SearchViG: Optimal Vision GNNs via Ramanujan Spectral Optimization

Mustafa Munir, Md Mostafijur Rahman, Xiwen Wei, Yuedong Yang, Radu Marculescu
Department of Electrical and Computer Engineering, The University of Texas at Austin
{mmunir, mostafijur.rahman, xiwenwei, albertyoung, radum}@utexas.edu

Abstract

Vision Graph Neural Networks (ViGs) are often limited by their reliance on a fixed, homogeneous graph construction rule applied across all network stages. To address this limitation, we introduce SearchViG, a novel framework that automatically designs optimal heterogeneous architectures by performing a Graph Construction Search (GCS) to produce the optimal graph topology for each stage within our designated search space. Our search is guided by a zero-shot, theoretically-grounded proxy: the spectral gap of the graph's adjacency matrix, which quantifies its Ramanujan-like expansion properties, provably linking it to superior information flow. SearchViG discovers new heterogeneous architectures that assign different graph topologies, number of neighbors, and hops between neighbors based on feature resolution. Our resulting models establish a new state-of-the-art Pareto frontier for Vision GNNs. For instance, our SearchViG-M achieves 83.3% top-1 accuracy, outperforming both Vision GNN-B (ViG-B) and Vision Hypergraph Neural Network-B (ViHGNN-B) while using over 70% fewer parameters and 80% fewer GMACs. This efficiency extends to downstream tasks, where our lightweight SearchViG-S obtains 43.4 mIoU, 43.5 AP^{box}, and 39.9 AP^{mask}, surpassing the much larger Pyramid Vision Transformer-Large (PVT-Large) across all metrics while using 80% fewer parameters. Code is available at https://github.com/SLDGroup/SearchViG.

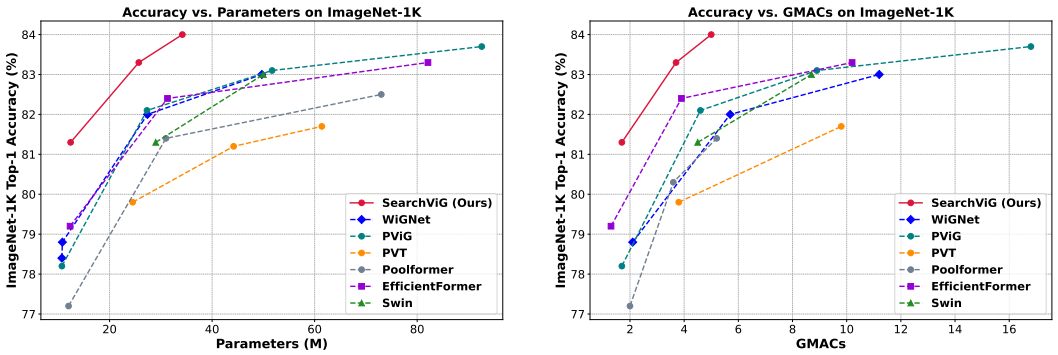
1 Introduction

The rapid growth of deep learning has led to remarkable successes across a diverse set of computer vision tasks, including image classification [1, 2], object detection [3, 4], instance segmentation [5], and semantic segmentation [6]. Key architectural paradigms driving this progress include convolutional neural networks (CNNs) [1, 7, 8], Vision Transformers (ViTs) [2, 9], and multi-layer perceptron (MLP)-based vision models [10, 11].

In CNNs and MLPs, input images are represented as a grid of pixels, whereas ViTs represent images as a sequence of patch embeddings. By splitting an image into fixed-size patches and linearly embedding them, ViTs leverage Transformer modules that were originally developed for natural language processing [12] to model long-range dependencies. Unlike CNNs and MLPs, which have local receptive fields, ViTs possess a global receptive field, which enables them to capture relationships between distant regions in an image.

Recently, Vision Graph Neural Networks (ViGs) have emerged as an alternative in visual representation learning [13]. ViGs represent an image as a graph, where nodes correspond to patches and edges are formed using a k-nearest neighbors (KNN) [14] graph construction strategy. This representation offers a more flexible connectivity pattern than the rigid grids of CNNs or the sequential ordering of ViTs. Extensions such as the Vision Hypergraph Neural Network (ViHGNN) [15] generalize ViGs by replacing pairwise connections with hyper-edges, enabling richer relational modeling. Other works [16–19] have explored dynamic and efficient graph construction, but they all still apply a single homogeneous graph construction rule across all network stages.

M. Munir et al., SearchViG: Optimal Vision GNNs via Ramanujan Spectral Optimization. *Proceedings of the Fourth Learning on Graphs Conference (LoG 2025)*, PMLR 269, Hybrid Event, December 10–12, 2025.



(a) Accuracy vs. Parameters (M) **(b)** Accuracy vs. GMACs

Figure 1: Comparison of model efficiency and performance on ImageNet-1K. SearchViG establishes a new state-of-the-art Pareto frontier, achieving superior top-1 accuracy with significantly fewer **(a)** parameters and **(b)** GMACs compared to competing vision models.

Despite these advances, all existing ViG-based architectures employ static, manually designed, or homogeneous graph construction policies across all network stages. However, feature semantics change substantially as the network progresses through different resolutions. A graph topology that is optimal for low-resolution, high-semantic features may not be optimal for high-resolution, low-semantic features. This mismatch can lead to suboptimal performance and under-utilization of the expressive capacity of graph-based models. The limitations of current ViG-based models are:

- **Homogeneous architectures:** Existing ViGs adopt the same graph topology across all network stages. This uniformity overlooks the fact that feature semantics vary with resolution: low-resolution features encode high-level semantics, while high-resolution features capture fine-grained details. A single topology cannot simultaneously optimize for both, leading to suboptimal representations.

- **Static graph construction rules:** Existing ViGs rely on manually designed and fixed graph construction rules, such as k-nearest neighbor or radius-based graphs [20]. These static rules cannot adapt to the evolving properties of features as the network deepens. As a result, the expressive capacity of graph-based models is underutilized, limiting their ability to fully exploit stage-specific feature structures.

To overcome these limitations, we propose a new class of heterogeneous architectures where the graph topology adapts to the changing properties of features at different network stages. To automatically design these models, we introduce *SearchViG*, a Graph Construction Search (GCS) framework guided by a zero-shot, theoretically grounded proxy, namely, the spectral gap of the graph's adjacency matrix, which measures expansion properties that have provable links to information flow [21, 22]. *SearchViG* automatically discovers the optimal graph construction module for each stage, producing heterogeneous topologies that are tailored to the evolving feature representations. Unlike prior ViGs, which are limited by fixed and homogeneous designs, *SearchViG* optimizes the number of neighbors, hop distance, and construction method per resolution, and can seamlessly incorporate new graph construction techniques into its search space. We summarize our contributions as follows:

- We propose **SearchViG**, a new framework that performs a zero-shot **Graph Construction Search (GCS)** to automatically discover optimal Vision GNN topologies without any training.

- We propose our spectral search to discover **heterogeneous** architectures that adapt their graph construction rules from static and homogeneous to dynamic, heterogeneous, and content-aware based on feature resolution.

- **SearchViG** establishes a new state-of-the-art Pareto frontier, outperforming existing Vision GNNs in both accuracy and efficiency as shown in Figure 1.

2 Related Work

2.1 Deep Learning Architectures

Convolutional Neural Networks (CNNs) have been the cornerstone of modern vision systems since AlexNet [8], with subsequent architectures like ResNet [1] and DenseNet [23] pushing accuracy via deeper or more densely connected networks. CNNs exploit local receptive fields and weight sharing to learn hierarchical feature representations efficiently. However, their grid-based convolutional structure limits the modeling of long-range or irregular dependencies. Extensions such as non-local neural networks [24] have been proposed to augment CNNs with global processing, thus underscoring the need to go beyond purely local operations.

Vision Transformers (ViTs) [2] introduce a fundamentally different architecture, processing images as a sequence of patch tokens with global self-attention. By design, ViTs can capture long-range interactions from early layers, addressing CNNs' limitation in long-distance modeling. Hierarchical variants like Swin Transformer [25] further improve efficiency by restricting self-attention to non-overlapping local windows and alternating window shifts to enable cross-region interaction. Pure MLP-based models (e.g., the MLP-Mixer [10]) have also been explored which mixes spatial information via all-MLP layers. But, these approaches come with trade-offs: ViTs and Mixers lack the inductive bias of locality and require very large training data and computations, and even with self-attention, they operate on a fixed grid of patches with a mostly uniform (fully-connected or windowed) attention pattern. This uniform structure may not optimally model irregular scene relationships; for instance, ViTs attend globally but do not explicitly distinguish which connections are useful. These limitations have inspired interest in more adaptive connectivity mechanisms, which paves the way for graph-based approaches.

2.2 Vision GNNs

ViGs have emerged in computer vision as a new paradigm to represent images beyond the rigid grid or sequence structures of CNNs and ViTs. In a ViG, image patches serve as nodes, and their relationships are encoded as edges, which enables flexible context modeling and long-range interactions [26].

A key research question for ViGs is how to construct the underlying graph of patches. The original ViG [13] employs a simple graph construction rule across all stages: a k-nearest neighbor graph in which all patch nodes share the same type of connection [13]. Subsequent work has attempted to improve graph construction efficiency and adaptivity. For example, MobileViG [20] introduces a static graph construction mechanism to reduce computational overhead for mobile vision applications. In parallel, other methods proposed clustered [17], windowed [19], and dynamic axial processing [16] to enable adaptivity while limiting computational cost. Despite improved efficiency, these extensions still operate on a single graph construction rule across all stages to propagate information.

Another line of research addressed a limitation of the above pairwise graph approaches: standard graphs encode only binary relationships between nodes. Using only pairwise edges can lead to an explosion of connections and fails to model higher-order dependencies among image regions. To overcome this issue, Vision Hypergraph Neural Network (ViHGNN) [15] represents images with a hypergraph structure. In a hypergraph, a single hyperedge can connect multiple patches simultaneously, thus enabling the modeling of group-wise relationships beyond simple pairs.

2.3 Neural Architecture Search

Neural Architecture Search (NAS) has emerged as a powerful approach to automate the design of high-performance deep networks. Multi-shot NAS methods explore architectures by training numerous candidates but suffer from high computational costs. In contrast one-shot NAS mitigates this by sharing weights among sub-networks through a supernet, achieving faster yet still training-dependent searches [27]. Recently, zero-shot NAS has gained attention for completely removing the training stage by ranking candidate architectures using computation-efficient proxies [28–34]. The effectiveness of zero-shot NAS depends on the quality of these proxies. Early works assess network expressivity via the number of linear regions [35, 36], while others adopt gradient-based metrics such as SynFlow [37], SNIP [38], and GraSP [39], derived from Taylor expansions of neural networks. Additional proxies like Zen-score [40] and Jacobian covariance [41] further capture architectural complexity. However, studies show that simple heuristics such as the number of parameters can outperform existing proxies [42], underscoring the need for more robust and theoretically grounded

zero-shot NAS approaches. Furthermore, current NAS based approaches focus on optimizing channel width, network depth, or kernel size but there are no NAS based methods to determine graph construction for Vision GNNs.

2.4 Need for Heterogeneous Graph Topologies

Despite recent progress, Vision GNNs, including ViG [13] and its extensions, still rely on *homogeneous graph construction rules*, where all node connections are defined by a single relational rule. However, image regions naturally exhibit diverse relationships: spatial adjacency, appearance similarity, and semantic affinity, to name a few. Imposing a uniform connectivity pattern, regardless of the underlying data's unique structure, creates bottlenecks that hinder the flow of information and limit the model's representational power. This aligns with a broader principle where prior work has shown the benefits of relating data characteristics to the design of networks [43, 44].

To address this gap, we introduce **SearchViG**, a framework that automatically discovers stage-specific, heterogeneous graph topologies rather than relying on a fixed rule. By searching over diverse construction strategies (e.g., local spatial neighbors, feature-driven similarity, logarithmic connections, etc.), SearchViG learns optimal graph construction tailored to evolving feature semantics. The resulting heterogeneous architectures capture complementary interaction types among patches, which enables more discriminative visual representations. Unlike prior models that enforce homogeneous connectivity patterns, SearchViG adapts the graph structure to the data and network depth, thus overcoming the limitations of homogeneous graphs and establishing a stronger foundation for graph-based vision understanding.

3 Methodology

3.1 Preliminaries

Our work is grounded in spectral graph theory, which studies the properties of a graph through the eigenvalues and eigenvectors of its associated matrices. This section introduces the concepts that motivate our use of a spectral gap as a proxy for discovering optimal graph topologies in ViGs.

Graph and Matrix Representations. A graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of $N = |\mathcal{V}|$ nodes and $\mathcal{E}$ is the set of edges. The structure of the graph is captured by its Adjacency Matrix $A \in \{0,1\}^{N \times N}$. We also define the Degree Matrix D as a diagonal matrix where $D_{ii} = \sum_j A_{ij}$. From these, the Normalized Laplacian is defined as $L_{\text{norm}} = I - D^{-1/2} A D^{-1/2}$. The spectra of both A and L_{norm} reveal insights into the graph's structure.

The Spectral Gap as a Measure of Connectivity. Spectral graph theory offers two primary ways to measure a graph's expansion properties via its spectral gap:

- The **Laplacian Spectral Gap** (μ_1) is the second-smallest eigenvalue of the Normalized Laplacian (L_{norm}). Also known as the *algebraic connectivity*, it is a measure of connectivity; a graph is connected if and only if $\mu_1 > 0$.
- The **Adjacency Spectral Gap** ($\lambda_1 - \lambda_2$) is the difference between the first and second largest eigenvalues of the Adjacency Matrix (A). This gap serves as a strong indicator of a graph's expansion properties.

Ramanujan Graphs as an Optimal Spectral Proxy. Our goal is to find graph topologies that are optimal for information flow, which leads to the concept of Expander Graphs [22]: a family of graphs that are both sparse yet highly connected. The properties of these graphs are deeply connected to the spectrum of their Adjacency Matrix, A. For a graph with adjacency eigenvalues sorted as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$, the Adjacency Spectral Gap, $\lambda_1 - \lambda_2$, is an indicator of its expansion quality.

The theoretical ideal for expander graphs is captured by the properties of Ramanujan Graphs [21, 45]. The Alon-Boppana bound [45] establishes a limit on the spectral expansion of any d-regular graph, stating that its second-largest eigenvalue must satisfy $\lambda_2 \geq 2\sqrt{d-1} - o(1)$. Ramanujan graphs are those that nearly meet this fundamental limit, being formally defined as d-regular graphs where all non-trivial eigenvalues λ_i are bounded by $|\lambda_i| \leq 2\sqrt{d-1}$.

These graphs are, in a spectral sense, the best possible expanders. Therefore, the adjacency spectral gap serves as a direct, zero-shot proxy to guide our search for optimal Vision GNN topologies. The graphs found are optimal with respect to the search space since a large spectral gap is provably linked to superior information flow [22] and because it mitigates the problems of oversmoothing [46] and oversquashing [47], as shown in Appendix A.

3.2 Spectral Analysis of Graph Structure

The choice of a graph construction rule is a central, yet underexplored, challenge in Vision GNN design. An effective rule must generate a topology that is conducive to stable and deep information propagation. The spectral properties of a graph serve as a powerful, training-free (zero-shot) indicator of this behavior. This section provides a detailed analysis of how the spectral gap is linked to the mitigation of common GNN failure modes, thus justifying its use as a strong objective for our Graph Construction Search. Detailed analysis can be found in our Appendix Sections A and B.

Resistance to Oversmoothing. A primary challenge in deep GNNs is oversmoothing [48], where repeated message passing causes node features to converge to a uniform, noninformative state. This phenomenon is governed by the convergence properties of the graph's propagation matrix, which is often a normalized version of the adjacency matrix (e.g., $P = D^{-1/2}AD^{-1/2}$). The rate of convergence to an over-smoothed state is controlled by the second largest eigenvalue of this propagation matrix. For regular graphs, this eigenvalue is directly proportional to λ_2, the second largest eigenvalue of the adjacency matrix A. A smaller λ_2 slows the convergence, thus preserving feature diversity for more layers. Since the adjacency spectral gap for a regular graph is $\lambda_1 - \lambda_2 = d - \lambda_2$, maximizing this gap is equivalent to minimizing λ_2. Therefore, our search for a large adjacency spectral gap favors graph topologies that are more resistant to oversmoothing.

Prevention of Oversquashing. The second critical challenge is oversquashing [47], a structural issue where information from a large receptive field is bottlenecked when passing through the graph. This is fundamentally a problem of graph expansion. As established in the preliminaries, the Adjacency Spectral Gap is a direct, theoretical measure of a graph's expansion properties. A graph with a large spectral gap is a good expander. By definition, an expander graph has no bottlenecks, ensuring that information and gradients can flow efficiently between any two regions of the graph. Therefore, maximizing the adjacency spectral gap is a principled method for discovering topologies that are structurally guaranteed to prevent oversquashing.

A Principled Zero-Shot Proxy. The analysis above demonstrates that the spectral gap is not merely a descriptive statistic, but is linked to the dynamic properties essential for effective GNNs. Because the spectral gap ($\lambda_1 - \lambda_2$) can be calculated at initialization without any training, it serves as an ideal **zero-shot proxy** for Vision GNN trainability. It allows our GCS to efficiently evaluate a vast design space and identify architectures that are predisposed to stable and powerful learning, making the discovery of well performing heterogeneous topologies computationally tractable. To our knowledge, SearchViG is the first framework to use the spectral gap as a zero-shot proxy for discovering optimal, stage-specific graph construction rules in Vision GNNs.

3.3 Graph Construction Search

We introduce a zero-shot **Graph Construction Search (GCS)** framework designed to identify the best graph construction policy for each stage of the SearchViG architecture. Our search is guided by a single, theoretically-grounded objective, which helps us find the stage-specific graph topology, $\mathcal{G}_s^*$, that maximizes the adjacency spectral gap. Formally, for each stage s in the architecture, SearchViG solves the following optimization problem:

$$\mathcal{G}_s^* = \arg\max_{\mathcal{G} \in \mathcal{C}_s} \ (\lambda_1(A(\mathcal{G})) - \lambda_2(A(\mathcal{G}))) \tag{1}$$

where $\mathcal{C}_s$ is the set of candidate graph construction methods for that stage, $A(\mathcal{G})$ is the adjacency matrix of graph G, and λ_1, λ_2 are the largest eigenvalues. The entire search process, detailed in Algorithm 1, is performed *without any model training*, making it computationally efficient. When we benchmark the graph construction cost compared to the training cost, we find it is less than 0.5% of the training cost in terms of GPU hours. A formal complexity analysis and detailed cost breakdown are provided in Appendix D.

Algorithm 1 Graph Construction Search (GCS)

Input: Set of stages (resolutions) $\mathcal{S}$, Set of candidate methods $\mathcal{M}$, Data sample D_{sample}
Output: Optimal heterogeneous ViG configuration $\mathcal{G}^*$ (the set of the stage-specific optimal rules $\mathcal{G}_s^*$)
$\mathcal{G}^* \leftarrow$ new empty map ▷ Initialize the final configuration map
for all stage $s \in \mathcal{S}$ **do**
  $best\_spectral\_gap \leftarrow -1$
  $\mathcal{G}_s^* \leftarrow$ None
  for all method $M_i \in \mathcal{M}$ **do**
    for all hyperparameter k of M_i **do**
      $spectral\_gaps \leftarrow []$
      for all $image \in D_{sample}$ **do**
        $A \leftarrow \text{ConstructAdjacencyMatrix}(image, s, M_i, k)$
        $(\lambda_1, \lambda_2) \leftarrow \text{LargestEigenvalues}(A)$
        $spectral\_gaps.\text{append}(\lambda_1 - \lambda_2)$
      end for
      $avg\_gap \leftarrow \text{mean}(spectral\_gaps)$
      if $avg\_gap > best\_spectral\_gap$ **then**
        $best\_spectral\_gap \leftarrow avg\_gap$
        $\mathcal{G}_s^* \leftarrow (M_i, k)$
      end if
    end for
  end for
  $\mathcal{G}^*[s] \leftarrow \mathcal{G}_s^*$ ▷ Add the best graph construction rule for stage s to the map
end for
return $\mathcal{G}^*$

Search Space. The search space for our GCS consists of a theoretically infinite search space of graph construction mechanisms as users can add additional graph construction mechanisms to the search process. For our experiments, we used a comprehensive set of static, dynamic, local, and global construction mechanisms. These include strided axial connections [20, 49, 50] and dynamic content-based methods (KNN [13], windowed KNN [19], Clustered-KNN [15, 17], dynamic axial connections [16], and similarity-thresholded [18]) methods. For each method, we also search over a set of hyperparameters, such as the hop distance or neighbor count K. This rich search space allows the GCS to explore a diverse range of possible topologies.

Search Process. The search is performed independently for each of the four stages of the SearchViG architecture, which correspond to feature resolutions of 56×56, 28×28, 14×14, and 7×7. For a given resolution, we iterate through every candidate graph construction method and hyperparameter in our search space. For each candidate, we generate graph topologies for a representative subset of 1000 ImageNet images. We then compute the Adjacency Spectral Gap for the largest connected component of each generated graph and average the results over all images. The policy that yields the highest average spectral gap is selected as the optimal constructor for that specific stage. This zero-shot evaluation allows us to efficiently find a powerful heterogeneous architecture.

3.4 Heterogeneous Graph Convolution

The Graph Construction Search (GCS) detailed in Section 3.3 yields a novel architectural paradigm for Vision GNNs. Instead of relying on a single, fixed graph construction rule, our search discovers an optimal, stage-specific rule for each resolution. This finding motivates the design of our architectural building block, the **Heterogeneous Graph Convolution (HGC) Block**.

An HGC Block is a module whose internal graph construction mechanism is not fixed, but is instead determined by the output of the GCS. For a stage s in the network, let the optimal graph construction rule discovered by our search be $\mathcal{G}_s^*$ with its corresponding hyperparameter K_s^*. The HGC block processes an input feature map $X^{(s)}$ through two main sub-modules: a Heterogeneous Grapher and a Feed-Forward Network (FFN), with residual connections. The operation is defined as follows:

$$X'^{(s)} = X^{(s)} + \text{Grapher}_s(X^{(s)}) \tag{2}$$

$$X^{(s+1)} = X'^{(s)} + \text{FFN}(X'^{(s)}) \tag{3}$$

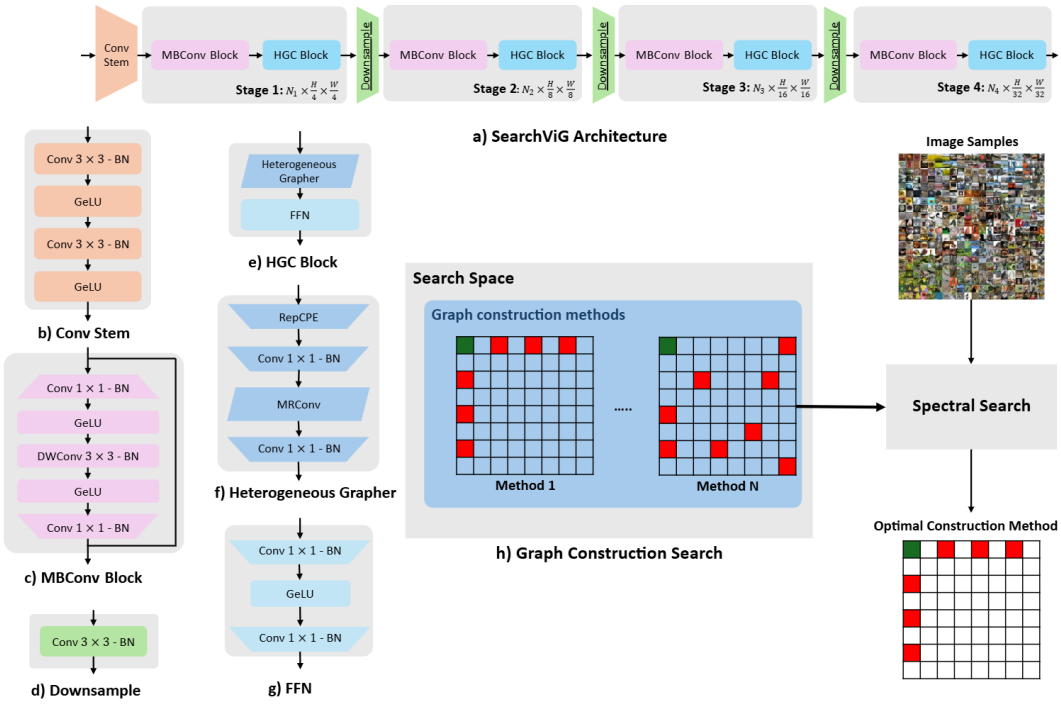


Figure 2: SearchViG Architecture. **(a)** The overall 4-stage hierarchical architecture. Each stage combines MBConv blocks for local processing with our HGC Block for global information mixing. **(b)** The initial convolutional stem for feature extraction. **(c)** The MBConv Block used for local feature processing. This block consists of pointwise convolutions, GeLU activation, and depthwise convolution (DWConv). **(d)** The downsample block used between stages to reduce spatial resolution. **(e)** The Heterogeneous Graph Convolution (HGC) Block. **(f)** The Heterogeneous Grapher module, which instantiates the optimal graph constructor for its stage. This block consists of reparameterizable conditional positional encoding (RepCPE), pointwise convolutions, and max-relative graph convolution (MRConv). **(g)** The Feed-Forward Network (FFN). **(h)** Our Graph Construction Search (GCS), which selects the best graph construction policy and hyperparameters from a set of candidates for each HGC Block.

The key to this block is the **Heterogeneous Grapher** module, Grapher_s, which is parameterized by the search result for its specific stage:

$$\text{Grapher}_s(X) = \phi_{out}(\text{GraphConv}_{\mathcal{G}_s^*, K_s^*}(\phi_{in}(X))) \tag{4}$$

where ϕ_{in} and ϕ_{out} represent feature-processing MLPs, and $\text{GraphConv}_{\mathcal{G}_s^*, K_s^*}$ is the graph convolution layer that utilizes the specific graph topology found by our search. This formulation allows each HGC block in the network to employ a different, specialized graph convolution policy.

3.5 SearchViG Architecture

The SearchViG architecture, shown in Figure 2(a), is a hierarchical design composed of a convolutional stem and four subsequent stages that progressively reduce spatial resolution while increasing channel width. The Stem (Figure 2(b)) uses two 3×3 convolutions, each with a stride of 2, to downsample the input and extract initial low-level features. Each of the four Stages contains a sequence of blocks for both local and global feature processing. For local processing, we use MBConv blocks (Figure 2(c)) based on the inverted residual structure [51]. MBConv blocks capture local patterns using a combination of depthwise [52] and pointwise convolutions, along with GeLU [53] activation.

For global processing, we introduce our novel **Heterogeneous Graph Convolution (HGC) Block** (Figure 2(e)). Crucially, the internal graph construction mechanism of the HGC block is tailored to each stage, having been discovered by our Graph Construction Search. This allows the network to apply the most effective graph topology (e.g., static, long-range, dynamic, clustered, etc.) at each level of feature abstraction. The HGC block is composed of the Heterogeneous Grapher and an FFN. As detailed in Figure 2(f), the Heterogeneous Grapher first enriches features with positional information

Table 1: Classification results on ImageNet-1k for SearchViG and other state-of-the-art models. Bold entries indicate results obtained for **SearchViG** proposed in this paper. The Top-1 accuracy results for SearchViG models show the mean ± standard deviation averaged over three experiments with different random seeds.

Model	Type	Resolution	Params (M)	GMACs	Epochs	Accuracy (%)
ResNet18 [1]	CNN	224×224	11.7	1.8	300	69.7
ResNet50 [1]	CNN	224×224	25.6	4.1	300	80.4
ConvNext-T [58]	CNN	224×224	28.6	7.4	300	82.7
EfficientFormer-L1 [59]	CNN-ViT	224×224	12.3	1.3	300	79.2
EfficientFormer-L3 [59]	CNN-ViT	224×224	31.3	3.9	300	82.4
EfficientFormer-L7 [59]	CNN-ViT	224×224	82.1	10.2	300	83.3
LeViT-192 [60]	CNN-ViT	224×224	10.9	0.7	1000	80.0
LeViT-384 [60]	CNN-ViT	224×224	39.1	2.4	1000	82.6
PVT-Small [61]	ViT	224×224	24.5	3.8	300	79.8
PVT-Large [61]	ViT	224×224	61.4	9.8	300	81.7
Swin-T [25]	ViT	224×224	29.0	4.5	300	81.3
Swin-S [25]	ViT	224×224	50.0	8.7	300	83.0
PoolFormer-s12 [62]	Pool	224×224	12.0	2.0	300	77.2
PoolFormer-s24 [62]	Pool	224×224	21.0	3.6	300	80.3
PoolFormer-s36 [62]	Pool	224×224	31.0	5.2	300	81.4
ViHGNN-S [15]	GNN	224×224	23.2	5.6	300	81.5
ViHGNN-B [15]	GNN	224×224	88.1	19.4	300	82.9
PViHGNN-Ti [15]	GNN	224×224	12.3	2.3	300	78.9
PViHGNN-S [15]	GNN	224×224	28.5	6.3	300	82.5
PViHGNN-B [15]	GNN	224×224	94.4	18.1	300	83.9
ViG-S [13]	GNN	224×224	22.7	4.5	300	80.4
ViG-B [13]	GNN	224×224	86.8	17.7	300	82.3
PViG-Ti [13]	GNN	224×224	10.7	1.7	300	78.2
PViG-S [13]	GNN	224×224	27.3	4.6	300	82.1
PViG-B [13]	GNN	224×224	92.6	16.8	300	83.7
WiGNet-Ti [19]	GNN	256×256	10.8	2.1	300	78.8
WiGNet-S [19]	GNN	256×256	27.4	5.7	300	82.0
WiGNet-M [19]	GNN	256×256	49.7	11.2	300	83.0
MobileViG-S [20]	CNN-GNN	224×224	7.2	1.0	300	78.2
MobileViG-M [20]	CNN-GNN	224×224	14.0	1.5	300	80.6
MobileViG-B [20]	CNN-GNN	224×224	26.7	2.8	300	82.6
SearchViG-S (Ours)	**CNN-GNN**	224×224	**12.4**	**1.7**	**300**	**81.3 ± 0.2**
SearchViG-M (Ours)	**CNN-GNN**	224×224	**25.7**	**3.7**	**300**	**83.3 ± 0.1**
SearchViG-B (Ours)	**CNN-GNN**	224×224	**34.2**	**5.0**	**300**	**84.0 ± 0.2**

via a reparameterizable conditional positional encoding (CPE) layer [54, 55]. The features then pass through pointwise convolution with batch normalization [56], max-relative graph convolution [57] for message passing, and a final pointwise convolution. Between each stage, a Downsample block (Figure 2(d)) halves the resolution and expands the channel dimension.

4 Experimental Results

We compare SearchViG with competing CNN, ViT, and Vision GNN architectures on the tasks of image classification, object detection, instance segmentation, and semantic segmentation to demonstrate its superior performance. For additional results on the CIFAR [63], OrganSMNIST [64], DermaMNIST [64], DeepCrack [65], Crack500 [66], CityScapes [67], and COCO-Stuff [68] benchmarks, please refer to the Appendix Section C.

4.1 Image Classification

All SearchViG models are trained from scratch on the ImageNet-1K dataset [69] using PyTorch [70] and the Timm library [71]. We train for 300 epochs using the AdamW optimizer [72, 73], a learning rate of $2e^{-3}$ with a cosine annealing schedule, and a standard training resolution of 224×224.

As shown in Table 1, our SearchViG models set a new state-of-the-art for Vision GNNs, outperforming prior works across all model sizes. Our smallest model, SearchViG-S, achieves an impressive 81.3% Top-1 accuracy with only 12.4M parameters and 1.7 GMACs. This significantly surpasses other GNNs in its class like PViG-Ti (+3.1%) and PViHGNN-Ti (+2.4%) with equal or lower parameters and GMACs. Our mid-sized model, SearchViG-M, reaches 83.3% accuracy, outperforming WigNet-M [19] by 0.3% with 67% fewer GMACs and ViG-B [13] by 1.0% with 79% fewer GMACs. Lastly, our SearchViG-B model achieves a remarkable 84.0% accuracy with only 5.0 GMACs. This result surpasses larger and more computationally expensive competitors from all paradigms, including Swin-S [25] (83.0%, 8.7 GMACs), EfficientFormer-L7 [59] (83.3%, 10.2 GMACs), and even the 94.4M parameter PViHGNN-B [15] (83.9%, 18.1 GMACs), establishing the superiority of SearchViG.

4.2 Object Detection and Instance Segmentation

To evaluate generalization to downstream tasks, we use SearchViG as a backbone in the Mask-RCNN framework [74] for object detection and instance segmentation tasks on the MS COCO 2017 dataset [75]. The model is initialized with ImageNet-1K pretrained weights from 300 epochs of training. We use the AdamW [72, 73] optimizer with an initial learning rate of $2e^{-4}$ and train the model for 12 epochs with a standard resolution (1333×800) following prior work [16, 20, 59, 76, 77].

The results in Table 2 demonstrate SearchViG's strong performance. SearchViG-S achieves 43.5 AP^{box} and 39.9 AP^{mask}, significantly outperforming other models like PoolFormer-S12 [62] by +6.2 AP^{box} and +5.3 AP^{mask}. Our SearchViG-B model achieves 46.5 AP^{box} and 42.4 AP^{mask}, surpassing similarly sized models like EfficientFormer-L3 [59] by +5.1 AP^{box} and +4.3 AP^{mask}.

Table 2: Object detection, instance segmentation, and semantic segmentation results of SearchViG and other backbones on MS COCO 2017 and ADE20K. Bold entries indicate results obtained using SearchViG proposed in this paper. A (-) denotes a model that did not report these results.

Backbone	Params (M)	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	$mIoU$
EfficientFormer-L1 [59]	12.3	37.9	60.3	41.0	35.4	57.3	37.3	38.9
PoolFormer-S12 [62]	12.0	37.3	59.0	40.1	34.6	55.8	36.9	37.2
FastViT-SA12 [78]	10.9	38.9	60.5	42.2	35.9	57.6	38.1	38.0
MobileViG-M [20]	14.0	41.3	62.8	45.1	38.1	60.1	40.8	-
PoolFormer-S24 [62]	21.0	40.1	62.2	43.4	37.0	59.1	39.6	40.3
SearchViG-S (Ours)	**12.4**	**43.5**	**65.6**	**47.4**	**39.9**	**62.3**	**43.5**	**43.4**
EfficientFormer-L3 [59]	31.3	41.4	63.9	44.7	38.1	61.0	40.4	43.5
EfficientFormer-L7 [59]	82.1	42.6	65.1	46.1	39.0	62.2	41.7	45.1
FastViT-SA36 [78]	30.4	43.8	65.1	47.9	39.4	62.0	42.3	42.9
Pyramid ViG-S [13]	27.3	42.6	65.2	46.0	39.4	62.4	41.6	-
Pyramid ViHGNN-S [15]	28.5	43.1	66.0	46.5	39.6	63.0	42.3	-
PVT-Large [61]	61.4	42.9	65.0	46.6	39.5	61.9	42.5	42.1
MobileViG-B [20]	26.7	42.0	64.3	46.0	38.9	61.4	41.6	-
SearchViG-B (Ours)	**34.2**	**46.5**	**68.7**	**51.5**	**42.4**	**65.9**	**45.6**	**47.9**

4.3 Semantic Segmentation

We further validate our models on semantic segmentation using the ADE20K dataset [79]. Following the methodologies of prior work [59, 62], we use SearchViG as a backbone for the Semantic FPN [80] segmentation decoder. The SearchViG backbones are initialized with their ImageNet-1K pre-trained weights and trained for 40K iterations. We use the AdamW [73] optimizer with a learning rate of 2×10^{-4} and a polynomial decay schedule with a power of 0.9. All models are trained with a resolution of 512×512 following prior work [16, 59, 77].

The results in Table 2 confirm the generalization of our SearchViG backbone to dense prediction tasks. Our SearchViG-S achieves 43.4 mIoU, outperforming other efficient models like EfficientFormer-L1 [59] by a margin of +4.5 mIoU with a comparable parameter count. Furthermore, our SearchViG-B model obtains a state-of-the-art 47.9 mIoU. This surpasses larger models like EfficientFormer-L7 [59] (+2.8 mIoU) while using less than half the parameters. The performance of SearchViG across these tasks shows its ability to generalize to downstream tasks.

4.4 Ablation Studies

We conduct ablation studies on the ImageNet-1K [69] dataset to validate our design choices. We use the SearchViG-S architecture as the baseline for these experiments as shown in Table 3.

Effectiveness of the Spectral Gap as a Search Metric. To verify that our spectral-aware search is a meaningful proxy for final model performance, we compare it against an architecture found by a random search baseline. We construct a RandomViG-S model by randomly selecting a graph construction policy and K-value for each stage from our search space. As shown in Table 3, the architecture discovered by our spectral-aware GCS significantly outperforms the randomly constructed one by 1.3% top-1 accuracy (81.3% vs. 80.0%), confirming that maximizing the spectral gap is an effective strategy for discovering high-performance topologies.

Impact of Heterogeneous Architecture. To verify that a heterogeneous graph topology is superior to a homogeneous one, we construct three homogeneous baselines by using a single graph policy (SVGA [20], KNN [13, 14], or Clustered KNN [15, 17]) across the stages. Table 3 shows that our heterogeneous SearchViG-S outperforms all homogeneous variants. Notably, it surpasses the strongest homogeneous baseline (Clustered KNN) by 0.8%, demonstrating that adapting the graph construction to the feature resolution is critical for achieving optimal performance.

Table 3: Ablation studies on SearchViG-S. We validate our spectral search metric and compare our discovered heterogeneous architecture against homogeneous and alternative search baselines.

Configuration	Parameters (M)	Top-1 Accuracy (%)
Homogeneous (SVGA Only)	12.4	80.4
Homogeneous (KNN Only)	12.4	80.4
Homogeneous (Clustered KNN Only)	12.4	80.5
RandomViG-S	12.4	80.0
SearchViG-S (Heterogeneous, Ours)	**12.4**	**81.3**

5 Conclusion

In this work, we have introduced SearchViG, a novel framework that challenges the prevailing paradigm of homogeneous graph construction in Vision GNNs. We proposed a zero-shot Graph Construction Search (GCS) guided by a theoretically-grounded spectral proxy to automatically discover optimal, stage-specific graph topologies. Our proposed heterogeneous architecture of SearchViG intelligently adapts its connectivity based on the resolution of the input. The resulting SearchViG models demonstrate the power of this heterogeneous design method, establishing a new state-of-the-art Pareto frontier for Vision GNNs on ImageNet-1K classification and downstream tasks. Most notably, our work provides the first principled method for automating the discovery of heterogeneous graph topologies, opening a new and promising direction for the future of Vision GNN design. Future work could explore expanding the search space with additional graph construction mechanisms or proposing new search policies for Vision GNN graph construction.

6 Acknowledgements

This work is supported in part by the NSF grant CCF-2531882 and a UT Cockrell School of Engineering Doctoral Fellowship.

References

[1] 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. 1, 3, 8, 22, 23, 24

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 3

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. 1

[4] 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. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002. 1

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012. 1, 3

[9] 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. 1

[10] 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. 1, 3

[11] Jiachen Li, Ali Hassani, Steven Walton, and Humphrey Shi. Convmlp: Hierarchical convolutional mlps for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6307–6316, 2023. 1

[12] 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. 1

[13] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *Advances in Neural Information Processing Systems*, 35:8291–8303, 2022. 1, 3, 4, 6, 8, 9, 10, 27

[14] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967. 1, 10

[15] Yan Han, Peihao Wang, Souvik Kundu, Ying Ding, and Zhangyang Wang. Vision hgnn: An image is more than a graph of nodes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19878–19888, 2023. 1, 3, 6, 8, 9, 10

[16] Mustafa Munir, William Avery, Md Mostafijur Rahman, and Radu Marculescu. Greedyvig: Dynamic axial graph construction for efficient vision gnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6118–6127, June 2024. 1, 3, 6, 9

[17] Dhruv Parikh, Jacob Fein-Ashley, Tian Ye, Rajgopal Kannan, and Viktor Prasanna. Clustervig: Efficient globally aware vision gnns via image partitioning. *arXiv preprint arXiv:2501.10640*, 2025. 3, 6, 10

[18] Ismael Elsharkawi, Hossam Sharara, and Ahmed Rafea. Svig: A similarity-thresholded approach for vision graph neural networks. *IEEE Access*, 2025. 6

[19] Gabriele Spadaro, Marco Grangetto, Attilio Fiandrotti, Enzo Tartaglione, and Jhony H Giraldo. Wignet: Windowed vision graph neural network. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 859–868. IEEE, 2025. 1, 3, 6, 8, 9

[20] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2211–2219, 2023. 2, 3, 6, 8, 9, 10, 17, 22, 23, 24, 27

[21] Duc NM Hoang, Shiwei Liu, Radu Marculescu, and Zhangyang Wang. Revisiting pruning at initialization through the lens of ramanujan graph. In *International Conference on Learning Representations*, 2023. 2, 4

[22] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006. 2, 4, 5

[23] 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. 3

[24] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[25] 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. 3, 8, 9, 22

[26] Chaoqi Chen, Yushuang Wu, Qiyuan Dai, Hong-Yu Zhou, Mutian Xu, Sibei Yang, Xiaoguang Han, and Yizhou Yu. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3

[27] Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, qiang liu, and Vikas Chandra. Nasvit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022. 3

[28] Meng-Ting Wu, Hung-I Lin, and Chun-Wei Tsai. A training-free genetic neural architecture search. In *Proceedings of the 2021 ACM International Conference on Intelligent Computing and Its Emerging Applications*, pages 65–70, 2021. 3

[29] Yao Shu, Zhongxiang Dai, Zhaoxuan Wu, and Bryan Kian Hsiang Low. Unifying and boosting gradient-based training-free neural architecture search. *Advances in Neural Information Processing Systems*, 35:33001–33015, 2022.

[30] Mojan Javaheripi, Shital Shah, Subhabrata Mukherjee, Tomasz Lukasz Religa, Caio Cesar Teodoro Mendes, Gustavo Henrique de Rosa, Sebastien Bubeck, Farinaz Koushanfar, and Debadeepta Dey. Litetransformersearch: Training-free on-device search for efficient autoregressive language models. In *First Conference on Automated Machine Learning (Late-Breaking Workshop)*, 2022.

[31] Qinqin Zhou, Kekai Sheng, Xiawu Zheng, Ke Li, Xing Sun, Yonghong Tian, Jie Chen, and Rongrong Ji. Training-free transformer architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10894–10903, 2022.

[32] Thorir Mar Ingolfsson, Mark Vero, Xiaying Wang, Lorenzo Lamberti, Luca Benini, and Matteo Spallanzani. Reducing neural architecture search spaces with training-free statistics and computational graph clustering. In *Proceedings of the 19th ACM International Conference on Computing Frontiers*, pages 213–214, 2022.

[33] Linh Tam Tran and Sung-Ho Bae. Training-free hardware-aware neural architecture search with reinforcement learning. *Journal of Broadcast Engineering*, 26(7):855–861, 2021.

[34] Tu Do and Ngoc Hoang Luong. Training-free multi-objective evolutionary neural architecture search via neural tangent kernel and number of linear regions. In *International Conference on Neural Information Processing*, pages 335–347. Springer, 2021. 3

[35] Joe Mellor, Jack Turner, Amos Storkey, and Elliot J Crowley. Neural architecture search without training. In *International conference on machine learning*, pages 7588–7598. PMLR, 2021. 3

[36] Kartikeya Bhardwaj, James Ward, Caleb Tung, Dibakar Gope, Lingchuan Meng, Igor Fedorov, Alex Chalfin, Paul Whatmough, and Danny Loh. Restructurable activation networks. *arXiv preprint arXiv:2208.08562*, 2022. 3

[37] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems*, 33:6377–6389, 2020. 3

[38] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018. 3

[39] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020. 3

[40] Ming Lin, Pichao Wang, Zhenhong Sun, Hesen Chen, Xiuyu Sun, Qi Qian, Hao Li, and Rong Jin. Zen-nas: A zero-shot nas for high-performance image recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 347–356, 2021. 3, 22, 23

[41] Vasco Lopes, Saeid Alirezazadeh, and Luís A Alexandre. Epe-nas: Efficient performance estimation without training for neural architecture search. In *International conference on artificial neural networks*, pages 552–563. Springer, 2021. 3

[42] Xuefei Ning, Changcheng Tang, Wenshuo Li, Zixuan Zhou, Shuang Liang, Huazhong Yang, and Yu Wang. Evaluating efficient performance estimators of neural architectures. *Advances in Neural Information Processing Systems*, 34:12265–12277, 2021. 3

[43] Mustafa Munir, Guihong Li, Md Mostafijur Rahman, Alex Zhang, and Radu Marculescu. From data to design: Leveraging frequency statistics for efficient neural network architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3208–3218, 2025. 4, 21

[44] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022. 4, 21

[45] Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. 4

[46] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *Proceedings of the AAAI conference on artificial intelligence*, 32 (1), 2018. 5, 17

[47] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021. 5, 18

[48] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023. 5, 17

[49] William Avery, Mustafa Munir, and Radu Marculescu. Scaling graph convolutions for mobile vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5857–5865, June 2024. 6

[50] Mustafa Munir, Alex Zhang, and Radu Marculescu. Multi-scale high-resolution logarithmic grapher module for efficient vision gnns. In *The Third Learning on Graphs Conference*, 2024. 6

[51] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 7

[52] 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. 7

[53] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 7

[54] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 8

[55] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Mobileone: An improved one millisecond mobile backbone. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7907–7917, 2023. 8

[56] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. pmlr, 2015. 8

[57] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9267–9276, 2019. 8

[58] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 8, 22, 23, 24

[59] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022. 8, 9, 27

[60] 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. 8

[61] 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. 8, 9

[62] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 8, 9

[63] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 8, 22

[64] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023. 8, 22, 23

[65] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE transactions on image processing*, 28(3): 1498–1512, 2018. 8, 22, 24

[66] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019. 8, 22, 24

[67] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 8, 22, 24

[68] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 8, 22, 24

[69] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. 8, 10, 24

[70] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 8, 22, 23

[71] Ross Wightman. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models, 2019. 8, 22, 23

[72] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8, 9

[73] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 8, 9, 22, 23, 24, 27

[74] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 9

[75] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 9

[76] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022. 9

[77] Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023. 9

[78] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 9

[79] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 9

[80] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 9

[81] Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In *Problems in analysis*, pages 195–200. Princeton University Press, 1969. 18

[82] Shmuel Friedland and Reinhard Nabben. On cheeger-type inequalities for weighted graphs. *Journal of Graph Theory*, 41(1):1–17, 2002. 18

[83] Tsz Chiu Kwok, Lap Chi Lau, and Kam Chuen Tung. Cheeger inequalities for vertex expansion and reweighted eigenvalues. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 366–377. IEEE, 2022. 18

[84] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Cheeger inequalities for directed graphs and hypergraphs using reweighted eigenvalues. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1834–1847, 2023. 18

[85] Tongtong Yuan, Weihong Deng, Jian Tang, Yinan Tang, and Binghui Chen. Signal-to-noise ratio: A robust distance metric for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4815–4824, 2019. 21

[86] G.B. Giannakis and M.K. Tsatsanis. Signal detection and classification using matched filtering and higher order statistics. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(7):1284–1296, 1990. doi: 10.1109/29.57557. 21

[87] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021. 22, 23

[88] Y. Yang, H. Huang, L. Su, and S. Zhang. SVGACrack: Sparse Vision Graph Attention Segmentation Networks Enabling Precise Pavement Crack Detection. In *2024 6th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP)*, pages 349–354, 2024. 24

[89] R. Ran, X. Xu, S. Qiu, X. Cui, and F. Wu. Crack-segnet: Surface crack detection in complex background using encoder-decoder architecture. In *Proceedings of the 2021 4th International Conference on Sensors, Signal and Image Processing*, pages 15–22, 2021. 24

[90] W. Wang and C. Su. Automatic concrete crack segmentation model based on transformer. *Automation in Construction*, 139:104275, 2022. 24

[91] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 24

[92] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 24

[93] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 24, 25

[94] Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural image statistics: A probabilistic approach to early computational vision.*, volume 39. Springer Science & Business Media, 2009. 26

[95] Daniel Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. *Advances in Neural Information Processing Systems*, 6, 1993.

[96] Daniel L Ruderman. Origins of scaling in natural images. *Vision research*, 37(23):3385–3398, 1997. 26

[97] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020. 27

[98] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 27

[99] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 27

[100] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 27

[101] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *Proceedings of the AAAI conference on artificial intelligence*, 34(07):13001–13008, 2020. 27

A Theoretical Analysis

A.1 Theoretical Justification for Spectral-Aware Graph Search

Our work, SearchViG, is motivated by the principle that a neural network's graph topology is critical for effective learning. We propose a Graph Construction Search (GCS) that, for the first time, creates a *heterogeneous* Vision GNN by selecting the optimal graph construction module at each resolution to produce topologies with superior spectral properties.

In this section, we provide the theoretical justification for our approach. We show that optimizing for the spectral gap leads to a graph structure with more stable gradient flow, addressing the key challenges of oversmoothing, oversquashing, and vanishing gradients in Vision GNNs.

A.2 Spectral Preliminaries for a GNN Layer

A single GNN layer updates a node feature matrix $H \in \mathbb{R}^{N\times C}$ by applying a graph-aware transformation:

$$H' = \sigma\left(PHW\right) \tag{5}$$

where W is a learnable weight matrix and σ is a non-linearity. The stability and effectiveness of this layer are intrinsically linked to the spectral properties of the propagation matrix $P \in \mathbb{R}^{N\times N}$, which mixes information across the graph's nodes.

The *propagation matrix* is the symmetrically normalized adjacency matrix, $P = D^{-1/2}AD^{-1/2}$, where D is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. The eigenvalues of P, denoted $1 = p_1 \geq p_2 \geq \cdots \geq p_N \geq -1$, dictate the layer's behavior as a graph filter. The properties of P are, in turn, governed by the spectrum of the Adjacency Matrix A.

Let the eigenvalues of the adjacency matrix A be sorted as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. Our search is guided by the *Adjacency Spectral Gap*, defined as $\lambda_1 - \lambda_2$. As established in our main paper, maximizing this gap is a means to find graphs with strong, Ramanujan-like expansion properties. The spectra of A and the propagation matrix P are closely related; for the simple case of a d-regular graph, the eigenvalues are related by $p_i = \lambda_i/d$. Therefore, a large adjacency spectral gap in A leads to a favorable spectral structure in P, which is critical for GNN stability. While our analysis focuses on the adjacency spectrum, we note its relation to the normalized Laplacian (L_{norm}), another key analytical tool. As defined in Section 3.1, the normalized Laplacian is $L_{\text{norm}} = I - D^{-1/2}AD^{-1/2}$. Given that the propagation matrix is $P = D^{-1/2}AD^{-1/2}$, this is equivalent to the equation $L_{\text{norm}} = I - P$, where I is the identity matrix.

Note on Graph Regularity. Exact results on the relation between adjacency and Laplacian spectral gaps (e.g., Ramanujan bounds) hold for d-regular graphs. In Vision GNNs, many graphs are irregular (e.g., similarity graphs), though some constructions like SVGA [20] with wrap-around connections are d-regular. For irregular graphs, the adjacency spectral gap is best viewed as a practical heuristic for expansion and stability, even though only the regular case provides formal guarantees.

A.3 The Impact of the Spectral Gap on Gradient Dynamics

We now provide a detailed analysis demonstrating how maximizing the spectral gap improves the stability of a Vision GNN.

A.3.1 Mitigating Oversmoothing

Oversmoothing [46, 48] describes the phenomenon where, as the number of GNN layers increases, the features of all nodes converge to a single, uninformative state, erasing discriminative information. This collapse of the feature space causes the problem of the vanishing gradients in deep GNNs.

Proposition. *A larger adjacency spectral gap corresponds to a graph structure that is more robust to oversmoothing [48].*

Proof. The oversmoothing [46, 48] process can be analyzed as governed by the propagation matrix $P = D^{-1/2}AD^{-1/2}$. Let $H_c = H - \bar{H}$ be the centered feature matrix, where $\bar{H}$ is the mean feature

vector. The convergence of these features to zero represents the loss of unique information. The rate of this convergence is controlled by p_2, the second-largest eigenvalue of P.

After a single GNN layer, the total magnitude of the centered features, as measured by the Frobenius norm, is bounded by the second-largest eigenvalue, p_2:

$$\|H_c'\|_F \leq p_2 \|H_c\|_F \tag{6}$$

A smaller value of p_2 signifies less information loss in a single step. In a deep GNN with L layers, this effect is compounded exponentially. The features after L layers are bounded by:

$$\|H_c^{(L)}\|_F \leq (p_2)^L \|H_c^{(0)}\|_F \tag{7}$$

As established in the preliminaries, for a d-regular graph, p_2 is directly proportional to the second-largest eigenvalue of the adjacency matrix, λ_2, via $p_2 = \lambda_2/d$. The adjacency spectral gap is $\lambda_1 - \lambda_2 = d - \lambda_2$. Therefore, maximizing the adjacency spectral gap is equivalent to minimizing λ_2, which in turn minimizes the convergence rate p_2.

By maximizing the adjacency spectral gap, our GCS discovers graph topologies with a smaller p_2. This slows the exponential decay of information, making the Vision GNN more resistant to oversmoothing. This is critical for deep models, as it helps prevent the network's end-to-end Jacobian matrix, $J = \frac{\partial H^{(L)}}{\partial H^{(0)}}$, from collapsing. For a linear GNN, this Jacobian is proportional to P^L (where L is the number of layers/depth in the GNN), so a smaller p_2 preserves its rank and mitigates vanishing gradients.

We note that minimizing p_2 is equivalent to maximizing the Laplacian spectral gap μ_1, since for the propagation matrix P, it holds that $p_2 = 1 - \mu_1$. Our search therefore uses the adjacency gap as a robust proxy to achieve these desirable spectral properties. ■

A.3.2 Preventing Oversquashing via Graph Expansion

Oversquashing [47] is a structural problem where the graph topology itself creates information bottlenecks, constricting the flow of messages and gradients between distant nodes.

Proposition. *Maximizing the spectral gap promotes graph structures that are robust to oversquashing.*

Proof. Oversquashing [47] is fundamentally a problem of poor graph expansion. The solution is to find topologies that are good expander graphs, which by definition lack the bottlenecks that cause information to be squashed. A high expansion constant ensures that there is a rich set of paths between any two regions of the graph. This structural property guarantees that gradients can propagate from the loss function back to any node in the computational graph without being "squashed" or attenuated by topological constrictions. This is critical for learning long-range dependencies within an image.

The foundational link between a graph's spectrum and its expansion properties is captured by Cheeger's Inequality [81]:

$$\frac{\mu_1}{2} \leq h(G) \leq \sqrt{2\mu_1} \tag{8}$$

where $h(G)$ is the Cheeger constant (a measure of bottlenecks) and μ_1 is the Laplacian spectral gap. This inequality provides a provable guarantee that maximizing μ_1 leads to better graph expansion. This result has since been generalized and extended to various domains relevant to modern graph learning, including weighted graphs [82], vertex expansion [83], and even hypergraphs [84].

While Cheeger's Inequality is defined using the Laplacian, maximizing the Laplacian spectral gap (μ_1) is spectrally related to maximizing the Adjacency Spectral Gap ($\lambda_1 - \lambda_2$). For a d-regular graph, where the largest eigenvalue is known to be $\lambda_1 = d$, the second-largest eigenvalue of the propagation matrix, p_2, is related to both spectra: $p_2 = \lambda_2/d$ and $p_2 = 1 - \mu_1$. This implies that maximizing the adjacency gap ($d - \lambda_2$) is equivalent to minimizing λ_2, which is equivalent to minimizing p_2 and maximizing μ_1.

Therefore, by maximizing the adjacency spectral gap, our GCS is optimizing for the same spectral properties that guarantee high expansion via Cheeger's Inequality. We choose the adjacency gap as our proxy not only for this strong theoretical connection, but also for its practical robustness, as it provides a meaningful measure even for **disconnected graphs** where μ_1 would be zero. ■

A.3.3 Detailed Gradient Flow and Spectral Conditioning

To provide a more rigorous basis for our claims, we now present a detailed mathematical analysis of the gradient signal during backpropagation. We demonstrate that maximizing the spectral gap, our core objective in SearchViG, is linked to optimizing the conditioning of the gradient propagation pathway, which is essential for stable and efficient learning in deep architectures.

Backpropagation as a Series of Vector-Jacobian Products. The backpropagation algorithm computes the gradient of the loss $\mathcal{L}$ with respect to the network's parameters by first finding the gradient with respect to the layer features. Let $\mathbf{g}^{(l)} = \frac{\partial \mathcal{L}}{\partial H^{(l)}}$ be the gradient vector (or its flattened representation) at layer l. The gradient is propagated backward from layer l to layer $l-1$ via the vector-Jacobian product:

$$\mathbf{g}^{(l-1)} = \mathbf{g}^{(l)} J^{(l-1)} \tag{9}$$

where $J^{(l-1)} = \frac{\partial H^{(l)}}{\partial H^{(l-1)}}$ is the Jacobian matrix of the layer l transformation. Propagating the gradient from the final layer L back to the input layer 0 involves a product of these Jacobians:

$$\mathbf{g}^{(0)} = \mathbf{g}^{(L)} \left(\prod_{l=L-1}^{0} J^{(l)} \right) \tag{10}$$

The stability of this process depends on the properties of this cumulative Jacobian product. The norm of the gradient vector is bounded at each step:

$$\|\mathbf{g}^{(l-1)}\|_2 \leq \|\mathbf{g}^{(l)}\|_2 \cdot \|J^{(l-1)}\|_2 \tag{11}$$

where $\|\cdot\|_2$ is the spectral norm, defined as the largest singular value of the matrix. If the spectral norms of the Jacobians are consistently greater than 1, gradients will explode; if they are consistently less than 1, gradients will vanish.

The Role of Graph Structure in Conditioning the Jacobian. For a GNN layer, the Jacobian $J^{(l)}$ can be expressed as:

$$J^{(l)} = (W^{(l)T} \otimes I_N) \cdot \text{diag}(\sigma'(\mathbf{a}^{(l)})) \cdot (I_C \otimes P) \tag{12}$$

where $\mathbf{a}^{(l)} = PH^{(l)}W^{(l)}$, and $\otimes$ is the Kronecker product. While this form is complex, its spectral norm is fundamentally influenced by the norms of its constituent parts. We can state that $\|J^{(l)}\|_2 \leq \|W^{(l)}\|_2 \cdot \|\text{diag}(\sigma')\|_2 \cdot \|P\|_2$. Assuming the learnable weights and activation derivatives are bounded (e.g., via weight decay and choice of σ), the spectral norm of the propagation matrix, $\|P\|_2$, becomes a structural regularizer. The conditioning of this Jacobian is fundamentally influenced by the spectral properties of the propagation matrix $P = D^{-1/2}AD^{-1/2}$. An ill-conditioned P can lead to an ill-conditioned J, thereby destabilizing the gradient flow. Our objective is therefore to find graph structures that produce a well-conditioned propagation matrix.

The more significant danger is gradient vanishing. This occurs if the cumulative Jacobian product is ill-conditioned, meaning its singular values are spread over many orders of magnitude. An ill-conditioned Jacobian will disproportionately shrink vectors aligned with its small singular vectors.

Spectral Gap as a Gradient Conditioner. Let us decompose the gradient vector $\mathbf{g}^{(l)}$ into a component parallel to the graph's stationary distribution (the eigenvector $\mathbf{v}_0$ of P for eigenvalue $\lambda_0 = 1$) and an orthogonal component: $\mathbf{g}^{(l)} = \mathbf{g}_{\parallel}^{(l)} + \mathbf{g}_{\perp}^{(l)}$. The parallel component, $\mathbf{g}_{\parallel}^{(l)}$, represents a uniform gradient across all nodes and is uninformative for learning node-specific features. The informative part of the signal is $\mathbf{g}_{\perp}^{(l)}$.

The propagation of the informative gradient component is governed by the non-trivial eigenvalues of the propagation matrix P. An ill-conditioned matrix P, characterized by a wide range of these eigenvalues, will distort the gradient. The conditioning of P in the informative subspace (orthogonal to the principal eigenvector) can be characterized by the ratio of its largest and smallest non-trivial eigenvalues, $\kappa_{\perp}(P) = p_2/|p_N|$, where a value close to 1 indicates good conditioning. As established in our proof for oversmoothing, maximizing the adjacency spectral gap ($\lambda_1 - \lambda_2$) is equivalent to minimizing p_2. Minimizing p_2 is the most critical factor in improving this condition number, as it pushes the entire non-trivial spectrum of P away from its dominant eigenvalue $p_1 = 1$. Furthermore,

our search for a large adjacency spectral gap is a proxy for finding graphs with Ramanujan-like properties, which by definition have tightly clustered non-trivial eigenvalues, leading to a well-conditioned P. This ensures that orthogonal components of the gradient are attenuated more uniformly during backpropagation, preserving the richness of the gradient signal and leading to more stable and efficient training.

A.3.4 Mathematical Derivation of Gradient Propagation

We provide a mathematical derivation to show how the spectral properties of a graph's propagation matrix are linked to the stability of the gradient signal during backpropagation.

1. The Layer-wise Gradient. Let the loss be $\mathcal{L}$ and the gradient vector with respect to the features at layer l be $\mathbf{g}^{(l)} = \frac{\partial \mathcal{L}}{\partial H^{(l)}}$. The backpropagation rule from layer l to $l-1$ is given by the chain rule:

$$\mathbf{g}^{(l-1)} = \mathbf{g}^{(l)} J^{(l-1)} \tag{13}$$

where $J^{(l-1)}$ is the layer's Jacobian. To isolate the effect of the graph structure, we analyze the propagation component, P. The structural update to the gradient, ignoring learnable weights, is:

$$\mathbf{g}_{\text{struct}}^{(l-1)} = P^T \mathbf{g}_{\text{struct}}^{(l)} \tag{14}$$

Since our propagation matrix $P = D^{-1/2} A D^{-1/2}$ is symmetric, $P^T = P$.

2. End-to-End Gradient Propagation. For a deep GNN with L layers, the gradient at the input layer is structurally dependent on the L-th power of the propagation matrix:

$$\mathbf{g}^{(0)} = P^L \mathbf{g}^{(L)} \tag{15}$$

The stability of learning depends entirely on the behavior of the matrix power P^L.

3. Gradient Norm Analysis. Using the property of the spectral norm, we can bound the norm of the output gradient:

$$\|\mathbf{g}^{(0)}\|_2 = \|P^L \mathbf{g}^{(L)}\|_2 \leq \|P^L\|_2 \cdot \|\mathbf{g}^{(L)}\|_2 \tag{16}$$

Further, since $\|P^L\|_2 \leq \|P\|_2^L$, and the spectral norm of our propagation matrix is $\|P\|_2 = p_1 = 1$, we get:

$$\|\mathbf{g}^{(0)}\|_2 \leq \|\mathbf{g}^{(L)}\|_2 \tag{17}$$

This demonstrates that the graph propagation step is non-expansive and acts as a regularizer against exploding gradients. The primary risk is the vanishing of the informative gradient signal.

4. Vanishing of the Informative Gradient Component. To analyze this, we decompose the gradient $\mathbf{g}^{(L)}$ into two orthogonal components: one parallel to the principal eigenvector of P (associated with $p_1 = 1$), which represents a uniform, non-informative signal, and one orthogonal to it, which carries the useful, node-specific information.

$$\mathbf{g}^{(L)} = \mathbf{g}_{\parallel}^{(L)} + \mathbf{g}_{\perp}^{(L)} \tag{18}$$

The non-informative component is preserved during backpropagation ($P^L \mathbf{g}_{\parallel}^{(L)} = \mathbf{g}_{\parallel}^{(L)}$). The informative component, however, evolves as:

$$\mathbf{g}_{\perp}^{(0)} = P^L \mathbf{g}_{\perp}^{(L)} \tag{19}$$

The action of P on this informative subspace is governed by its second-largest eigenvalue, p_2. We can thus bound the norm of the informative gradient:

$$\|\mathbf{g}_{\perp}^{(0)}\|_2 = \|P^L \mathbf{g}_{\perp}^{(L)}\|_2 \leq \|P_{\perp}^L\|_2 \cdot \|\mathbf{g}_{\perp}^{(L)}\|_2 \tag{20}$$

$$\leq (p_2)^L \cdot \|\mathbf{g}_{\perp}^{(L)}\|_2 \tag{21}$$

This leads to the final, critical relationship:

$$\|\mathbf{g}_{\perp}^{(0)}\|_2 \leq (p_2)^L \cdot \|\mathbf{g}_{\perp}^{(L)}\|_2 \tag{22}$$

Equation 22 shows that the norm of the informative gradient component vanishes exponentially at a rate controlled by p_2. To preserve the gradient signal, p_2 must be minimized. As established previously, maximizing the adjacency spectral gap is equivalent to minimizing p_2. Therefore, our search method optimizes the graph structure to mitigate the vanishing of informative gradients.

Limitations of Gradient Analysis. Our gradient flow analysis focuses on the structural component of backpropagation to isolate the effect of graph topology. This analysis does not include the interactions between learnable weight matrices and activation functions. Furthermore, our analysis relies on the symmetry of the propagation matrix ($P^T = P$), a condition met by the undirected graphs in our work, though we note this assumption does not extend to all graph construction methods (e.g., those with directed graphs). While this analysis provides strong intuition for why spectral properties matter, the full gradient dynamics in deep networks involve additional complexities not captured here.

A.4 The Heterogeneous GCS Objective

The theoretical analysis above provides a firm justification for our search strategy. The objective of our Graph Construction Search (GCS) is to create a novel heterogeneous Vision GNN. As formally stated in Equation 1 of the main paper, for each stage s of the network, characterized by a specific feature resolution, SearchViG solves the following optimization problem:

$$\mathcal{G}_s^* = \arg\max_{\mathcal{G} \in \mathcal{C}_s} \ (\lambda_1(A(\mathcal{G})) - \lambda_2(A(\mathcal{G}))) \tag{1}$$

where $\mathcal{C}_s$ is the set of candidate graph construction methods for that stage (e.g., k-NN, axial, static, attention-based), $A(\mathcal{G})$ is the adjacency matrix of a graph $\mathcal{G}$, and λ_1, λ_2 are its largest and second-largest eigenvalues, respectively. This principled, zero-shot approach allows for the creation of an architecture where each stage is equipped with a graph structure optimized for stable and efficient information flow.

B Connecting Graph Expansion to Signal-to-Noise Ratio

The theoretical analysis in the main paper justifies our spectral-aware search through the lens of GNN stability (i.e., mitigating oversmoothing and oversquashing). Here, we provide an alternative but complementary justification from a signal processing perspective. We argue that maximizing the adjacency spectral gap can be interpreted as implicitly optimizing the graph structure to improve the Signal-to-Noise Ratio (SNR) of the feature representations, which is linked to lower network loss.

SNR as a Proxy for Network Performance. The SNR of a network's feature representations is inversely related to the training loss (e.g., Mean Squared Error) [43]. Formally, the Signal-to-Noise Ratio (SNR) is the ratio of the power of the desired signal to the power of the corrupting noise:

$$\text{SNR} = \frac{\mathbb{E}[\text{signal}^2]}{\mathbb{E}[\text{noise}^2]} \tag{23}$$

A higher SNR indicates that the meaningful signal is strong relative to corrupting noise, implying more effective learning and lower loss [43, 85, 86]. Therefore, architectural choices that intrinsically promote a higher SNR are desirable.

Graph Propagation as a Low-Pass Filter. We can view the GNN propagation step as a filtering operation on a signal. The node features H represent the signal, and the graph topology defines the filter, captured by the propagation matrix $P = D^{-1/2}AD^{-1/2}$, where D is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. In this context, we can decompose the feature signal into two components:

- **Signal (H_{signal}):** The useful, low-frequency information that is smooth across neighboring nodes. This signal primarily aligns with the eigenvectors of the adjacency matrix A corresponding to its largest eigenvalues, especially the principal eigenvector associated with λ_1.
- **Noise (H_{noise}):** Random perturbations or irrelevant, high-frequency details that vary sharply between nodes. This noise aligns with eigenvectors corresponding to smaller eigenvalues ($\lambda_2, \lambda_3, \dots$).

The propagation matrix P acts as a low-pass filter, where the message passing operation amplifies the low-frequency signal components while attenuating high-frequency noise [44].

Maximizing the Spectral Gap as Implicit SNR Optimization. Our GCS is designed to find graphs with a large adjacency spectral gap ($\lambda_1 - \lambda_2$). This is the defining characteristic of an expander graph.

Such a graph has a dominant principal eigenvalue (λ_1) that is well-separated from the rest of the spectrum.

This large spectral gap creates a cleaner separation for the graph filter. The "pass-band" for the low-frequency signal (associated with λ_1) is kept distinct from the "stop-band" where the high-frequency noise (associated with λ_2 and below) resides. When features are propagated through this graph structure, the signal component is preserved while the noise component is more effectively suppressed. This increases the ratio of signal power to noise power in the resulting features.

Therefore, by searching for graphs with the largest possible spectral gap, our method is implicitly discovering graph topologies that function as superior low-pass filters. This optimization of the graph structure for signal and noise separation contributes to improving the SNR of the learned representations, providing a signal processing-based justification for why our method leads to better-performing models. We note, however, that this decomposition is a conceptual model, and we do not formally prove a direct optimization of SNR.

C Additional Experimental Results

To further validate our results we perform image classification on the CIFAR-100 and CIFAR-10 [63] datasets. To compare to other zero-shot NAS methods we compare to Zen-NAS [40] and TE-NAS [87]. To show broader generalization to real-world tasks, we also provide results on OrganSMNIST [64], DermaMNIST [64], DeepCrack [65], Crack500 [66], CityScapes [67], and COCO-Stuff [68] benchmarks.

C.1 CIFAR-100 Image Classification Results

We conduct image classification experiments on the CIFAR-100 [63] dataset, training from scratch for 200 epochs. We report the top-1 accuracy on the test set and implement all models using PyTorch [70] and the Timm library [71] with the AdamW [73] optimizer and a cosine annealing schedule.

Table 4: Classification results on CIFAR-100 for our SearchViG-S and other competing models.

Model	Type	Parameters (M)	Accuracy (%)
ResNet-50 [1]	CNN	23.7	80.9
ConvNeXt-T [58]	CNN	28.0	82.5
MobileViG-Ti [20]	CNN-GNN	4.3	80.2
MobileViG-B [20]	CNN-GNN	25.4	83.8
Swin-T [25]	ViT	28.0	74.9
SearchViG-S (Ours)	**CNN-GNN**	**11.0**	**84.3**

As shown in Table 4, our SearchViG-S model achieves state-of-the-art performance among efficient models on CIFAR-100. With only 11.0M parameters, it obtains 84.3% Top-1 accuracy, outperforming MobileViG-B by 0.5% while using 56.7% fewer parameters, demonstrating its superior efficiency.

C.2 CIFAR-10 Image Classification Results

We conduct further image classification experiments on the CIFAR-10 [63] dataset, which consists of 10 object classes, training from scratch for 200 epochs.

Table 5: Classification results on CIFAR-10 for our SearchViG-S and other competing models.

Model	Type	Parameters (M)	Accuracy (%)
ConvNeXt-T [58]	CNN	28.0	97.1
MobileViG-Ti [20]	CNN-GNN	4.3	95.6
MobileViG-B [20]	CNN-GNN	25.3	96.7
Swin-T [25]	ViT	28.0	91.1
SearchViG-S (Ours)	**CNN-GNN**	**11.0**	**97.1**

As shown in Table 5, SearchViG-S demonstrates exceptional efficiency on CIFAR-10. It matches the state-of-the-art accuracy of ConvNeXt-T (97.1%) while using $\approx 61\%$ fewer parameters (11.0M

vs. 28.0M). Compared to Swin-T, our model is both significantly more accurate (+6.0%) and more efficient (reduction of $\approx 61\%$ in parameters).

C.3 Comparison to Other Zero-Shot NAS Methods

We note that other zero-shot NAS methods (e.g., Zen-NAS [40], TE-NAS [87]) are **complementary** to ours rather than direct competitors, as they address fundamentally different problems and search spaces. Zen-NAS and TE-NAS focus on optimizing the depth, width, and operations within CNN architectures. In contrast, our SearchViG is the **first to tackle the unique challenge of finding optimal graph construction rules** within Vision GNNs, as summarized in Table 6.

Table 6: Comparison of Zero-Shot NAS Paradigms. SearchViG operates in a distinct search space from prior works.

Aspect	SearchViG (Ours)	Zen-NAS [40]	TE-NAS [87]
Primary Goal	Vision GNN Architectures	CNN Architectures	CNN Architectures
Search Space	Graph Construction	CNN Blocks	CNN Blocks
Zero-Shot Proxy	Adjacency Spectral Gap	Zen-Score	NTK and Linear Regions

In terms of performance, SearchViG compares favorably. We achieve **84.0% accuracy with 34.2M parameters (SearchViG-B)** and **83.3% with 25.7M parameters (SearchViG-M)**. In contrast, the largest Zen-NAS model [40] achieves 83.0% accuracy with 183M parameters. This means SearchViG-B achieves 1.0% higher accuracy with an 81.3% reduction in parameters, and SearchViG-M achieves 0.3% higher accuracy with an 86% reduction in parameters.

Compared to TE-NAS, our method demonstrates superior final accuracy as well (81.3% for our smallest architecture vs. 75.5% for TE-NAS [87] on ImageNet-1K).

C.4 Broader Generalization to Real-World Tasks

To further demonstrate the practical value and robust generalization of our method, we conducted new experiments on six diverse, real-world datasets spanning medical imaging, pavement crack detection, and challenging semantic segmentation benchmarks. The results confirm that SearchViG achieves state-of-the-art performance across these distinct domains.

Medical Image Classification

We evaluated SearchViG-S on the DermaMNIST and OrganSMNIST medical classification tasks from the MedMNISTv2 suite [64]. The first, OrganSMNIST, consists of 11 organ classes from 13,932 training and 2,452 validation abdominal CT images. The second, DermaMNIST, consists of 7 skin lesion classes from 7,007 training and 1,003 validation Dermatoscope images. For both datasets, we train models from scratch for 200 epochs using the AdamW optimizer [73] and a cosine annealing schedule. All implementations use PyTorch [70] and the Timm library [71].

As shown in Table 7 and Table 8, SearchViG-S outperforms established CNNs and other Vision GNNs on both medical benchmarks. Notably, on OrganSMNIST, SearchViG-S surpasses the much larger ConvNeXt-T by 0.8% accuracy while using over 60% fewer parameters, demonstrating the strong transferability of our architecture to specialized domains.

Table 7: Results on the OrganSMNIST medical image classification task.

Model	Type	Parameters (M)	Accuracy (%)
ResNet-34 [1]	CNN	20	91.1
ConvNeXt-T [58]	CNN	28	91.6
MobileViG-S [20]	CNN-GNN	6	91.0
SearchViG-S (Ours)	**CNN-GNN**	**11**	**92.4**

Table 8: Results on the DermaMNIST medical image classification task.

Model	Type	Parameters (M)	Accuracy (%)
ResNet-34 [1]	CNN	20	75.5
ConvNeXt-T [58]	CNN	28	78.1
MobileViG-S [20]	CNN-GNN	6	75.9
SearchViG-S (Ours)	**CNN-GNN**	**11**	**78.2**

Pavement Crack Segmentation

We benchmarked our SearchViG-B model on the DeepCrack [65] and Crack500 [66] datasets. The DeepCrack [65] dataset consists of 537 images, each with a size of 544 $\times$ 384 pixels, featuring complex backgrounds and various crack patterns. Following the setup of [88], 300 images are allocated for training, and 237 for testing. The Crack500 [66] dataset contains 3,368 pavement crack images, each with a resolution of 640 $\times$ 360 pixels, captured under various lighting and weather conditions. Following the setup of [88] we used 2,244 images for training and 1,124 for testing.

We evaluate model performance using precision, recall, and F1-score following the experimental setup of [88]. Where TP, FP, and FN represent true positives, false positives, and false negatives, respectively, the metrics are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{24}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{25}$$

Table 9: Comparison of crack segmentation performance on the DeepCrack and Crack500 datasets.

Methods	DeepCrack			Crack500		
	Precision	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
CrackSegNet [89]	83.8%	82.5%	80.3%	64.2%	80.7%	68.4%
SegCrack [90]	83.4%	82.2%	80.3%	67.8%	77.8%	70.0%
SETR [91]	79.5%	83.3%	79.5%	67.3%	73.3%	66.6%
SVGACrack [88]	85.0%	83.8%	81.8%	69.3%	80.2%	72.0%
SearchViG-B (Ours)	**85.5%**	**84.3%**	**82.3%**	**70.1%**	**80.8%**	**72.8%**

As detailed in Table 9, our model outperforms competing model on both Crack500 [66] and DeepCrack [65] datasets, demonstrating its strong capability for fine-grained segmentation tasks crucial for infrastructure maintenance. On the challenging Crack500 [66] dataset, SearchViG-B achieves a 72.8% F1-score, outperforming the next best method, SVGACrack [88], by a significant margin of 0.8%. This robust performance on both benchmarks underscores the effectiveness of our discovered topologies for precise, real-world segmentation.

Dense Semantic Segmentation

Finally, we evaluated our models on the challenging Cityscapes [67] and COCO-Stuff [68] benchmarks. Cityscapes [67] is a driving dataset for semantic segmentation consisting of 5000 fine-annotated high resolution images with 19 categories. COCO-Stuff [68] covers 172 labels and consists of 164k images: 118k for training, 5k for validation, and 40k for testing. We use the MMSegmentation [92] framework and the encoder is pre-trained on ImageNet-1K [69] and the decoder is randomly initialized. We apply data augmentation including random resizing, random horizontal flipping, and random cropping following the experimental setup of [93]. We train the models using the AdamW [73] optimizer for 160K iterations on Cityscapes and 80K iterations on COCO-Stuff, respectively. The learning rate starts at 6×10^{-5} and follows a polynomial decay schedule. Our experimental setup matches that of [93] for fair comparison.

As shown in Table 10, our SearchViG-S and SearchViG-B models outperform the state-of-the-art MiT [93] backbone on these challenging benchmarks. SearchViG-S surpasses the similarly sized

MiT-B1 by +1.2% mIoU on Cityscapes and +1.3% on COCO-Stuff. Furthermore, our SearchViG-B model achieves 82.1% mIoU on Cityscapes, outperforming the larger MiT-B3 while using 27% fewer parameters, which confirms the robust generalization of our architecture to complex scene parsing.

Table 10: SearchViG performance on downstream semantic segmentation tasks on Cityscapes and COCO-Stuff datasets.

Dataset	Backbone	Params (M)	mIoU (%)
Cityscapes	MiT-B1 [93]	13.7	78.5
	MiT-B2 [93]	27.5	81.0
	MiT-B3 [93]	47.3	81.7
	SearchViG-S (Ours)	**12.4**	**79.7**
	SearchViG-B (Ours)	**34.2**	**82.1**
COCO-Stuff	MiT-B1 [93]	13.7	40.2
	MiT-B2 [93]	27.5	44.6
	MiT-B3 [93]	47.3	45.5
	SearchViG-S (Ours)	**12.4**	**41.5**
	SearchViG-B (Ours)	**34.2**	**45.6**

D Complexity and Cost Analysis

Theoretical Complexity Analysis

The scaling complexity of our Graph Construction Search (GCS) is dictated by its core analytical step: the eigenvalue computation. While the search evaluates a fixed set of S graph configurations, the total complexity is best understood as $O(S \cdot C_\lambda)$, where C_λ represents the cost of computing the top two eigenvalues of an $N \times N$ adjacency matrix.

The scaling of this process is therefore primarily dependent on C_λ with respect to the number of nodes, N. Because the graphs generated in our search space are **sparse**, this computation is highly efficient. By employing an iterative solver the cost C_λ is **near-linear** in the number of nodes (i.e., approximately $O(N)$). This low-cost scaling of the core analytical step ensures our entire search process is computationally tractable, with a **total complexity of approximately** $O(N)$.

Empirical Cost Analysis

A key advantage of our framework is its efficiency. The entire zero-shot GCS requires only **4 GPU hours** on a single NVIDIA A6000 GPU. As detailed in Table 11, this search cost is negligible compared to the model training time, representing less than 0.5% of the total training cost for our SearchViG-B model. This confirms that our spectral proxy is an effective and efficient method for discovering optimal topologies without any training. For our experimental setup we used 1000 random samples of images, but if we reduce this to 250 random samples of images we can reduce the search cost to 0.105% of the total training time.

Table 11: Computational cost analysis on an A6000 GPU in terms of GPU hours.

Component	Cost (GPU Hours)
SearchViG-B Training	950
PViG-S Training	1200
ViHGNN-S Training	1300
Search Cost / Training Cost	**0.42%**

D.1 Search Process Sensitivity and Stability

We performed a sensitivity analysis to validate the robustness of our search process. We ran our GCS 15 times on random subsets of ImageNet images, with the sample size (N) varying for each set of runs. The results, summarized in Table 12, demonstrate that our search is highly stable.

Table 12: GCS Stability Analysis. Results of 15 GCS runs per sample size.

Sample Size (N)	Identical Topologies Discovered
N = 1000 images	15/15
N = 500 images	15/15
N = 250 images	15/15
N = 100 images	13/15
N = 50 images	11/15
N = 10 images	8/15

This high degree of stability, especially for $N \geq 250$, is consistent with the well-established principle that natural images follow specific statistical regularities [94–96], which our spectral proxy effectively captures.

E Correlation of Spectral Proxy and Final Performance

To provide formal analysis on the correlation between our proxy and final performance, we computed the rank correlation between our zero-shot spectral proxy and the final trained accuracy. The analysis was conducted across 9 different graph construction methods (Ours, KNN, Clustered KNN, windowed KNN, random, static, dynamic axial, logarithmic connections, and dynamic local and logarithmic connections) while keeping the underlying network architecture fixed. This approach allows us to isolate and measure the specific impact of the graph topology on performance. All models were evaluated on ImageNet-1K. The results yield a **Spearman's rank correlation coefficient (ρ) of 0.85 and a Kendall's Tau (τ) of 0.722.** Both of these values indicate a strong positive correlation, providing formal quantitative evidence that a higher adjacency spectral gap for the graph construction is a reliable predictor of higher final accuracy after training.

F Hyperparameter Settings

The detailed hyperparameter settings used for our ImageNet-1K training are provided in Table 13. The hyperparameter settings match those of Vision GNN [13], MobileViG [20], and EfficientFormer [59] for fair comparison.

Table 13: Training hyperparameters for ImageNet-1K.

Hyperparameter	**Value**
Epochs	300
Optimizer	AdamW [73]
Batch Size	1024
Start Learning Rate (LR)	2e-3
LR Schedule	Cosine
Warmup Epochs	20
Weight Decay	0.05
Repeated Augment [97]	✓
RandAugment [98]	✓
Mixup Prob. [99]	0.8
Cutmix Prob. [100]	1.0
Random Erasing Prob. [101]	0.25
Exponential Moving Average	0.99996

G Future Work

The SearchViG framework introduces the first zero-shot method for discovering heterogeneous Vision GNN architectures. This opens several promising avenues for future research.

G.1 Scalability to Larger Datasets

Dataset Scalability. Our current search is performed on a subset of ImageNet-1K. Future work could investigate the scalability and robustness of the discovered topologies when the search is performed on larger and more diverse datasets. While our sensitivity analysis suggests a small subset is sufficient, validating this on larger-scale data is beneficial. Furthermore, the search cost, while low, scales with the number of samples and graph construction methods. Exploring more advanced subset selection strategies to capture maximum data diversity with minimal samples would be beneficial.

G.2 Extending the Search Framework

Layer-wise vs. Stage-wise Heterogeneity. Our framework's search space includes **dynamic, content-aware graph construction mechanisms.** However, SearchViG discovers an optimal graph that is fixed for all layers within a given stage. A natural extension is to apply the Graph Construction Search at a **layer-wise granularity**. This would allow the graph topology to evolve not just when the resolution changes, but also as features are refined within a single stage. This would create a finer-grained and potentially more powerful heterogeneous architecture, though it would also increase the search cost.

Multi-Objective Search Proxies. Our search is guided by a single, theoretically-grounded proxy. A compelling future direction is to move to a **multi-objective search**, where the spectral gap could be combined with other proxies that capture different properties, such as trainability or efficiency.