[Team 14] Final Report
Hypergraph Vision Transformers: Images are More than Nodes, More than Edges

Donghwan Seo
Biomedical Engineering
2020250065
skjs002@korea.ac.kr

Jeongmin Lee
Computer Science and Engineering
2023320060
a23493307@gmail.com

SeongHyeock Pyeon
Computer Science and Engineering
2024320141
04dot15@gmail.com

1. Introduction

This report presents a review of a research paper conducted as a team project for the Fall 2025 Deep Learning course and is written in compliance with the provided requirements.

2. Motivation

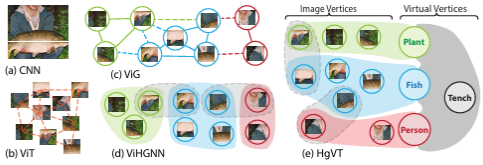


Figure 1. Motivation for hypergraph-based modeling.

The Hypergraph Vision Transformer (HgVT) is proposed to extend the representational capacity of current vision architectures, particularly in modeling relations beyond pairwise interactions. Convolutional networks are built upon local filters, enabling effective extraction of localized spatial details but making it difficult for them to capture long-range dependencies or semantic groupings. Vision Transformers incorporate global self-attention, yet the attention mechanism remains fundamentally pairwise: every patch token interacts with every other token, but the model does not explicitly encodes a multi-patch structure, such as a visual concept shared among a subset of patches. Previous graph-based vision networks attempt to impose relational structure by constructing graphs or hypergraphs through clustering methods, but such procedures are computationally heavy, fixed once computed, and unable to adjust as the network representation expands. HgVT is designed in response to these limitations.

3. Overview of the Proposed Idea

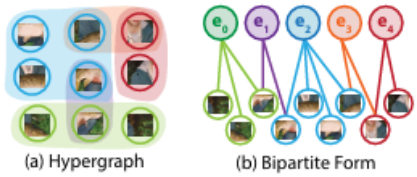


Figure 2. Overall bipartite hypergraph design of HgVT.

HgVT reinterprets an input image as a learnable bipartite hypergraph containing image vertices and virtual vertices, along with two types of hyperedges. Image patches correspond to image vertices, while virtual vertices serve as abstract semantic tokens. Primary hyperedges connect both image and virtual vertices, and virtual hyperedges connect only among virtual vertices.

In each transformer block, HgVT reconstructs the hypergraph by estimating vertex–hyperedge membership through a similarity-driven mechanism. The soft adjacency matrix is defined as

$$A = \sigma(\alpha S), \qquad S = \tilde{\mathbf{X}}_{\text{adj}}^{(V)} \left(\tilde{\mathbf{X}}_{\text{adj}}^{(E)}\right)^{\top},$$

where normalized adjacency features are given by

$$\tilde{\mathbf{X}}_{\text{adj}} = \frac{\mathbf{X}_{\text{adj}}}{\|\mathbf{X}_{\text{adj}}\|_2}.$$

A sharpened sigmoid produces near-binary outputs, and a hard incidence matrix is extracted via thresholding:

$$\hat{A}_{ij} = \begin{cases} 1, & A_{ij} > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

This process is repeated independently at every layer, allowing the hypergraph structure to evolve as feature representations become increasingly abstract.

Together, these innovations produce a representation capable of expressing complex visual structures and higher-order reasoning.

4. Explanation of the Method

4.1. Architectural Structure

HgVT divides the input image into patches, projecting each into a feature vector that becomes an image vertex. The architecture also includes learnable virtual vertices and two types of hyperedges. Let the vertex and hyperedge feature matrices be

$$\mathbf{X}^{(V)} \in \mathbb{R}^{|V| \times d_v}, \qquad \mathbf{X}^{(E)} \in \mathbb{R}^{|E| \times d_e},$$

with separate adjacency features

$$\mathbf{X}^{(V)}_{\text{adj}}, \qquad \mathbf{X}^{(E)}_{\text{adj}}.$$

These adjacency embeddings guide the computation of membership between vertices and hyperedges.

4.2. Dynamic Hypergraph Construction

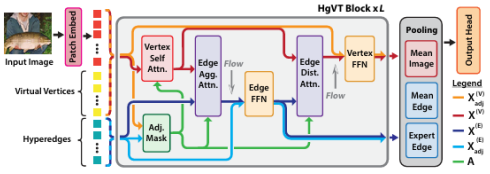


Figure 3. Dynamic hypergraph reconstruction inside each block.

In every transformer block, HgVT constructs a new hypergraph by computing cosine similarity between normalized adjacency projections of vertices and hyperedges. A sharpened sigmoid produces near-binary assignments, and thresholding yields a sparse incidence matrix. Virtual hyperedges are restricted to connect only to virtual vertices, maintaining separation between spatial and abstract components.

4.3. Message Passing and Attention

Once the hypergraph is built, attention proceeds in three constrained stages:

1. **Vertex self-attention.** Vertices attend only to those sharing at least one hyperedge, producing naturally sparse and semantically structured attention patterns.
2. **Hyperedge aggregation.** Hyperedges aggregate information via cross-attention, where hyperedges query the set of vertices. Membership strengths modulate the attention logits.
3. **Redistribution to vertices.** Aggregated hyperedge features are redistributed through cross-attention, enabling vertices to integrate higher-order signals through their hyperedge memberships.

4.4. Feature Transformation and Regularization

Vertices and hyperedges are fed into independent feed-forward networks, allowing them to develop distinct semantic functions. Because the hypergraph is learned from similarity alone, regularization terms stabilize the structure:

- **Diversity regularization** encourages orthogonality among virtual vertices and hyperedges to prevent redundancy.
- **Population regularization** constrains the number of vertices per hyperedge, preventing collapse into overly large or unused hyperedges.

4.5. Global Representation via Expert-Style Pooling

HgVT treats virtual hyperedges as experts for global representation. Each virtual hyperedge produces a confidence score via a learned projection and softmax. During inference, only the most confident experts contribute to the final embedding, improving interpretability and promoting specialization.

5. Survey of Related Work

Structured and Routed Attention in Vision Transformers Recent Vision Transformers increasingly replace fully dense self-attention with structured or routed interactions that allocate computation to a subset of informative regions. Bi-level routing attention, as in BiFormer [5], performs coarse-to-fine token selection to limit fine-grained attention to routed subsets, improving efficiency while preserving global context. Similarly, TransNeXt [3] introduces inductive biases that emphasize salient regions to achieve robust perception at scale. These approaches reflect a broader shift toward structured attention mechanisms that guide token interactions beyond uniform all-to-all attention.

Token Reduction and Dynamic Computation A complementary line of work improves efficiency by reducing token redundancy or making computation conditional. Token Merging (ToMe) [1] progressively merges similar tokens using lightweight matching, often without retraining.

Related dynamic computation methods, including sparse mixture-of-experts formulations [2], scale model capacity while keeping per-sample computation bounded. Unlike purely compressive strategies, structure-aware approaches aim to retain relational information that can support downstream semantic tasks.

Hypergraph-augmented Transformers Recent studies explore integrating hypergraph structure into transformers to capture higher-order relations among tokens. HGFormer [4] constructs hypergraphs via neighborhood sampling and applies topology-aware attention to encourage region-level grouping. Other hypergraph-based formulations further demonstrate the potential of multi-way relational modeling beyond pairwise token interactions. These efforts position hypergraph-augmented transformers as a promising direction for structured representation learning in vision.

6. Critical Review

HgVT is well-motivated in that it targets a concrete bottleneck in prior vision hypergraph models—the reliance on repeated, expensive clustering for structure construction—and proposes a principled alternative that integrates bipartite hypergraphs directly into a transformer backbone. While the core idea is sound and the results are competitive, several limitations weaken the paper's claim as a broadly impactful architectural contribution.

Limited architectural generality The empirical evaluation is restricted to isotropic, single-scale architectures, with pyramidal and multi-scale designs explicitly excluded due to the difficulty of down-sampling virtual tokens. This substantially limits the generality of the claims, as modern vision backbones and dense prediction tasks predominantly rely on hierarchical representations. Without empirical evidence or a concrete integration strategy for multi-scale pipelines, it remains unclear whether HgVT extends beyond a narrowly defined architectural setting.

Misalignment between structural modeling and task objectives The paper introduces custom metrics to analyze the learned hypergraph structure, but it does not clearly justify why these quantities reflect task-relevant behavior. The reported weak anti-correlation between structure quality and Top-1 accuracy suggests that the structural objectives are only loosely coupled to the primary learning goal. In addition, adjacency construction relies on a fixed cosine-similarity kernel with a non-learnable sharpening function, constraining the model's ability to represent task-specific semantic relationships. As a result, the structural analysis remains largely diagnostic rather than explanatory.

Unclear semantic role of virtual vertices The analysis indicates that introducing virtual vertices increases inter-cluster separation while reducing intra-cluster compactness. This raises ambiguity about their semantic role: rather than acting as higher-level semantic abstractions, virtual vertices may partially introduce structural noise. Although expert pooling is proposed as a mitigating mechanism, the paper does not provide a systematic characterization of what semantics hyperedges encode or how stable these representations are across layers and inputs.

Insufficient validation of efficiency and robustness Although HgVT is positioned as an efficient alternative to clustering-based approaches, the evaluation does not isolate the contribution of the hypergraph mechanism from architectural or scaling adjustments. Efficiency claims rely primarily on FLOPs and parameter counts, which may not fully capture practical costs such as memory overhead, dynamic adjacency construction, or training stability. Furthermore, limited analysis of robustness—such as sensitivity to hyperparameters or random seeds—weakens confidence in reproducibility and real-world deployment.

7. Improvement Directions

While HgVT demonstrates strong performance through dynamic hypergraph construction and efficient vertex–hyperedge communication, several architectural extensions may further enhance its adaptability, semantic consistency, and applicability to dense prediction tasks. We outline three promising directions below.

7.1. Learnable Hyperedge Queries Instead of Fixed Cosine Similarity

HgVT currently relies on sharpened cosine similarity between vertex and hyperedge adjacency features to form the soft membership matrix. Although effective, this fixed similarity metric may limit the expressiveness of hyperedge formation. A natural extension is to replace cosine similarity with a *learnable* query–key projection:

$$k_i = \phi_V\left(X_i^{(V)}\right), \qquad q_j = \phi_E\left(X_j^{(E)}\right),$$

where ϕ_V and ϕ_E are lightweight MLPs or low-rank linear mappings that project vertices and hyperedges into a shared "adjacency space." The adjacency matrix can then be computed as:

$$A = \sigma\left(\alpha \cdot \frac{\phi_V(X^{(V)})\,\phi_E(X^{(E)})^\top}{\sqrt{d_a}}\right),$$

allowing the model to *learn* how hyperedges should query vertices, instead of relying on a hand-designed similarity

measure. Such parameterized adjacency formation may improve semantic clustering, particularly for fine-grained categories or datasets with heterogeneous appearance distributions.

7.2. Pyramidal HgVT for Dense Prediction Tasks

The current HgVT architecture is isotropic, processing all image tokens at a fixed spatial resolution. To adapt HgVT to dense prediction tasks such as semantic segmentation or object detection, a *pyramidal* design may be introduced. In this framework, image vertices are progressively downsampled across stages ($\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, etc.), while virtual vertices and hyperedges remain as global, resolution-agnostic tokens that persist throughout the hierarchy.

Prior to each down-sampling step, a mild smoothing operation (e.g., Gaussian blur) may be applied to reduce aliasing artifacts, following the classical construction of Gaussian image pyramids. These resolution-reduced vertices allow the network to capture both fine- and coarse-scale structures, while the persistent hyperedges provide global semantic context to all levels. This enables bottom-up detail aggregation together with top-down semantic propagation:

$$X_{\ell+1}^{(V)} = \text{Downsample}(X_{\ell}^{(V)}), \qquad X_{\ell+1}^{(E)} = X_{\ell}^{(E)}.$$

A pyramidal HgVT therefore combines the benefits of multi-scale feature hierarchies with hypergraph-driven message passing, potentially improving dense prediction accuracy without sacrificing computational efficiency.

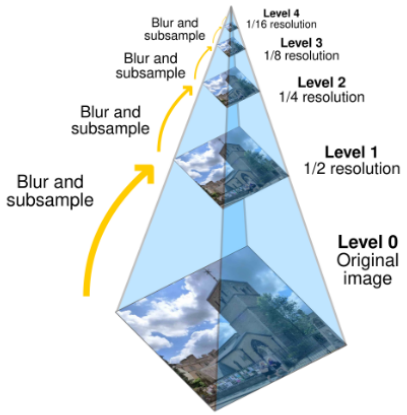


Figure 4. Illustration of a pyramidal HgVT design. Image vertices undergo iterative blur-and-downsample operations across levels, while virtual vertices and hyperedges remain as resolution-invariant global tokens that provide semantic context to all stages.

7.3. Contrastive Hyperedge Supervision for More Robust Grouping

Although HgVT forms hyperedges through dynamic adjacency, the process remains fully implicit. Introducing a lightweight contrastive supervision signal may further stabilize and sharpen hyperedge semantics. Specifically, vertices belonging to the same class or region can be encouraged to attend to similar hyperedges, while unrelated vertices are pushed apart:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp\left(\text{sim}(h_i, h_j)/\tau\right)}{\sum_k \exp\left(\text{sim}(h_i, h_k)/\tau\right)},$$

where h_i denotes the hyperedge embedding associated with vertex i. This auxiliary loss can help hyperedges acquire more stable roles (e.g., "texture," "global shape," "object part"), ultimately improving both classification and retrieval performance.

8. Novelty and Strengths

8.1. Key Contributions

Solution to Computational Bottlenecks via Dynamic Construction HgVT introduces a novel dynamic hypergraph construction mechanism that replaces the computationally expensive, non-differentiable clustering algorithms used in prior Vision GNNs. By utilizing a differentiable, attention-based querying process to rebuild the graph structure within each transformer block , the authors effectively resolve the trade-off between adaptability and efficiency. This contribution is strategically significant because it transforms the hypergraph from a static pre-processing artifact into a fully adaptive, end-to-end learnable component, streamlining the architecture significantly compared to previous iterations.

Capturing Higher-Order Relationships with Bipartite Structures A key innovation is the formulation of a bipartite structure where hyperedges function as active communication pools rather than passive connectors. By explicitly separating image vertices from virtual hyperedges and implementing structured message passing $\mathcal{V} \to \mathcal{E}$ and $\mathcal{E} \to \mathcal{V}$, the model enforces a strong inductive bias that mirrors the hierarchical nature of visual semantics. This design provides a robust alternative to standard pairwise attention, allowing the network to capture higherorder relationships and semantic groupings that standard ViTs often struggle to model explicitly.

Framework-Oriented Contribution From a paper-level perspective, a clear strength of this work is that it is presented as a framework-oriented study rather than a narrowly model-centric one. The paper does not focus solely on

introducing a single architecture and validating it through benchmark performance. Instead, it frames the proposed approach as a general way of thinking about structure and semantics in vision models, and demonstrates its implications through multiple forms of analysis, including classification, ablation, structural evaluation, and retrieval. This gives the paper a broader scope and a clearer identity, positioning it as a reusable framework rather than a task-specific improvement.

8.2. Overall Evaluation

Overall, this paper presents a well-motivated approach to integrating hypergraph structures into vision transformers. By addressing key limitations of existing vision GNN and hypergraph models, the proposed HgVT architecture tightly couples hypergraph construction, attention masking, and pooling. Empirical results on ImageNet classification and image retrieval demonstrate that these ideas are effective in practice, achieving performance and efficiency competitive with strong isotropic baselines such as ViHGNN and DeiT, while outperforming a retrieval-specific baseline on ImageNet-1k.

Model	Params	FLOPs	ImNet Top-1	ReaL Top-1	V2 Top-1
❉ ResMLP-S12 conv3x3 [52]	16.7M	3.2B	77.0	84.0	65.5
❉ ConvMixer-768/32 [55]	21.1M	20.9B	80.2	–	–
❉ ConvMixer-1536/20 [55]	51.6M	51.1B	81.4	–	–
♦DINOv1-S [2]	21.7M	4.6B	77.0	–	–
♦ViT-B/16 [12]	86.4M	55.5B	77.9	83.6	–
♦DeiT-Ti [53]	5.7M	1.3B	72.2	80.1	60.4
♦DeiT-S [53]	22.1M	4.6B	79.8	85.7	68.5
♦DeiT-B [53]	86.4M	17.6B	81.8	86.7	71.5
★ViG-Ti [16]	7.1M	1.3B	73.9	–	–
★ViG-S [16]	22.7M	4.5B	80.4	–	–
★ViG-B [16]	86.8M	17.7B	82.3	–	–
■ViHGNN-Ti [17]	8.2M	1.8B	74.3	–	–
■ViHGNN-S [17]	23.2M	5.6B	81.5	–	–
■ViHGNN-B [17]	88.1M	19.4B	82.9	–	–
▲HgVT-Ti (ours)	7.7M	1.8B	**76.2**	**83.2**	**64.3**
▲HgVT-S (ours)	22.9M	5.5B	81.2	**86.7**	**70.1**

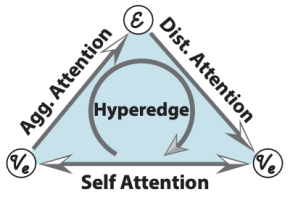
Figure 5. Comparison of classification accuracy and efficiency on ImageNet-1k, ImageNet-ReaL, and ImageNet-V2 for isotropic vision models. HgVT achieves competitive accuracy–efficiency trade-offs relative to ViHGNN and DeiT, with improved robustness on ReaL and V2.

At the same time, the work leaves several avenues open. The restriction to isotropic models, the acknowledged difficulty of tuning the hypergraph-related regularizations and pooling, and the observed tension between graph-structure metrics and classification accuracy all point to complexity that future work will need to manage. The absence of experiments on dense prediction tasks and pyramidal architectures also limits the current scope of the claims, even though the methodological ideas appear broadly applicable.

9. Q&A

Q1. HgVT aggressively compresses hundreds of patches into a small number of hyperedges, which should normally cause significant loss of fine-grained visual information. Yet the model reports almost no accuracy drop compared to ViT. How does the method preserve fine-grained details despite this strong information bottleneck? Is there a theoretical justification for why such heavy compression does not degrade performance?

A1. HgVT does not replace patch tokens with hyperedges in a lossy manner. Instead, it keeps all patch vertices throughout the network and uses a smaller set of hyperedges as learned communication pools that route information between patches. Patch vertices are still updated directly, while vertex-to-hyperedge and hyperedge-to-vertex interactions enable structured aggregation without discarding fine-grained details.



(a) Communication Pool.

Figure 6. Communication Pool between hyperedge-vertex.

Because membership is soft, a patch can contribute to multiple hyperedges rather than being forced into a single hard cluster. Although the paper does not provide a formal theoretical guarantee, it motivates this design as a way to reduce attention cost while preserving patch-level representations, and empirically shows little to no accuracy drop compared to ViT on ImageNet.

Q2. HgVT recomputes its hypergraph adjacency from evolving features at every training step, meaning the model's structure changes continuously during training. Such dynamic graphs are known to cause unstable gradients, yet the paper provides no convergence analysis. How does HgVT ensure stable training under these conditions?

A2. While no convergence analysis is provided, the paper incorporates several stabilizing design choices. Adjacency is computed from L2-normalized features, reducing sensitivity to feature scale changes. Cosine similarities are sharpened and thresholded into hard masks for sparse attention, which limits jitter in connectivity. In addition, diversity regularization prevents collapse of virtual features, and population regularization keeps hyperedge densities within a reasonable range. Together, these mechanisms are intended

to make training stable in practice despite the dynamically changing structure.

Q3. HgVT constructs hyperedges solely based on cosine similarity between patch features, which means spatially distant patches can be grouped together. This may lead to a loss of the original spatial structure of the image. Since the paper does not present experiments or analysis on this issue, how does HgVT mitigate or compensate for potential spatial information loss in practice?
A3. Because hyperedge membership is driven by feature similarity, HgVT can indeed group spatially distant patches. The paper does not explicitly analyze spatial structure preservation, but mitigates this issue architecturally by never discarding patch vertices, allowing spatially local information to remain encoded at the patch level. Hyperedges act as an additional relational pathway rather than replacing the patch grid. Since adjacency is recomputed per block, harmful long-range groupings can be weakened through learning, and population regularization further discourages overly broad hyperedges.

Q4. While reading the HgVT paper, I noticed that the model builds hyperedges dynamically using cosine similarity instead of fixed clustering. Since this can change per image and per layer, is there a risk that the hypergraph becomes unstable or overly sensitive to small feature changes? How do the authors control that instability?
A4. Frequent recomputation of adjacency can indeed make the hypergraph sensitive to small feature changes. The authors address this through L2 normalization of adjacency features, sharpening and thresholding of cosine similarities to obtain stable hard masks, and regularization terms. Population regularization constrains hyperedge density, while diversity regularization prevents virtual representations from collapsing. These mechanisms reduce structural degeneracy and sensitivity, although they do not constitute a formal stability guarantee.

Q5. HgVT introduces virtual vertices and virtual hyperedges to build a hierarchical structure. In particular, why are the virtual hyperedges used for classification restricted to connect only to virtual vertices rather than to actual image patch vertices, and how does this restrictive communication pathway contribute to improved performance?
A5. The model enforces a hierarchical communication pattern by restricting classification hyperedges to connect only with virtual vertices. This separates patch-level visual processing from class-level abstraction and allows virtual tokens to specialize as global summarizers not tied to specific patches. Multiple virtual hyperedges are then combined using an expert-style pooling mechanism, encouraging diverse and complementary class-level representations rather than a single redundant classifier token.

Q6. Since hyperedges are defined as subsets of vertices, the number of hyperedges can be much larger than the number of vertices. However, as mentioned in the paper, the $\mathcal{O}(|V|E)$ time complexity relies on the condition $E < |V|$. How can we prevent the case $E > |V|$, or alternatively, how can efficiency be maintained when $E > |V|$?
A6. The paper primarily enforces $E < |V|$ through architectural design by fixing a small hyperedge budget relative to the number of vertices. This keeps the bilinear adjacency computation manageable. If E were to exceed $|V|$, the same computation would become a bottleneck, so efficiency would rely on practical mechanisms such as thresholding adjacency to enable sparse attention and using expert-style routing at inference to avoid aggregating over all hyperedges. Overall, HgVT prioritizes controlling hyperedge capacity by design and exploiting sparsification to maintain efficiency.

10. Individual Contribution

Donghwan Seo
- Coordinating team discussions and ensuring consistency across the analysis, presentation, and written materials.
- Summarizing the core ideas of the paper, including the motivation, contributions, and key results.
- Providing a detailed explanation of the proposed method and architecture, with emphasis on hypergraph construction and dynamic attention mechanisms.
- Preparing and delivering the presentation, as well as leading the question-and-answer session.

Jeongmin Lee
- Prepared and presented the presentation materials, focusing on the improvement and extension of the proposed method.
- Wrote the Critical Review section, analyzing the strengths and limitations of the paper.
- Surveyed and organized relevant Related Work to position the paper within existing literature.
- Compiled and answered the Q&A section, addressing technical questions and potential concerns regarding the proposed approach.

SeongHyeock Pyeon
- Prepared presentation materials on Model Architecture and Critical Issues, delivered the presentation
- Delivered a presentation on Model Architecture and Critical Issues with team
- Wrote the Novelty and Strengths section, while analyzing the paper's methodology and results.

References

[1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster, 2023. 2

[2] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. 2022. 3

[3] Dai Shi. Transnext: Robust foveal visual perception for vision transformers. In *CVPR*, 2024. 2

[4] Hao Wang, Shuo Zhang, and Biao Leng. Hgformer: Topology-aware vision transformer with hypergraph learning, 2025. 3

[5] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson Lau. Biformer: Vision transformer with bi-level routing attention. In *CVPR*, 2023. 2