VISION HGNN: AN ELECTRON-MICROGRAPH IS WORTH HYPERGRAPH OF HYPERNODES

[Sakhinana Sagar Srinivas[1]**, Rajat Kumar Sarkar**[1]**]**[*]**, Sreeja Gangasani**[2][†]**,**
Venkataramana Runkana[1]
TCS Research[1], Indian Institute of Technology[2]
sagar.sakhinana@tcs.com, rajat.sarkar1@tcs.com
111901023@smail.iitpkd.ac.in, venkat.runkana@tcs.com

ABSTRACT

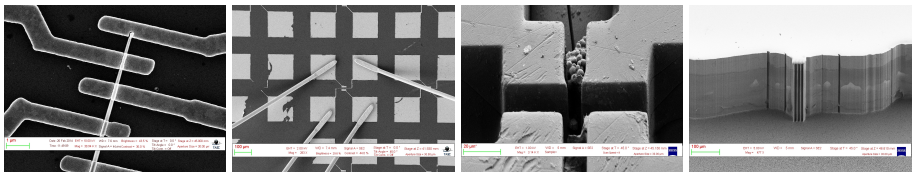
Material characterization using electron micrographs is a crucial but challenging task with applications in various fields, such as semiconductors, quantum materials, batteries, etc. The challenges in categorizing electron micrographs include but are not limited to the complexity of patterns, high level of detail, and imbalanced data distribution(long-tail distribution). Existing methods have difficulty in modeling the complex relational structure in electron micrographs, hindering their ability to effectively capture the complex relationships between different spatial regions of micrographs. We propose a hypergraph neural network(HgNN) backbone architecture, a conceptually alternative approach, to better model the complex relationships in electron micrographs and improve material characterization accuracy. By utilizing cost-effective GPU hardware, our proposed framework outperforms popular baselines. The results of the ablation studies demonstrate that the proposed framework is effective in achieving state-of-the-art performance on benchmark datasets and efficient in terms of computational and memory requirements for handling large-scale electron micrograph-based datasets.

1 INTRODUCTION

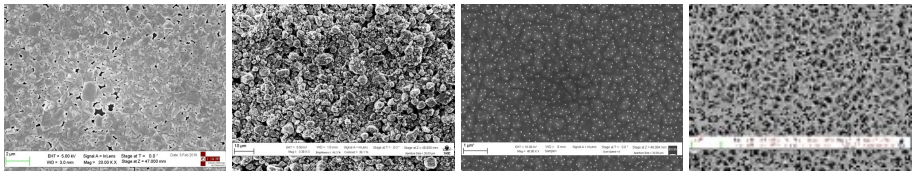
Accurate development, characterization, and testing of miniaturized semiconductor devices are essential in leading-edge chip design to ensure their proper functioning and performance. State-of-the-art imaging and analysis techniques(Holt & Joy (2013)) play a critical role in the fabrication, inspection, and testing of the next-generation miniaturized semiconductor devices, such as those with a feature size of 7nm or smaller, as they help ensure their quality and reliability. The advanced imaging of miniature devices in the semiconductor industry typically utilizes a broad spectrum of electron beam tools, including Scanning Electron Microscopy(SEM), Transmission Electron Microscopy(TEM), Reflective Electron Microscopy(REM), and others. Electron microscopes, as ultra-modern imaging tools, produce high-magnification and high-resolution images of material specimens, known as electron micrographs, to perform microstructural characterization or identification of materials, which is essential for the accurate design fabrication, and evaluation of miniaturized semiconductor devices. However, classifying electron micrographs is challenging due to high intra-class variance, low inter-class dissimilarity, and multiple spatial scales of visual patterns. Figure 1 illustrates the various challenges in the automatic nanomaterial identification task. The de facto standard neural-network architectures for vision tasks such as ConvNets(Iandola et al. (2016), He et al. (2016)), Vision transformers(ViTs, Dosovitskiy et al. (2020), Liu et al. (2021), d'Ascoli et al. (2021), Chen et al. (2021b)), hybrid architectures(Wu et al. (2021), Graham et al. (2021), Wang et al. (2022)) and MLP-based vision models(Tolstikhin et al. (2021), Touvron et al. (2021a)) do not explicitly model the higher-order dependencies between the multiple grid-like patches(also referred to as tokens) of the electron micrographs. Nevertheless, this work aims to explore an alternative effective, efficient neural-network architecture beyond traditional methods for modeling the fine-grained interrelations among the spatially and semantically dependent regions(patches) of the electron micrographs for automatic nanomaterial identification tasks via the hypergraph framework. We utilize the hypergraphs as a mathematical model(Ouvrard (2020); Feng et al. (2019); Gao et al. (2022); Yadati et al. (2019)) for a structured representation of the electron micrographs to learn the hierarchical relations among the spatial regions(patches) unconstrained by their spatial location in the micrograph. We begin

[*]Conceived, designed, implemented the research and drafted the manuscript
[†]Performed computational experiments, interpretation and visualization analysis of the results

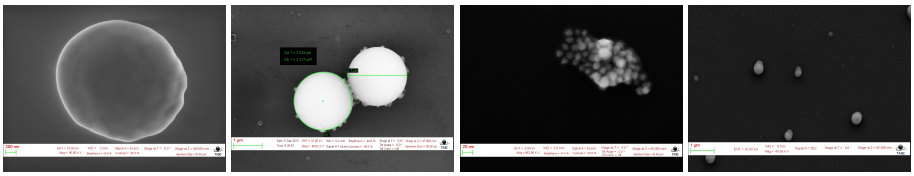
with a hypothesis, that the electron micrographs have an inherent hypergraph structure that capture a wide range of structural and property information from sub-hypergraphs of various sizes, and represents the higher-order dependencies between the patches in the electron micrographs. We present the Vision Hypergraph Neural Networks(for short, Vision-HgNN) designed to automatically encode the dominant structural and feature information of the visual hypergraphs and then learn relation structure-aware hypergraph-level embeddings to perform effective neural relational inference on the downstream multi-class classification task. The proposed visual hypergraph representation learning framework is designed to identify discrete visual elements(low-level entities) and their higher-order dependencies, and to prune redundant visual elements to learn an efficient representation of scale-variant visual elements(high-level entities) for improved perception and reasoning of the visual content in hypergraph-structured micrographs to enhance classification performance. The proposed framework is intended to offer better generalization and scalability for large-scale electron microscopy image corpus-based classification tasks.



(a) High intra-class dissimilarity in electron micrographs of *MEMS* device.



(b) The high inter-class similarity in different material categories(left to right: *films, powder, particles, porous sponges*).

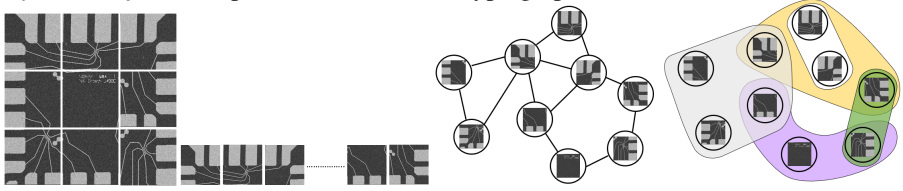


(c) Multi-spatial scales of patterns in electron micrographs of *particles*.

Figure 1: The figure depicts the various challenges in the electron micrograph classification task on the SEM dataset(Aversa et al. (2018)).

2 Problem Statement

Consider a dataset consisting of visual hypergraph-label pairs $(\mathcal{G}_i, y_i)$ where the ground-truth label of $\mathcal{G}_i$ is denoted by y_i. The objective of the classification task is to learn a novel mapping neural network function $f : \mathcal{G}_i \rightarrow y_i$ that maps the discrete visual hypergraphs to the set of predefined categories.



(a) Grid (b) Sequence (c) Graph (d) Hypergraph

Figure 2: For illustration purpose, an electron micrograph(MEMS device, Aversa et al. (2018)) was split into 3×3 patches. This figure depicts a regular grid, a sequence, a graph, and a hypergraph representation of an electron micrograph. (a) ConvNets operate on a grid of pixels, (b) ViTs operate on a sequence of grid-like patches, (c) GNNs operate on visual graphs where patches are viewed as nodes, and (d) HgNNs operate on visual hypergraphs where the patches represent the hypernodes to perform classification tasks. The visual graph and hypergraph structure representations are learned through the nearest neighbor search algorithm. They are linked based on the visual content and are not necessarily determined by their spatial location in the micrograph. The edges in the graph model pair-wise relations among the patches, while hyperedges model multi-dyadic relationships.

3 PROPOSED APPROACH

As illustrated in Figure 3, our framework consists of the following modules. (a) hypergraph structure learning module, for brevity, **HgSL** backbone, to learn the discrete visual hypergraph representations of the electron micrographs through pairwise proximity function. (b) a local and global neighborhood connectivity-driven hypergraph attention network, for brevity, the **HgAT** backbone is designed to capture short, and moderate-range dependencies, i.e., encapsulates the hypergraph's structural and feature information in the hypernode-level embeddings. (c) a self-attention mechanism-based-hypergraph transformer network, for brevity, an **HgT** backbone with no hypergraph spatial priors to learn all pairwise hypernode interactions for better learning of the long-range pairwise dependencies. (d) the hypergraph read-out module, for brevity, the **HgRo** backbone, performs the global average pooling that collapses hypernode-level embeddings to obtain the single hypergraph-level embedding. (e) a linear projection layer and a normalized exponential function transform the hypergraph-level embedding to a multinomial probability distribution over the predefined electron micrograph categories to predict the visual hypergraph category.

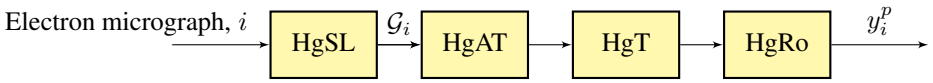


Figure 3: The isotropic Vision-HgNN architecture. y_i^p denotes the model predictions.

3.1 HYPERGRAPH STRUCTURE LEARNING(HGSL)

The HgSL operates in two phases. At first, it performs tokenization of electron micrographs. Next, it optimizes the discrete visual hypergraph structure through a differentiable approach to learn the more robust and optimal representation through the nearest neighbor search technique and formulate the posterior classification task as message-passing schemes with hypergraph neural networks.

3.1.1 ELECTRON MICROGRAPH TOKENIZER

We split an electron micrograph with the size $h \times w \times c$, where (h, w) is the resolution of the RGB image, and c is the number of channels into non-overlapping n uniform patches, where the size of each patch is $p \times p \times c$ and p is patch size. We reshape the patches to obtain feature matrix $\mathbf{X}' \in \mathbb{R}^{n \times p^2 c}$. We linearly transform the feature matrix, $\mathbf{X}'$, through a trainable embedding layer $\mathbf{E}$ to compute a refined feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ as described below,

$$\mathbf{X} = \mathbf{X}'\mathbf{E};\ \mathbf{E} \in \mathbb{R}^{p^2 c \times d} \tag{1}$$

The row i of the feature matrix $\mathbf{X} = [\mathbf{x}_p^1; \ldots; \mathbf{x}_p^n]$ represents the (low-dimensional) feature representation for patch $\boldsymbol{x}_p^i \in \mathbb{R}^d$, $i = 1, 2, \ldots, n$, where d is the predefined feature dimension.

3.1.2 HYPERGRAPH REPRESENTATION

We represent the patches as the unordered hypernodes of an undirected visual hypergraph denoted as $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. For each hypernode v_i, we form an undirected hyperedge e_p from the hypernode v_i to v_j; if v_j is among the top-K visual-semantic-nearest neighbors of v_i. Thus we obtain n hyperedges incident with K + 1 non-repeating hypernodes. The hyperedges describe the relations and capture more complex relationships and interdependencies among hypernodes in the visual hypergraphs. We then obtain a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ where $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ denotes the set of hyperedges and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the hypernode feature matrix. Each row i in $\mathbf{X}$ represents the hypernode feature vector, $x_{v_i} \in \mathbb{R}^d$. Note: x_{v_i} is the patch feature representation, x_p^i. The incidence matrix, $\mathbf{H} \in \mathbb{R}^{n \times n}$, describes the hypergraph structure. $\mathbf{H}_{i,\,p} = 1$ if the hyperedge p incident with hypernode i and otherwise 0. The hyperparameter K $< n$ determines the sparsity of the visual hypergraph. Let $\mathcal{N}_{p,i} = \{v_i | \mathbf{H}_{i,p} = 1\}$ represent the subset of hypernodes v_i incident with any hyperedge p. The intra-edge neighborhood of the hypernode i is given by $\mathcal{N}_{p,i} \backslash i$. It is a localized group of perceptually similar patches and captures higher-order relationships. The inter-edge neighborhood of hypernode i, $\mathcal{N}_{i,p} = \{e_p | \mathbf{H}_{i,p} = 1\}$, spans the spectrum of the set of hyperedges e_p incident with hypernode i. There is no natural ordering of the hypernodes in the hypergraph. To preserve patch-locality information in the main hypergraph, we linearly add the trainable position embeddings($\mathbf{E}_{pos}$) to the hypernode feature vectors to enable position awareness. The HgSL module computes the hypernode's positional embeddings based on the intra- and inter-edge neighborhood in the hypergraph. Equation 2 shows the transformed feature vector of the hypernodes.

$$[\mathbf{x}_{v_1}; \ldots; \mathbf{x}_{v_n}] = [\mathbf{x}_{v_1}, \ldots, \mathbf{x}_{v_n}] + \mathbf{E}_{pos};\ \mathbf{E}_{pos} \in \mathbb{R}^{n \times d} \tag{2}$$

3.2 Model Architecture

Figure 4 depicts the Vision-HgNN framework. The Hypergraph attention network(HgAT) performs the message-passing schemes on the visual hypergraphs to obtain (low-dimensional) hypernode embeddings. The Hypergraph transformer(HgT) utilizes the self-attention mechanism for transforming the hypernode embeddings determined by the HgAT operator to compute refined hypernode embeddings($\mathbf{z}_{v_1}^{L_{\text{HgT}}}, \ldots, \mathbf{z}_{v_n}^{L_{\text{HgT}}}$). We discussed the HgAT and HgT operators in the appendix. The HgRo module performs the average pooling of hypernode embeddings($\mathbf{z}_{v_1}^{L_{\text{HgT}}}, \ldots, \mathbf{z}_{v_n}^{L_{\text{HgT}}}$) to obtain hypergraph-level embedding $\mathbf{z}^{L_{\text{HgT}}} \in \mathbb{R}^d$. We apply a linear projection and softmax to transform $\mathbf{z}^{L_{\text{HgT}}}$ for determining the model predictions $y_i^p = \text{softmax}\big(\mathrm{W}^{out}\mathbf{z}^{L_{\text{HgT}}}\big)$, where $\mathrm{W}^{out} \in \mathbb{R}^{d\times d}$.

3.3 Algorithmic Architecture

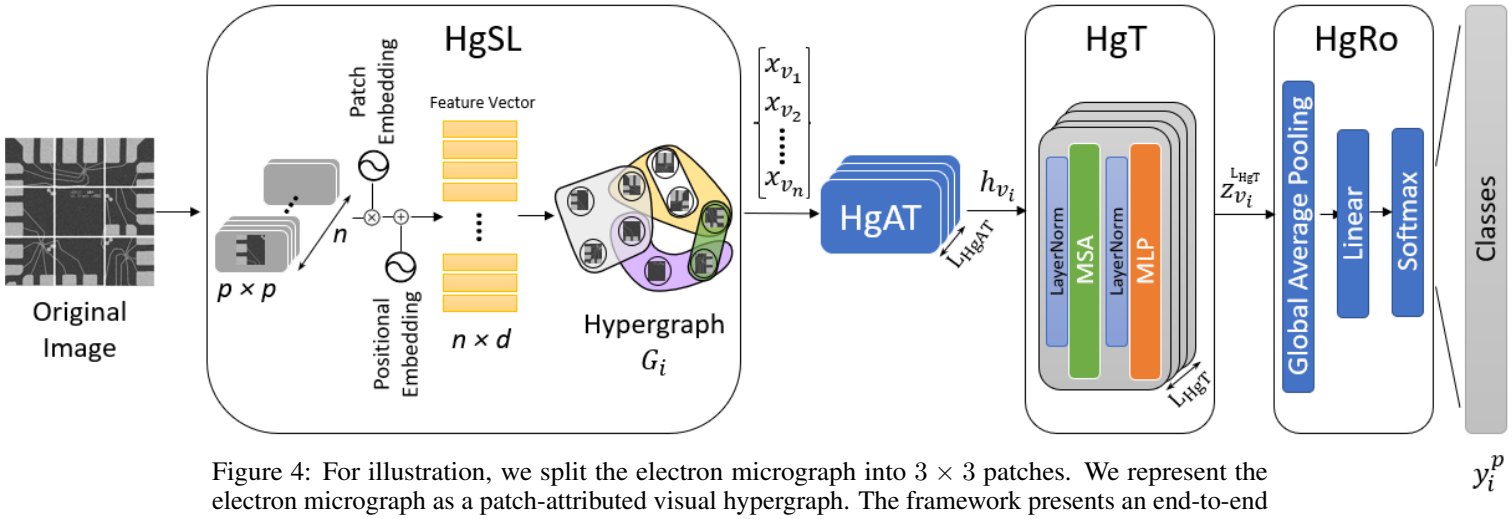


Figure 4: For illustration, we split the electron micrograph into 3×3 patches. We represent the electron micrograph as a patch-attributed visual hypergraph. The framework presents an end-to-end visual hypergraph representation learning with Hypergraph Neural Networks for categorization tasks.

4 Experiments and Results

4.1 Datasets

We conduct experiments on the SEM dataset(Aversa et al. (2018)) for automatic nanomaterials identification. The human-annotated dataset contains a set of 10 categories belonging to a wide range of nanomaterials spanning a broad range of particles, nanowires, patterned surfaces, etc., for a total of ≈21,283 electron micrographs. The initial experimental results are reported by Modarres et al. (2017) on the subset of the complete dataset. Due to the unavailability of the subset dataset publicly, we conducted experiments on the original dataset(Aversa et al. (2018)), which contains 12% more samples. The dataset curators(Aversa et al. (2018)) had not shared the predefined train/validation/test splits, so we leveraged the k-fold cross-validation technique to evaluate our model performance for competitive benchmarking with the varied baseline models. Additionally, we utilized several open-source material benchmark datasets to demonstrate the effectiveness of our proposed method.

	Algorithms	Parameters	Top-1	Top-2	Top-3	Top-5
ConvNets	AlexNet(Krizhevsky et al. (2017))	57.0M	0.519	0.593	0.681	0.785
	DenseNet(Huang et al. (2017))	0.24M	0.521	0.722	0.869	0.912
	ResNet(He et al. (2016))	0.27M	0.524	0.773	0.904	0.915
	VGG(Simonyan & Zisserman (2014))	34.4M	0.523	0.657	0.729	0.783
	GoogleNet(Szegedy et al. (2015))	0.26M	0.573	0.857	0.914	0.942
	SqueezeNet(Iandola et al. (2016))	0.74M	0.467	0.476	0.621	0.684
Vision Transformers(ViTs)	CCT(Hassani et al. (2021))	0.41M	0.586	0.795	0.886	0.952
	ConViT(d'Ascoli et al. (2021))	0.60M	0.596	0.724	0.836	0.941
	PVTC(Wang et al. (2022))	1.30M	0.572	0.758	0.826	0.917
	SwinT(Liu et al. (2021))	27.8M	0.658	0.763	0.904	0.928
	VanillaViT(Dosovitskiy et al. (2020))	1.79M	0.638	0.834	0.868	0.943
	CaiT(Touvron et al. (2021c))	0.38M	0.627	0.741	0.879	0.945
	LeViT(Graham et al. (2021))	16.8M	0.625	0.776	0.863	0.957
	NesT(Zhang et al. (2022))	16.1M	0.643	0.835	0.923	0.952
	PatchMerger(Renggli et al. (2022))	3.26M	0.561	0.703	0.842	0.939
	RegionViT(Chen et al. (2021a))	12.2M	0.589	0.812	0.863	0.936
	T2TViT(Yuan et al. (2021))	10.3M	0.672	0.841	0.911	0.927
	ViT-SD(Lee et al. (2021))	4.47M	0.609	0.746	0.873	0.949
	Vision-HgNN	0.91M	**0.819**	**0.864**	**0.942**	**0.994**

	Algorithms	Parameters	Top-1	Top-2	Top-3	Top-5
GCL	GBT(Bielak et al. (2021))	0.71M	0.524	0.612	0.703	0.794
	GRACE(Zhu et al. (2020))	0.74M	0.602	0.639	0.726	0.784
	BGRL(Thakoor et al. (2021))	0.69M	0.589	0.647	0.705	0.739
	InfoG(Sun et al. (2019))	0.68M	0.572	0.643	0.717	0.767
Graph Convolution Networks	APPNP(Klicpera et al. (2018))	0.74M	0.625	0.726	0.825	0.855
	AGNN(Thekumparampil et al. (2018))	0.52M	0.533	0.745	0.851	0.954
	ARMA(Bianchi et al. (2021))	0.45M	0.549	0.763	0.857	0.937
	DNA(Fey (2019))	0.84M	0.607	0.683	0.772	0.913
	GAT(Veličković et al. (2017))	0.63M	0.524	0.689	0.814	0.926
	GGC(Li et al. (2015))	0.81M	0.617	0.813	0.845	0.951
	GC(Morris et al. (2019))	0.59M	0.606	0.769	0.913	0.962
	GCN2C(Chen et al. (2020a))	0.62M	0.703	0.829	0.874	0.957
	CC(Defferrard et al. (2016))	0.50M	0.566	0.782	0.847	0.913
	GUNet(Gao & Ji (2019))	0.96M	0.635	0.746	0.873	0.928
	MPNN(Gilmer et al. (2017))	0.52M	0.662	0.825	0.896	0.972
	RGGC(Bresson & Laurent (2017))	0.66M	0.658	0.744	0.906	0.947
	SGAT(Kim & Oh (2022))	0.55M	0.592	0.694	0.892	0.955
	TAGC(Du et al. (2017))	0.57M	0.637	0.753	0.827	0.962
	Vision-HgNN	0.91M	**0.819**	**0.864**	**0.942**	**0.994**

	Algorithms	Parameters	Top-1	Top-2	Top-3	Top-5
VCL	Barlowtwins(Zbontar et al. (2021))	8.99M	0.176	0.264	0.337	0.449
	SimCLR(Chen et al. (2020b))	8.73M	0.189	0.243	0.408	0.475
	Byol(Grill et al. (2020))	8.86M	0.163	0.245	0.323	0.437
	Moco(He et al. (2020))	8.73M	0.174	0.196	0.264	0.468
	Nnclr(Dwibedi et al. (2021))	9.12M	0.153	0.271	0.441	0.538
	SimSiam(Chen & He (2021))	9.01M	0.196	0.301	0.416	0.561
	Vision-HgNN	0.91M	**0.819**	**0.864**	**0.942**	**0.994**

	Algorithms	Parameters	Top-1	Top-2	Top-3	Top-5
ViTs	CVT(Wu et al. (2021))	0.26M	0.551	0.764	0.843	0.965
	CrossViT(Chen et al. (2021b))	0.84M	0.493	0.738	0.842	0.963
	ATS(Fayyaz et al. (2021))	3.26M	0.536	0.725	0.805	0.942
	DeepViT(Zhou et al. (2021))	3.26M	0.537	0.762	0.893	0.961
	Distallation(Touvron et al. (2021b))	2.06M	0.524	0.733	0.859	0.953
	PiT(Heo et al. (2021))	4.48M	0.547	0.716	0.845	0.962
	Vision-HgNN	0.91M	**0.819**	**0.864**	**0.942**	**0.994**

Table 1: Comparative study of our proposed method and the baseline algorithms.

4.2 Benchmarking algorithms

We train the Vision-HgNN framework through a supervised learning approach for joint visual hypergraph inference and category prediction of micrographs. Table 1 provides a performance comparison of the Vision-HgNN framework with other baseline models, which include ConvNets, GNNs(Rozemberczki et al. (2021); Fey & Lenssen (2019)), and ViTs(al. (2022b;a)) architectures. In addition, we utilize the different self-supervised learning algorithms: Vision Contrastive Learning(VCL, et al. (2020))) and Graph Contrastive Learning(GCL, Zhu et al. (2021))) for comparison with our proposed method. We ensure a fair and rigorous comparison between the Vision-HgNN framework and the baseline algorithms by generating the results under identical experimental settings. The evaluation metric is the Top-N accuracy, where $N \in \{1, 2, 3, 5\}$. The standard deviation values are less than at most 4% of the mean value. The proposed method demonstrates the best performance with a high Top-1 accuracy score of 81.9% and a Top-5 score of 99.4%. The Vision-HgNN model brings a significant relative improvement of 21.87% and 16.50% in the Top-1 scores compared to the next-best baseline models T2TViT(Yuan et al. (2021)) among ViTs and GCN2C(Chen et al. (2020a)) among GNNs. The ablation studies, hyperparameters optimization, and other additional experimental results are reported and discussed in the appendix.

5 Conclusion

The challenge associated with the design of chips smaller than 7 nanometers is the increased complexity of the manufacturing process. As feature sizes shrink, the tolerance for errors in the manufacturing process decreases, making it more difficult to produce high-quality chips with consistent performance. In addition, the smaller feature sizes can lead to increased variability in the performance of the transistors, which can negatively impact the overall performance and reliability of the chip. Indeed, state-of-the-art imaging and analysis techniques are crucial in the development of next-generation semiconductor devices with feature sizes of 7nm or smaller. These techniques play a prominent role in the fabrication, inspection, and testing processes and are essential for driving the development of advanced microelectronics technologies. The high-resolution imaging of the device structures and materials allows for the identification of potential defects and deviations from design specifications, which can then be addressed through process optimization or design modification. In addition, these challenges present significant opportunities for innovation and the development of automatic material characterization methods for electron micrographs, which is essential for ensuring the quality and reliability of semiconductor devices. We conduct the first comprehensive study of the hypergraph-neural networks for electron micrograph classification tasks to improve the accuracy and efficiency of material characterization in various applications. We learn the optimal visual hypergraph structure to capture complex relationships and interactions between different spatial regions in an electron micrograph, allowing for a more robust and accurate representation of the micrograph. Hypergraph Neural Networks(HgNNs) operate on visual hypergraphs, where the patches in the micrograph are represented as hypernodes. The hyperedges in the hypergraph represent relationships between multiple patches, allowing for the capture of higher-order relationships between the patches in the micrograph. The experimental results corroborate our approach of augmenting HgAT layer stacks with a subsequent fully-connected transformer module(HgT) to achieve better performance compared to the state-of-art methods on automatic electron micrographs classification tasks. For future work, we would endeavor to generalize our framework on other electron micrograph datasets like REM, TEM, FE-SEM, STEM, etc., for anomaly detection, segmentation, etc.

References

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

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

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

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.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Jinheon Baek, Minki Kang, and Sung Ju Hwang. Accurate learning of graph representations with graph multiset pooling. *arXiv preprint arXiv:2102.11533*, 2021.

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.

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.

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

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

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

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*, pp. 357–366, 2021b.

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

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

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

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.

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.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4027–4035, 2021.

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

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.

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.

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

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*, pp. 9588–9597, 2021.

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

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.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3558–3565, 2019.

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

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

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

Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. Hgnn: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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*, pp. 1263–1272. PMLR, 2017.

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*, pp. 12259–12269, 2021.

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.

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.

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*, pp. 770–778, 2016.

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*, pp. 9729–9738, 2020.

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*, pp. 11936–11945, 2021.

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

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*, pp. 4700–4708, 2017.

EunJeong Hwang, Veronika Thost, Shib Sankar Dasgupta, and Tengfei Ma. Revisiting virtual nodes in graph neural networks for link prediction. 2021.

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.

Katsuhiko Ishiguro, Shin-ichi Maeda, and Masanori Koyama. Graph warp module: an auxiliary module for boosting the power of graph neural networks. *arXiv preprint arXiv:1902.01020*, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Siduo Jiang, Cristopher Benge, and William Casey King. Bertvision–a parameter-efficient approach for question answering. *arXiv preprint arXiv:2202.12210*, 2022.

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

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

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pp. 3499–3508. PMLR, 2019.

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

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *International conference on machine learning*, pp. 3734–3743. PMLR, 2019.

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

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

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

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*, pp. 10012–10022, 2021.

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.

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, pp. 4602–4609, 2019.

Xavier Ouvrard. Hypergraphs: an introduction and review. *arXiv preprint arXiv:2002.05014*, 2020.

Lin Pan, Chung-Wei Hang, Avirup Sil, and Saloni Potdar. Improved text classification via contrastive adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11130–11138, 2022.

Trang Pham, Truyen Tran, Hoa Dam, and Svetha Venkatesh. Graph classification via deep learning with virtual nodes. *arXiv preprint arXiv:1708.04357*, 2017.

Ladislav Rampášek and Guy Wolf. Hierarchical graph neural nets can capture long-range interactions. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2021.

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.

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*, pp. 4564–4573, 2021.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.

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

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.

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*, pp. 1–9, 2015.

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.

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.

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.

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. *arXiv preprint arXiv:2105.03404*, 2021a.

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*, pp. 10347–10357. PMLR, 2021b.

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*, pp. 32–42, 2021c.

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.

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

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

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*, pp. 1–10, 2022.

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*, pp. 22–31, 2021.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462. PMLR, 2018.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems*, 32, 2019.

Hongyuan Yu, Ting Li, Weichen Yu, Jianguo Li, Yan Huang, Liang Wang, and Alex Liu. Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2362–2368. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/328. URL https://doi.org/10.24963/ijcai.2022/328. Main Track.

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*, pp. 558–567, 2021.

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*, pp. 12310–12320. PMLR, 2021.

Weiqi Zhang, Chen Zhang, and Fugee Tsung. Grelen: Multivariate time series anomaly detection from the perspective of graph relational learning.

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.

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.

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

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

A APPENDIX

A.1 MODULES DESCRIPTION

A.1.1 HYPERGRAPH ATTENTION NETWORK(HGAT)

The HgAT(Veličković et al. (2017); Brody et al. (2021)) architecture extends the traditional convolution operation on the visual hypergraphs. It combines both local and global attention mechanisms to learn robust and expressive representations of visual hypergraphs. The hypergraph encoder(HgAT) is designed to utilize the hypergraph structure($\mathbf{H}$) and feature matrix($\mathbf{X}$) to compute the hypernode embeddings $\mathbf{h}_{v_i} \in \mathbb{R}^d, \forall i \in \mathcal{V}$. We learn the optimal embeddings $\mathbf{h}_{v_i}$ to preserve the high-level visual content embedded in the structural characteristics and feature attributes of the hypergraph. The layer-wise HgAT operator performs local and global-neighborhood aggregation for utilizing the powerful relational inductive bias of spatial equivariance encoded by the hypergraph's connectivity to model the fine-grained correlations explicitly between visual patches. We perform the attention-based intra-edge neighborhood aggregation for learning the latent hyperedge embeddings as,

$$\mathbf{h}_{e_p}^{(\ell)} = \sum_{z=1}^{\mathcal{Z}} \sigma\big(\sum_{i \in \mathcal{N}_{p,i}} \alpha_{p,i}^{(\ell,z)} \mathbf{W}_0^{(z)} \mathbf{h}_{v_i}^{(\ell-1,z)} \big), \ell = 1 \dots L_{\text{HgAT}} \tag{3}$$

where the superscript ℓ denotes the layer and for scenario ℓ = 1, $\mathbf{h}_{v_i}^{(0,z)} = \mathbf{x}_{v_i}$. $\mathbf{h}_{e_p} \in \mathbb{R}^d$ denotes the hyperedge embeddings. We compute multiple-independent embeddings in each layer with different trainable parameters and output summed-up embeddings. σ is the sigmoid function. The attention coefficient $\alpha_{p,i}$ determines the relative importance of the hypernode i incident with the hyperedge, p and is computed by,

$$\alpha_{p,i}^{(\ell,z)} = \frac{\exp\left(e_{p,i}^{(\ell,z)}\right)}{\sum_{i \in \mathcal{N}_{p,i}} \exp\left(e_{p,i}^{(\ell,z)}\right)}; e_{p,i}^{(\ell,z)} = \text{ReLU}\left(\mathbf{W}_0^{(z)} \mathbf{h}_{v_i}^{(\ell-1,z)}\right) \tag{4}$$

where $e_{p,i}$ denote the attention score. We then model the complex relations between hyperedges and hypernodes by performing the attention-based inter-edge neighborhood aggregation for learning the expressive hypernode embeddings as described by,

$$\mathbf{h}_{v_i}^{(\ell)} = \sum_{z=1}^{\mathcal{Z}} \text{ReLU}\left(\mathbf{W}_0^{(z)} \mathbf{h}_{v_i}^{(\ell-1,z)} + \sum_{p \in \mathcal{N}_{i,p}} \beta_{i,p}^{(\ell,z)} \mathbf{W}_1^{(z)} \mathbf{h}_{e_p}^{(\ell,z)}\right), \ell = 1 \dots L_{\text{HgAT}} \tag{5}$$

where $\mathbf{W}_0^{(z)}, \mathbf{W}_1^{(z)} \in \mathbb{R}^{d \times d}$ are learnable weight matrices. We utilize the ReLU function to introduce non-linearity for updating the hypernode-level embeddings. The normalized attention scores $\beta_{i,p}$ specifies the importance of hyperedge p incident with hypernode i and are computed by,

$$\beta_{i,p}^{(\ell,z)} = \frac{\exp(\phi_{i,p}^{(\ell,z)})}{\sum_{p \in \mathcal{N}_{i,p}} \exp(\phi_{i,p}^{(\ell,z)})}; \phi_{i,p}^{(\ell,z)} = \text{ReLU}\left(\mathbf{W}_3^{(z)} \cdot \left(\mathbf{W}_2^{(z)} \mathbf{h}_{v_i}^{(\ell-1,z)} \oplus \mathbf{W}_2^{(z)} \mathbf{h}_{e_p}^{(\ell,z)}\right)\right) \tag{6}$$

where $\mathbf{W}_2^{(z)} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_3^{(z)} \in \mathbb{R}^{2d}$ are weight matrix and vector. $\oplus$ denotes the concatenation operator. $\phi_{i,p}$ is the unnormalized attention score. Stacking multiple layers broadens the receptive field, but performance degrades due to over-squashing(Alon & Yahav (2020)) and over-smoothing issues(Li et al. (2018)). In between each layer, we apply batch norm and dropout for regularization. We concatenate the embeddings of each HgAT layer, transform it through a linear projection, and serve as input to the HgT network.

A.1.2 HYPERGRAPH TRANSFORMER(HGT)

The HgT operator generalizes the transformer neural networks(Vaswani et al. (2017)) to arbitrary sparse hypergraph structures with full attention as a desired inductive bias for generalization. The permutation-invariant HgT module models the pairwise relations between all hypernodes and updates the hypernode-level embeddings by exploiting global contextual information in the visual hypergraphs. The HgT module with no structural priors acts as a drop-in replacement for various other methods of stacking multiple HgNN layers with residual connections(Fey (2019), Xu et al. (2018)), virtual hypernode mechanisms(Gilmer et al. (2017); Pham et al. (2017)), or hierarchical pooling schemes(Rampášek & Wolf (2021), Lee et al. (2019)) to model the long-range correlations in the visual hypergraph. The HgT operator incentivizes learning the fine-grained relations to facilitate learning of expressive embeddings by spanning large receptive fields to effectively capture the high-level semantic information embedded in the hypergraph structure. The transformer encoder(Vaswani et al. (2017)) consists of alternating layers of multiheaded self-attention(MSA) and MLP blocks. We

apply Layernorm(LN(Ba et al. (2016))) for regularization and residual connections after every block. We skip the details for conciseness and to avoid notion clutter. Inspired by ResNets(He et al. (2016)), we add skip-connections through an initial connection strategy to relieve the vanishing gradients and over-smoothing issues.

$$\mathbf{h}'^{(\ell)}_{v_i} = \mathrm{MSA}\left(\mathrm{LN}\left(\mathbf{h}^{(\ell-1)}_{v_i} + \mathbf{x}^i_p\mathbf{E}\right)\right) + \mathbf{h}^{(\ell-1)}_{v_i}, \qquad \ell = 1 \ldots L_{\mathrm{HgT}} \tag{7}$$

$$\mathbf{z}^{\ell}_{v_i} = \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{h}'^{(\ell)}_{v_i}\right)\right) + \mathbf{h}'^{(\ell)}_{v_i}, \qquad \ell = 1 \ldots L_{\mathrm{HgT}} \tag{8}$$

We do not add position embeddings in skip-connections as HgAT operator had encoded the structural information into the hypernode embeddings. HgT overcomes the inherent information bottleneck of HgAT representational capacity for effective hypergraph summarization. It does so by learning hypernode-to-hypernode relations beyond the original sparse structure and distills the long-range information in the downstream layers to learn task-specific expressive hypergraph embeddings.

A.2 Algorithmic Architecture

Algorithm 1 summarizes the hypergraph structure learning(HgSL) module. While Algorithm 2 gives an overview of the Vision-HgNN framework. Our implementation utilizes the HgAT module to encode discrete visual hypergraphs to compute the hypernode embeddings, and the HgT module refines the embeddings through the self-attention mechanism. The HgRo module computes the hypergraph-level embedding to facilitate the classification task

Algorithm 1: Hypergraph structure learning

Input: Electron micrograph, $\mathbf{X}' \in \mathbb{R}^{h\times w\times c}$, where h is the height, w is the width, and c is the number of channels, patch size p, patch feature representation size (d)

Output: Visual hypergraph $\mathcal{G}$, incidence matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$, feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$

Model Parameters: Patch and Position embedding matrices $\mathbf{E} \in \mathbb{R}^{p^2c\times d}$, $\mathbf{E}_{pos} \in \mathbb{R}^{n\times d}$

1: reshape electron micrograph, $\mathbf{X}' \in \mathbb{R}^{h\times w\times c} \rightarrow \mathbf{X}' \in \mathbb{R}^{n\times p^2c}, \mathbf{x}^i_p \in \mathbb{R}^{p^2c}, i = 1, \ldots, n$

2: $[\mathbf{x}_{\mathbf{v_i}}, \ldots, \mathbf{x}_{v_n}] = \left[\mathbf{x}^1_p; \mathbf{x}^2_p; \cdots; \mathbf{x}^n_p\right]\mathbf{E}, \mathbf{x}_{\mathbf{v_i}} \in \mathbb{R}^d,\ i = 1, \ldots, n$ `// linear transformation`

3: $[\mathbf{x}_{\mathbf{v_i}}, \ldots, \mathbf{x}_{v_n}] = [\mathbf{x}_{\mathbf{v_i}}, \ldots, \mathbf{x}_{v_n}] + \mathbf{E}_{pos}$, `// add position embeddings`

4: $\mathbf{X} = [\mathbf{x}_{\mathbf{v_i}}, \ldots, \mathbf{x}_{v_n}], \mathbf{X} \in \mathbb{R}^{n\times d}$ `// feature matrix`

for *patch* $i = 1, \ldots, n$ **do**

  for *patch* $j = 1, \ldots, n$ **do**

    $d^{(p)}_{i,j} = \left(\sum_{z=1}^{d} |\mathbf{x}^{(\mathbf{z})}_{\mathbf{v_i}} - \mathbf{x}^{(\mathbf{z})}_{\mathbf{v_j}}|^2\right)^{1/2}; i \neq j$ `// ` $d^{(p)}_{i,j}$ `denotes the distance similarity measure between a pair of hypernodes,` i `and` j `connected by any hyperedge p`

  end

end

5: $\mathbf{H}_{j,p} = \mathbb{1}\{j \in \text{Top-K}\left(\min\left\{d^{(p)}_{i,j}\right\}, j \in \mathcal{V}\right)\}|\mathbf{H}_{i,p} = 1$

`//` Top-K `returns the indices of the K-nearest hypernodes of i`

Algorithm 2: Vision-HgNN framework

Input: Visual hypergraph $\mathcal{G}$, incidence matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$, the number of hypergraph attention layers L_{HgAT}, number of transformer layers L_{HgT}, feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$
Output: model predictions of electron micrographs category, y^p
Model Parameters: $\mathbf{W}_0^{(z)}, \mathbf{W}_1^{(z)}, \mathbf{W}_2^{(z)}, \mathbf{W}^{\mathbf{out}} \in \mathbb{R}^{n \times d}, \mathbf{E} \in \mathbb{R}^{p^2 c \times d}$
for *layer* $\ell = 1, \ldots, L_{HgAT}$ **do**
  evaluate $\alpha_{p,i}^{(\ell,z)}$ `// attention coefficient`
  $\mathbf{h}_{e_p}^{(\ell)} = \sum_{z=1}^{\mathcal{Z}} \sigma\left(\sum_{i \in \mathcal{N}_{p,i}} \alpha_{p,i}^{(\ell,z)} \mathbf{W}_0^{(z)} \mathbf{h}_{v_i}^{(\ell-1,z)}\right)$ `// hyperedge embeddings`
  evaluate $\beta_{i,p}^{(\ell,z)}$ `// attention coefficient`
  $\mathbf{h}_{v_i}^{(\ell)} = \sum_{z=1}^{\mathcal{Z}} \text{ReLU}\left(\mathbf{W}_0^{(z)} \mathbf{h}_{v_i}^{(\ell-1,z)} + \sum_{p \in \mathcal{N}_{i,p}} \beta_{i,p}^{(\ell,z)} \mathbf{W}_1^{(z)} \mathbf{h}_{e_p}^{(\ell,z)}\right)$ `// hypernode embeddings`
end
compute $\mathbf{h}_{v_i} = [\mathbf{h}_{v_i}^{(\ell)} || \ldots || \mathbf{h}_{v_i}^{(L_{\text{HgAT}})}]\mathbf{W}, \mathbf{W} \in \mathbb{R}^{d \cdot L_{\text{HgAT}} \times d}$
for *layer* $\ell = 1, \ldots, L_{HgT}$ **do**
  $\mathbf{h'}_{v_i}^{(\ell)} = \text{MSA}\left(\text{LN}\left(\mathbf{h}_{v_i}^{(\ell-1)} + \mathbf{x}_p^i \mathbf{E}\right)\right) + \mathbf{h}_{v_i}^{(\ell-1)}$ `// apply multi-head self-attention`
  $\mathbf{z}_{v_i}^{\ell} = \text{MLP}\left(\text{LN}\left(\mathbf{h'}_{v_i}^{(\ell)}\right)\right) + \mathbf{h'}_{v_i}^{(\ell)}$ `// refine hypernode-level embeddings`
end
$\mathbf{z}^{L_{\text{HgT}}} = \text{READOUT}(\{\mathbf{z}_{v_1}^{L_{\text{HgT}}}, \ldots, \mathbf{z}_{v_n}^{L_{\text{HgT}}}\})$ `// Aggregate hypernode embeddings`
$y_i^p = \text{softmax}\left(\mathbf{W}^{\mathbf{out}} \mathbf{z}^{L_{\text{HgT}}}\right)$ `// Category predictions`

A.3 Experimental setup

The data pre-processing involves per electron micrograph standard intensity normalization with a mean and covariance of 0.5 across all the channels to obtain normalized electron micrographs to the [-1, 1] range. The size of each electron micrograph in the SEM dataset (Aversa et al. (2018)) is $1024 \times 768 \times 3$ pixels. We resize electron micrographs to obtain a relatively lower spatial resolution, $256 \times 256 \times 3$ pixels, and split the downscaled electron micrographs into non-overlapping uniform patches of size $32 \times 32 \times 3$ pixels. The total number of patches(n) for each electron micrograph is 64. The position embedding($\mathbf{E}_{pos}$) and patch embedding($\mathbf{E}$) have a dimensionality size(d) of 128. We obtain the visual hypergraph representations of the electron micrographs through the Top-K nearest neighbor search algorithm with the optimal K value is 20. We utilize the k-fold cross-validation technique to evaluate the performance of our proposed method with k $= 10$. The training dataset comprises consecutive "k-2" of the folds. The validation and test dataset contains each with "1" in the remaining "2" folds. We implement an early stopping technique on the validation set to prevent the model from over-fitting and for model selection. The model is trained for 100 epochs to learn from the training dataset. We set the initial learning rate as $1e^{-3}$. The recommended batch size is 48. The optimal number of layers of HgAT and HgT operators, i.e., L_{HgAT} and L_{HgT}, are 2. The number of multiple-independent replicas($\mathcal{Z}$) is 4 for the HgAT operator. We utilize a learning rate scheduler to drop the learning rate by half if the Top-1 accuracy shows no improvement on the validation dataset for a waiting number of 10 epochs. We run the adam optimization algorithm to minimize the cross-entropy loss between the ground-truth labels and the model predictions. We train our model and baseline methods on multiple NVIDIA Tesla T4, Nvidia Tesla v100, and GeForce RTX 2080 GPUs built upon the PyTorch framework. We evaluate the model performance and report the evaluation metrics on the test dataset. Each computational experiment runs for a unique random seed. In this work, we report the ensemble average of the results obtained from five computational experiments. The experimental results reported are the average value of the different random seeds-based experimental run outputs.

A.4 Baseline Settings

We construct the visual graphs representation of the electron micrographs through the Top-K nearest neighbor search technique, where the patches had viewed as nodes and the edges model the pairwise associations between the semantic nearest-neighbor nodes. The baseline GNNs (Rozemberczki

et al. (2021); Fey & Lenssen (2019)) operate on the visual graphs to perform the classification task trained through the supervised learning approach. We utilize the graph contrastive learning(GCL, Zhu et al. (2021)) algorithms to learn the unsupervised node embeddings. The node-level graph encoder of the GCL algorithms had modeled by the GAT (Veličković et al. (2017)) algorithm. We compute the graph-level embedding through the sum-pooling of the node-level embeddings. The random Forest(RF) algorithm utilizes the unsupervised graph-level embeddings to predict the electron micrograph categories trained through the supervised learning technique. We report the RF model classification accuracy on the holdout data to evaluate the quality of unsupervised embeddings. The baseline ConvNets operate on the regular grid of electron micrographs to perform classification tasks trained through a supervised learning approach. The baseline ViTs (al. (2022b;a)) were trained on the supervised learning classification task to learn from the sequence of patches of each electron micrograph. Furthermore, we leverage visual-contrastive learning(VCL, et al. (2020)) techniques, i.e., computer-vision-based self-supervised algorithms for performing contrastive learning of visual representations to report classification accuracy. We utilized the ResNet backbone architecture for feature extraction. To reduce the baseline model complexity, we set the dimensionality size(d) of the patches to 64. We leverage the 10-fold cross-validation technique to evaluate the performance of the baseline methods. We train the baseline models for 100 epochs. The batch size is 48.

A.5 Study of Modules

We conduct detailed ablation studies to shed light on the relative contribution of modules for the improved overall performance of our framework. We gradually exclude the modules to design several variants of our framework and then investigate the variant model's performance compared to the Vision-HgNN model on the SEM dataset to demonstrate the efficacy and support the rationale of our modules. We refer to w/o HgAT and w/o HgT as Vision-HgNN models without HgAT and HgT modules, respectively. Table 2 shows the results of the ablation studies.

Algorithms	Top-1	Top-2	Top-3	Top-5	Avg-Precision	Avg-Recall	Avg-F1 Score
Vision-HgNN	**0.819±0.005**	**0.864±0.009**	**0.942±0.005**	**0.990±0.001**	**0.784±0.006**	**0.718±0.008**	**0.731±0.004**
w/o HgAT	0.682±0.008	0.783±0.007	0.900±0.013	0.968±0.007	0.713±0.008	0.662±0.005	0.657±0.005
w/o HgT	0.497±0.013	0.644±0.010	0.834±0.008	0.946±0.009	0.588±0.007	0.543±0.002	0.548±0.001

Table 2: The table reports the results of ablation studies.

Category	Multi-class metrics		
	Precision	Recall	F1 Score
Biological	0.700±0.016	0.728±0.004	0.713±0.007
Tips	0.705±0.014	0.656±0.007	0.608±0.016
Fibres	0.922±0.033	0.625±0.010	0.751±0.011
Porous Sponge	0.833±0.009	0.852±0.009	0.842±0.009
Films Coated Surface	0.849±0.002	0.837±0.003	0.843±0.000
Patterned surface	0.766±0.007	0.823±0.020	0.793±0.013
Nanowires	0.751±0.009	0.756±0.001	0.754±0.004
Particles	0.816±0.017	0.481±0.055	0.572±0.033
MEMS devices	0.676±0.003	0.705±0.000	0.670±0.013
Powder	0.822±0.013	0.721±0.002	0.767±0.004

Table 3: The table reports the performance of the Vision-HgNN framework on each electron-micrograph category classification task of the SEM dataset.

We additionally report the average precision, recall, and F1 score across the electron micrograph categories to evaluate the framework performance on the highly unbalanced SEM dataset. The w/o HgAT variant model reports a drop in performance of 9.06% in the precision score, 7.79% in the recall, and 10.12% in the F1 score w.r.t. Vision-HgNN model. Likewise, regarding w.r.t precision, recall, and F1 score, we observe a 25%, 24.37%, and 25.03% decline in w/o HgT variant model performance compared to the Vision-HgNN model. The variant model's performance disentangles the relative gains of each module and corroborates the hypothesis of the joint optimization of modules for better learning on the visual hypergraphs. The experimental results support our modules, HgAT and HgT effectiveness for capturing the short-range and high-level, long-range correlations, respectively, for the improved overall performance of our proposed framework. Table 3 reports the unaveraged multi-class evaluation metrics such as precision, recall, and F1 score of the Vision-HgNN framework performance on each predefined electron micrograph category. The results reported in Table 3 demonstrate that our proposed framework generalizes well despite the complexity of patterns across

the broad spectrum of electron micrograph categories, with a relatively higher score for the more labeled electron micrograph categories w.r.t. to the fewer labeled categories. The less electron micrograph-specific relational inductive bias offers an advantage for the Vision-HgNN framework to perform better on the classification task than the traditional methods.

A.6 Ablation Studies

Our proposed Vision-HgNN framework consists of HgSL, HgAT, HgT, and HgRo modules. We study the impact of each module in great detail that is responsible for the enhanced performance of our framework by substituting the modules with well-known algorithms of similar functionality to design replaced models. We compare the performance of the replaced models with our proposed framework to support the efficacy of our modules.

Study of HgSL Module:- We study the effectiveness of the HgSL module in modeling the complex relations among the spatially and semantically dependent visual patches for a structured representation of electron micrographs. The HgSL module learns the optimal K-uniform bi-directed hypergraph structure of the electron micrographs through the top-K nearest neighbor search algorithm. The most popular graph-based approaches for constructing the optimal graph structure are classified into two categories, (a) constructing a K-regular bi-directed graph or (b) the K-irregular bi-directed graph. The K-irregular graph construction techniques overcome the limitations of K-uniform graph structure learning techniques, which are not continuously differentiable. The former algorithmic approach in the literature includes (a) the top-K nearest neighbor search strategy using cosine similarity(CS,Deng & Hooi (2021)) and (b) structure learning through implicit correlations of node embeddings(MTGNN, Wu et al. (2020)). The later algorithmic techniques include (a) parametrization of the adjacency matrix and sample through the Gumbel reparameterization trick (Jang et al. (2016); Kool et al. (2019)) to output the link probability between nodes(GPT, Shang et al. (2021)), (b) regularized graph generation(RGG, Yu et al. (2022)) to learn sparse implicit graph by dropping redundant connections between nodes, and (c) graph relational learning using the self-attention mechanism(GRL, Zhang et al.) to construct a graph from observed data. We refer to the Vision-HgNN model for which the HgSL module is modeled with the different operators as follows,

- **w/** CS: Vision-HgNN model with the CS operator.
- **w/** MTGNN: Vision-HgNN model with the MTGNN operator.
- **w/** GPT: Vision-HgNN model with the GPT operator.
- **w/** RGG: Vision-HgNN model with the RGG operator.
- **w/** GRL: Vision-HgNN model with the GRL operator.

Table 4 reports the performance of the replaced models compared to the Vision-HgNN framework. The Top-1 scores of the substituted models, **w/** MTGNN, w/ GPT, w/ RGG, w/ GRL declined by 10.74%, 12.94%, 15.63% , 12.69% on SEM dataset compared to the Vision-HgNN model. The impact of w/ CS is marginal and achieves on-par performance compared to the Vision-HgNN model. The results show the advantages of utilizing the HgSL module modeled with the top-K nearest neighbor search technique, a simple yet effective similarity learning technique based on the Euclidean distance between patches for capturing the underlying higher-order relational information in visual hypergraphs.

Algorithms	Top-1	Top-2	Top-3	Top-5	Avg-Precision	Avg-Recall	Avg-F1 Score
Vision-HgNN	0.819	0.864	0.942	0.994	0.784	0.718	0.731
w/ CS	0.790	0.833	0.909	0.959	0.756	0.693	0.706
w/ MTGNN	0.731	0.763	0.843	0.897	0.699	0.639	0.644
w/ GPT	0.713	0.744	0.822	0.875	0.682	0.616	0.632
w/ RGG	0.691	0.718	0.788	0.844	0.668	0.621	0.627
w/ GRL	0.715	0.749	0.826	0.862	0.694	0.633	0.645

Table 4: The table reports the comparative study of the various structure learning techniques.

Study of HgAT Module:- (a) We study the importance of the **attention mechanism** in the hypernode-level hypergraph encoder(HgAT) to compute the expressive hypernode embeddings($\mathbf{h}_{v_i}$) of visual hypergraphs. We disable the attention mechanism of the layerwise HgAT operator. We perform the unweighted sum-pooling operation on the neural messages in the intra- and inter-neighborhood aggregation scheme for computing hypergraph embeddings. We refer to the Vision-HgNN model in the absence of the attention-mechanism for determining the hyperedge($\mathbf{h}_{e_p}$) and hypernode($\mathbf{h}_{v_i}$) embeddings as follows,

- **w/o** $\alpha_{\mathbf{p},\mathbf{i}}, \beta_{\mathbf{i},\mathbf{p}}$:**Vision-HgNN** model without the attention mechanism.

Eliminating the attention mechanism degrades the replaced model performance, as evident in Table 5. In particular, it decreases the Top-1 accuracy w.r.t. the Vision HgNN model by more than 13.79% on the SEM dataset. The attention mechanism automatically learns the relative importance among the incident hypernodes and hyperedges during the local and global neighborhood aggregation. It provides beneficial inductive bias to capture the dominant-visual patterns in the electron micrographs across the categories. The hypergraph attention mechanism models the fine-grained correlations for encoding the higher-order relationships in the embeddings for better learning the visual hypergraphs.

Algorithms	Top-1	Top-2	Top-3	Top-5	Avg-Precision	Avg-Recall	Avg-F1 Score
Vision-HgNN	0.819	0.864	0.942	0.994	0.784	0.718	0.731
w/o $\alpha_{\mathbf{p,i}}, \beta_{\mathbf{i,p}}$	0.706	0.742	0.812	0.827	0.667	0.619	0.630

Table 5: The table shows the comparative study of the model's performance with and without the attention mechanism in HgAT.

(b) We analyze the usefulness of **latent master-hypernode** to encode the long-range hypernode relations in the visual hypergraph embeddings for enhancing classification performance. We add a superhypernode(or virtual master hypernode, Gilmer et al. (2017); Ishiguro et al. (2019); Pham et al. (2017); Hwang et al. (2021)) for augmenting the visual hypergraphs. The master hypernode had connected to all hyperedges of the augmented hypergraph. It provides an additional route for neural message-passing schema on visual hypergraphs. The master hypernode embeddings contain the global latent information of the visual hypergraphs. Each hyperedge reads and writes to transform the master hypernode embedding through the intra- and inter-neighborhood aggregation-based message-passing attention networks. We design a replaced model by disabling the HgT module to operate on the augmented hypergraphs as follows,

- **w/** Virtual: **Vision-HgNN** model without the HgT module to operate on augmented hypergraphs.

The degradation of the variant performance is evident in Table 6, which shows a drop of 11.72% in the Top-1 accuracy compared to the Vision HgNN model. The latent "master" hypernode is ineffective for capturing long-range hypernode-to-hypernode relations to encode the spatial dependencies among patches in the hypergraphs embeddings for effectively learning on the visual hypergraphs-topology compared to the HgT module.

Algorithms	Top-1	Top-2	Top-3	Top-5	Avg-Precision	Avg-Recall	Avg-F1 Score
Vision-HgNN	0.819	0.864	0.942	0.994	0.784	0.718	0.731
w/ Virtual	0.723	0.761	0.832	0.847	0.684	0.634	0.648

Table 6: The table shows the comparative study of the model's performance with and without the latent "master" hypernode.

Algorithms	Top-1	Top-2	Top-3	Top-5	Avg-Precision	Avg-Recall	Avg-F1 Score
Vision-HgNN	0.819	0.864	0.942	0.994	0.784	0.718	0.731
w/ TopK Pooling	0.695	0.731	0.803	0.814	0.657	0.603	0.620
w/ SAG Pooling	0.732	0.770	0.843	0.866	0.699	0.642	0.656

Table 7: The table shows the comparative study of the model's performance with different hierarchical pooling schemes.

(c) We study the effectiveness of **hierarchical pooling schemes**(Gao & Ji (2019); Lee et al. (2019)) for better learning the long-range, higher-order dependencies in visual hypergraphs. The spatial pooling operator learns the hierarchical representations in a two-stage approach to model the large portions in the visual hypergraphs. (a) assigns a score to all hypernodes and drops the low-scoring hypernodes and the corresponding hyperedges, which contain noise or less prominent patterns. (b) samples the high-scoring hypernodes to obtain hierarchical-induced visual subhypergraphs. (c) performs the hierarchical message passing schemes on pooled hypergraphs for encoding the hypergraph's local and global structure information to compute higher-order representations of the visual hypergraphs. We obtain a replaced model by (a) disabling the HgT module, (b) stacking three layers of pooling operators(pooling ratio(p_r) as 0.75) interleaved with the HgAT module, and refer to the Vision-HgNN model as follows,

- **w/** TopK Pooling: **Vision-HgNN** model with the TopK Pooling operator (Gao & Ji (2019)).

- **w/** SAG Pooling: **Vision-HgNN** model with the SAG Pooling operator(Lee et al. (2019)).

The performance degradation of the replaced models was evident in the SEM dataset, as indicated in Table 7. The replaced models, w/ TopK Pooling, w/ SAG Pooling yields an overall 15.14%, 10.62% relative lower Top-1 accuracy compared to the Vision-HgNN model. The self-attention mechanism-based HgT module is more effective for capturing long-range spatial dependencies by computing all pairwise interactions among hypernodes through a position-agnostic fashion for learning about visual hypergraphs.

Study of HgT Module:- Inspired by using a special-classification token(<CLS>) in transformer architectures (Jiang et al. (2022); Pan et al. (2022); Devlin et al. (2018)) for sentence-level classification tasks. We will append a special-patch(<CLS>) embedding along with the hypernode's embeddings corresponding to the visual hypergraphs as the input set of embeddings sequence to the HgT module. Note: We do not add the special patch(<CLS>) as a virtual hypernode to the visual hypergraph. This method overcomes the drawbacks of the information bottleneck problems of the virtual hypernode augmentation technique. The former algorithm prevents learning pairwise relationships between hypernodes except with the virtual hypernode. The special-patch incorporated variant model encodes the one-to-one relations between the <CLS> patch and every other patch of the visual hypergraphs in the hypergraph embeddings with the self-attention module. We hypothesize that the special-patch(<CLS>) embedding contains complete visual hypergraph information. We apply the softmax operator to special-patch embedding and predict the category of the visual hypergraphs. Thus, we obtain a replaced model with the special-patch-inspired readout mechanism coupled with the HgT module as follows,

- w/ <CLS>: The Vision-HgNN model with <CLS> patch in HgT module and disabled HgRo module.

Algorithms	Top-1	Top-2	Top-3	Top-5	Avg-Precision	Avg-Recall	Avg-F1 Score
Vision-HgNN	0.819	0.864	0.942	0.994	0.784	0.718	0.731
w/ <CLS>	0.747	0.779	0.839	0.917	0.707	0.672	0.677

Table 8: The table reports the comparative study of the special-token-based hypergraph readout mechanism with our proposed method.

The experimental results had illustrated in Table 8. We notice a drop in the variant performance compared to our proposed method. The Top-1 accuracy of the design variant, w/ <CLS>, declined by 8.79% on the SEM dataset compared to the Vision-HgNN model.

Algorithms	Top-1	Top-2	Top-3	Top-5	Avg-Precision	Avg-Recall	Avg-F1 Score
Vision-HgNN	0.819	0.864	0.942	0.994	0.784	0.718	0.731
w/GMT	0.774	0.816	0.890	0.939	0.741	0.679	0.691
w/GA	0.786	0.828	0.884	0.954	0.752	0.703	0.716
w/Set2Set	0.719	0.741	0.818	0.882	0.698	0.650	0.658
w/GSM	0.737	0.753	0.841	0.909	0.716	0.666	0.675

Table 9: The table reports the comparative study of the hypergraph readout baseline operators.

Study of HgRo Module:- We probe the HgRo module's effectiveness compared to the well-known algorithms of identical functionality. The hypergraph readout module performs global average pooling on the hypernode-level embeddings to compute the visual hypergraph-level embedding. We utilize well-known methods as baseline operators to perform global-pooling operations on the hypergraphs. The list includes GraphMultisetTransformer(GMT, Baek et al. (2021)), GlobalAttention(GA, Li et al. (2015)), Set2Set(Vinyals et al. (2015)), and Global Summation Pooling(GSM). We refer to the **Vision-HgNN** model with the baseline operators for modeling the global readout function(HgRo module) as follows:

- w/ GMT: The Vision-HgNN model with GMT operator.
- w/ GA: The Vision-HgNN model with GA operator.
- w/ Set2Set: The Vision-HgNN model with Set2Set operator.
- w/ GSM: The Vision-HgNN model with GSM operator.

The results reported in Table 9 across all the evaluation metrics demonstrate no significant improvements in the replaced model's performance compared to our proposed method with the global average pooling operator. Overall, our hypergraph readout module proves effective by learning to compute the optimal hypergraph-level representations while maximally preserving the visual hypergraph's global-contextual information.

A.7 HYPERPARAMETER STUDIES

We perform an in-depth hyperparameter tuning to determine the optimal hyperparameters of our framework. The algorithm hyperparameters are (1) the dimensionality of embedding(d). (2) The stack of HgAT operators(L_{HgAT}) and the HgT operators(L_{HgT}). Hyperparameters had chosen from the following ranges: embedding dimension(d) $\in [32, 256]$, $L_{\text{HgAT}} \in [1, 8]$, and $L_{\text{HgT}} \in [1, 6]$. We perform hyperparameter optimization using the grid-search technique to yield the optimal performance of our proposed method on the validation dataset in terms of the Top-1 classification accuracy. For each experiment, we change the hyperparameter under investigation to determine the impact on the model performance. The optimal hyperparameters determined from the study are as follows, d is 128, L_{HgAT}, and L_{HgAT} is 2.

d	32	64	128	256
	0.632	0.740	0.813	0.802

L_{HgAT}	1	2	4	8
	0.712	0.813	0.673	0.520

L_{HgT}	1	2	4	6
	0.754	0.813	0.830	0.798

Table 10: The table reports the experimental results of the hyperparameter study.

A.8 SEM DATASET

The SEM dataset contains ten diverse electron micrograph categories of nanomaterials such as *biological, fibers, films, MEMS, patterned surfaces, etc*. Table 11 shows the unequal distributions in the total count of each electron micrograph category. A few illustrative electron micrographs belonging to the different nanomaterials had shown in Figure 5.

Category	**Number of images**
Biological	973(4.57%)
Tips	1625(7.63%)
Fibres	163(0.76%)
Porous Sponge	182(0.85%)
Films Coated Surface	327(1.53%)
Patterned surface	4756(22.34%)
Nanowires	3821(17.95%)
Particles	3926(18.44%)
MEMS devices and electrodes	4591(21.57%)
Powder	918(4.31%)
Total	21282

Table 11: Summary of SEM dataset: The listing of electron micrographs count per category.

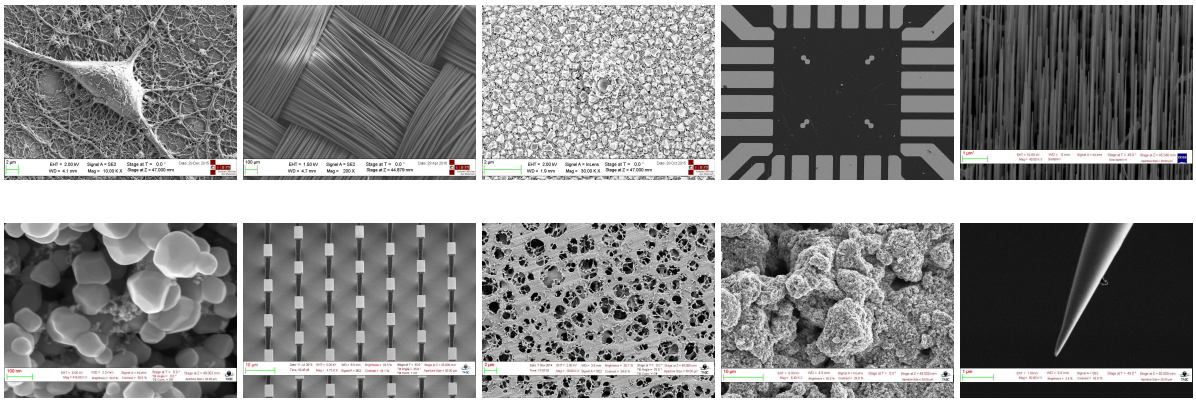


Figure 5: The figure shows the electron micrograph categories in the SEM dataset (Aversa et al. (2018))(left to right in the first row: *biological, fibers, films, MEMS, nanowires*; left to right in the second row: *particles, patterned surface, porous sponges, powder, tips*).

A.8.1 BENCHMARKING ON OPEN-SOURCE MATERIAL DATASETS

- **NEU-SDD**[1] (Deshpande et al. (2020)) is a hot-rolled steel-strip surface defects database consisting of 1800 gray-scale electron micrographs with a resolution of 200×200 pixels. The dataset had categorized into six surface defect classes, where each class has 300 electron micrographs. The list includes pitted surfaces, scratches, rolled-in scale, crazing, patches, and inclusion defects. For illustration, each category's representative electron micrograph had shown in Figure 6. We compare the performance of our proposed method with several baseline algorithms on the multiclass classification task.
- **CMI**[2] is a benchmark dataset for automating corrosion assessment of materials, which consists of 600 corroding panel-based electron micrographs of resolution 512×512 pixels. The human-annotated dataset had manually assigned a rating to each electron micrograph by the corrosion experts following the ASTM-D1654 standards. The discrete rating variables range from 5 to 9, where the corrosion rating of 9 implies that the panel is in the initial stage of corrosion. Each corrosion rating-based panel has 120 electron micrographs, and Figure 7 shows the representative electron micrographs for each corrosion rating. We report the performance of our proposed method compared to the several baseline algorithms on the multiclass classification task.
- **KTH-TIPS**[3] is a texture database that contains 810 electron micrographs of ten different materials. The electron micrograph resolution in the dataset is 200×200 pixels. In each material category, the corresponding electron micrographs have varying illumination, pose, and scale of the material under examination. The ten materials categories are *sponge, orange peel, styrofoam, cotton, cracker, linen, brown bread, sandpaper, crumpled aluminum foil, and corduroy*. A few example electron micrographs per category had shown in Figure 8. We evaluate and report the performance of our proposed method compared to the several baseline algorithms on the multiclass classification task.

Table 12 shows the performance comparison of our proposed method w.r.t to the baselines across all the datasets. The experimental results demonstrate the efficacy of our proposed method.

	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
	Vision-HGNN	**0.975**	**0.981**	**0.987**

Table 12: Performance comparison on the datasets.

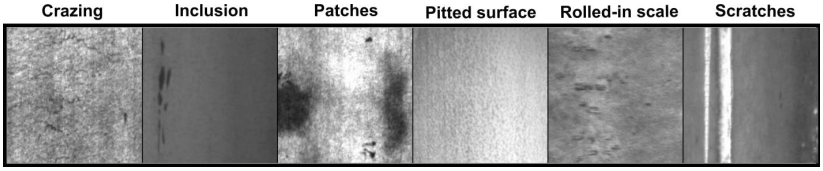


Figure 6: The NEU-SDD dataset: We illustrate the electron micrographs of six defect categories of hot-rolled steel strip 1(Deshpande et al. (2020)).

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

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

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

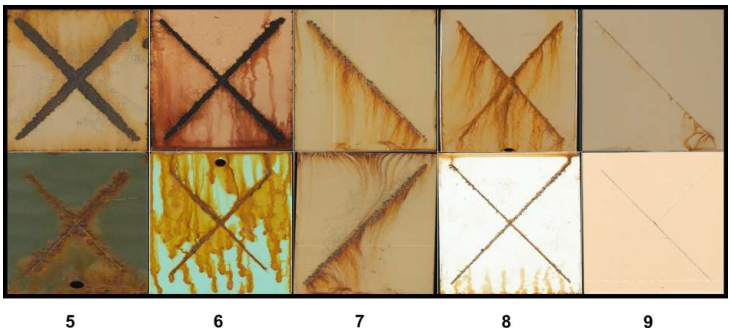


Figure 7: The CMI dataset: The representative electron micrographs of five corrosion rating classes 2.

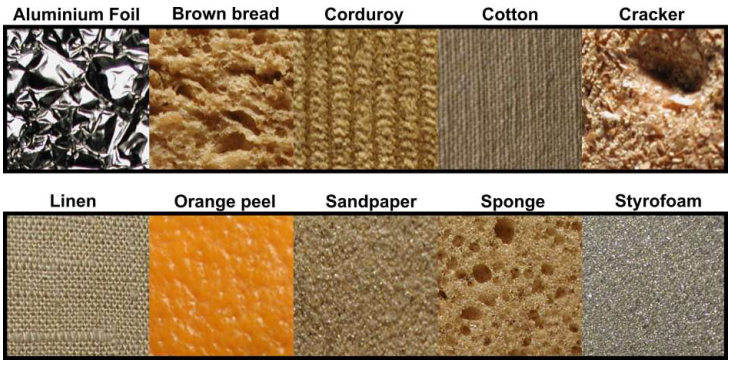


Figure 8: The KTH-TIPS dataset: Illustration of electron micrographs belonging to ten different materials 3.