, opening up the possibility for innovative techniques that combine language models and graph representation learning algorithms to improve nanomaterial identification applications. To address this, our approach capitalizes on zero-shot chain-of-thought (Zero-Shot CoT) prompting of LLMs to generate technical descriptions of nanomaterials. We pre-train smaller LMs on the generated textual descriptions using the masked language modeling (MLM) technique (i.e., pre-training for domain-customization) to learn expressive token embeddings for a better understanding of language structure and semantics. We then fine-tune smaller LMs for downstream supervised multi-class classification task (i.e., fine-tuning for task adaptation) to compute context-aware token embeddings. We employ weighted sum-pooling attention mechanisms to obtain contextualized text-level embeddings from token embeddings, which are used to perform inference in the nanomaterial identification task. Our work evaluates two LLMs: GPT-3.5-turbo, and Google BARD[1]. GPT-3.5-turbo, a newer and larger extension of GPT-3.5 model from OpenAI, excels in various language tasks and shows cost-effectiveness, while Google BARD is significantly larger than GPT-3.5 models. We also utilize a pre-trained small-scale LM, DeBERTa[2] [52], which is an improved version of the BERT architecture. The technical details of these language models are given in Table 1.

[1] https://bard.google.com

[2] For more information, refer to the DeBERTa model documentation available at `https://huggingface.co/docs/transformers/index`.

In the GPT-3.5-turbo and BARD, text generation diversity is mainly influenced by two parameters: Top-p (nucleus sampling) and temperature. Top-p sets a probability threshold for token inclusion, filtering out excessively rare or common tokens to balance the output. The temperature parameter dictates the randomness of generated text; high values foster creativity, while low values ensure focused and deterministic outputs. In our experiments, we set Top-p to 1 and temperature to 0 for accurate and controlled text generation.

Table 1: Technical specifications of the LLMs and LMs. The *Cost* category indicates the price for using 1k tokens, while the *Date of Last Update* category denotes the the most recent date the knowledge base of the LLMs was updated.

Model	Organization	Cost	Date of Last Update	Vocabulary Size
ChatGPT	Open-AI	0.002$	Jun. 2021	175B
BARD	Google	Free	Undisclosed	1,560B
DeBERTa	Hugging Face	Free	N/A	50M

Zero-Shot CoT LLMs Prompting: We access LLMs via the LMaaS platform, using text-based API interactions. We employ open-ended natural language prompts with task-specific instructions to query the LLMs, thereby generating detailed textual descriptions pertaining to the structure, properties, and applications of given nanomaterials. Utilizing a tailored zero-shot prompt template, we guide the LLMs through a series of chain-of-thought prompts[101], extracting comprehensive domain knowledge embedded within the language model parameters to generate rich, detailed technical descriptions of nanomaterials. The customized CoT prompt format is as follows:

> **Prompt 1:** Introduction: Provide an overview of the nanomaterial category and its significance in various fields. **Prompt 2:** Definition and Structure: Define the nanomaterial category and describe its typical structure at the nanoscale. **Prompt 3:** Synthesis Methods: Explore different methods used to synthesize or fabricate nanomaterials in this category. Discuss their advantages and limitations. **Prompt 4:** Properties: Highlight the unique physical, chemical, and electronic properties exhibited by nanomaterials in this category. Discuss how these properties differ from their bulk counterparts. **Prompt 5:** Applications: Explore the wide range of applications where nanomaterials in this category are utilized. Discuss their potential impact in fields such as electronics, energy, medicine, environmental remediation, etc. **Prompt 6:** Surface Modification: Describe the strategies used to modify the surface properties of nanomaterials in this category, such as functionalization, coating, or doping. Explain how these modifications enhance their performance or enable specific applications. **Prompt 7:** Toxicity and Safety: Address the potential health and environmental concerns associated with nanomaterials in this category. Discuss studies on their toxicity, risk assessment, and safety measures to mitigate any potential hazards. **Prompt 8:** Future Directions: Discuss current research trends and future prospects for nanomaterials in this category. Highlight emerging technologies, challenges, and areas of active exploration.

Querying the LLMs generates technical descriptions of nanomaterial categories. It provides valuable insights into the characteristics, properties, and applications of different types of nanomaterials.

> (**LLMs Response**) [Textual Outputs]

In the following section, we will present our approach to integrating detailed textual descriptions into a small-scale LM for pre-training through the masked language modeling (MLM) technique, and fine-tuning for domain customization on the downstream supervised nanomaterial identification task.

Domain Customization: Fine-Tuning LMs Our approach employs a smaller language model (LM) to interpret and encode the textual outputs generated by a larger language model (LLM). We leverage the smaller LM as an intermediate network to bridge the LLMs and downstream classification layers. The encoder-only LMs[77] are fine-tuned using a self-supervised learning approach known as masked language modeling (MLM). In this approach, the large corpus of LLM textual outputs is processed by randomly masking out tokens in each sentence. The model is then trained to predict the masked words, given the context of the surrounding non-masked words. This process helps the model learn the statistical relationships between words and phrases, thereby facilitating the generation of coherent language representations. Briefly, we pre-train smaller general-purpose language models (referred to as LM_{expl}) using the MLM technique for domain customization, enhancing language-based contextual understanding and semantic relationship extraction for aiding downstream applications. We then fine-tune the smaller LM for downstream task-specific adaptation to encapsulate the explanations generated by LLMs. Post pre-training on MLM technique, we input the text sequences generated by LLMs (denoted as $\mathcal{S}_{\text{expl}}$) into the LM_{expl} model, which then generates expressive, context-aware embeddings for each token in the sentence, capturing the semantic relationships between the tokens as follows:

$$h_{\text{expl}} = \text{LM}_{\text{expl}}(s_{\text{expl}}) \tag{2}$$

where the context-aware embeddings are denoted as $h_{\text{expl}} \in \mathbb{R}^{m \times d}$, where m represents the number of tokens in $\mathcal{S}_{\text{expl}}$ and d is token embedding dimension. We then perform sum-pooling attention mechanism to compute a weighted sum of these token embeddings to encode the textual explanations to obtain an text-level fixed-length embedding as follows:

$$\alpha_i = \text{softmax}(q_i); \quad q_i = \mathbf{u}^T h_{\text{expl}}^{(i)} \tag{3}$$

$$h^{\text{text}} = \sum_{i=0}^{m} \alpha_i h_{\text{expl}}^{(i)} \tag{4}$$

where $\mathbf{u}$ is a differentiable vector. The text-level embedding $h^{\text{text}} \in \mathbb{R}^d$ captures the essence or core of the domain knowledge as a whole, extracted from the foundational LLMs for each nanomaterial. We calculate the relevance score between the text-level embedding(h^{text}) and the electron micrograph representations(h_{fus}) obtained from the hierarchical network fusion(HNF, refer to section 3.2), as detailed below,

$$\beta^* = \arg\max_c [\text{softmax}(q_k h_{\text{fus}})]; \quad q_k = \mathbf{v}^T [h_1^{text} || \cdot || h_c^{text}] \tag{5}$$

where the subscript, c denotes the the total number of nanomaterial categories and $\mathbf{v}$ is a trainable parameter. The above operator computes the list of scores or probabilities for each nanomaterial, and the $\arg\max$ operator selects the nanomaterial for which the probability score is maximized. We then select the appropriate/relevant nanomaterial text-level embedding conditioned on hierarchical embedding (h_{fus}) as follows:

$$h_{\text{fus}}^{text} = h_{\beta^*}^{text} \tag{6}$$

β^* denotes the nanomaterial label with the highest probability. This is essentially a matching mechanism that tries to find the best pairwise alignment among the various nanomaterial text-level embeddings ($h_1^{text}, \ldots, h_c^{text}$) and the hierarchical embedding (h_{fus}) obtained from the hierarchical network fusion (HNF). We utilize backpropagation error in the downstream supervised multi-classification task to fine-tune the smaller LMs to maximize the pairwise alignment between the complementary hierarchical embedding (h_{fus}) and its corresponding text-level embedding h_{fus}^{text}. To put it briefly, h_{fus}^{text} incorporates the expert knowledge obtained from foundational LLMs for the appropriate nanomaterial underlying the electron micrographs.

3.4 Overall Method

Figure 2 provides an overview of the "MultiFusion-LLM" framework. Our proposed framework comprises three distinct methods: a) **Hierarchical Network Fusion (HNF)** tokenizes micrographs into patches to obtain patch sequences and construct vision graphs. It introduces a <*cls*> token into the patch sequence and a virtual node for the vision graph to capture global characteristics. The network has a multi-layered structure; each layer of the network consists of bidirectional Neural ODEs and graph Chebyshev networks, and regulates the information flow through a gating mechanism to learn hierarchical embeddings with increasing patch sizes across each layer. It computes cross-modal embeddings, denoted as $\mathbf{h}_{fus}$, by integrating embeddings between modalities at different patch resolutions, thereby facilitating the exchange of information and integration of knowledge. For more detailed information, please refer to section 3.2. b) **LLMs for Incorporating Domain Knowledge:** We generate technical descriptions of nanomaterials, capturing a wide range of information including structure, properties, and applications using Zero-Shot CoT prompting of LLMs. To illustrate, Table 8 provides a glimpse of the LLM-retrieved text obtained from GPT-3.5 turbo, specifically generated to address natural language queries regarding MEMS devices. Initially, we pre-train a smaller LM on the generated descriptions through masked language modeling (MLM). Later, we fine-tune this small-scale LM on a downstream supervised task to encapsulate the generated explanations. We then utilize the weighted sum-pooling attention mechanism to compute domain-specific knowledge-incorporated text-level embeddings, denoted as $\mathbf{h}_{fus}^{\text{text}}$. For additional details, please refer to subsection 3.3. (c) We employ the **multi-head attention mechanism (MHA)**[95] to fuse text-level embeddings $\mathbf{h}_{fus}^{\text{text}}$ with hierarchical embeddings $\mathbf{h}_{fus}$, enabling the capture of contextually relevant information and achieving semantic alignment across different cross-domain embeddings. Simultaneously, by focusing on and aligning high-level textual descriptions (text-level embeddings) with detailed visual representations (hierarchical embeddings), we ensure a comprehensive understanding and analysis of electron micrographs from both descriptive and visual perspectives. This approach helps mitigate the inherent limitations arising from high intra-class dissimilarity, high inter-class similarity, and spatial heterogeneity in visual patterns across the electron micrographs, ultimately enhancing the performance of nanomaterial identification tasks. We compute the Query, Key, Value projections for the text-level embedding $\mathbf{h}_{fus}^{\text{text}}$ for each head h as follows:

$$Q_{\text{text}}^h = \mathbf{h}_{fus}^{\text{text}} W_{Q_{\text{text}}}^h; K_{\text{text}}^h = \mathbf{h}_{fus}^{\text{text}} W_{K_{\text{text}}}^h; V_{\text{text}}^h = \mathbf{h}_{fus}^{\text{text}} W_{V_{\text{text}}}^h \tag{7}$$

Similarly, the Query, Key, Value projections for hierarchical embedding $\mathbf{h}_{fus}$ for each head h as follows:

$$Q_{\text{fus}}^h = \mathbf{h}_{fus} W_{Q_{\text{fus}}}^h; K_{\text{fus}}^h = \mathbf{h}_{fus} W_{K_{\text{fus}}}^h; V_{\text{fus}}^h = \mathbf{h}_{fus} W_{V_{\text{fus}}}^h \tag{8}$$

We concatenate keys and values of text-level and hierarchical embeddings to create a unified representation.

$$K_{\text{concat}}^h = [K_{\text{text}}^h, K_{\text{fus}}^h]; V_{\text{concat}}^h = [V_{\text{text}}^h, V_{\text{fus}}^h] \tag{9}$$

We apply Softmax attention to integrate complementary information from the cross-domain embeddings, focusing on relevant information and aligning them semantically.

$$A_{\text{cross}}^h = \text{Softmax}\left(\frac{(Q_{\text{text}}^h + Q_{\text{fus}}^h){K_{\text{concat}}^h}^T}{\sqrt{d_h}}\right) \tag{10}$$

Each head outputs a new vector representation that highlights the most relevant features in the mono-domain embeddings, tailored to specific aspects of the data.

$$O_{\text{cross}}^h = A_{\text{cross}}^h V_{\text{concat}}^h \tag{11}$$

Finally, we concatenate and linearly transform all head-specific outputs to create the final unified cross-modal embedding.

$$O_{\text{concat}} = [O_{\text{cross}}^1, O_{\text{cross}}^2, \ldots, O_{\text{cross}}^H] \tag{12}$$

$$y_{\text{cross}} = O_{\text{concat}} W_{O_{\text{cross}}} \tag{13}$$

$$\mathrm{p}_i = \text{softmax}\big(\mathrm{W} y_{\text{cross}}\big) \tag{14}$$

where $W_{Q_{\text{text}}}^h$, $W_{K_{\text{text}}}^h$, $W_{V_{\text{text}}}^h$, $W_{V_{\text{fus}}}^h$, $W_{Q_{\text{fus}}}^h$, $W_{K_{\text{fus}}}^h$, $W_{O_{\text{cross}}}$ and W are the trainable weight matrices. d_h represents the dimensionality of the key/query/value for each head, and H is the number of heads. p_i represents the probability distribution across nanomaterial categories, we apply the argmax operation to p_i to determine the framework's predictions for the nanomaterial category. In summary, we conduct Zero-shot CoT prompting of LLMs to generate technical descriptions of nanomaterials and pre-train small-scale LMs using masked language modeling (MLM). Next, we jointly optimize the smaller pre-trained LM and the hierarchical network fusion (HNF) method on supervised learning tasks. The objective is to minimize the cross-entropy loss and enhance multi-class classification accuracy. In summary, the MHA offers a multi-faceted approach to capture and align varied information sources, making it a powerful tool for multi-modal data integration and analysis. It allows for a robust, synergistic, and comprehensive representation of data, especially in contexts like nanomaterial analysis where both modalities offer complementary insights.

4 Experiments And Results

4.1 Datasets

Our study primarily utilized the SEM dataset[4] to automate nanomaterial identification. The expert-annotated dataset spans across 10 distinct categories, representing a broad range of nanomaterials such as *particles, nanowires, patterned surfaces, among others*. In total, it contains approximately 21,283 electron micrographs. Figure 4 provides a visual representation of the different nanomaterial categories included in the SEM dataset. Despite the initial findings by [74] on a subset of the original dataset, our research was based on the complete dataset since the subset was not publicly accessible. Although the original dataset curators, [4], did not provide predefined splits for training, validation, and testing, we utilized the k-fold cross-validation method to evaluate our framework's performance. This strategy facilitated a fair comparison with popular baseline models in a competitive benchmark setting. Furthermore, we extended our evaluation by leveraging several open-source material benchmark datasets relevant to our study. These datasets were used to showcase the efficacy of our proposed framework and its applicability in a broader context beyond the SEM dataset.

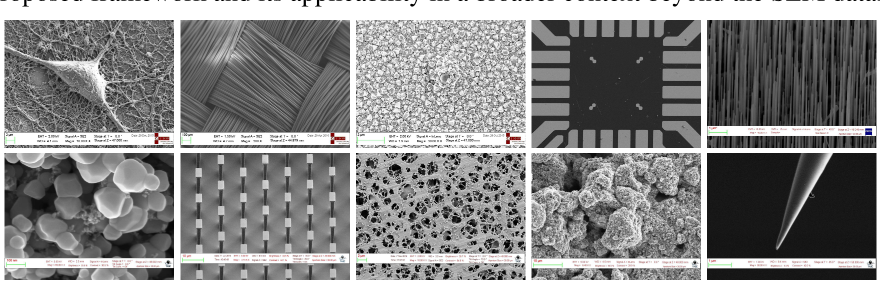


Figure 4: The figure depicts the different types of nanomaterials found in the SEM dataset ([4]) (left to right in the first row: *biological, fibers, films, MEMS, nanowires*; left to right in the second row: *particles, patterned surface, porous sponges, powder, tips*).

4.2 Results

We evaluated the effectiveness of our proposed framework through a comprehensive performance analysis, comparing it to commonly used computer vision baseline models. Our comparisons included supervised learning models such as ConvNets and ViTs (as referenced in [2, 1]), along with self-supervised learning techniques like Vision Contrastive Learning (VCL, as discussed in [34]). Table 2 reports the experimental results from our study. To ensure a fair and rigorous comparison, we conducted experiments with consistent settings across all algorithms, measuring performance using the Top-N accuracy metric and evaluating specifically for $N \in \{1, 2, 3, 5\}$. Our proposed framework outperforms the baseline models, showing a substantial relative improvement of 25.8% in the Top-1 score and a marginal improvement of 5.34% in the Top-5 score compared to the next-best baseline model, T2TViT ([110]).

Table 2: The table shows the comparison of our proposed method with baseline algorithms, including vision-based supervised ConvNets, ViTs, and self-supervised learning (VSL) algorithms.

	Algorithms	**Parameters**	**Top-1**	**Top-2**	**Top-3**	**Top-5**
ConvNets	AlexNet([65])	5.70E+07	0.493	0.582	0.673	0.793
	DenseNet([57])	2.39E+05	0.539	0.750	0.875	0.906
	ResNet([50])	2.72E+05	0.512	0.766	0.891	0.906
	VGG([81])	3.44E+07	0.517	0.644	0.717	0.779
	GoogleNet([84])	2.61E+05	0.560	0.844	0.906	0.938
	SqueezeNet([58])	7.41E+05	0.436	0.469	0.609	0.656
VSL	Barlowtwins[111]	8.99E+06	0.138	0.250	0.328	0.453
	SimCLR[22]	8.73E+06	0.157	0.234	0.359	0.469
	byol[43]	8.86E+06	0.130	0.234	0.281	0.422
	moco[49]	8.73E+06	0.158	0.188	0.250	0.438
	nnclr[33]	9.12E+06	0.144	0.266	0.313	0.531
	simsiam[23]	9.01E+6	0.170	0.266	0.391	0.500
Vision Transformers(ViTs)	CCT[47]	4.10E+05	0.600	0.781	0.875	0.969
	CVT[102]	2.56E+05	0.537	0.750	0.828	0.953
	ConViT[26]	6.00E+05	0.582	0.734	0.828	0.938
	ConvVT[102]	9.23E+04	0.291	0.563	0.734	0.875
	CrossViT[17]	8.35E+05	0.466	0.719	0.828	0.938
	PVTC[99]	1.30E+06	0.567	0.766	0.813	0.922
	SwinT[71]	2.78E+07	0.675	0.766	0.891	0.938
	VanillaViT[31]	1.79E+06	0.623	0.828	0.859	0.938
	Visformer[24]	1.21E+05	0.371	0.578	0.641	0.797
	ATS[36]	3.26E+06	0.511	0.703	0.828	0.938
	CaiT[92]	3.84E+07	0.616	0.750	0.906	0.938
	DeepViT[113]	3.26E+06	0.512	0.734	0.875	0.938
	Dino[15]	2.02E+07	0.047	0.219	0.391	0.432
	Distallation[91]	2.06E+06	0.516	0.719	0.844	0.938
	LeViT[42]	1.68E+07	0.597	0.813	0.875	0.953
	MA[48]	3.87E+06	0.192	0.288	0.350	0.459
	NesT[112]	1.61E+07	0.636	0.828	0.891	0.953
	PatchMerger[78]	3.26E+06	0.549	0.719	0.859	0.922
	PiT[54]	4.48E+06	0.520	0.703	0.828	0.953
	RegionViT[16]	1.22E+07	0.575	0.797	0.859	0.922
	SMIM[104]	2.38E+06	0.163	0.297	0.453	0.609
	T2TViT[110]	1.03E+07	0.702	0.859	0.906	0.938
	ViT-SD[68]	4.47E+06	0.613	0.766	0.906	0.953
	MultiFusion-LLM W/GPT-3.5	2.39E+07	**0.947**	**0.965**	**0.986**	**0.991**
	MultiFusion-LLM W/Google Bard	2.39E+07	0.852	0.899	0.927	0.953

5 Conclusion

To conclude, we have conducted the first in-depth study aimed at achieving state-of-the-art performance in nanomaterial characterization. This study introduces the innovative `MultiFusion-LLM` framework, a robust solution to the challenges associated with nanomaterial identification in electron micrographs. By synergistically integrating multi-modal representations and leveraging the analytical prowess of large language models, it promises more nuanced and accurate classification. Our comprehensive framework has outperformed traditional methods, showcasing cutting-edge performance on cost-efficient GPU hardware. Furthermore, it has demonstrated effectiveness and computational efficiency, particularly with large datasets, thereby accelerating high-throughput screening and advancing research holding implications for the advancement of the semiconductor industries.

6 Technical Appendix

Table 3 presents experimental findings comparing the proposed framework's performance to various supervised learning-based baseline models, including several GNN architectures ([79, 38]), and we use Graph Contrastive Learning (GCL, [114]) algorithms for additional comparison. Our proposed framework achieves SOTA performance on the benchmark dataset [4] compared to the baselines.

Table 3: The table presents the results of a comparative study between our proposed method and supervised-learning based GNNs, as well as self-supervised graph contrastive learning (GCL) algorithms, on the SEM dataset [4].

	Algorithms	Parameters	Top-1	Top-2	Top-3	Top-5
GSL	GBT[8]	7.09E+05	0.513	0.595	0.686	0.778
	GRACE[115]	7.44E+05	0.581	0.646	0.711	0.773
	BGRL[87]	6.92E+05	0.573	0.629	0.671	0.728
	InfoGraph[82]	6.82E+05	0.560	0.631	0.694	0.756
Graph Neural Networks	APPNP[64]	7.35E+05	0.604	0.713	0.792	0.823
	AGNN[88]	5.22E+05	0.517	0.733	0.841	0.943
	ARMA[7]	4.57E+05	0.553	0.747	0.848	0.925
	DNA[37]	8.48E+05	0.593	0.677	0.786	0.891
	GAT[96]	6.31E+05	0.507	0.724	0.807	0.914
	GGConv[69]	8.05E+05	0.583	0.778	0.841	0.944
	GraphConv[75]	5.85E+05	0.623	0.787	0.875	0.953
	GCN2Conv[19]	6.18E+05	0.697	0.813	0.867	0.945
	ChebConv[28]	5.00E+05	0.547	0.762	0.834	0.896
	GraphConv[75]	6.79E+05	0.533	0.727	0.847	0.961
	GraphUNet[39]	9.57E+05	0.622	0.738	0.866	0.912
	MPNN[40]	5.22E+05	0.643	0.792	0.873	0.959
	RGGConv[9]	6.58E+05	0.633	0.727	0.886	0.928
	SuperGAT[61]	5.54E+05	0.561	0.676	0.863	0.935
	TAGConv[32]	5.74E+05	0.614	0.739	0.803	0.946
	MultiFusion-LLM W/GPT-3.5	2.39E+07	**0.947**	**0.965**	**0.986**	**0.991**
	MultiFusion-LLM W/Google Bard	2.39E+07	0.852	0.899	0.927	0.953

6.1 Experimental Setup

The SEM dataset[4] consists of electron micrographs with dimensions of $1024 \times 768 \times 3$ pixels. To facilitate our analysis, we downscale these micrographs to $224 \times 224 \times 3$ pixels. As part of the data preprocessing, we normalize the electron micrographs by adjusting the mean and covariance to achieve a value of 0.5 across all channels. This normalization results in the micrographs falling within the range of [-1, 1]. We tokenize the downscaled and normalized micrographs into discrete, non-overlapping patches. Subsequently, we represent the electron micrographs as patch sequences and construct vision graphs using the Top-K nearest neighbor search algorithm. Specifically, we set the value of K to 10, 6, and 4 for each layer in the hierarchical network fusion (HNF) method, resulting in a total of three layers. This process generates multi-scale vision graphs and patch sequences with patch resolutions increasing of 16, 28, and 32 pixels. The patch dimension (d_{pos}) and position embedding dimension (d) are both set to 64. The framework is evaluated using a 10-fold cross-validation strategy and trained for 50 epochs with an initial learning rate of $1e^{-3}$ and a batch size of 48. We have a few more hyperparameters set for the cross-modal attention layer with the number of attention heads(H) to 4, and the dimensionality of Key/Query/Value (d_h) is 16. To enhance the performance of the `MultiFusion-LLM` framework, we employ two key strategies: (a) early stopping on the validation set, which halts training when the framework's performance on the validation data plateaus to prevent overfitting; and (b) a learning rate scheduler that systematically reduces the learning rate by half if the validation loss stagnates for five consecutive epochs. Reducing the learning rate can help the framework converge to a better solution and avoid overfitting. In addition, we utilize the Adam optimization algorithm [62] to update the trainable parameters of the framework. Our proposed framework enhances the accuracy of multi-class classification tasks by seamlessly integrating both large language models (LLMs) and small-scale language models (LMs). The framework fully leverages the capabilities of LLMs in generating technical descriptions of nanomaterials, an approach that can significantly exploit domain-specific linguistic insights critical for nanomaterial identification tasks. The framework interacts with off-the-shelf LLMs through a Language Model as a Service (LaMaaS) platform through the text-based API interactions. In this study, we utilized GPT-3.5-turbo and Google Bard as representative LLMs. The hyperparameters for our framework were not individually fine-tuned for each LLM. Instead, they were consistently applied across all LLMs. This method underscores our framework's generality, ease of use, and compatibility with existing off-the-shelf LLMs. For decoder-only LLMs, the maximum output token sequence length is 4096 for GPT-3.5-turbo and 4000 for Google Bard. To optimize computational resource use, the system is trained on eight V100 GPUs, each boasting 8 GB of GPU memory, utilizing the PyTorch framework. This configuration ensures the training process is completed within a reasonable timeframe. Given the potentially high computational cost of using prompting with LLMs, we conducted each experiment twice and reported the averaged results.

6.2 Ablation Study

Figure 5 illustrates the overview of the framework. Our proposed framework comprises three distinct methods: (a) The Hierarchical Network Fusion (HNF) is a multi-layered, cascading network architecture designed to enhance the classification accuracy of electron micrographs. It integrates two complementary representations at multiple layers: (a) patch sequences, which assist in capturing spatial dependencies among patches beyond pairwise dependencies, and (b) vision graphs, which capture the local pairwise patch relationships. These techniques provide a detailed multi-scale representation of the micrographs, encapsulating both fine-grained and coarse-grained details. HNF uses an inverted pyramid structure, incorporating increasing patch sizes at each layer, and utilizes bidirectional Neural ODEs and Graph chebyshev convolution(GCC) networks for iterative patch embeddings refinement and the computation of the optimal node-level embeddings, respectively. A mixture-of-experts technique further optimizes the integration of these cross-domain modalities, fostering efficient knowledge exchange and improving classification accuracy by effectively modeling structural, semantic, and causal information from both techniques. (b) Using Zero-shot CoT prompting with LLMs, we generate detailed technical descriptions of nanomaterials. We pre-train smaller LMs using masked language modeling (MLM) on these descriptions to facilitate domain-specific customization. These pre-trained LMs are then fine-tuned for task-specific adaptation to generate contextualized token embeddings. We apply a sum-pooling attention mechanism to obtain text-level embeddings from these token embeddings, thereby capturing the vast domain-specific knowledge embedded in the generated textual descriptions. (c) We use the cross-modal multi-head attention mechanism to integrate and align information from different modalities — specifically, from hierarchical network fusion (HNF) and language models — into a coherent and unified representation that captures complex, hierarchical, and potentially cross-modal patterns, emphasizing relevant features to enhance the accuracy of the multi-class classification task.

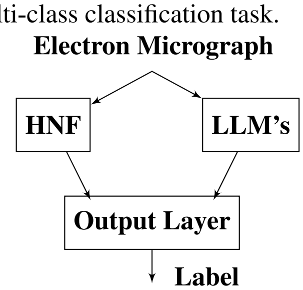


Figure 5: Overall, the architecture of our framework involves using zero-shot CoT prompting with LLMs to generate technical textual descriptions and pre-train smaller language models (LMs) using masked language modeling (MLM). We then jointly optimize the smaller LM along with the HNF method in supervised learning tasks, aiming to minimize cross-entropy loss and improve multi-class classification accuracy.

Algorithms	Avg-Precision	Avg-Recall	Avg-F1 Score
MultiFusion-LLM W-GPT4	**0.941**	**0.945**	**0.939**
w/o HNF	0.776	0.753	0.745
w/o LLMs	0.714	0.726	0.721
w/o MHA	0.827	0.831	0.823

Table 4: In the ablation study, we systematically disable individual methods to assess their respective contributions and importance. The goal of this study is to understand the impact or significance of specific methods on the overall performance of the framework. The experimental findings reveal the significance of the disabled methods, as indicated by the consistent decrease in performance metrics of the ablated variants compared to the baseline. These results substantiate our hypothesis regarding the joint optimization of HNF (see subsection 3.2) and LLMs (see subsection 3.3) methods, demonstrating improved framework performance.

To perform ablation studies, we systematically disabled certain methods to create various ablated variants, which were subsequently evaluated using the SEM dataset [4], with our original framework serving as the baseline for comparison. This approach enables us to verify the effectiveness of our methods, substantiate their design decisions, and justify their inclusion in the framework. A substantial decrease in performance of the ablated variants, compared to the baseline, underscores the significance of the omitted method. The ablated variants that exclude the hierarchical network fusion (HNF), large language models (LLMs), and the multi-head attention layer are denoted as proposed framework "w/o HNF", "w/o LLMs", and "w/o MHA" respectively. The abbreviation "w/o" stands for "without". For the case of "w/o MHA", we concatenate the cross-domain embeddings and transform them through a linear layer to predict the label. The findings from the ablation study are presented in Table 4. On the SEM dataset[4], the "w/o HNF" variant shows a substantial decline in performance relative to the baseline, evidenced by a significant drop of 17.53% in **Avg-Precision**. Similarly, the "w/o LLMs" variant performs much worse than the baseline, with a drop of 24.12% in **Avg-Precision**. In addition, the "w/o MHA" variant exhibited a notable deterioration in performance compared to the baseline, manifested by a substantial decrease of 11.9% in **Avg-Precision**. This is attributed to the overly simplified linear operator in the output layer. The results of our ablation study clearly illustrate the crucial role of each omitted method, with the ablated variants demonstrating a consistent decline in performance metrics compared to the baseline.

6.3 An In-Depth Empirical Insights into Nanomaterial Classification

We have conducted additional experiments to gauge the efficacy of our framework, which sheds light on its ability to categorize electron micrographs across various nanomaterial categories. The experimental results, presented in Table 5, demonstrate that our proposed framework can generalize to a wide range of nanomaterials, including those with complex patterns. We evaluated the performance of our framework using the SEM dataset[4], employing standard metrics such as precision (P in $\%$), recall (R in $\%$), and F1-score (F1 in $\%$). We adopt a multi-metric approach to ensure a fair and thorough comparison with baseline models. To facilitate this, we utilize a confusion matrix encompassing various metrics for multi-class classification. This confusion matrix aids in scrutinizing our framework's performance by offering insights into how it categorizes electron micrographs across different nanomaterial categories. The metrics included in the confusion matrix are as follows: True Positives (TP) represent micrographs that are correctly classified as belonging to a specific category. False Negatives (FN) represent micrographs that actually belong to a category but are incorrectly classified or missed. True Negatives (TN) represent micrographs that are correctly identified as not belonging to a particular category. False Positives (FP) represent micrographs that are mistakenly classified as belonging to a category despite not actually belonging to that category. These metrics evaluate the accuracy and effectiveness of our framework in micrograph categorization. Precision (TP / (FP + TP)) measures the proportion of correctly classified micrographs for a specific category, while recall (TP / (FN + TP)) measures the proportion of all micrographs of a category that were accurately identified. The F1-score is computed as the balanced mean of precision and recall. It is important to note that the SEM dataset is highly class-imbalanced. Our framework demonstrates a relatively higher score in the classification of nanomaterial categories with a large number of labeled instances compared to those with fewer. This favorable performance of our proposed framework can be attributed to its reduced dependency on nanomaterial-specific relational inductive bias, setting it apart from traditional methods.

Category	**Multi-class metrics**		
	Precision	**Recall**	**F1 Score**
Biological	0.931±0.009	0.943±0.007	0.935±0.013
Tips	0.909±0.005	0.919±0.008	0.916±0.011
Fibres	0.979±0.007	0.965±0.012	0.963±0.014
Porous Sponge	0.929±0.014	0.941±0.013	0.925±0.010
Films Coated Surface	0.938±0.005	0.934±0.009	0.941±0.008
Patterned surface	0.946±0.016	0.942±0.006	0.941±0.014
Nanowires	0.938±0.012	0.945±0.007	0.948±0.011
Particles	0.935±0.006	0.937±0.011	0.929±0.023
MEMS devices	0.939±0.011	0.932±0.008	0.923±0.009
Powder	0.941±0.014	0.928±0.009	0.917±0.011

Table 5: The table illustrates the effectiveness of our proposed framework in identifying individual nanomaterial categories within the SEM dataset.

6.4 Baseline Algorithms

We have categorized our baseline methods into four distinct groups: Graph Neural Networks (GNNs) ([79, 38]), Graph Contrastive Learning (GCL) [114]), Convolutional Neural Networks (ConvNets)[2, 1], Vision Transformers (ViTs) ([2, 1]) and Vision Contrastive Learning (VCL) ([34]) algorithms

. We construct vision graphs to represent electron micrographs using the Top-K nearest neighbor search technique. In this representation, patches are treated as nodes, and pairwise associations between semantically similar nearest-neighbor nodes are represented as edges. For the baselines, we avoid constructing multi-scale vision graphs with increasing patch resolutions. Instead, we set the patch size to 32 pixels to reduce the complexity of the baseline models and set K to 5 for finding the nearest neighbors. The baseline Graph Neural Networks (GNNs)[79, 38]) are used for the multi-class classification task on vision graphs through supervised learning. The graph contrastive learning (GCL) algorithms ([114]) utilize several graph data augmentation strategies to create multiple correlated views of a vision graph. GCL aims to maximize the similarity between positively correlated views of a graph while minimizing dissimilarity with others, thereby learning invariant self-supervised node-level embeddings. The GCL algorithms employ the Graph Attention Network (GAT) ([96]) as the node-level graph encoder. Graph-level embeddings are generated by performing sum-pooling on the node-level embeddings. During inference, the Random Forest (RF) algorithm utilizes these robust self-supervised graph-level embeddings to predict nanomaterial categories, having been trained using supervised learning. To evaluate the effectiveness of the unsupervised embeddings, we measure the classification accuracy of the RF model on the holdout data. In addition, we employ baseline ConvNets ([2, 1]) operating on the regular grid of pixels in electron micrographs for classification tasks using supervised learning. We also utilize baseline Vision Transformers (ViTs) ([2, 1]) trained through supervised learning to analyze patch sequences within each electron micrograph for classification tasks. Furthermore, we utilize visual-contrastive learning (VCL) techniques ([34]), which are self-supervised algorithms designed for contrastive learning in computer vision tasks. We employ the ResNet backbone architecture for feature extraction.

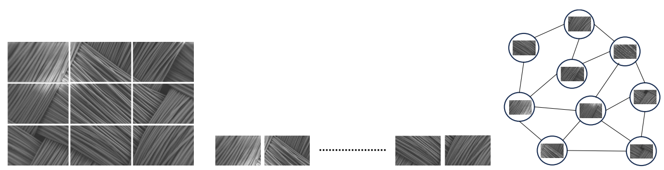


Figure 6: In this illustrative example, we divided an electron micrograph (MEMS device, [4]) into a grid of 3 × 3 patches. The image presents various representations of the micrograph, including a regular grid, a sequence, and a graph representation from left to right, respectively. Different approaches for processing these representations include ConvNets that operate on pixel grids, ViTs that operate on patch sequences, and GNNs that operate on vision graphs. These graphs represent patches as nodes and are constructed using a nearest neighbor search algorithm, connecting patches based on visual similarity rather than spatial proximity. Each method offers a unique perspective for analyzing electron micrographs, providing distinct advantages and insights into patterns.

6.5 Hyperparameter Studies

We performed an in-depth hyperparameter tuning to determine the optimal hyperparameters for our framework. The hyperparameters of the algorithm are: (1) the dimensionality of the embedding (d), and (2) batch size (b). The hyperparameters were chosen from the following ranges: embedding dimension (d) $\in [32, 64, 128, 256]$ and batch size (b) $\in [32, 48, 64, 96]$. We conducted hyperparameter optimization using the random-search technique to achieve the optimal performance of our proposed framework on the validation dataset, measured in terms of Top-1 classification accuracy. For each experiment, we altered the hyperparameter under investigation to ascertain its impact on the framework's performance. The study determined that the optimal hyperparameters are $d = 64$ and $b = 48$.

(d, b)	$(32, 48)$	$(64, 48)$	$(128, 48)$	$(256, 48)$
	0.941	0.947	0.935	0.927

(d, b)	$(64, 32)$	$(64, 48)$	$(64, 64)$	$(64, 96)$
	0.943	0.947	0.939	0.936

Table 6: The table reports the experimental findings of the hyperparameter study.

6.6 Benchmarking with open-source material datasets

- **NEU-SDD**[3] ([29]) is a comprehensive database comprising 1800 grayscale electron micrographs of surface defects on hot-rolled steel strips. The dataset is divided into six distinct defect classes, each containing 300 micrographs with a resolution of 200×200 pixels. The defect categories include *pitted surfaces, scratches, rolled-in scale, crazing, patches, and inclusion defects*. Figure 7 displays representative images from each category. We conducted

[3]Datasource: `http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html`

a comparative analysis using various standard algorithms to evaluate the effectiveness of our proposed approach, specifically in the domain of multi-class classification tasks for surface defect identification.

- **CMI**[4] consists of 600 high-resolution electron micrographs depicting corroding panels. Each micrograph has been annotated by corrosion experts following ASTM-D1654 standards, assigning discrete ratings ranging from 5 to 9. The dataset includes 120 distinct micrographs for each corrosion rating with a spatial resolution of 512×512 pixels. Figure 8 illustrates a selection of representative images for each rating. Our proposed method for multiclass classification task is evaluated by comparing its performance against several standard algorithms.
- **KTH-TIPS**[5] represents a comprehensive texture dataset, comprising 810 electron micrographs, each depicting one of ten distinct material types. These micrographs, with a resolution of 200 x 200 pixels, encompass a wide range of materials captured under different illuminations, poses, and scales. The diverse material categories encompass textures such as *sponge, orange peel, styrofoam, cotton, cracker, linen, brown bread, sandpaper, crumpled aluminum foil, and corduroy*. Figure 9 showcases a selection of sample images from each category. In order to assess and demonstrate the efficacy of our proposed method, we conduct a comparative analysis of its performance against various standard algorithms within the domain of multi-class identification tasks.

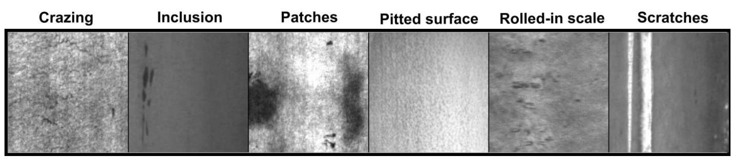


Figure 7: The NEU-SDD dataset contains six distinct defect categories found in hot-rolled steel strips, which are described in reference 3([29]).

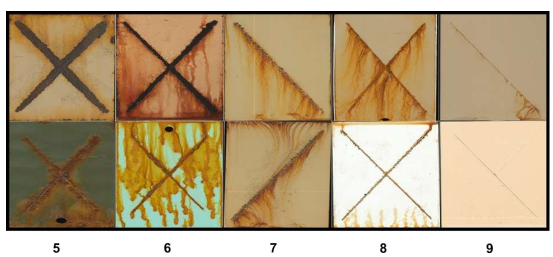


Figure 8: The CMI dataset is a collection of electron micrographs that represent five corrosion rating categories. These categories are described in reference 4.

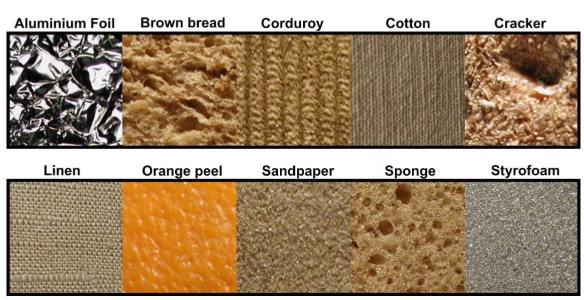


Figure 9: The KTH-TIPS dataset contains samples of electron micrographs of ten distinct materials. These materials are described in reference 5.

Table 7 presents a comprehensive comparison of the performance achieved by our proposed approach in contrast to various baseline methods, evaluated across all datasets. The experimental results demonstrate that our method achieves state-of-the-art performance on all datasets, underscoring the efficacy and robustness of our framework.

[4]`https://arl.wpi.edu/corrosion_dataset`

[5]`https://www.csc.kth.se/cvap/databases/kth-tips/index.html`

	Algorithms	NEU-SDD	CMI	KTH-TIPS
Baselines	ResNet	0.906	0.928	0.941
	GoogleNet	0.936	0.928	0.929
	SqueezeNet	0.955	0.943	0.963
	VanillaViT	0.962	0.968	0.972
	MultiFusion-LLM	**0.993**	**0.989**	**0.991**

Table 7: The table presents the comparative evaluation of our proposed framework's performance against several benchmark algorithms on a variety of datasets.

6.7 Graph Chebyshev convolution

The graph convolution is a powerful tool in the realm of learning from graph-structured data. The spectral graph convolution[85] is a popular approach, but it can be computationally expensive for large graphs. To tackle this issue, Chebyshev graph convolution[28] offers a more scalable approach that can be used to achieve similar performance in capturing the local connectivity and spectral properties of the graph. More precisely, Graph Chebyshev Convolution is a method that approximates the spectral graph convolution by using Chebyshev polynomials. Graph Chebyshev Convolution allows us to apply convolutional filters on graph-structured data based on the Chebyshev polynomial approximation of the graph Laplacian. The Chebyshev polynomials are calculated based on the normalized Laplacian matrix of the graph. The normalized Laplacian matrix, denoted as $\hat{L}$, is defined as:

$$\hat{L} = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2} \tag{15}$$

where $\hat{A}$ is the normalized adjacency matrix and $\hat{D}$ is the diagonal degree matrix of the graph. The Chebyshev approximation of the graph Laplacian up to any desired degree is obtained by using a truncated expansion of Chebyshev polynomials, denoted as $T_k(\hat{L})$, where k represents the degree of the polynomial. These polynomials are computed recursively using the following recurrence relation as follows:

$$T_k(\hat{L}) = \begin{cases} I, & \text{if } k = 0 \\ \hat{L}, & \text{if } k = 1 \\ 2\hat{L}T_{k-1}(\hat{L}) - T_{k-2}(\hat{L}), & \text{otherwise} \end{cases}$$

where I is the identity matrix. Given an input graph feature matrix $I \in \mathbb{R}^{n\times d}$, where n denotes the number of patches and d is the patch embedding dimension, and the Chebyshev polynomials denoted by $T_k(\hat{L})$. The Chebyshev graph convolution operation can be defined as follows:

$$E = \sigma\left(\sum_{k=0}^{K-1} T_k(\hat{L}) I \Theta_k\right) \tag{16}$$

where $\sigma(\cdot)$ is a non-linear ReLU activation function applied element-wise and $\Theta_k \in \mathbb{R}^{d\times d}$ is the parameter matrix (weights) for the k-th order Chebyshev polynomial. It is important to note that the parameter matrices Θ_k are typically learnable and optimized during the training process to adaptively capture the global graph characteristics. K denotes the maximum order of the Chebyshev polynomials and influences the expressive power of the approximation. $E \in \mathbb{R}^{n\times d}$ is the transformed node feature matrix, which captures the local structure and relationships within the graph, where $e_i \in \mathbb{R}^{d}$ denotes the node embedding.

6.8 Neural Ordinary Differential Equations (NODE)

Neural Ordinary Differential Equations (Neural ODE) [20] represent a deep neural network model designed for continuous-time systems, in contrast to traditional discrete-time neural networks. In the Neural ODE framework, we denote the hidden state of a dynamic system at a given time t as $\mathbf{z}(t)$. The objective is to determine the evolution of $\mathbf{z}(t)$ by calculating its derivative with respect to time to capture the temporal dynamics of the system. This derivative is represented by a parameterized neural network function, denoted as $f(\mathbf{z}(t), t, \theta)$, as follows:

$$\frac{d\mathbf{z}(t)}{dt} = f(\mathbf{z}(t), t, \theta) \tag{17}$$

Here, θ represents the parameters of the neural network $f(\cdot)$. To compute the output of the Neural ODE framework, an ODE solver takes the initial hidden state $\mathbf{z}(t_0)$ at the starting time point t_0 and integrates the hidden state derivative over time to produce the hidden state $\mathbf{z}(t_1)$ at the specified output time point t_1, as described below:

$$\mathbf{z}(t_1) = \mathbf{z}(t_0) + \int_{t_0}^{t_1} f(\mathbf{z}(t), t, \theta) dt \tag{18}$$

In summary, by formulating neural networks as continuous-depth models through Neural ODEs, this framework can generate the hidden state of a dynamic system at any given time point and effectively handle continuous-time data. This characteristic makes it particularly useful for modeling continuous-time dynamic systems. Furthermore, to reduce memory requirements during backpropagation, Chen et al. [20] introduced the adjoint sensitivity method for Neural ODEs. An adjoint, denoted as $\mathbf{a}(t) = \frac{\partial \mathcal{L}}{\partial \mathbf{z}(t)}$, is defined, where $\mathcal{L}$ represents the loss function. The gradient of $\mathcal{L}$ with respect to the network parameters θ can be directly computed using the adjoint and an ODE solver:

$$\frac{d\mathcal{L}}{d\theta} = -\int_{t_1}^{t_0} \mathbf{a}(t)^T \frac{\partial f(\mathbf{z}(t), t, \theta)}{\partial \theta} dt \tag{19}$$

In essence, the adjoint sensitivity method solves an augmented ODE backward in time, enabling the computation of gradients without the need for backpropagation through the ODE solver operations. This means that the model doesn't have to store intermediate results (partial derivatives) from forward propagation, resulting in a constant memory cost as a function of the depth. In this work, we incorporate Neural ODEs into computer vision tasks for electron micrograph classification by segmenting an electron micrograph into a sequence of patches. The sequence length is determined by the total number of patches generated through the tokenization of the electron micrograph, with each patch serving as an individual token in the sequence. Treating the input sequence of patches as a continuous-time system enables Neural ODEs to capture the evolution of the patch embeddings smoothly and continuously. Moreover, this approach facilitates the causal modeling of spatial relationships and transformations between consecutive patches by encoding them into patch embeddings. In this work, we model the neural network $f(\cdot)$ using a transformer encoder[95]. It consists of a stack of encoder layers, each containing self-attention mechanisms and feed-forward neural networks. The encoder layers capture the relationships and dependencies between the patches in the image. We learn the bidirectional representation of sequences to capture information from both the past and future context of a given patch in a sequence. Our bidirectional representation learning approach incorporates two separate Neural ODEs: one that processes the sequence from left to right (forward pass) and another that processes the sequence from right to left (backward pass). Each pass maintains its own hidden state, and the outputs of both passes are combined through a gating mechanism. Let's denote the forward Neural ODE estimate of the patch embedding at time point t_1 as $\mathbf{z}_f(t_1)$ and the backward Neural ODE as $\mathbf{z}_b(t_1)$ using Equation 18. A gating mechanism is implemented to regulate the information flow from $\mathbf{z}_f(t_1)$ and $\mathbf{z}_b(t_1)$, which produces a weighted combination of representations $h_{(t_1)}$. The gating mechanism is described as follows:

$$g = \sigma\big(f'(\mathbf{z}_f(t_1)) + f''(\mathbf{z}_b(t_1))\big) \tag{20}$$

$$h_{(t_1)} = \sigma\big(g(\mathbf{z}_f(t_1)) + (1-g)(\mathbf{z}_b(t_1))\big) \tag{21}$$

where f' and f'' are linear projections. In our work, the use of adaptive ODE solvers can lead to significant time consumption. To ensure manageable training time, we use fixed-grid ODE solvers in combination with the Interpolated Reverse Dynamic Method (IRDM) proposed by Daulbaev et al.[27]. The IRDM employs Barycentric Lagrange interpolation[6] on a Chebyshev grid[94] to approximate the solution of patch embeddings during the reverse-mode differentiation (referred to as backpropagation) through the ODE solver. By incorporating IRDM, we can reduce the computational time during backpropagation while maintaining satisfactory learning accuracy. Specifically, we adopt a fixed-grid ODE solver, namely the fourth-order Runge-Kutta method[12], and implement the interpolated reverse dynamic method with 3 Chebyshev nodes. This approach enables us to ensure tractable training time without compromising precision.

6.9 Related Work

In this section, we will first review the backbone architectures used in computer vision. Next, we will survey the evolution of graph neural networks, with a particular focus on Graph Convolutional Networks (GCN)[63] and their utilization in vision tasks. The landscape of computer vision has been significantly shaped by convolutional networks(i.e., ConvNets or CNNs), which have brought about a seismic shift in the field and have established themselves as the predominant architecture (LeCun et al.[67], Krizhevsky et al.[65], He et al.[50]). LeNet[67] significantly influenced the development and popularity of ConvNets, paving the way for more advanced and deeper networks in subsequent years across a broad spectrum of vision tasks, including image classification[65], object detection[35], and semantic segmentation[72]. Over the past decade, groundbreaking advancements such as ResNet[50], MobileNet[56], and NAS[116, 109] have further shaped the landscape of CNN architectures. The advent of the vision transformer(ViT)[31, 45, 14, 18] has been a trailblazer, leading to the development of a myriad of improved ViT variants[31]. These improvements encompass pyramid architectures[71, 98], local attention mechanisms[46, 71], and position encoding techniques[103]. Inspired by the vision transformer, researchers have also explored the potential of Multilayer Perceptrons (MLP) in computer vision tasks[90, 89]. By incorporating tailored modules[21, 70, 44, 86], MLP-based techniques have demonstrated exceptional performance in general vision tasks, including object detection and segmentation. Graph Neural Networks (GNNs) originated from the early work of Scarselli et al.[80] and Gori et al.[41], introducing the concept of spatial graph convolutional networks with non-recursive layers[73] to learn from graph-structured data. Since then, spatial GCNs have seen numerous adaptations and improvements, as proposed in previous works such as Niepert et al.[76], Atwood et al.[3], and Gilmer et al.[40]. Spectral GCNs, grounded in spectral graph theory, were first introduced in a study by Bruna et al.[11]. Subsequent methods to enhance these networks have been proposed in studies by Kipf et al.[63], Henaff et al.[53], and Defferrard et al.[28]. In the realm of computer vision[105, 66, 60, 97], Graph Convolutional Networks (GCNs) have been applied to diverse tasks including point cloud classification, scene graph generation, and action recognition. Point clouds refer to sets of 3D points derived from LiDAR scans, where GCNs have been leveraged for classification and segmentation[66, 100, 108]. The process of scene graph generation involves parsing an input image into a graph representation that delineates objects and their interrelationships, often integrating object detection with GCN techniques[107]. Furthermore, GCNs have been instrumental in facilitating human action recognition tasks by analyzing graphs representing linked human joints[106, 59]. Current frameworks in the semiconductor manufacturing sector fall short in various aspects, especially when compared to recently proposed advancements. Many existing solutions fail to capitalize on the detailed analysis achievable through the synergy of patch sequences and vision graphs at different scales in electron micrographs. Moreover, these frameworks typically analyze data at a singular scale, missing the opportunities that a multi-scale approach could offer in enhancing classification accuracy. Furthermore, the industry has yet to fully embrace the utilization of large language models (LLMs) in generating technical descriptions of nanomaterials, a strategy that can significantly deepen domain-specific insights critical for nanomaterial identification tasks. This glaring gap in the integration of image-based and linguistic insights renders current architectures less comprehensive and nuanced, potentially impeding breakthroughs in the semiconductor industry. The proposed “MultiFusion-LLM" framework represents a pioneering approach to automatic nanomaterial identification, addressing the prevailing challenges in semiconductor manufacturing processes. At its core, it employs a hierarchical network fusion (HNF) architecture that amalgamates two diverse representations of electron micrographs: patch sequences and vision graphs. These representations undergo iterative refinement through a layered, inverted pyramid structure, integrating multi-scale information to enhance classification accuracy. Additionally, the framework introduces a zero-shot chain-of-thought (Zero-Shot CoT) prompting technique that utilizes large language models (LLMs) to generate detailed technical descriptions of nanomaterials. These descriptions, rich in domain-specific knowledge, serve as auxiliary information that bolsters identification tasks. By encapsulating both image-based and linguistic insights, the framework aims to facilitate a more comprehensive and nuanced analysis of electron micrographs, thereby holding significant promise for advancements in the semiconductor industry.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	LLM-retrieved text
Introduction: Provide an overview of the MEMS and its significance in various fields?	Introduction: Micro-Electro-Mechanical Systems, commonly known as MEMS, is a cutting-edge technology that integrates miniaturized mechanical and electromechanical elements with electronics on a single silicon chip or substrate. These systems typically consist of components with dimensions ranging from micrometers to millimeters. MEMS devices have revolutionized various industries and fields due to their unique capabilities, compact size, low power consumption, and cost-effectiveness. They enable the creation of tiny, sophisticated systems with enhanced functionalities, making them indispensable in numerous applications. Significance in Various Fields: Consumer Electronics: MEMS devices have played a crucial role in the advancement of consumer electronics. Accelerometers, gyroscopes, and magnetometers in smartphones enable motion sensing, orientation detection, and navigation. MEMS microphones are widely used for voice and audio input in smartphones, laptops, and other gadgets. Automotive: MEMS sensors are fundamental to the functioning of modern vehicles. They are used in airbag deployment systems, tire pressure monitoring systems, electronic stability control, and more. MEMS accelerometers and gyroscopes provide data for advanced driver assistance systems (ADAS) and vehicle navigation. Aerospace and Defense: MEMS technology has revolutionized aerospace and defense applications. Inertial sensors, such as accelerometers and gyroscope arrays, are vital for aircraft and spacecraft stabilization, guidance, and control. MEMS-based pressure sensors are used in altimeters and barometers, and microbolometers for infrared imaging. Industrial Applications: MEMS devices find extensive use in industrial automation and control systems. They are used for precise measurement, monitoring, and feedback control in manufacturing processes, structural health monitoring, and predictive maintenance. Robotics: MEMS sensors are essential components in robotics and autonomous systems. They provide feedback for motion control, balance, and obstacle detection, enabling robots to navigate and interact with their environment safely. Telecommunications: MEMS technology has been instrumental in the miniaturization of RF switches, tunable capacitors, and filters, leading to improved performance and efficiency in wireless communication devices. Energy Harvesting: MEMS energy harvesters are used to capture and convert ambient energy sources such as vibrations, solar, and thermal energy into usable electrical power for low-power electronic devices. The significance of MEMS in these and many other fields stems from their ability to offer scalable, high-performance, and cost-effective solutions for various applications. The continuous advancement in MEMS fabrication techniques and materials has opened up new possibilities for even more innovative and diverse applications in the future. As research and development in this field continue, we can expect MEMS to remain at the forefront of technological progress across numerous industries.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	LLM-retrieved text
Definition and Structure: Define the MEMS and describe its typical structure at the nanoscale.?	Definition of MEMS: Micro-Electro-Mechanical Systems (MEMS) refers to a class of miniaturized devices and systems that integrate micro-scale mechanical components, electrical circuits, and sometimes optical elements on a single substrate. These components are typically fabricated using semiconductor manufacturing techniques, similar to those used for integrated circuits (ICs). MEMS devices are characterized by their small size, with components ranging from micrometers to millimeters, and they often operate in the micro/nano scale. Typical Structure of MEMS at the Nanoscale: MEMS devices exhibit a wide variety of structures and designs, each tailored to their specific application. At the nanoscale, MEMS structures often involve mechanical components with dimensions in the micrometer range and electrical connections at the nanoscale level. The following are the key components commonly found in MEMS devices at the nanoscale: Substrate: The foundation of a MEMS device is typically a silicon substrate. Silicon wafers are commonly used due to their excellent mechanical properties, compatibility with semiconductor processes, and availability in various sizes. Mechanical Components: MEMS devices consist of micro or nanoscale mechanical elements that can move or respond to external stimuli. These components may include beams, cantilevers, diaphragms, membranes, gears, and more. The design and material of these structures depend on the specific application. Actuators: Actuators are responsible for generating mechanical motion in MEMS devices. They can be electrostatic, piezoelectric, thermal, magnetic, or other types, depending on the required movement and sensitivity. Sensors: MEMS sensors are vital for detecting and measuring physical quantities such as acceleration, pressure, temperature, humidity, etc. Common MEMS sensors include accelerometers, gyroscopes, pressure sensors, and temperature sensors. Transducers: Transducers convert one form of energy into another. In MEMS, they are often used to transform mechanical motion or physical parameters into electrical signals or vice versa. Interconnects: Nanoscale electrical interconnects provide the necessary connections between the MEMS components and the external circuits or systems. These connections are usually made using metal traces or vias. Control Electronics: For some MEMS devices, integrated control electronics are present on the same chip to process, amplify, and transmit signals to and from the mechanical components and sensors. The fabrication of these nano/micro-scale components involves various advanced processes, including photolithography, etching, deposition, and bonding. The choice of materials, such as silicon, polymers, metals, and composites, is critical to achieving the desired mechanical and electrical properties. The combination of these components and their precise integration on a single substrate allows MEMS devices to perform a wide range of functions across various industries, making them a fundamental technology in the world of miniaturized electronics and sensors.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	LLM-retrieved text
Synthesis Methods: Explore different methods used to synthesize or fabricate MEMS devices in this category. Discuss their advantages and limitations.?	Various methods are used to synthesize or fabricate MEMS devices, each with its own advantages and limitations. Below are some commonly used synthesis methods: Bulk Micromachining: Advantages: (a) Well-established and widely used method, (b) Suitable for fabricating simple, 2D MEMS structures and (c) Relatively low cost and straightforward process. Limitations: (a) Limited design flexibility for complex 3D structures, (b) Fabrication of delicate or high-aspect-ratio structures can be challenging, and (c) Not suitable for integrating multiple layers of different materials. Surface Micromachining: Advantages: (a) Enables the fabrication of complex 3D structures, (b) Suitable for high-volume production due to batch processing capabilities, and (c) Can integrate multiple layers of different materials. Limitations: (a) Typically limited to planar structures, (b) Requires sacrificial layer techniques, which can be more complicated and time-consuming than bulk micromachining, and (c) Material choices are often limited to specific combinations suitable for sacrificial release. LIGA (Lithography, Electroplating, and Molding): Advantages: (a) Capable of fabricating high-aspect-ratio structures with excellent precision, (b) Suitable for mass production of microstructures, and (c) Compatible with various materials. Limitations: (a) Complex and costly process due to multiple steps involved, (b) Limited to metallic materials for electroplating, and (c) Difficulties in creating free-standing structures. Soft Lithography: Advantages: (a) Simple and cost-effective technique, (b) Suitable for rapid prototyping and small-scale production, and (c) Enables the use of a wide range of materials, including polymers and biomaterials. Limitations: (a) Limited to simple structures and low aspect ratios, (b) Resolution and precision may not be as high as other techniques, and (c) Not suitable for high-temperature or harsh environments. Silicon-on-Insulator (SOI) Technology: Advantages: (a) Provides high mechanical stability and isolation, (b) Suitable for integrating MEMS and CMOS components on the same chip, and (c) Excellent electrical isolation between components. Limitations: (a) Higher cost compared to standard silicon wafers, (b) Limited availability of thick SOI wafers, restricting the aspect ratio of structures, and (c) Challenges in wafer bonding and handling. Nanolithography and Nanofabrication: Advantages: (a) Enables fabrication of nanoscale MEMS devices with high precision, (b) Allows for the development of novel and highly sensitive sensors and actuators, and (c) Can be used to create complex nanostructures and patterns. Limitations: (a) Expensive and requires specialized equipment, (b) Limited to research and development due to high costs and complexity, and (c) Mass production of nanoscale MEMS devices remains challenging. 3D Printing/Additive Manufacturing: Advantages: (a) Enables rapid prototyping and customization of MEMS devices, (b) Supports the creation of complex 3D geometries and intricate structures, and (c) Reduces the need for complex lithography and etching steps. Limitations: (a) Limited resolution compared to traditional lithography-based methods, (b) Material choices may be restricted for some 3D printing techniques, and (c) Post-processing steps might be required for improved surface finishes and accuracy. Overall, the choice of synthesis method depends on the requirement of material properties, volume of production, and budget constraints.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	LLM-retrieved text
Properties: Highlight the unique physical, chemical, and electronic properties exhibited by MEMS in this nanomaerial category. Discuss how these properties differ from their bulk counterparts.?	MEMS (Micro-Electro-Mechanical Systems) exhibit several unique physical, chemical, and electronic properties at the nanoscale, which differentiate them from their bulk counterparts. Below are some of the key properties: Size and Scale: The most obvious property of MEMS is their miniature size, typically ranging from micrometers to millimeters. This reduced scale allows for compact, lightweight, and portable devices, which is not achievable in their bulk counterparts. Surface-to-Volume Ratio: At the nanoscale, MEMS devices have a higher surface-to-volume ratio compared to bulk devices. This increased surface area enhances interactions with the surrounding environment, leading to improved sensitivity in sensors and more efficient reactions in chemical and biological applications. Mechanical Properties: MEMS structures often exhibit different mechanical properties compared to their bulk counterparts. For example, nanoscale mechanical components can have enhanced flexibility, higher resilience, and reduced inertia due to their reduced mass and size. These properties are advantageous in applications such as accelerometers and gyroscopes. Thermal Properties: The reduced dimensions of MEMS components result in lower thermal mass and faster response to changes in temperature. This property is vital for accurate temperature sensing and thermal management in compact devices. Electrical Properties: MEMS devices at the nanoscale can exhibit unique electrical properties, such as high sensitivity to electrical signals. Piezoelectric and electrostatic properties are commonly utilized for sensing and actuation in MEMS devices. Optical Properties: Some MEMS devices, especially those incorporating optical elements, can exhibit novel optical properties due to the interaction of light with nanoscale structures. This enables the development of miniaturized optical sensors, displays, and communication systems. Material Properties: At the nanoscale, materials can exhibit different properties than at the bulk level. For example, nanomaterials may have altered mechanical, electrical, and optical characteristics due to quantum effects and surface interactions. MEMS devices can leverage these unique material properties for specific functionalities. Sensitivity: MEMS sensors at the nanoscale are highly sensitive to various stimuli, such as mechanical, electrical, thermal, and chemical changes. This sensitivity enables precise measurements and detections, making MEMS devices suitable for a wide range of applications, including healthcare, environmental monitoring, and industrial automation. Energy Efficiency: The reduced size and lower power requirements of MEMS devices at the nanoscale make them more energy-efficient compared to bulk systems. This energy efficiency is crucial for portable and battery-operated devices. Integration: MEMS devices can integrate multiple functionalities, such as sensing, actuation, and signal processing, on a single chip. This integration capability allows for the development of complex, multifunctional systems in a compact form factor. In summary, the unique physical, chemical, and electronic properties of MEMS at the nanoscale make them ideal for a wide range of applications, including sensors, actuators, and biomedical devices.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	**LLM-retrieved text**
Applications: Explore the wide range of applications where MEMs are utilized. Discuss their potential impact in fields such as electronics, energy, medicine, environmental remediation, etc.	MEMS (Micro-Electro-Mechanical Systems) have found extensive applications across various industries due to their unique properties and miniaturized form factor. Here are some of the key applications of MEMS in different fields: Electronics and Consumer Devices: (a) MEMS accelerometers and gyroscopes in smartphones and wearable devices enable motion sensing and orientation detection for gaming, navigation, and fitness tracking, (b) MEMS microphones are used in smartphones, tablets, and other consumer electronics for voice recording and noise cancellation, and (c) MEMS-based displays, such as digital micromirror devices(DMDs), are utilized in projectors and digital cinema. Healthcare and Medicine: (a) MEMS-based lab-on-a-chip devices allow for rapid and precise analysis of biological samples, enabling diagnostics, DNA sequencing, and disease detection, (b) Microfluidic MEMS devices are used for drug delivery systems, implantable medical devices, and micro-pumps for controlled drug release, and (c) MEMS sensors monitor vital signs in wearable health devices and provide real-time patient data for telemedicine applications. Automotive and Transportation: (a) MEMS accelerometers and gyroscopes are essential components in automotive safety systems, such as airbags, electronic stability control, and tire pressure monitoring systems, (b) MEMS pressure sensors are used in engine management and emissions control systems to optimize performance and fuel efficiency, and (c) MEMS-based inertial navigation systems provide precise navigation and positioning for autonomous vehicles and drones. Environmental Monitoring: (a) MEMS sensors are used for monitoring air quality, temperature, humidity, and gas concentrations in environmental monitoring systems, and (b) MEMS devices enable remote sensing and data collection for climate research and weather forecasting. Aerospace and Defense: (a) MEMS gyroscopes and accelerometers are critical components in aerospace applications for attitude control, navigation, and guidance systems, (b) MEMS pressure sensors are used in altitude and airspeed measurements in aircraft, and (c) MEMS-based infrared imaging devices are utilized in night vision systems for military and security applications. Energy Harvesting:(a) MEMS energy harvesters convert ambient energy, such as vibrations or thermal gradients, into electrical power, providing a sustainable energy source for low-power electronics and IoT devices. Industrial Automation: (a) MEMS-based sensors are used for condition monitoring, predictive maintenance, and feedback control in manufacturing and industrial processes, improving efficiency and reducing downtime. Robotics: MEMS sensors and actuators enable precise motion control and sensing in robots, making them more autonomous and capable of interacting with their environment. The potential impact of MEMS in these fields is immense. They contribute to increased efficiency, enhanced functionality, reduced energy consumption, and improved safety in various applications. As MEMS technology continues to advance, we can expect further integration, miniaturization, and performance improvements, leading to even more innovative applications across industries and benefiting society as a whole.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	**LLM-retrieved text**
Applications: Explore the wide range of applications where MEMs are utilized. Discuss their potential impact in fields such as electronics, energy, medicine, environmental remediation, etc.	Surface modification plays a crucial role in tailoring the properties of MEMS devices in the nanomaterials category. It involves altering the surface characteristics of the MEMS components to enhance their performance or enable specific applications. Here are some common strategies used for surface modification: Functionalization: Functionalization involves attaching or grafting specific molecules or functional groups onto the surface of MEMS devices. This process enhances the surface's chemical reactivity and allows for specific interactions with target substances. Functionalization can be achieved through chemical reactions or self-assembled monolayers (SAMs). Some applications of functionalization in MEMS include: Biomolecule Immobilization: Functionalizing the surface with biomolecules, such as antibodies or DNA probes, enables biosensing applications for disease detection and medical diagnostics. Gas Sensing: The surface functionalization of MEMS gas sensors with specific materials enhances their selectivity and sensitivity to target gases, making them suitable for environmental monitoring and industrial safety. Coating: Surface coating involves depositing thin layers of materials onto the MEMS surface to alter its properties. Coatings can be functional (active) or protective (passive) in nature. Some coating methods include physical vapor deposition (PVD), chemical vapor deposition (CVD), and atomic layer deposition (ALD). Coatings can enhance MEMS performance in various ways, such as: Anti-Stiction Coatings: Coating the MEMS surface with lubricants or hydrophobic materials reduces stiction and friction, which is crucial for reliable operation in micro-mechanical devices. Chemical Protection: Coatings can protect MEMS devices from chemical corrosion or degradation, increasing their durability and lifespan in harsh environments. Optical Coatings: Coatings with specific optical properties can enhance light transmission or reflection, improving the efficiency of optical MEMS devices like micro-mirrors and displays. Doping: Doping involves introducing impurity atoms into the material's surface to modify its electrical properties. This strategy is commonly used in semiconductors to create p-type or n-type regions and adjust the device's conductivity. Doping is critical for MEMS devices with electronic functionality, such as sensors and actuators. By carefully controlling the doping process, the sensitivity, response time, and power consumption of MEMS devices can be optimized for specific applications. Nanopatterning: Nanopatterning involves creating specific patterns or nanostructures on the MEMS surface. Techniques like electron beam lithography, nanoimprint lithography, and nanocontact printing can be used to fabricate intricate patterns. Nanopatterning enables applications such as: Enhanced Adhesion: Nanostructures can improve the adhesion between MEMS components and bonding surfaces, reducing the risk of delamination or failure. Superhydrophobicity: Nanostructured surfaces can achieve superhydrophobic properties, repelling water and other liquids. This is useful in self-cleaning or water-repellent MEMS devices. Overall, surface modification techniques allows to fine-tune the material's properties, improve performance.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	LLM-retrieved text
Toxicity and Safety: Address the potential health and environmental concerns associated with MEMS in this nanomaterials category. Discuss studies on their toxicity, risk assessment, and safety measures to mitigate any potential hazards.	As with any emerging technology, there are concerns about the potential health and environmental impacts associated with MEMS in the nanomaterials category. While MEMS devices themselves are generally considered safe, certain nanomaterials and fabrication processes used in their production can raise toxicity and safety issues. It's essential to address these concerns through rigorous studies, risk assessments, and implementation of safety measures. Here are some key points to consider: Nanomaterial Toxicity: Nanomaterials, due to their small size and increased surface area, may exhibit different properties and behaviors compared to their bulk counterparts. Some nanomaterials can potentially pose health and environmental risks if they are released into the environment or come into contact with living organisms. Toxicity studies are essential to assess the potential harm caused by exposure to nanomaterials used in MEMS devices. Risk Assessment: Researchers and regulatory bodies conduct risk assessments to evaluate the potential hazards associated with the use of nanomaterials in MEMS devices. These assessments consider exposure pathways, potential toxicity, and the likelihood of adverse effects. Risk assessment helps in identifying potential risks and implementing appropriate safety measures to minimize or eliminate hazards. Safety Measures: To mitigate potential health and environmental risks, safety measures can be implemented throughout the lifecycle of MEMS devices. These measures include: Engineering Controls: Implementing engineering controls during the fabrication process to minimize exposure to hazardous materials and ensure safe handling and disposal of nanomaterials. Personal Protective Equipment (PPE): Providing employees with appropriate PPE to prevent inhalation or skin contact with nanomaterials during fabrication or handling of MEMS devices. Workplace Safety Protocols: Establishing workplace safety protocols and guidelines for the safe handling, storage, and disposal of nanomaterials and MEMS devices. Environmental Regulation: Complying with environmental regulations and guidelines for the safe disposal of waste materials generated during MEMS fabrication. Environmental Impact Assessment: Environmental impact assessments evaluate the potential environmental impacts of introducing MEMS devices into various applications. These assessments help identify potential risks and provide insights into how to design MEMS devices with minimal environmental impact. Continued Research: Continued research into the toxicity of nanomaterials and the potential hazards associated with MEMS devices is essential. As the technology advances, new materials and fabrication processes may emerge, warranting further investigation to ensure safety. In conclusion, while MEMS devices in the nanomaterials category offer exciting possibilities for various applications, it is crucial to address potential health and environmental concerns associated with nanomaterials. Conducting thorough toxicity studies, risk assessments, and implementing safety measures are essential steps to ensure the responsible and sustainable development and use of MEMS technology. By prioritizing safety and environmental considerations, the benefits of MEMS can be harnessed while minimizing any potential risks.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	LLM-retrieved text
Future Directions: Discuss current research trends and future prospects for MEMS . Highlight emerging technologies, challenges, and areas of active exploration.	Current research trends and future prospects for MEMS (Micro-Electro-Mechanical Systems) are driven by a combination of technological advancements, societal demands, and industry needs. Here are some key areas of active exploration and emerging technologies in the field of MEMS: Internet of Things (IoT) and Smart Devices: The proliferation of IoT and smart devices is fueling the demand for MEMS sensors and actuators that are smaller, more power-efficient, and capable of providing precise data. Research is focused on developing low-power, miniaturized MEMS devices for applications in smart homes, wearable devices, environmental monitoring, and industrial automation, (b) Energy Harvesting: Energy harvesting using MEMS devices is a promising area of research. Researchers are exploring the use of MEMS energy harvesters to capture ambient energy from vibrations, thermal gradients, and solar radiation to power low-energy electronic devices and sensors, (C) MEMS-NEMS Integration: The integration of MEMS with NEMS (Nano-Electro-Mechanical Systems) is an area of active exploration. NEMS components, due to their smaller size and increased sensitivity, can enhance the performance and capabilities of MEMS devices, especially in sensing and actuation applications, (d) Nanomaterials and Nanofabrication: Advances in nanomaterials and nanofabrication techniques are enabling the development of novel MEMS devices with enhanced functionalities and improved performance. Researchers are exploring nanomaterial-based MEMS devices for applications in gas sensing, chemical detection, and bio-imaging, and (e) 3D Printing and Additive Manufacturing: 3D printing and additive manufacturing are being investigated for rapid prototyping and fabrication of complex MEMS structures. This technology allows for greater design flexibility and customization, facilitating innovation in MEMS development. Challenges: Despite the promising future of MEMS technology, some challenges need to be addressed: (a) Integration Complexity: As MEMS devices become more sophisticated and multifunctional, integration challenges arise. The integration of different materials, electronics, and sensors on a single chip requires precise fabrication techniques and design optimization. (b) Reliability and Longevity: MEMS devices are often used in harsh environments, and ensuring their reliability and longevity under such conditions is crucial. Materials compatibility, stiction, and packaging issues need to be carefully addressed, (c) Standardization: The lack of standardized processes and testing methods for MEMS devices can hinder their widespread adoption and commercialization. Standardization efforts are essential to ensure consistent performance and compatibility across different MEMS devices, and (d) Scaling and Manufacturing Costs: Scaling down MEMS devices to nanoscale dimensions presents manufacturing challenges and can increase production costs. Cost-effective fabrication methods for mass production are crucial for widespread adoption.

Table 8: LLM-retrieved text of GPT-3.5 for natural language queries about MEMS devices.

Query prompt	**LLM-retrieved text**
	Future Prospects: The future prospects for MEMS technology are promising, with potential impact in various sectors: (a) Healthcare and Medical Devices: MEMS-based medical devices are expected to revolutionize diagnostics, treatment, and personalized medicine, leading to better patient outcomes and healthcare efficiency, (b) Autonomous Systems: MEMS sensors and actuators will play a crucial role in enabling autonomous vehicles, drones, and robotics, advancing automation and safety across industries, (c) Environmental Monitoring and Sustainability: MEMS devices will continue to contribute to environmental monitoring and sustainability efforts by providing accurate data on pollution levels, climate change, and resource management, (d) Smart Cities: MEMS technology will be integral to the development of smart cities, enabling efficient infrastructure management, traffic control, and environmental monitoring, and (e) Wearable and Implantable Devices: MEMS-based wearable and implantable devices will enable continuous health monitoring and intervention, improving overall well-being and quality of life. In conclusion, the future of MEMS technology holds great promise, with advancements in nanomaterials, 3D printing, IoT, and healthcare applications driving innovation. Addressing current challenges and promoting collaboration between researchers, industry, and regulatory bodies will be key to unlocking the full potential of MEMS and realizing its transformative impact across various sectors.

References

[1] Neelay Shahet al. Vformer: A modular pytorch library for vision transformers. *GitHub. Note: https://github.com/SforAiDl/vformer*, 2022.

[2] Phil Wang al. Vision transformer - pytorch. *GitHub. Note: https://github.com/lucidrains/vit-pytorch*, 2022.

[3] James Atwood and Don Towsley. Diffusion-convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[4] Rossella Aversa, Mohammad Hadi Modarres, Stefano Cozzini, Regina Ciancio, and Alberto Chiusole. The first annotated set of scanning electron microscopy images for nanoscience. *Scientific data*, 5(1):1–10, 2018.

[5] Edoardo Barba, Niccolò Campolungo, and Roberto Navigli. Dmlm: Descriptive masked language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12770–12788, 2023.

[6] Jean-Paul Berrut and Lloyd N Trefethen. Barycentric lagrange interpolation. *SIAM review*, 46(3):501–517, 2004.

[7] Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[8] Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V Chawla. Graph barlow twins: A self-supervised representation learning framework for graphs. *arXiv preprint arXiv:2106.02466*, 2021.

[9] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[12] John Butcher. Runge-kutta methods. *Scholarpedia*, 2(9):3147, 2007.

[13] Joan Villoslada Camps and Angel Saz-Carranza. The european chips act: Europe's quest for semiconductor autonomy.

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[15] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[16] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. *arXiv preprint arXiv:2106.02689*, 2021.

[17] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 357–366, 2021.

[18] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.

[19] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR, 2020.

[20] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[21] Shoufa Chen, Enze Xie, Chongjian Ge, Runjian Chen, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*, 2021.

[22] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[23] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[24] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 589–598, 2021.

[25] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[26] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[27] Talgat Daulbaev, Alexandr Katrutsa, Larisa Markeeva, Julia Gusak, Andrzej Cichocki, and Ivan Oseledets. Interpolated adjoint method for neural odes. *arXiv preprint arXiv:2003.05271*, 2020.

[28] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[29] Aditya M Deshpande, Ali A Minai, and Manish Kumar. One-shot recognition of manufacturing defects in steel surfaces. *Procedia Manufacturing*, 48:1064–1071, 2020.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[32] Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*, 2017.

[33] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021.

[34] Igor Susmelj et al. Lightly. *GitHub. Note: https://github.com/lightly-ai/lightly*, 2020.

[35] RCNN Faster. Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 9199(10.5555):2969239–2969250, 2015.

[36] Mohsen Fayyaz, Soroush Abbasi Kouhpayegani, Farnoush Rezaei Jafari, Eric Sommerlade, Hamid Reza Vaezi Joze, Hamed Pirsiavash, and Juergen Gall. Ats: Adaptive token sampling for efficient vision transformers. *arXiv preprint arXiv:2111.15667*, 2021.

[37] Matthias Fey. Just jump: Dynamic neighborhood aggregation in graph neural networks. *arXiv preprint arXiv:1904.04849*, 2019.

[38] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[39] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019.

[40] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[41] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[42] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021.

[43] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[44] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 826–836, 2022.

[45] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[46] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

[47] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021.

[48] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[49] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[51] Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. *Advances in Neural Information Processing Systems*, 35:7264–7276, 2022.

[52] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[53] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[54] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021.

[55] David Basil Holt and David C Joy. *SEM microcharacterization of semiconductors*. Academic Press, 2013.

[56] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[57] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[58] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[59] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 5308–5317, 2016.

[60] Yongcheng Jing, Yining Mao, Yiding Yang, Yibing Zhan, Mingli Song, Xinchao Wang, and Dacheng Tao. Learning graph neural networks for image style transfer. In *European Conference on Computer Vision*, pages 111–128. Springer, 2022.

[61] Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. *arXiv preprint arXiv:2204.04879*, 2022.

[62] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[63] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[64] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[66] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018.

[67] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[68] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[69] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[70] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. *arXiv preprint arXiv:2107.08391*, 2021.

[71] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[72] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[73] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

[74] Mohammad Hadi Modarres, Rossella Aversa, Stefano Cozzini, Regina Ciancio, Angelo Leto, and Giuseppe Piero Brandino. Neural network for nanoscience scanning electron microscope image recognition. *Scientific reports*, 7(1):1–12, 2017.

[75] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

[76] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

[77] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

[78] Cedric Renggli, André Susano Pinto, Neil Houlsby, Basil Mustafa, Joan Puigcerver, and Carlos Riquelme. Learning to merge tokens in vision transformers. *arXiv preprint arXiv:2202.12015*, 2022.

[79] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, , Guzman Lopez, Nicolas Collignon, and Rik Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, page 4564–4573, 2021.

[80] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[81] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[82] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019.

[83] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[84] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[85] Yuichi Tanaka. Graph spectral filtering. *Graph Spectral Image Processing*, page 3, 2021.

[86] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10935–10944, 2022.

[87] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

[88] Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735*, 2018.

[89] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

[90] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5314–5321, 2022.

[91] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[92] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021.

[93] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[94] Eugene E Tyrtyshnikov. *A brief introduction to numerical analysis*. Springer Science & Business Media, 1997.

[95] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[96] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[97] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3056–3065, 2019.

[98] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[99] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, pages 1–10, 2022.

[100] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[101] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[102] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[103] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10041, 2021.

[104] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021.

[105] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

[106] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[107] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018.

[108] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7074–7083, 2020.

[109] Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. Cars: Continuous evolution for efficient neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1829–1838, 2020.

[110] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

[111] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[112] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. 2022.

[113] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[114] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An Empirical Study of Graph Contrastive Learning. *arXiv.org*, September 2021.

[115] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

[116] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.