Vision Correlators: Correlation-Driven Visual Understanding with Hypergraphs

Mengqi Lei[1] **Guohuan Xie**[1] **Siqi Li**[1*] **Shihui Ying**[2]
Shaoyi Du[3] **Jun-Hai Yong**[1] **Nanning Zheng**[3] **Yue Gao**[1]

[1]{BNRist, THUIBCS, School of Software}, Tsinghua University
[2]Shanghai Institute of Applied Mathematics and Mechanics,
School of Mechanics and Engineering Science, Shanghai University
[3]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

leimq25@mails.tsinghua.edu.cn, stuxiemol@gmail.com, lisiqi19971013@gmail.com
shying@shu.edu.cn, dushaoyi@xjtu.edu.cn, yongjh@tsinghua.edu.cn
nnzheng@mail.xjtu.edu.cn, kevin.gaoy@gmail.com

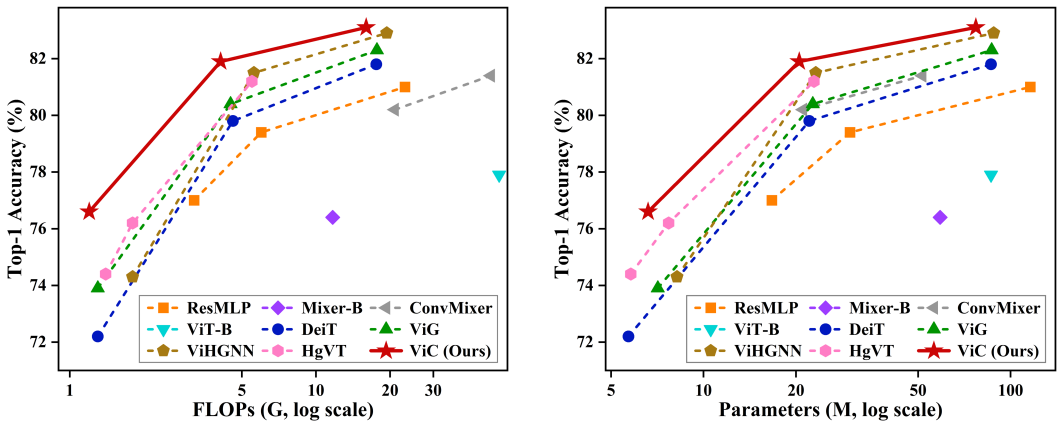

Figure 1: Comparison of the proposed Vision Correlator (ViC) with other representative methods on ImageNet: accuracy-FLOPs (left) and accuracy-parameter (right) trade-offs.

Abstract

Modern visual backbones, despite their architectural diversity, largely reduce relational reasoning to pairwise message passing on graphs and therefore primarily capture 2-order interactions. However, visual understanding often depends on latent yet crucial high-order semantic correlations among multiple regions, which cannot be adequately expressed by pairwise modeling alone. We first formalize this limitation by proving an irreducibility result: on a fixed vertex set, there exist nonlinear hypergraph message passing layers that cannot be represented exactly by any graph message passing layer unless the hypergraph satisfies highly restrictive structural conditions. This shows that high-order correlation modeling offers a fundamental expressive advantage rather than a simple approximation to graph-based backbones. Therefore, we propose a hypergraph-based visual correlation modeling framework, namely Correlation Induction–Correlation Propagation, and design Vision Correlator (ViC) according to the framework. In the Correlation Induction stage, we propose an Anchor-based Hypergraph Generation strategy that uses anchors as semantic centers and incorporates topological information

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

to generate scalable hypergraph structures. In the Correlation Propagation stage, we further design Differential Mixed Aggregation to achieve efficient hyperedge representation and vertex feature updating, and introduce hyperedge dropout to improve the stability of structure learning. Extensive experiments with isotropic and pyramid variants show that our ViC achieves a better accuracy-efficiency trade-off than strong Transformer and graph-based baselines. Code and models are available at: `https://github.com/Mengqi-Lei/vision-correlator`.

1 Introduction

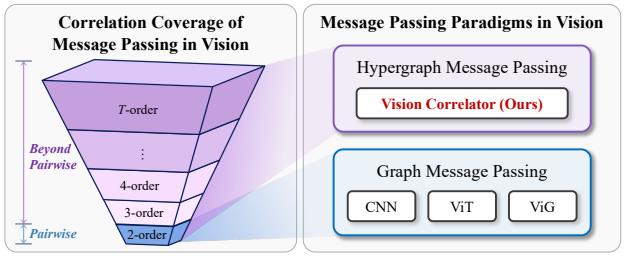

Figure 2: Correlation coverage and message passing paradigms in visual understanding.

In recent years, visual backbone networks [1, 2] have made steady progress in tasks such as image classification [3] and object detection [4, 5, 6, 7]. From convolutional networks [8] to Vision Transformers (ViT) [9], and more recently to graph-based visual models [10], this progress has been accompanied by an evolution in how visual tokens interact with each other. Although these approaches differ significantly in their specific architectures, when re-examined from the perspective of correlation modeling, most of them can be interpreted as message passing between visual tokens on graphs, as shown in Fig. 2. More specifically, convolutions can be viewed as neighborhood message passing on a local grid graph [11, 12], self-attention as dynamic message passing on a fully connected graph [13, 14], and frameworks directly based on graph neural networks (GNNs) as performing inter-node communication more explicitly on graph structures [15]. Therefore, the development of visual backbones can be seen as a progressive expansion of visual correlation modeling from local neighborhoods to global interactions and explicit relational structures. Nevertheless, these paradigms still share a fundamental premise at the level of relational expression: they primarily model pairwise correlations.

However, pairwise correlations only cover a limited part of the correlation space required by real-world visual understanding [16, 17, 18, 19]. Many key visual patterns are not determined solely by the interaction between two local regions, but rather emerge from high-order semantic correlations jointly formed among multiple regions, multiple components, and multiple semantic entities. Compared to a large number of dense and redundant pairwise correlations, such high-order correlations are often distributed across the entire scene and possess greater discriminative value for scene understanding [18, 19]. While relying solely on pairwise interactions can effectively model local structures, it struggles to fully capture the multi-regional semantic coordination required for visual understanding. In other words, a fundamental limitation of existing visual backbones lies in the fact that their correlation coverage is largely constrained to pairwise structures.

This paper first formalizes this limitation from a theoretical perspective. We prove that, *for a given set of vertices, there exists a nonlinear hypergraph message passing layer whose propagation results cannot be exactly equated to those of any graph message passing layer, unless the corresponding hypergraph satisfies extremely strong structural constraints*. This result indicates that modeling high-order correlations is not merely a simple approximation or enhancement of graph message passing, but rather possesses irreducible expressive advantages. Specifically, at a comparable scale, graph message passing methods are structurally constrained by pairwise correlations, making their coverage of the correlation space inferior to that enabled by hypergraph message passing methods. Therefore, moving from pairwise graph message passing to hypergraph message passing is a necessary step for expanding the correlation coverage of visual understanding.

A natural way to expand the correlation coverage of visual understanding is to introduce hypergraphs [20, 21]. However, existing hypergraph-based visual methods [22, 23] typically rely on predefined high-order structural forms, or use hypergraph computation as an auxiliary module within existing backbones, and thus lack a general framework for adaptively discovering and propagating visual correlations of different orders. Therefore, in this paper, we reformulate visual understanding as a hypergraph-based visual correlation modeling framework, namely Correlation Induction–Correlation Propagation: first inducing multi-order (second and higher orders) correlation structures from visual tokens, and then propagating information across the induced hypergraph structure.

Following the hypergraph-based visual correlation modeling framework above, we propose an effective visual backbone, Vision Correlator (ViC). ViC consists of two stages. In the Correlation Induction stage, we construct a unified hypergraph structure across different semantic scales through Anchor-based Hypergraph Generation. On the one hand, each token itself serves as a local token anchor, and local correlations are induced through nearest-neighbor queries. On the other hand, a set of learnable semantic anchors is introduced to induce global high-order hyperedges for capturing cross-regional and cross-part semantic correlations. In the Correlation Propagation stage, we first construct hyperedge representations based on the induced global hyperedges, and then propose Differential Mixed Aggregation to unify local neighbor differences and global hyperedge-level semantic differences within a single vertex update. Additionally, we design a hyperedge dropout mechanism to enhance the stability and robustness of hypergraph structure learning.

To demonstrate the generality of the proposed framework, we instantiate ViC into two variants: Isotropic ViC, which maintains constant resolution, and Pyramid ViC, designed for multi-scale visual tasks. Experimental results on ImageNet [24] and MS COCO [25] demonstrate that ViC achieves a favorable accuracy-efficiency trade-off compared with other representative visual backbones, as shown in Fig. 1

The main contributions of this paper are as follows:

- We revisit mainstream visual backbones from the perspective of correlation modeling, pointing out that most existing methods primarily capture pairwise correlations, and theoretically demonstrate the irreducible expressive advantage of hypergraph message passing paradigm for high-order correlation modeling.
- We propose a hypergraph-based visual correlation modeling framework that reformulates visual understanding as Correlation Induction–Correlation Propagation, providing a more general paradigm for modeling visual correlations, covering both pairwise and beyond-pairwise correlations.
- Following the hypergraph-based visual correlation modeling framework, we propose a novel visual backbone, Vision Correlator (ViC), which efficiently models both local fine-grained and global high-order semantic correlations.
- Extensive experiments with isotropic and pyramid variants of ViC demonstrate favorable accuracy-efficiency trade-offs on both ImageNet and MS COCO datasets.

2 Related Work

2.1 Visual Backbones with Pairwise Correlation Modeling

Although modern visual backbones differ significantly in their network architectures, from the perspective of correlation modeling, most methods still primarily revolve around pairwise correlations. Convolutional networks [26, 27, 28] perform local pairwise interactions within spatial neighborhoods using convolutional kernels. Vision Transformers [9, 29, 30, 31, 32] partition images into patch tokens and establish global token-token interactions through self-attention. DeiT [33] has further advanced the efficient training of such models on standard visual recognition datasets. Meanwhile, patch-based MLP-like models such as MLP-Mixer [34] and ResMLP [35], although they do not explicitly employ graph structures or attention mechanisms, can still be interpreted as relying primarily on 2-order, *i.e.*, pairwise, cross-token interactions for information exchange, rather than explicitly modeling high-order semantic dependencies across multiple regions.

Closer to the setup of this paper are graph-based visual backbones [10], which more explicitly formulate visual representation learning as correlation modeling on graphs. ViG [15] treats image patches as graph nodes, constructs the graph via K-nearest neighbors, and organizes visual representations through graph message passing. Subsequent work such as GreedyViG [36] further improves graph structure modeling from the perspective of computational efficiency. Although these methods depict visual correlations more explicitly than CNNs or ViTs, their propagation targets remain vertex pairs connected by edges, and their correlation coverage is therefore still fundamentally pairwise. Unlike these works, this paper does not focus on proposing another pairwise interaction unit. Instead, starting from a more fundamental perspective of correlation modeling, we discuss the common limitation that existing visual backbones are generally constrained to pairwise correlation structures, and further explore how to explicitly model high-order semantic correlations at the backbone level.

2.2 Hypergraph Computation

Hypergraphs, as a generalization of graphs, allow each hyperedge to simultaneously connect an arbitrary number of vertices [37, 38]. Consequently, hypergraphs provide a more natural structural tool for high-order correlation modeling than ordinary graphs [39, 40]. HGNN [20] introduced hyperedge convolution into neural networks, explicitly demonstrating that hypergraphs can represent high-order data correlations beyond pairwise interactions. Its extended variant, HGNN+ [21], further shows the flexibility of hypergraph computation in multimodal and multi-view information fusion. UniGNN [41], in turn, advances the integration of graph and hypergraph neural networks from a unified message-passing perspective. These works established the computational foundation for hypergraph neural networks and demonstrated the potential advantages of high-order correlation modeling in structural expressiveness. Following this direction, the computer vision field has begun to gradually adopt hypergraph-based methods [22, 23, 42, 18, 19]. ViHGNN [22] extends pairwise graph relations between image patches to hypergraph relations to model more complex inter-patch correlations. HgVT [23] introduces a hierarchical bipartite hypergraph structure into Vision Transformers, enabling hyperedge features to continuously interact with image patches within the same framework. Overall, these works have demonstrated that extending visual correlations from pairwise edges to hyperedges can effectively enhance the ability of visual models to capture complex semantic structures.

Recent work has also begun to further explore soft correlations and task-oriented applications [43, 44]. For example, SoftHGNN [18] enhances high-order correlation modeling in general visual recognition through learnable soft hyperedges, while YOLOv13 [19] introduces a hypergraph-enhanced mechanism into the real-time object detection framework. However, many existing methods either rely on predefined or task-specific high-order structures, or treat hypergraph computation as an augmentation module embedded within CNNs or Transformers, rather than formulating the visual backbone itself as a general correlation modeling framework. By contrast, this paper does not focus on integrating hypergraph computation as an add-on to an existing framework; instead, it reorganizes the visual backbone itself into a hypergraph-based Correlation Induction–Correlation Propagation framework, enabling adaptive discovery and propagation of both pairwise and beyond-pairwise high-order visual correlations.

3 Theoretical Motivation for High-order Correlation Modeling

Most existing visual backbones, despite their architectural diversity, can be cast as graph-based message passing architectures, and therefore fundamentally operate on pairwise correlations. This naturally raises a basic question: if visual understanding depends on high-order semantic correlations among multiple regions, can such correlations be represented exactly by pairwise graph message passing on the same vertex set? Our answer is negative. In this section, we briefly present the theoretical result that motivates the proposed hypergraph-based correlation modeling framework; the complete formal statement, definitions, and proof are provided in Appendix A.

Theorem 3.1 (Irreducibility of high-order hypergraph message passing)**.** *On a fixed vertex set, there exist nonlinear single-layer hypergraph message passing operators that cannot be represented exactly by any single-layer pairwise graph message passing operator, unless the underlying hypergraph satisfies a highly restrictive structural condition.*

Theorem 3.1 shows that high-order correlation modeling is not merely a stronger parameterization of graph-based relational learning, but has genuinely distinct expressive power. Moreover, in Appendix A, we further show that if all single-layer hypergraph message passing operators on a given hypergraph were exactly representable by pairwise graph message passing, then the hypergraph would have to satisfy a strong local symmetry condition that is extremely restrictive and unlikely to hold in natural settings from a combinatorial perspective. This theoretical gap provides the basis for moving from graph-based pairwise correlation modeling to hypergraph-based high-order correlation modeling. Specifically, at a comparable scale of message passing, graph message passing methods are structurally constrained by pairwise correlations, making their coverage of the correlation space inferior to that enabled by hypergraph message passing methods. Motivated by this observation, we formulate visual understanding as a hypergraph-based Correlation Induction–Correlation Propagation framework, and instantiate it as Vision Correlator in the following section.

4 Methodology

4.1 Overview of Vision Correlator

Given the input feature $X = [x_1, x_2, \ldots, x_N]^\top \in \mathbb{R}^{N\times C}$, where N denotes the number of visual tokens and C denotes the channel dimension, the goal of ViC is to learn effective dependencies among tokens under a unified correlation modeling framework, and thereby produce high-quality visual representations. As shown in Fig. 3, a ViC block consists of two stages. The first stage is **Correlation Induction**. Starting from the input tokens, this stage constructs two types of correlation support sets, namely local and global ones, based on token anchors and semantic anchors, respectively, thereby forming a unified hypergraph-based correlation structure representation. Local correlations mainly serve fine-grained pattern modeling and spatial neighborhood modeling, whereas global correlations focus on cross-region long-range dependencies and high-order semantic collaboration. The second stage is **Correlation Propagation**. On the induced hypergraph structure, ViC first aggregates vertex-related information to form hyperedge-level representations, and then feeds back correlation information at different semantic ranges into vertex representation updates through a unified Differential Mixed Aggregation operator, thereby achieving collaborative modeling of local details and global semantics. Meanwhile, ViC further improves the stability and robustness of structural learning through hyperedge dropout.

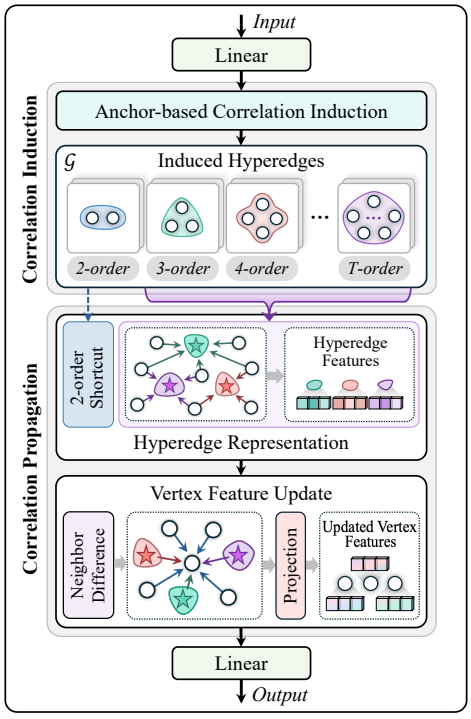

Figure 3: Architecture of the ViC block.

4.2 Correlation Induction

In the correlation induction stage, we design an anchor-based mechanism to induce correlations of different orders among visual tokens at different semantic scales. On the one hand, each token itself is treated as a local anchor, and local dense correlations are induced through neighborhood querying to characterize fine-grained texture details. On the other hand, a set of learnable semantic anchors is introduced to capture cross-region long-range dependencies and higher-level semantic compositions.

Local correlation induction with token anchors. Local correlation induction aims to organize fine-grained neighborhood relations among visual tokens. For each input token, we treat the token itself as a token anchor and establish local dense correlations through nearest-neighbor querying. Specifically, for each vertex x_i, we retrieve its k_{loc} most relevant neighbors in the feature space, denoted by $\mathcal{N}_i = \text{TopK}\left(\text{sim}(x_i, x_j)\right)_{j=1}^{N}$, where $\text{sim}(\cdot,\cdot)$ denotes the vertex similarity function, and k_{loc} is the local neighborhood size. Accordingly, the local correlation set can be obtained as $\mathcal{E}^{\text{loc}} = \{(i,j) \mid j \in \mathcal{N}_i\}$. Such local correlations correspond to 2-order correlations. They can effectively preserve fine-grained patterns such as textures and edges, thereby providing a local geometric foundation for visual representations.

Global correlation induction with semantic anchors. Based on the local topological relations, ViC further induces global high-order semantic correlations through a set of learnable semantic anchors to characterize semantic collaboration across regions and parts. To preserve local structural priors, we first inject topological information into vertex features using a Laplacian operator:

$$\tilde{x}_i = \frac{\alpha}{|\mathcal{N}_i|} \sum_{x_j \in \mathcal{N}_i} x_j + (1-\alpha) x_i. \tag{1}$$

Here, $\alpha \in [0,1]$ controls the balance between the original features and the local topological prior. After obtaining the structure-enhanced vertex representation $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N]^\top \in \mathbb{R}^{N\times C}$, we introduce M learnable semantic anchors, $A = [a_1, a_2, \ldots, a_M]^\top \in \mathbb{R}^{M\times C}$, where each $a_m \in \mathbb{R}^C$ denotes a semantic center. We then generate the global hypergraph structure by leveraging

the correlation between the structure-enhanced vertex representations and the semantic anchors. Specifically, we define the correlation score matrix as:

$$S = \tilde{X}A^\top/\sqrt{C} \in \mathbb{R}^{N\times M}, \tag{2}$$

where S_{im} denotes the correlation score between the i-th token and the m-th semantic anchor. To improve computational efficiency, for each semantic anchor a_m, we retain only the top-k_{glb} scoring tokens as its vertex support set: $\mathcal{V}_m = \mathrm{TopK}\left(S_{im}\right)_{i=1}^{N}$. From this vertex support set, the m-th global hyperedge can be induced as $e_m^{\mathrm{glb}} = \{x_i \mid i \in \mathcal{V}_m\}$. Furthermore, all global hyperedges together constitute the global correlation set: $\mathcal{E}^{\mathrm{glb}} = \{e_m^{\mathrm{glb}}\}_{m=1}^{M}$. Through cCorrelation Induction at different semantic scales, ViC jointly models local fine-grained correlations and global high-order semantic correlations across regions and parts. At the same time, this design keeps the scale of the hypergraph structure controllable, thereby ensuring computational efficiency.

4.3 Correlation Propagation

In the Correlation Propagation stage, ViC updates visual token representations based on the local correlation set $\mathcal{E}^{\mathrm{loc}}$ and the global hyperedge set $\mathcal{E}^{\mathrm{glb}}$ induced in Sec. 4.2. To this end, ViC first constructs hyperedge representations, and then performs message aggregation over multi-order relations in a single node updating step through the proposed Differential Mixed Aggregation. In addition, hyperedge dropout is employed to further improve the stability of structural learning.

4.3.1 Hyperedge Representation

Given the hyperedge set generated in the global correlation induction stage, $\mathcal{E}^{\mathrm{glb}} = \{e_m^{\mathrm{glb}}\}_{m=1}^{M}$, where each hyperedge e_m^{glb} corresponds to a vertex support set $\mathcal{V}_m$ organized by a semantic anchor, we first construct a representation for each high-order hyperedge. Specifically, based on the correlation score S_{im} between vertex x_i and semantic anchor a_m, we define the normalized weight:

$$\omega_{im} = \frac{\exp(S_{im})}{\sum_{j\in\mathcal{V}_m}\exp(S_{jm})}, \qquad i \in \mathcal{V}_m, \tag{3}$$

and obtain the representation of the m-th hyperedge via weighted aggregation:

$$h_m = \sum_{i\in\mathcal{V}_m} \omega_{im} x_i, \tag{4}$$

where $h_m \in \mathbb{R}^C$ denotes the feature representation of this hyperedge. The representations of all global hyperedges are denoted by $H = [h_1, h_2, \ldots, h_M]^\top \in \mathbb{R}^{M\times C}$.

It is worth noting that, for local 2-order correlations, we do not explicitly construct hyperedge representations. Instead, we adopt a 2-order shortcut that directly acts in the vertex updating phase. The key observation here is that the role of local 2-order correlations in ViC is to model fine-grained relative differences between adjacent tokens so as to preserve local texture detail patterns. Therefore, it suffices to focus on the relative difference between the two involved vertices. Accordingly, we consider a class of relative-compatible aggregation functions with respect to the target vertex x_i, *i.e.*, there exists a function Γ such that:

$$\rho(x_i, x_j) - x_i = \Gamma(x_j - x_i). \tag{5}$$

That is, the aggregated feedback message depends only on the difference of the neighboring vertex relative to the current vertex. Within this relative-compatible family, the shortcut is an exact relation-wise reparameterization of the corresponding two-stage propagation. This condition is satisfied by mean aggregation, fixed normalized weighted averaging, and affine aggregation $\rho(x_i, x_j) = Ax_i + Bx_j + b$ with $A + B = I$. Under this condition, the two-phase hyperedge propagation on local 2-order correlations can be strictly rewritten as a direct pairwise shortcut. Specifically, for a local 2-order correlation $(i, j) \in \mathcal{E}^{\mathrm{loc}}$, let the corresponding two-stage propagation be:

$$h_{ij} = \rho(x_i, x_j), \qquad x_i' = \psi(h_{ij} - x_i), \tag{6}$$

where $\rho(\cdot,\cdot)$ denotes the vertex-to-hyperedge aggregation and $\psi(\cdot)$ denotes the hyperedge-to-vertex update mapping. Since $h_{ij} - x_i = \rho(x_i, x_j) - x_i = \Gamma(x_j - x_i)$, substituting this into the above update yields $x_i' = \psi(\Gamma(x_j - x_i))$. Further defining $\psi' = \psi \circ \Gamma$, we obtain $x_i' = \psi'(x_j - x_i)$. Therefore, for local 2-order aggregations satisfying the above relative-compatible condition, explicitly performing the two-stage $V \to E \to V$ propagation is strictly equivalent to directly performing message passing based on the relative difference $x_j - x_i$. Based on this result, ViC constructs explicit hyperedge representations only for high-order global semantic correlations, while incorporating local

2-order correlation into vertex updating via a shortcut, thereby improving computational efficiency while preserving a unified propagation objective.

4.3.2 Vertex Feature Updating via Differential Mixed Aggregation

After obtaining the high-order hyperedge representations H, ViC adopts the proposed Differential Mixed Aggregation to uniformly aggregate messages from correlations of different orders during the vertex updating phase. For each vertex x_i, we denote its local 2-order neighborhood by $\mathcal{N}_i$, and denote the index set of global high-order hyperedges incident to it by $\mathcal{M}_i = \{m \mid x_i \in e_m^{\text{glb}}\}$. The former corresponds to the local pairwise relations provided by the 2-order shortcut, while the latter corresponds to the high-order semantic structures induced by the semantic anchors. To handle these two types of messages within a single propagation operator, we define the unified correlation support set of vertex x_i as:

$$\mathcal{S}_i = \{x_j \mid j \in \mathcal{N}_i\} \cup \{h_m \mid m \in \mathcal{M}_i\}. \tag{7}$$

Based on this support set, ViC adopts a unified vertex updating operator:

$$x_i' = \phi\left(\left[x_i \mid \max_{u \in \mathcal{S}_i}(u - x_i)\right]\right), \tag{8}$$

where $\max(\cdot)$ denotes channel-wise maximization, $[\cdot \mid \cdot]$ denotes feature concatenation, and $\phi(\cdot)$ is a learnable mapping function. In this formulation, the local neighboring vertex x_j provides fine-grained 2-order differential information, while the global high-order hyperedge representation h_m provides multi-region semantic differential information. Both are processed in the same differential aggregation operator, enabling local structural details and global high-order semantics to interact collaboratively within a single vertex update.

4.3.3 Hyperedge Dropout

To enhance the stability of structural learning, ViC further introduces a hyperedge dropout mechanism during training. Given the global hyperedge set $\mathcal{E}^{\text{glb}}$, we randomly mask a subset of vertex–hyperedge incidence relations and perform hyperedge representation construction as well as hyperedge-to-vertex message feedback only based on the retained incidences. Let $z_{im} \in \{0, 1\}$ denote the random retention variable for the incidence relation between vertex x_i and hyperedge e_m^{glb}. Then, during training, the representation of the m-th hyperedge can be written as

$$h_m = \sum_{i \in \mathcal{V}_m} \hat{\omega}_{im} x_i, \qquad \hat{\omega}_{im} = \frac{z_{im} \exp(S_{im})}{\sum_{j \in \mathcal{V}_m} z_{jm} \exp(S_{jm})}. \tag{9}$$

Accordingly, when a vertex receives messages from global hyperedges, aggregation is also performed only over the retained incidence relations. During inference, we remove this random masking process and use the complete hyperedge structure for prediction.

4.4 Instantiations of ViC

As a general implementation of the proposed correlation modeling framework, ViC does not rely on a specific network hierarchy and can be naturally instantiated as different types of visual backbones. We consider two representative instantiations: the **isotropic** instantiation and the **pyramid** instantiation.

Isotropic ViC keeps the feature resolution unchanged throughout the entire network and is suitable for constructing structurally homogeneous visual backbones. Specifically, a single ViC encoder consists of one ViC block followed by a feed-forward network implemented by an MLP. We stack multiple ViC encoders in a serial manner, and adopt a progressively increasing k_{glb} in the ViC blocks of deeper layers, so as to further enrich the modeled order and the diversity of correlations in the deep ViC network. Such a design makes Isotropic ViC structurally comparable to isotropic Transformers, MLP-like backbones, and existing isotropic graph-based backbones, thereby enabling a direct evaluation of the modeling gain brought by the ViC block itself.

For broader visual tasks, especially scenarios such as object detection that require multi-scale representations, pyramid-style backbones are often more advantageous. To this end, we further construct Pyramid ViC. Specifically, Pyramid ViC consists of multiple stages. Each stage contains several ViC encoders, while adjacent stages progressively reduce the spatial resolution through downsampling and increase the channel dimension, thereby forming hierarchical visual representations. In this way, Pyramid ViC extends the correlation modeling capability of ViC to different spatial scales.

5 Experimental Results

5.1 Experimental Setup

Datasets. We evaluate ViC on two widely used benchmarks covering both image classification and object detection. For image classification, we adopt the ImageNet-1K [24] benchmark, which contains approximately 1.2 million training images and 50,000 validation images from 1,000 object categories. Following standard practice, we train all classification models on the training split and report Top-1 and Top-5 accuracy on the validation split. For object detection, we evaluate ViC on the MS COCO [25] benchmark. Following the common setting, we use the `train2017` split for training and report detection performance on the `val2017` split. We adopt the standard COCO evaluation protocol and report bounding box average precision (AP), including mAP that is averaged over IoU thresholds from 0.5 to 0.95, as well as AP_{50} and AP_{75}.

Table 1: Comparison with other isotropic networks on ImageNet-1K.

Method	Params (M)	FLOPs (G)	Top-1	Top-5
ResMLP-S12 conv3x3 [35]	16.7	3.2	77.0	-
ConvMixer-768/32 [45]	21.1	20.9	80.2	-
ConvMixer-1536/20 [45]	51.6	51.4	81.4	-
ViT-B/16 [9]	86.4	55.5	77.9	-
DeiT-Ti [33]	5.7	1.3	72.2	91.1
DeiT-S [33]	22.1	4.6	79.8	95.0
DeiT-B [33]	86.4	17.6	81.8	95.7
ResMLP-S24 [35]	30.0	6.0	79.4	94.5
ResMLP-B24 [35]	116.0	23.0	81.0	95.0
MLP-Mixer-B/16 [34]	59.0	11.7	76.4	-
Isotropic ViG-Ti [15]	7.1	1.3	73.9	92.0
Isotropic ViG-S [15]	22.7	4.5	80.4	95.2
Isotropic ViG-B [15]	86.8	17.7	82.3	95.9
ViHGNN-Ti [22]	8.2	1.8	74.3	92.5
ViHGNN-S [22]	23.2	5.6	81.5	95.7
ViHGNN-B [22]	88.1	19.4	82.9	96.2
HgVT-Mi [23]	5.8	1.4	74.4	-
HgVT-Ti [23]	7.7	1.8	76.2	-
HgVT-S [23]	22.9	5.5	81.2	-
Isotropic ViC-T (Ours)	6.6	1.2	**76.5**	**93.4**
Isotropic ViC-S (Ours)	20.5	4.1	**81.9**	**95.8**
Isotropic ViC-B (Ours)	77.1	16.0	**83.1**	**96.3**

Implementation details. To accommodate different performance requirements and computational resources, our Isotropic ViC and Pyramid ViC each include three variants: Tiny (T), Small (S), and Base (B). For ImageNet classification, following prior works [15, 22], we adopt a data augmentation pipeline including RandAugment [46], Mixup [47], CutMix [48], random erasing [49], and repeated augmentation [50]. The details of the hyperparameter settings are described and shown in Tab. 6 in the Appendix. For MS COCO object detection, we use RetinaNet [51] and Mask R-CNN [52] as the detection frameworks, with our Pyramid ViC serving as the backbone. Following prior works [30, 53, 15, 22], all models are trained under the "1×" schedule. All models in this paper are trained on 8 NVIDIA V100 GPUs.

5.2 Comparative Experiments on ImageNet

Tables 1 and 2 report the comparison results of ViC against isotropic and pyramid visual backbones on ImageNet-1K. Under the isotropic setting, ViC consistently exhibits a superior accuracy–efficiency trade-off across different model scales. In the lightweight regime, ViC-T achieves 76.5% Top-1 accuracy, outperforming ViG-Ti and ViHGNN-Ti by 2.6% and 2.2%, respectively, while reducing the parameter count by 7.6% and 19.5%. ViC-S attains 81.9% Top-1 accuracy, surpassing the larger-scale ViT-B by 4%, while reducing the parameter count by 76.3% and the FLOPs by 92.6%. In the large-model regime, ViC-B achieves 83.1% Top-1 accuracy, still reaching the best accuracy with fewer parameters and lower computational cost.

Under the pyramid setting, Pyramid ViC also maintains a stable advantage. Pyramid ViC-T reaches 79.4% Top-1 accuracy with 9.9M parameters and 1.4G FLOPs, improving over Pyramid ViG-Ti by 1.2% while reducing the parameter count by 7.4% and the FLOPs by 17.6%. Pyramid ViC-S achieves 82.8% Top-1 accuracy, outperforming Pyramid ViG-S at 82.1% and Pyra-

Table 2: Comparison with other pyramid networks on ImageNet-1K.

Model	Params (M)	FLOPs (G)	Top-1	Top-5
ResNet-18 [26]	11.7	1.8	70.6	89.7
ResNet-50 [26]	25.6	4.1	79.8	95.0
ResNet-152 [26]	60.2	11.5	81.8	95.9
PVT-Tiny [30]	13.2	1.9	75.1	-
PVT-Small [30]	24.5	3.8	79.8	-
PVT-Medium [30]	44.2	6.7	81.2	-
PVT-Large [30]	61.4	9.8	81.7	-
PVTv2-B1 [53]	13.1	2.1	78.7	-
PVTv2-B2 [53]	25.4	4.0	82.0	-
PVTv2-B4 [53]	62.6	10.1	83.6	-
CvT-13 [54]	20	4.5	81.6	-
CvT-21 [54]	32	7.1	82.5	-
Swin-T [29]	29	4.5	81.3	95.5
Swin-S [29]	50	8.7	83.0	96.2
Swin-B [29]	88	15.4	83.5	96.5
CycleMLP-B2 [55]	27	3.9	81.6	-
CycleMLP-B3 [55]	38	6.9	82.4	-
CycleMLP-B4 [55]	52	10.1	83.0	-
Poolformer-S12 [56]	12	2.0	77.2	93.5
Poolformer-S36 [56]	31	5.2	81.4	95.5
Poolformer-M48 [56]	73	11.9	82.5	96.0
Pyramid ViG-Ti [15]	10.7	1.7	78.2	94.2
Pyramid ViG-S [15]	27.3	4.6	82.1	96.0
Pyramid ViG-B [15]	92.6	16.8	83.7	96.5
Pyramid ViHGNN-Ti [22]	12.3	2.3	78.9	94.6
Pyramid ViHGNN-S [22]	28.5	6.3	82.5	96.3
Pyramid ViHGNN-B [22]	94.4	18.1	83.9	96.7
HgVT-Mi [23]	5.8	1.4	74.4	-
HgVT-Ti [23]	7.7	1.8	76.2	-
HgVT-S [23]	22.9	5.5	81.2	-
Pyramid ViC-T (Ours)	9.9	1.4	**79.4**	**94.8**
Pyramid ViC-S (Ours)	24.8	3.9	**82.8**	**96.3**
Pyramid ViC-B (Ours)	83.3	14.6	**83.9**	**96.7**

mid ViHGNN-S at 82.5%, while also incurring lower computational cost. For large models, Pyramid ViC-B achieves 83.9% Top-1 accuracy, matching Pyramid ViHGNN-B, while reducing the parameter count and FLOPs by 11.8% and 19.3%, respectively, and surpassing Swin-B at 83.5%. Overall, ViC consistently delivers stable performance gains under both isotropic and pyramid instantiations while substantially reducing model parameters and computational cost.

Furthermore, we visualize the token-to-token similarity matrices in the deep representations of different models, as shown in Fig. 4. In the figure, brighter areas represent higher similarity between tokens, while darker red areas represent lower similarity. It can be observed that the token-level similarity matrices of ViG and ViHGNN are dominated by relatively bright regions overall, indicating that a large number of tokens exhibit high similarity in the deep space and their representations progressively become homogeneous. In contrast, ViC's similarity matrix shows more darker off-diagonal regions, indicating that different tokens maintain lower similarity and stronger discriminability. This phenomenon suggests that ViC can preserve greater feature richness during deep feature propagation, preventing different token representations from collapsing into similar representation spaces, and thus effectively alleviating the common over-smoothing problem [57] in graph and hypergraph neural networks. See the Appendix Sec. E for more discussion on feature richness.

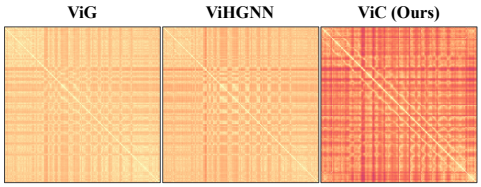

Figure 4: Visualization of token-to-token similarity in the deep layers of different models.

5.3 Comparative Experiments on MS COCO

Table 3 reports the comparison results of different backbones on MS COCO under the Mask R-CNN and RetinaNet frameworks. Overall, Pyramid ViC-S achieves a favorable accuracy–efficiency trade-off under both detection frameworks. Under the Mask R-CNN framework, Pyramid ViC-S attains 43.7% mAP, outperforming Pyramid ViG-S at 42.6% mAP and Pyramid ViHGNN-S at 43.1% mAP, while reducing the parameter count by 5.5% and 8.5%, and the FLOPs by 3.1% and 4.0%, respectively. Under the RetinaNet framework, Pyramid ViC-S achieves 43.0% mAP, surpassing all other methods. Compared with Pyramid ViG-S, it improves mAP by 1.2% while reducing the parameter count and FLOPs by 6.9% and 3.3%, respectively; compared with Pyramid ViHGNN-S, Pyramid ViC-S further improves mAP by 0.8%, while reducing the parameter count and FLOPs by 11.1% and 4.8%, respectively. These results indicate that Pyramid ViC can consistently improve downstream object detection performance while maintaining lower model complexity.

Table 3: Comparison of different backbones under the Mask R-CNN and RetinaNet frameworks.

Backbone	**Mask R-CNN**					**RetinaNet**				
	Params (M)	**FLOPs (G)**	**mAP**	**AP$_{50}$**	**AP$_{75}$**	**Params (M)**	**FLOPs (G)**	**mAP**	**AP$_{50}$**	**AP$_{75}$**
ResNet-50 [26]	44.2	260.1	38.0	58.6	41.4	37.7	239.3	36.3	55.3	38.6
PVT-Small [30]	44.1	245.1	40.4	62.9	43.8	56.4	319.0	39.9	59.6	42.7
CycleMLP-B2 [55]	46.5	249.5	42.1	64.0	45.7	34.2	226.5	40.4	61.3	44.2
PoolFormer-S24 [56]	41.0	–	40.1	62.2	43.4	36.5	230.9	40.6	61.4	43.2
Swin-T [29]	47.8	264.0	42.2	64.6	46.2	38.5	244.8	41.5	62.1	44.2
Pyramid ViG-S [15]	45.8	258.8	42.6	65.2	46.0	36.2	240.0	41.8	63.1	44.7
Pyramid ViHGNN-S [22]	47.3	261.4	43.1	66.0	46.5	37.9	243.7	42.2	63.8	45.1
Pyramid ViC-S (Ours)	43.3	250.9	**43.7**	**66.1**	**48.1**	33.7	232.1	**43.0**	**63.8**	**46.3**

5.4 Ablation Study

Ablation on the main components of ViC. Table 4 reports the ablation results on the main components of ViC. When only local correlations are used, the model achieves 73.8% Top-1 accuracy. Using only global correlations improves the performance to 75.6%, showing that global high-order semantic correlations already provide stronger discriminative cues than local pairwise relations alone. Combining local and global correlations further boosts the accuracy to 76.4%, which indicates that the two branches are complementary: the local branch captures fine-grained structural details, while the global branch models cross-region semantic collaboration. Finally, adding hyperedge dropout further improves the Top-1 accuracy to 76.5% without introducing additional parameters or FLOPs, confirming that it effectively enhances the stability of hypergraph structure learning.

Comparison of different hypergraph operators. Table 5 further compares the proposed Differential Mixed Aggregation (DMA) with several representative hypergraph operators. Replacing DMA with HGConv [20], HGConv+, *etc*. consistently leads to inferior results, with Top-1 accuracy ranging from 73.8% to 74.3%. By contrast, our DMA achieves the best accuracy while also requiring the fewest parameters and the lowest computational cost. We attribute this to their different propagation biases: conventional operators mainly rely on hyperedge-level smoothing, which behaves like low-pass filtering and may wash out informative local details. In contrast, DMA introduces differential aggregation, which better preserves discriminative variations while incorporating global high-order correlations. This leads to the best accuracy-efficiency trade-off among all compared operators.

Table 4: Ablation on the main components of ViC. "Loc.": local correlations; "Glb.": global correlations; "Hy. Dropout": hyperedge dropout.

Loc.	Glb.	Hy. Dropout	Top-1	Params (M)	FLOPs (G)
✓			73.8	6.57	1.16
	✓		75.6	6.62	1.09
✓	✓		76.4	6.62	1.18
✓	✓	✓	**76.5**	6.62	1.18

Table 5: Comparison of using different hypergraph operators.

Operator	Top-1	Params (M)	FLOPs (G)
HGConv [20]	73.9	7.29	1.33
HGConv+ [21]	74.1	7.74	1.34
HyperGCN [58]	73.8	6.84	1.31
HyperAttn [59]	74.3	8.20	2.12
DMA (Ours)	**76.5**	**6.62**	**1.18**

5.5 Qualitative Visualizations

Furthermore, we visualize the correlations modeled by our ViC, as shown in Fig. 5. In the figure, column e) shows all the correlations induced within a ViC block. The column b), c), and d) display several sets of different global high-order relations. Notably, we observe that different hyperedges spontaneously evolves into different roles in correlation modeling. They can be roughly grouped into three types: the hyperedges in b) mainly focus on high-order correlations among various regions or parts of the foreground object; those in c) mainly focus on high-order correlations among different background regions; and those in d) mainly focus on cooperative correlations between foreground and background regions. The rich correlations captured together contribute to ViC's superior visual understanding ability.

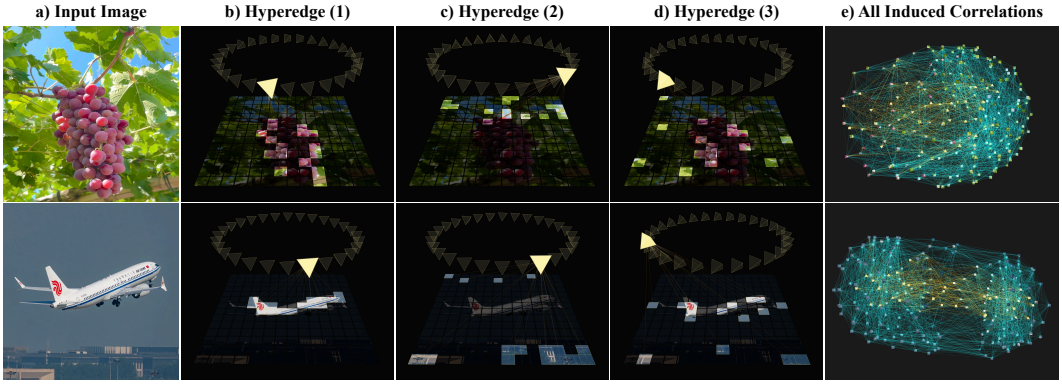

Figure 5: Visualization of the correlations modeled by our ViC. The yellow tetrahedrons are the semantic anchors, and the yellow lines extending from it represent the global correlations induced by semantic anchors; the blue lines represent the local correlations induced by the token anchors.

6 Conclusion

In this paper, we revisit visual backbones from a correlation modeling perspective and identify the limited coverage of pairwise graph message passing. We further show that hypergraph message passing provides irreducible expressive power for high-order correlation modeling, motivating a hypergraph-based Correlation Induction–Correlation Propagation framework for visual understanding. We implement this framework as Vision Correlator (ViC), an efficient implementation that induces local and global high-order correlations, and propagates them via Differential Mixed Aggregation. Extensive experiments on image classification and object detection demonstrate that ViC achieves a strong accuracy-efficiency trade-off against competitive baselines. We hope this work offers a useful step toward more expressive visual backbones with explicit high-order correlation modeling.

Acknowledgments and Disclosure of Funding

This work was supported by the Brain Science and Brain-like Intelligence Technology–National Science and Technology Major Project under Grant No. 2025ZD0217300, the National Natural Science Foundation of China under Grant Nos. U25A20532, 62501358, U24A20252, 62327808, and 12531019, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant No. JYB2025XDXM504, and the Guangdong Major Project of Basic and Applied Basic Research under Grant No. 2023B0303000009.

References

[1] Omar Elharrouss, Younes Akbari, Noor Almadeed, and Somaya Al-Maadeed. Backbones-review: Feature extractor networks for deep learning and deep reinforcement learning approaches in computer vision. *Comput. Sci. Rev.*, 53:100645, 2024.

[2] Xiangtai Li, Henghui Ding, Haobo Yuan, Wenwei Zhang, Jiangmiao Pang, Guangliang Cheng, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):10138–10163, 2024.

[3] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12):8927–8948, 2022.

[4] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(11):13467–13488, 2023.

[5] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceed. IEEE*, 111(3):257–276, 2023.

[6] Mengqi Lei, Shuokun Cheng, Wei Bao, Shaoyi Du, Jun-Hai Yong, Siqi Li, and Yue Gao. Count anything. *arXiv preprint arXiv:2605.30846*, 2026.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with Transformers. In *Eur. Conf. Comput. Vis.*, pages 213–229, 2020.

[8] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neur. Net. Learn. Sys.*, 33(12):6999–7019, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16×16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2020.

[10] 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 Trans. Pattern Anal. Mach. Intell.*, 46(12):10297–10318, 2024.

[11] Dominique Beaini, Saro Passaro, Vincent Létourneau, Will Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. pages 748–758, 2021.

[12] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. SplineCNN: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 869–877, 2018.

[13] Chaitanya K Joshi. Transformers are graph neural networks. *arXiv preprint arXiv:2506.22084*, 2025.

[14] Chen Cai, Truong Son Hy, Rose Yu, and Yusu Wang. On the connection between mpnn and graph Transformer. pages 3408–3430, 2023.

[15] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision GNN: An image is worth graph of nodes. *Adv. Neural Inform. Process. Syst.*, 35:8291–8303, 2022.

[16] Jiajun Hong, Jianan Wei, and Wenguan Wang. Learning human-object interaction as groups. In *Adv. Neural Inform. Process. Syst.*, 2025.

[17] Trong-Thuan Nguyen, Pha Nguyen, Jackson Cothren, Alper Yilmaz, and Khoa Luu. HyperGLM: Hypergraph for video scene graph generation and anticipation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 29150–29160, 2025.

[18] Mengqi Lei, Yihong Wu, Siqi Li, Xinhu Zheng, Juan Wang, Yue Gao, and Shaoyi Du. SoftHGNN: Soft hypergraph neural networks for general visual recognition. *arXiv preprint arXiv:2505.15325*, 2025.

[19] Mengqi Lei, Siqi Li, Yihong Wu, Han Hu, You Zhou, Xinhu Zheng, Guiguang Ding, Shaoyi Du, Zongze Wu, and Yue Gao. YOLOv13: Real-time object detection with hypergraph-enhanced adaptive visual perception. *arXiv preprint arXiv:2506.17733*, 2025.

[20] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI*, pages 3558–3565, 2019.

[21] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. HGNN+: General hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3181–3199, 2022.

[22] Yan Han, Peihao Wang, Souvik Kundu, Ying Ding, and Zhangyang Wang. Vision HGNN: An image is more than a graph of nodes. In *Int. Conf. Comput. Vis.*, pages 19878–19888, 2023.

[23] Joshua Fixelle. Hypergraph Vision Transformers: Images are more than nodes, more than edges. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9751–9761, 2025.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[25] 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 *Eur. Conf. Comput. Vis.*, pages 740–755, 2014.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[27] Yuhong Li, Xiaofan Zhang, and Deming Chen. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1091–1100, 2018.

[28] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[29] 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 *Int. Conf. Comput. Vis.*, pages 10012–10022, 2021.

[30] 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 *Int. Conf. Comput. Vis.*, pages 568–578, 2021.

[31] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *Int. Conf. Learn. Represent.*, pages 138846–138923, 2026.

[32] Yufei Guo, Yinan Wu, Haoran Duan, Guiguang Ding, and Jungong Han. PhaseLoRA: Control-regime-conditioned low-rank adaptation for continuous-action vision-language-action policies. *arXiv preprint arXiv:2608.15285*, 2026.

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image Transformers & distillation through attention. pages 10347–10357, 2021.

[34] 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. *Adv. Neural Inform. Process. Syst.*, 34:24261–24272, 2021.

[35] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. ResMLP: Feedforward networks for image classification with data-efficient training. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):5314–5321, 2022.

[36] Mustafa Munir, William Avery, Md Mostafijur Rahman, and Radu Marculescu. GreedyViG: Dynamic axial graph construction for efficient vision gnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6118–6127, 2024.

[37] Yue Gao, Shuyi Ji, Xiangmin Han, and Qionghai Dai. Hypergraph computation. *Engineering*, 2024.

[38] Mengqi Lei, Guohuan Xie, Shihui Ying, Shaoyi Du, Jun-Hai Yong, Chuan Shi, Ling Tian, Siqi Li, and Yue Gao. Hypergraph as language. *arXiv preprint arXiv:2605.21858*, 2026.

[39] Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Comp. Surv.*, 56(1):1–38, 2023.

[40] Sunwoo Kim, Soo Yong Lee, Yue Gao, Alessia Antelmi, Mirko Polato, and Kijung Shin. A survey on hypergraph neural networks: An in-depth and step-by-step guide. In *ACM SIGKDD*, pages 6534–6544, 2024.

[41] Jing Huang and Jie Yang. UniGNN: A unified framework for graph and hypergraph neural networks. In *IJCAI*, pages 2563–2569, 2021.

[42] Linfei Li, Pei Tan, Siqi Li, Changqing Zou, and Yue Gao. Hyper-PCN: Hypergraph-based point cloud completion via high-order correlation modeling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 39121–39130, 2026.

[43] Leo Thomas Ramos and Angel D Sappa. A decade of you only look once (YOLO) for object detection: A review. *IEEE Access*, 13:192747–192794, 2025.

[44] Siqi Li, Yongji Zhang, Wei Bao, and Yue Gao. Event-based facial expression recognition via large vision-language models. *Pattern Recognition*, page 113300, 2026.

[45] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

[46] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, pages 702–703, 2020.

[47] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. MixUp: Beyond empirical risk minimization. In *Int. Conf. Learn. Represent.*, 2018.

[48] 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 *Int. Conf. Comput. Vis.*, pages 6023–6032, 2019.

[49] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, volume 34, pages 13001–13008, 2020.

[50] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8129–8138, 2020.

[51] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Int. Conf. Comput. Vis.*, pages 2980–2988, 2017.

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Int. Conf. Comput. Vis.*, pages 2961–2969, 2017.

[53] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT v2: Improved baselines with pyramid vision transformer. *Comput. Visual Media*, 8(3):415–424, 2022.

[54] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CVT: Introducing convolutions to vision Transformers. In *Int. Conf. Comput. Vis.*, pages 22–31, 2021.

[55] Shoufa Chen, Enze Xie, Chongjian Ge, Runjian Chen, Ding Liang, and Ping Luo. CycleMLP: A MLP-like architecture for dense visual predictions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12):14284–14300, 2023.

[56] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10819–10829, 2022.

[57] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. DeepGCNs: Can GCNs go as deep as CNNs? In *Int. Conf. Comput. Vis.*, pages 9267–9276, 2019.

[58] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method for training graph convolutional networks on hypergraphs. *Adv. Neural Inform. Process. Syst.*, 32, 2019.

[59] Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

[60] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2019.

[61] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are AllSet: A multiset function framework for hypergraph neural networks. In *Int. Conf. Learn. Represent.*, 2022.

[62] Richard Forsyth. Zoo. UCI Machine Learning Repository, 1990.

[63] Yihe Dong, Will Sawin, and Yoshua Bengio. HNHN: Hypergraph networks with hyperedge neurons. *arXiv preprint arXiv:2006.12278*, 2020.

[64] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *Int. World Wide Web Conf.*, pages 2022–2032, 2019.

Appendix

Contents

A Proof of the Irreducibility Result **15**

A.1 Formal setup 15

A.2 A strict separation theorem 16

A.3 A necessary structural condition for universal graph representability 19

A.4 Why the necessary condition is highly restrictive 20

A.5 Connection to Differential Mixed Aggregation 20

B More Details of Hypergraph Construction and Propagation **21**

C Implementation Details **21**

D Additional Visual Evaluations **22**

E Additional Ablation Results **23**

F Model Efficiency **25**

G Extension to Hypergraph Benchmarks **25**

H Discussion **27**

A Proof of the Irreducibility Result

In this section, we provide a rigorous formalization and complete proof of the irreducibility result stated in the main text. Throughout, we compare two classes of message passing layers under the following setting: both are defined on the *same finite vertex set* V; the graph layer is restricted to a single-layer pairwise message passing scheme; no auxiliary vertices are introduced; and no additional edge attributes are allowed to explicitly encode the full hyperedge structure.

To remain consistent with Sec. 3, we keep C as the input feature dimension. Moreover, since the main text uses $H \in \mathbb{R}^{M\times C}$ to denote the matrix of global hyperedge representations, we denote the underlying hypergraph in this appendix by $\mathcal{K} = (V, \mathcal{E})$ instead of H.

A.1 Formal setup

Let V be a finite vertex set, let $C \in \mathbb{N}$ denote the input feature dimension, and let $D \in \mathbb{N}$ denote a generic output feature dimension. For each $v \in V$, let $x_v \in \mathbb{R}^C$ be its feature vector, and write $X = (x_v)_{v\in V} \in (\mathbb{R}^C)^V$.

Definition A.1 (Hypergraph, local residual family, and local support set)**.** Let $\mathcal{K} = (V, \mathcal{E})$ be a finite simple hypergraph, where "simple" means that $\mathcal{E}$ contains no repeated hyperedges. For each vertex $v \in V$, define its *local residual family* by

$$\mathcal{F}_v := \{\, e \setminus \{v\} : e \in \mathcal{E},\ v \in e \,\} \subseteq 2^{V\setminus\{v\}}, \tag{10}$$

and its *local support set* by

$$U_v := \bigcup_{A\in\mathcal{F}_v} A \subseteq V \setminus \{v\}. \tag{11}$$

Definition A.2 (Single-layer pairwise graph message passing layer)**.** A mapping

$$\mathcal{G} : (\mathbb{R}^C)^V \to (\mathbb{R}^D)^V \tag{12}$$

is called a *single-layer pairwise graph message passing layer* on the vertex set V if there exist a simple undirected graph $G = (V, E)$, a positive integer r, and functions

$$\mu_G : \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}^r, \qquad \eta_G : \mathbb{R}^C \times \mathbb{R}^r \to \mathbb{R}^D, \tag{13}$$

such that for every $v \in V$,

$$(\mathcal{G}(X))_v = \eta_G\left(x_v, \sum_{u \in N_G(v)} \mu_G(x_v, x_u)\right), \tag{14}$$

where $N_G(v)$ denotes the neighbor set of v in G.

Definition A.3 (Single-layer hypergraph message passing layer)**.** Let $\mathrm{Mult}_f(\mathbb{R}^C)$ denote the collection of all finite multisets over $\mathbb{R}^C$, and let $\{\{z_1, \ldots, z_k\}\}$ denote the multiset formed by $z_1, \ldots, z_k$. A mapping

$$\mathcal{T} : (\mathbb{R}^C)^V \to (\mathbb{R}^D)^V \tag{15}$$

is called a *single-layer hypergraph message passing layer* on $\mathcal{K} = (V, \mathcal{E})$ if there exist functions

$$\mu_{\mathcal{K}} : \mathbb{R}^C \times \mathrm{Mult}_f(\mathbb{R}^C) \to \mathbb{R}^D, \qquad \eta_{\mathcal{K}} : \mathbb{R}^C \times \mathbb{R}^D \to \mathbb{R}^D, \tag{16}$$

where $\mu_{\mathcal{K}}$ is permutation-invariant in its multiset argument, such that for every $v \in V$,

$$(\mathcal{T}(X))_v = \eta_{\mathcal{K}}\left(x_v, \sum_{e \in \mathcal{E}:\, v \in e} \mu_{\mathcal{K}}\big(x_v, \{\{x_u : u \in e \setminus \{v\}\}\}\big)\right). \tag{17}$$

Definition A.4 (Exact representation)**.** A hypergraph layer $\mathcal{T}$ defined on $\mathcal{K}$ is said to be *exactly represented* by a graph layer if there exists a single-layer pairwise graph message passing layer $\mathcal{G}$ defined on the same vertex set V such that

$$\mathcal{G}(X) = \mathcal{T}(X), \qquad \forall\, X \in (\mathbb{R}^C)^V. \tag{18}$$

Here, "exact" means equality as functions, rather than approximation.

Lemma A.5 (Local multiset invariance of graph layers)**.** *Let $\mathcal{G}$ be a single-layer pairwise graph message passing layer defined on a graph $G = (V, E)$. If $X, Y \in (\mathbb{R}^C)^V$ satisfy*

$$\begin{aligned} x_v &= y_v, \\ \{\{x_u : u \in N_G(v)\}\} &= \{\{y_u : u \in N_G(v)\}\}, \end{aligned} \tag{19}$$

then

$$(\mathcal{G}(X))_v = (\mathcal{G}(Y))_v. \tag{20}$$

Proof. By Definition A.2,

$$\begin{aligned} (\mathcal{G}(X))_v &= \eta_G\left(x_v, \sum_{u \in N_G(v)} \mu_G(x_v, x_u)\right), \\ (\mathcal{G}(Y))_v &= \eta_G\left(y_v, \sum_{u \in N_G(v)} \mu_G(y_v, y_u)\right). \end{aligned} \tag{21}$$

Since $x_v = y_v$, the first argument of both η_G and μ_G is the same in the two expressions. Moreover,

$$\{\{x_u : u \in N_G(v)\}\} = \{\{y_u : u \in N_G(v)\}\}, \tag{22}$$

so the summands are the same up to permutation. Hence,

$$\sum_{u \in N_G(v)} \mu_G(x_v, x_u) = \sum_{u \in N_G(v)} \mu_G(y_v, y_u). \tag{23}$$

Substituting this into η_G yields $(\mathcal{G}(X))_v = (\mathcal{G}(Y))_v$. □

A.2 A strict separation theorem

Theorem A.6 (Strict separation under local equal-cardinality mismatch)**.** *Let $\mathcal{K} = (V, \mathcal{E})$ be a finite simple hypergraph, and let $C \geq 1$. Suppose there exist a vertex $v \in V$ and two sets $P, Q \subseteq U_v$ such that*

$$|P| = |Q|, \qquad P \in \mathcal{F}_v, \qquad Q \notin \mathcal{F}_v. \tag{24}$$

Then there exists a nonlinear *single-layer hypergraph message passing layer*

$$\mathcal{T} : (\mathbb{R}^C)^V \to (\mathbb{R}^2)^V \tag{25}$$

defined on $\mathcal{K}$ that cannot be exactly represented by any single-layer pairwise graph message passing layer on the same vertex set V.

Proof. Let

$$e_1 = (1, 0, \ldots, 0) \in \mathbb{R}^C, \tag{26}$$

and define the linear functional

$$\ell : \mathbb{R}^C \to \mathbb{R}, \qquad \ell(z) := z_1, \tag{27}$$

that is, the projection onto the first coordinate. Define

$$\widehat{\mu} : \mathbb{R}^C \times \mathrm{Mult}_f(\mathbb{R}^C) \to \mathbb{R}^2, \qquad \widehat{\mu}(x, \mathcal{B}) := \left(\prod_{z \in \mathcal{B}} \ell(z),\ \sum_{z \in \mathcal{B}} \ell(z) \right), \tag{28}$$

and

$$\widehat{\eta} : \mathbb{R}^C \times \mathbb{R}^2 \to \mathbb{R}^2, \qquad \widehat{\eta}(x, s) := s. \tag{29}$$

Here the empty product is defined as 1, and the empty sum as 0. By Definition A.3, these functions define a hypergraph layer

$$(\mathcal{T}(X))_u = \sum_{e \in \mathcal{E}:\, u \in e} \left(\prod_{w \in e \setminus \{u\}} \ell(x_w),\ \sum_{w \in e \setminus \{u\}} \ell(x_w) \right). \tag{30}$$

We first note that $\mathcal{T}$ is nonlinear. Indeed, the assumption of the theorem implies that $\mathcal{F}_v$ cannot be determined solely by cardinality. In particular, $\mathcal{F}_v$ cannot consist only of subsets of size at most one. Otherwise, by the definition of U_v, every singleton $\{u\}$ with $u \in U_v$ would belong to $\mathcal{F}_v$, while the only 0-subset is $\emptyset$; hence membership in $\mathcal{F}_v$ would depend only on cardinality, contradicting the existence of $P, Q \subseteq U_v$ with $|P| = |Q|$ but different membership statuses. Therefore $\mathcal{F}_v$ contains at least one set of size at least 2, so the first coordinate of $\widehat{\mu}$ contains a genuine multiplicative interaction term. Thus the constructed layer is nonlinear.

We now prove that no single-layer pairwise graph layer can exactly represent $\mathcal{T}$. Assume, for contradiction, that there exists a single-layer pairwise graph message passing layer

$$\mathcal{G} : (\mathbb{R}^C)^V \to (\mathbb{R}^2)^V \tag{31}$$

that exactly represents $\mathcal{T}$. Then, by Definition A.2, there exist a simple graph $G = (V, E)$, a positive integer r, and functions

$$\mu_G : \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}^r, \qquad \eta_G : \mathbb{R}^C \times \mathbb{R}^r \to \mathbb{R}^2 \tag{32}$$

such that

$$\mathcal{G}(X) = \mathcal{T}(X), \qquad \forall\, X \in (\mathbb{R}^C)^V. \tag{33}$$

For a vector $(\mathcal{T}(X))_v \in \mathbb{R}^2$, we denote its first and second coordinates by $(\mathcal{T}(X))_{v,1}$ and $(\mathcal{T}(X))_{v,2}$, respectively.

Step 1: $U_v \subseteq N_G(v)$. We claim that

$$U_v \subseteq N_G(v). \tag{34}$$

Suppose otherwise. Then there exists some $w \in U_v \setminus N_G(v)$. Since $w \in U_v$, by Definition A.1 there is at least one set $R \in \mathcal{F}_v$ such that $w \in R$. Define

$$d_v(w) := \big|\{\, R \in \mathcal{F}_v : w \in R \,\}\big|. \tag{35}$$

Then $d_v(w) \geq 1$.

Now construct two inputs $X, X' \in (\mathbb{R}^C)^V$ by setting

$$x_v = x'_v = \mathbf{0}, \qquad x_w = e_1, \qquad x'_w = 2e_1, \tag{36}$$

and, for every $u \in V \setminus \{v, w\}$,

$$x_u = x'_u = \mathbf{0}. \tag{37}$$

Since $w \notin N_G(v)$ and all other neighbors of v have the same features under X and X', we have

$$\begin{aligned} x_v &= x'_v, \\ \{\{x_u : u \in N_G(v)\}\} &= \{\{x'_u : u \in N_G(v)\}\}. \end{aligned} \tag{38}$$

By Lemma A.5, $(\mathcal{G}(X))_v = (\mathcal{G}(X'))_v$. By exact representation, this implies

$$(\mathcal{T}(X))_v = (\mathcal{T}(X'))_v. \tag{39}$$

However,

$$\begin{aligned} (\mathcal{T}(X))_{v,2} &= \sum_{R \in \mathcal{F}_v} \sum_{u \in R} \ell(x_u) = d_v(w), \\ (\mathcal{T}(X'))_{v,2} &= \sum_{R \in \mathcal{F}_v} \sum_{u \in R} \ell(x'_u) = 2d_v(w). \end{aligned} \tag{40}$$

Since $d_v(w) \geq 1$, these two quantities are different, yielding a contradiction. Hence $U_v \subseteq N_G(v)$.

Step 2: contradiction via a permutation argument. Since $|P| = |Q|$, there exists a bijection $\pi_0 : P \to Q$. Extend it to a permutation $\pi : U_v \to U_v$ such that $\pi(P) = Q$. Then

$$\pi(\mathcal{F}_v) \neq \mathcal{F}_v, \tag{41}$$

because $P \in \mathcal{F}_v$ while $Q = \pi(P) \notin \mathcal{F}_v$. Equivalently,

$$\pi^{-1}(\mathcal{F}_v) \neq \mathcal{F}_v. \tag{42}$$

Choose an injection $\kappa : U_v \to \{0, 1, \ldots, |U_v| - 1\}$, and define, for each $u \in U_v$,

$$a_u := 2^{2^{\kappa(u)}}. \tag{43}$$

Now define an input $X \in (\mathbb{R}^C)^V$ by

$$x_u = \begin{cases} a_u e_1, & u \in U_v, \\ \mathbf{0}, & u \notin U_v. \end{cases} \tag{44}$$

Extend π to a permutation of the whole vertex set V by letting it act as the identity on $V \setminus U_v$, and define another input $Y \in (\mathbb{R}^C)^V$ by

$$y_u := x_{\pi^{-1}(u)}, \qquad \forall\, u \in V. \tag{45}$$

Because $U_v \subseteq N_G(v)$ and all vertices in $N_G(v) \setminus U_v$ have zero features under both X and Y, we have

$$\begin{aligned} x_v &= y_v, \\ \{\{x_u : u \in N_G(v)\}\} &= \{\{y_u : u \in N_G(v)\}\}. \end{aligned} \tag{46}$$

Lemma A.5 therefore gives $(\mathcal{G}(X))_v = (\mathcal{G}(Y))_v$. By exact representation, this implies

$$(\mathcal{T}(X))_v = (\mathcal{T}(Y))_v. \tag{47}$$

We now show that this is impossible.

First,

$$(\mathcal{T}(X))_{v,1} = \sum_{R \in \mathcal{F}_v} \prod_{u \in R} a_u = \sum_{R \in \mathcal{F}_v} 2^{\sum_{u \in R} 2^{\kappa(u)}}. \tag{48}$$

Second,

$$\begin{aligned} (\mathcal{T}(Y))_{v,1} &= \sum_{R \in \mathcal{F}_v} \prod_{u \in R} \ell(y_u) = \sum_{R \in \mathcal{F}_v} \prod_{u \in R} a_{\pi^{-1}(u)} \\ &= \sum_{R' \in \pi^{-1}(\mathcal{F}_v)} 2^{\sum_{u \in R'} 2^{\kappa(u)}}. \end{aligned} \tag{49}$$

Now consider the map $R \longmapsto \sum_{u \in R} 2^{\kappa(u)}$. This map is injective on 2^{U_v}, because different subsets have different binary encodings. Hence the quantities $2^{\sum_{u \in R} 2^{\kappa(u)}}$ are distinct powers of two for different $R \subseteq U_v$, and their sums uniquely encode the underlying subset families. Since $\pi^{-1}(\mathcal{F}_v) \neq \mathcal{F}_v$, it follows that

$$(\mathcal{T}(X))_{v,1} \neq (\mathcal{T}(Y))_{v,1}, \tag{50}$$

and therefore $(\mathcal{T}(X))_v \neq (\mathcal{T}(Y))_v$. This contradicts the equality derived above. Hence no single-layer pairwise graph message passing layer can exactly represent the constructed $\mathcal{T}$. □

Corollary A.7 (Existence of nonlinear hypergraph layers that are not graph-representable)**.** *Under the constraints of a fixed vertex set, no auxiliary vertices, and single-layer pairwise graph message passing, there exist nonlinear hypergraph message passing layers that cannot be exactly represented by any graph message passing layer.*

Proof. Take

$$V = \{v, a, b, c\}, \qquad \mathcal{E} = \big\{\{v, a, b\}, \{v, a, c\}\big\}. \tag{51}$$

Then

$$\mathcal{F}_v = \big\{\{a, b\}, \{a, c\}\big\}, \qquad U_v = \{a, b, c\}. \tag{52}$$

Let

$$P = \{a, b\} \in \mathcal{F}_v, \qquad Q = \{b, c\} \notin \mathcal{F}_v. \tag{53}$$

Clearly, $|P| = |Q| = 2$. Therefore Theorem A.6 applies, and there exists a nonlinear single-layer hypergraph message passing layer on this hypergraph that cannot be exactly represented by any single-layer pairwise graph message passing layer on the same vertex set. □

A.3 A necessary structural condition for universal graph representability

Lemma A.8 (Equivalent characterizations of cardinality-only families)**.** *Let U be a finite set and let $\mathcal{F} \subseteq 2^U$. Then the following are equivalent:*

1. *For every permutation $\pi : U \to U$, one has $\pi(\mathcal{F}) = \mathcal{F}$.*
2. *For any $C_1, C_2 \subseteq U$, if $|C_1| = |C_2|$, then $C_1 \in \mathcal{F} \iff C_2 \in \mathcal{F}$.*
3. *There exists an index set $R \subseteq \{0, 1, \ldots, |U|\}$ such that*
$$\mathcal{F} = \bigcup_{r \in R} \binom{U}{r}, \tag{54}$$
where $\binom{U}{r}$ denotes the family of all r-subsets of U.

Proof. We first prove $(1) \Rightarrow (2)$. If $|C_1| = |C_2|$, then there exists a permutation $\pi : U \to U$ such that $\pi(C_1) = C_2$. If $C_1 \in \mathcal{F}$, then by (1), $C_2 = \pi(C_1) \in \mathcal{F}$. The converse follows by symmetry, so (2) holds.

Next we prove $(2) \Rightarrow (3)$. Define
$$R := \{\, r \in \{0, 1, \ldots, |U|\} : \exists\, C \subseteq U,\ |C| = r,\ C \in \mathcal{F} \,\}. \tag{55}$$
If $C \in \mathcal{F}$, then $|C| \in R$, and hence $C \in \bigcup_{r \in R} \binom{U}{r}$. Conversely, if $C \in \binom{U}{r}$ with $r \in R$, then there exists $D \in \mathcal{F}$ such that $|D| = r = |C|$. By (2), this implies $C \in \mathcal{F}$. Therefore,
$$\mathcal{F} = \bigcup_{r \in R} \binom{U}{r}. \tag{56}$$

Finally, we prove $(3) \Rightarrow (1)$. Any permutation preserves subset cardinality, and thus for every r, $\pi\Big(\binom{U}{r}\Big) = \binom{U}{r}$. Hence,
$$\begin{aligned} \pi(\mathcal{F}) &= \pi\left(\bigcup_{r \in R} \binom{U}{r}\right) = \bigcup_{r \in R} \pi\left(\binom{U}{r}\right) \\ &= \bigcup_{r \in R} \binom{U}{r} = \mathcal{F}. \end{aligned} \tag{57}$$
Thus the three conditions are equivalent. □

Definition A.9 (Local complete symmetry)**.** Let $\mathcal{K} = (V, \mathcal{E})$ be a finite simple hypergraph. If, for some vertex $v \in V$, the local residual family $\mathcal{F}_v$ satisfies any of the equivalent conditions in Lemma A.8, then we say that $\mathcal{F}_v$ has *local complete symmetry*. If this holds for every $v \in V$, then we say that the hypergraph $\mathcal{K}$ has *local complete symmetry*.

Equivalently, $\mathcal{F}_v$ has local complete symmetry if and only if, after fixing v, whether a subset $B \subseteq U_v$ belongs to $\mathcal{F}_v$ depends only on its cardinality $|B|$, and not on the identities of the elements in B.

Theorem A.10 (Necessary structural condition for universal graph representability)**.** *Let $\mathcal{K} = (V, \mathcal{E})$ be a finite simple hypergraph, and let $C \geq 1$. If every single-layer hypergraph message passing layer $\mathcal{T} : (\mathbb{R}^C)^V \to (\mathbb{R}^2)^V$ defined on $\mathcal{K}$ can be exactly represented by some single-layer pairwise graph message passing layer on the same vertex set V, then $\mathcal{K}$ must have local complete symmetry. Equivalently, for every $v \in V$, there exists a set $R_v \subseteq \{0, 1, \ldots, |U_v|\}$ such that*
$$\mathcal{F}_v = \bigcup_{r \in R_v} \binom{U_v}{r}. \tag{58}$$

Proof. We argue by contradiction. Suppose the conclusion were false. Then there exists some vertex $v \in V$ such that $\mathcal{F}_v$ does not have local complete symmetry. By Lemma A.8, there exist two sets $P, Q \subseteq U_v$ such that
$$|P| = |Q|, \qquad P \in \mathcal{F}_v, \qquad Q \notin \mathcal{F}_v. \tag{59}$$
Then Theorem A.6 applies, yielding a single-layer hypergraph message passing layer $\mathcal{T} : (\mathbb{R}^C)^V \to (\mathbb{R}^2)^V$ that cannot be exactly represented by any single-layer pairwise graph message passing layer on the same vertex set. This contradicts the assumption. Therefore $\mathcal{K}$ must have local complete symmetry. □

A.4 Why the necessary condition is highly restrictive

Proposition A.11 (Combinatorial sparsity of local complete symmetry)**.** *Let U be a finite set of size m. Then:*

1. *The total number of subfamilies $\mathcal{F} \subseteq 2^U$ is 2^{2^m}.*
2. *The number of subfamilies satisfying the cardinality-only condition $\mathcal{F} = \bigcup_{r \in R} \binom{U}{r}$ is 2^{m+1}.*
3. *Consequently, among all local patterns $\mathcal{F} \subseteq 2^U$, the proportion of those having local complete symmetry is exactly 2^{m+1-2^m}.*

Proof. Since 2^U contains 2^m elements and $\mathcal{F}$ is a subset of 2^U, the total number of possible subfamilies is $2^{|2^U|} = 2^{2^m}$. On the other hand, any subfamily of the form $\mathcal{F} = \bigcup_{r \in R} \binom{U}{r}$ is uniquely determined by the index set $R \subseteq \{0, 1, \ldots, m\}$. There are exactly 2^{m+1} such choices of R, and therefore exactly 2^{m+1} such subfamilies. Dividing the two quantities yields

$$\frac{2^{m+1}}{2^{2^m}} = 2^{m+1-2^m}. \tag{60}$$

□

Remark A.12 (Interpretation and scope). Proposition A.11 shows that local complete symmetry is an extremely strong combinatorial constraint: if the local support set around a vertex has size m, then the admissible local patterns occupy only a 2^{m+1-2^m} fraction of all possible local patterns, which decays doubly exponentially as m grows. This quantitatively justifies the main text's statement that universal graph representability would require the hypergraph to satisfy highly restrictive structural conditions.

Combining Theorem A.6 and Theorem A.10, we obtain the logical statement used in the paper: under a fixed vertex set, without auxiliary vertices, and within the class of single-layer pairwise graph message passing layers, there exist nonlinear hypergraph message passing layers that cannot be exactly represented by any graph message passing layer; conversely, if one requires *every* single-layer hypergraph message passing layer on a given hypergraph to be exactly representable by a graph message passing layer, then that hypergraph must satisfy local complete symmetry, a highly restrictive necessary condition.

A.5 Connection to Differential Mixed Aggregation

We now connect the preceding separation result to the DMA operator in Sec. 4.3. We treat the global hyperedges $\mathcal{E}^{\mathrm{glb}}$, their normalized incidence weights W, and the local support sets $\mathcal{N} = (\mathcal{N}_u)_{u \in V}$ as fixed inputs to Correlation Propagation. For a chosen mapping ϕ, let $\mathcal{T}^{\mathrm{DMA}}$ denote the resulting operator. Exact representation is required for all vertex features while these quantities remain fixed.

Proposition A.13 (Non-representability of conditional DMA)**.** *There exist a finite vertex set V, fixed global hyperedges, normalized incidence weights, local support sets, and a linear mapping ϕ such that the conditional DMA operator $\mathcal{T}^{\mathrm{DMA}} : \mathbb{R}^V \to \mathbb{R}^V$ cannot be exactly represented by any single-layer pairwise graph message passing layer in Definition A.2 on the same vertex set.*

Proof. Let V contain four distinct vertices v, a, b, c, and take

$$\mathcal{E}^{\mathrm{glb}} = \{e_1, e_2\}, \qquad e_1 = \{v, a, b\}, \qquad e_2 = \{v, a, c\}. \tag{61}$$

Assign uniform weights $1/3$ within each hyperedge. Choose nonempty local support sets satisfying $v \in \mathcal{N}_v$ and $\mathcal{N}_v \cap \{a, b, c\} = \emptyset$, and let $\phi([x \mid d]) = d$. The scalar DMA output at v is then

$$(\mathcal{T}^{\mathrm{DMA}}(X))_v = \max\left\{ \max_{u \in \mathcal{N}_v} (x_u - x_v),\ \frac{x_a + x_b - 2x_v}{3},\ \frac{x_a + x_c - 2x_v}{3} \right\}. \tag{62}$$

Suppose that a graph layer $\mathcal{G}$ exactly represents $\mathcal{T}^{\mathrm{DMA}}$. Starting from the zero input, changing only x_w to 1 for any $w \in \{a, b, c\}$ changes the DMA output at v from 0 to $1/3$. If $w \notin N_G(v)$, the graph output at v would remain unchanged. Hence,

$$\{a, b, c\} \subseteq N_G(v). \tag{63}$$

Now consider two inputs $X, Y \in \mathbb{R}^V$ with

$$(x_a, x_b, x_c) = (2, 1, 0), \qquad (y_a, y_b, y_c) = (0, 2, 1), \tag{64}$$

and $x_u = y_u = 0$ for every $u \in V \setminus \{a, b, c\}$. The center features agree, and the graph-neighbor features form the same multiset under both inputs. Lemma A.5 therefore gives $(\mathcal{G}(X))_v = (\mathcal{G}(Y))_v$. However, all local differential messages at v are zero, whereas the global hyperedge messages yield

$$(\mathcal{T}^{\mathrm{DMA}}(X))_v = 1, \qquad (\mathcal{T}^{\mathrm{DMA}}(Y))_v = \frac{2}{3}. \tag{65}$$

This contradicts exact representation and proves the claim. □

Remark A.14 (Scope of the DMA connection). Proposition A.13 shows that the conditional DMA family contains an operator sensitive to hyperedge membership and overlap that cannot be represented by the stated graph class. Its scope is the propagation stage with fixed supports and weights; extending the result to the full input-dependent Correlation Induction–Correlation Propagation map requires a separate analysis.

B More Details of Hypergraph Construction and Propagation

Topology-enhanced features are used to compute correlation scores, while hyperedge representations aggregate the features before topological injection. This introduces local feature-space context into correlation induction.

For local neighbor queries at the same resolution, the center token is included in the candidate set. When selected, it contributes a zero differential message and is counted as one of the k_{loc} entries.

During training, hyperedge dropout masks connections used for vertex-to-hyperedge aggregation and hyperedge-to-vertex feedback. For each nonempty candidate set, at least one connection is retained to prevent empty aggregation. Vertex-to-hyperedge weights are renormalized over the retained vertices; if only one vertex remains, its normalized weight is one. Masked feedback messages are excluded from the channel-wise maximum. Random masking is disabled during inference.

DMA uses channel-wise max to select the largest differential response from local neighbors and global hyperedges in each feature channel, while retaining the target vertex feature through concatenation. This allows different channels to select informative responses from different relations without uniformly averaging all messages.

C Implementation Details

To accommodate different performance requirements and computational budgets, both Isotropic ViC and Pyramid ViC are instantiated in three model sizes, namely Tiny (T), Small (S), and Base (B). All models are trained on 8 NVIDIA V100 GPUs.

ImageNet classification. For ImageNet-1K classification, the detailed experiment setting is shown in Tab. 6. We train all models for 400 epochs using AdamW [60] with a batch size of 1024. The initial learning rate is set to 2×10^{-3}, and a cosine annealing schedule is adopted with 20 warmup epochs. The weight decay is set to 0.05, and label smoothing is set to 0.1. We use stochastic depth with drop path rates of 0.1/0.1/0.3 for the T/S/B models, respectively. Following common practice, the data augmentation pipeline includes repeated augmentation, RandAugment [46], Mixup [47] with probability 0.8, CutMix [48] with probability 1.0, and random erasing [49] with probability 0.25. We further maintain an exponential moving average (EMA) of model parameters with decay 0.99996 throughout training.

For the hypergraph-related configurations, let L denote the number of ViC encoder layers in the network. The number of semantic anchors is progressively increased across depth as $M = \mathrm{LinSpace}(16, 32, L)$. Similarly, the local neighborhood size is set as $k_{\mathrm{loc}} = \mathrm{LinSpace}(9, 18, L)$, which gradually enlarges the local receptive field in deeper layers. The global support size is defined as $k_{\mathrm{glb}} = \mathrm{LinSpace}(0.5M, 0.5M, L)$, *i.e.*, the global support size is set to half of the anchor number at each layer. The topological injection coefficient is fixed to $\alpha = 0.1$, and the hyperedge dropout rate is set to 0.2 for all model scales.

MS COCO object detection. For MS COCO object detection, we use RetinaNet [51] and Mask R-CNN [52] as the detection frameworks, with Pyramid ViC serving as the backbone. Following

Table 6: Training hyper-parameters for ImageNet.

Hyper-parameters	**T**	**S**	**B**
Epochs		400	
Optimizer		AdamW [60]	
Batch size		1024	
Start learning rate		2e-3	
Learning rate schedule		CosineAnnealing	
Warmup epochs		20	
Weight decay		0.05	
Label smoothing		0.1	
Stochastic path	0.1	0.1	0.3
Repeated augment		✓	
RandAugment		✓	
Mixup prob.		0.8	
Cutmix prob.		1.0	
Random erasing prob.		0.25	
Exponential moving average		0.99996	
M		$\mathrm{LinSpace}(16, 32, L)$	
k_{loc}		$\mathrm{LinSpace}(9, 18, L)$	
k_{glb}		$\mathrm{LinSpace}(0.5M, 0.5M, L)$	
α		0.1	
Hyperedge dropout rate		0.2	

prior works [30, 53, 15, 22], all detection models are trained under the standard "1×" schedule on the COCO 2017 training set and evaluated on the validation set. We keep the detector settings consistent with the corresponding baseline implementations, so that the comparison mainly reflects the difference brought by the backbone design. We initialize the backbone with the ImageNet-pretrained Pyramid ViC checkpoint and transfer all ViC-specific structural hyperparameters unchanged, including M, k_{loc}, k_{glb}, α, and the hyperedge-dropout rate. No COCO-specific search of these structural hyperparameters is performed.

D Additional Visual Evaluations

Instance segmentation on MS COCO. We additionally report instance-segmentation results from the same Mask R-CNN experiments used for the object detection evaluation in Table 3. The models are evaluated on COCO val2017 under the same $1\times$ training schedule, with no changes to the backbone architecture or ViC-specific structural hyperparameters. We report mask AP, AP_{50}, and AP_{75}, denoted by AP^m, AP^m_{50}, and AP^m_{75}, respectively.

Table 7: Instance-segmentation results on MS COCO using Mask R-CNN under the $1\times$ schedule.

Backbone	Params (M)	FLOPs (G)	AP^m	AP^m_{50}	AP^m_{75}
ResNet-50 [26]	44.2	260.1	34.4	55.1	36.7
PVT-Small [30]	44.1	245.1	37.8	60.1	40.3
CycleMLP-B2 [55]	46.5	249.5	38.9	61.2	41.8
Swin-T [29]	47.8	264.0	39.1	61.6	42.0
Pyramid ViG-S [15]	45.8	258.8	39.4	62.4	41.6
Pyramid ViHGNN-S [22]	47.3	261.4	39.6	63.0	42.3
Pyramid ViC-S (Ours)	43.3	250.9	**39.7**	**63.2**	**42.4**

As shown in Table 7, Pyramid ViC-S achieves 39.7 mask AP, compared with 39.4 for Pyramid ViG-S and 39.6 for Pyramid ViHGNN-S, while using fewer parameters and FLOPs than both backbones. These results extend the evaluation of ViC to pixel-level instance prediction.

Evaluation at different input resolutions. We evaluate ViC-T/S at input resolutions of 192×192, 224×224, 256×256, and 320×320 on the full ImageNet-1K validation set. The same checkpoints trained at 224×224 are used for all resolutions. Absolute positional embeddings are resized by bicubic

interpolation, with no additional training or resolution-specific hyperparameter tuning. ViHGNN-T/S are evaluated under the same protocol.

Table 8: ImageNet-1K Top-1 accuracy (%) at different input resolutions. All checkpoints are trained at 224×224 and evaluated without additional training.

Resolution	ViHGNN-T	ViHGNN-S	ViC-T	ViC-S
192×192	71.1	79.4	74.4	80.3
224×224	74.3	81.5	76.5	81.9
256×256	74.0	81.1	76.7	81.7
320×320	73.8	80.7	76.3	81.5

Table 8 shows that ViC outperforms the corresponding ViHGNN variants at all four resolutions. Relative to the results at 224×224, the largest accuracy drops for ViC-T/S are 2.1/1.6 percentage points, compared with 3.2/2.1 for ViHGNN-T/S. These results indicate better accuracy retention across the evaluated resolutions without retraining or resolution-specific tuning.

E Additional Ablation Results

Contribution of Global Hyperedge Aggregation. We evaluate two controls under the same Isotropic ViC-T architecture and ImageNet-1K training recipe as Table 4. Both retain the local branch, DMA, and anchor-based support selection. Global-pairwise replaces each global hyperedge representation with a representative token from its support set, while Latent-prototype replaces it with a learned prototype of the same feature dimension. We also include Local-only, which retains local differential aggregation but removes the global hyperedges.

Table 9: Controls for global hyperedge aggregation in Isotropic ViC-T on ImageNet-1K.

Variant	Params (M)	FLOPs (G)	Top-1 (%)
Local-only	6.57	1.16	73.8
Global-pairwise	6.62	1.18	74.9
Latent-prototype	6.62	1.18	74.7
Full ViC	6.62	1.18	**76.5**

As shown in Table 9, removing the global hyperedges reduces Top-1 accuracy by 2.7 percentage points. Full ViC also outperforms Global-pairwise and Latent-prototype by 1.6 and 1.8 percentage points, respectively, at comparable parameter and FLOP budgets. These results support the contribution of input-dependent multi-token aggregation beyond anchor-based support selection or static prototype features.

Feature richness analysis. We further analyze the token feature richness across network depth by comparing ViC-S with ViG-S. Following a controlled setting, we randomly sample 100 categories from the ImageNet validation set and further randomly select 3 images per category, resulting in 300 images in total. For each block, we compute two token-level statistics and report the average results over all sampled images.

Given the token feature matrix of a block $X \in \mathbb{R}^{N \times C}$, where N is the number of tokens and C is the channel dimension, we first consider *relative feature diversity*, defined as:

$$\mathrm{Div}_{\mathrm{rel}}(X) = \frac{\|X - \mathbf{1}\bar{x}^\top\|_F}{\|X\|_F}, \qquad \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{66}$$

where $\bar{x}$ is the mean token feature. This metric measures the proportion of feature energy contributed by inter-token variations. A larger value indicates that the token representations remain more diverse, while a smaller value suggests that the tokens become increasingly homogeneous.

As a complementary measure, we also compute the *mean off-diagonal cosine similarity*. Specifically, we perform row-wise ℓ_2 normalization on X to obtain $\hat{X}$, compute the token-token cosine similarity

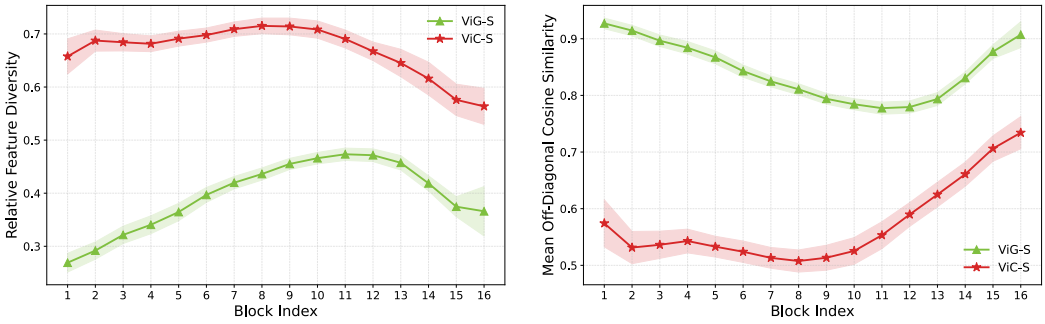

Figure 6: Comparison of feature richness between ViG-S and ViC-S across network depth. Left: relative feature diversity. Right: mean off-diagonal cosine similarity. ViC-S consistently exhibits higher feature diversity and lower inter-token similarity, indicating richer token representations and less homogenization than ViG-S.

matrix $S = \hat{X}\hat{X}^\top$, and then average all off-diagonal entries:

$$\mathrm{Sim}_{\mathrm{off}}(X) = \frac{1}{N(N-1)} \sum_{i \neq j} S_{ij}. \tag{67}$$

This metric reflects how similar different tokens are to each other. A larger value indicates stronger token homogenization and thus a more severe over-smoothing tendency.

As shown in Fig. 6, ViC-S consistently exhibits higher relative feature diversity than ViG-S across all blocks, indicating that ViC better preserves informative token-wise variations during deep propagation. This advantage is further supported by the mean off-diagonal cosine similarity, where ViC-S remains consistently lower than ViG-S, suggesting reduced token homogenization. Overall, these results show that the Differential Mixed Aggregation in ViC is more effective at maintaining feature richness than conventional graph-based propagation.

Effect of semantic anchor number. Table 10 studies the effect of different semantic anchor settings. When the anchor number is too small, *e.g.*, [4→8], the induced global hypergraph is not expressive enough, leading to inferior performance. Increasing the anchor number to [8→16] and [16→32] steadily improves the Top-1 accuracy, showing that a richer set of semantic anchors helps capture more informative high-order semantic correlations. When further increasing it to [32→64], the accuracy saturates at 76.5%, indicating that simply enlarging the number of anchors brings limited additional benefit. Therefore, we use [16→32] as the default setting, which provides a good balance between effectiveness and structural complexity.

Table 10: Effect of different M settings on Top-1 accuracy.

M	[4→8]	[8→16]	[16→32]	[32→64]
Top-1	75.8	76.3	**76.5**	**76.5**

Effect of local neighborhood size. Table 11 studies the effect of different local neighborhood settings. When the local neighborhood size is too small, *e.g.*, [3→6], the local branch cannot capture sufficient structural context, resulting in inferior performance. Increasing k_{loc} to [9→18] improves the Top-1 accuracy to 76.5%, showing that a moderate local receptive field is beneficial for modeling informative local correlations. However, further enlarging it to [15→30] slightly degrades the performance to 76.4%, suggesting that overly large local neighborhoods may introduce redundant or less relevant correlations. Therefore, we use [9→18] as the default setting.

Table 11: Effect of different k_{loc} settings on Top-1 accuracy.

k_{loc}	[3→6]	[9→18]	[15→30]
Top-1	75.7	**76.5**	76.4

Effect of hyperedge dropout rate. Table 12 studies the effect of different hyperedge dropout rates. Without dropout, the model achieves 76.3% Top-1 accuracy. Introducing a moderate dropout rate consistently improves the performance, and the best result is obtained at 0.2 and 0.3. This suggests that hyperedge dropout helps regularize the induced hypergraph structure and improves the stability of high-order correlation learning. When the dropout rate is further increased to 0.5, the performance slightly drops, indicating that excessive dropout may remove too much useful structural information. Therefore, we use 0.2 as the default setting.

Table 12: Effect of different dropout rates on Top-1 accuracy.

Dropout Rate	0	0.1	0.2	0.3	0.5
Top-1	76.3	76.4	**76.5**	**76.5**	76.4

Effect of topological injection coefficient. Table 13 studies the effect of different topological injection coefficients. Without topological injection, the model achieves 76.3% Top-1 accuracy. Introducing a small coefficient ($\alpha = 0.1$) improves the performance to 76.5%, showing that local topological priors help enhance the induced correlation structure. However, further increasing α leads to a gradual performance drop, suggesting that overly strong topological injection may over-constrain the learned correlations and weaken the flexibility of semantic modeling. Therefore, we use $\alpha = 0.1$ as the default setting.

Table 13: Effect of different α values on Top-1 accuracy.

α	0	0.1	0.3	0.5
Top-1	76.3	**76.5**	76.4	76.2

Stability Across Random Seeds. We repeat Isotropic ViC-T training on ImageNet-1K with three random seeds under the same training protocol. Table 14 reports the individual results and their mean and standard deviation.

Table 14: ImageNet-1K results of Isotropic ViC-T across three random seeds. The standard deviation is computed with divisor $n = 3$.

Metric	Run 1	Run 2	Run 3	Mean $\pm$ Std.
Top-1 (%)	76.49	76.40	76.63	76.51 ± 0.09

The three runs span 0.23 percentage points, and their mean closely matches the reported 76.5% Top-1 accuracy. These results indicate limited variation across the evaluated seeds for this configuration.

F Model Efficiency

We evaluate single-image inference on the same NVIDIA RTX 4090 GPU with a batch size of 1 and an input resolution of 224×224. Table 15 reports inference latency and peak GPU memory usage across model scales.

At the corresponding T/S/B scales, Pyramid ViC achieves approximately 1.4–$2.5\times$ lower latency than DVHGNN while using less peak memory. Both isotropic and pyramid ViC variants also exhibit lower latency and memory usage than their corresponding ViHGNN counterparts. These measurements support the practical efficiency of ViC under the evaluated single-image inference setting.

G Extension to Hypergraph Benchmarks

To further examine whether the proposed method is specific to visual recognition or can generalize to generic high-order data, we extend it to standard hypergraph benchmarks. Thus, here we name the extended version as **Correlator**.

Table 15: Single-image inference efficiency on an NVIDIA RTX 4090 with 224×224 inputs and batch size 1.

Model	Params (M)	Latency (ms)	Memory (MiB)
DVHGNN-T	11.1	22.6	207.9
DVHGNN-S	30.2	35.8	329.6
DVHGNN-B	92.8	80.8	746.6
ViHGNN-T	8.2	122.0	56.9
ViHGNN-S	23.2	169.1	150.1
ViHGNN-B	88.1	160.8	523.0
Pyramid ViHGNN-T	12.3	350.8	205.3
Pyramid ViHGNN-S	28.5	346.2	300.8
Pyramid ViHGNN-B	94.4	470.6	675.5
ViC-T	6.6	15.3	52.4
ViC-S	20.5	20.5	134.7
ViC-B	77.1	21.1	467.3
Pyramid ViC-T	9.9	16.1	195.1
Pyramid ViC-S	24.8	16.3	292.3
Pyramid ViC-B	83.3	31.7	633.2

Following the experimental protocol of *You are AllSet* [61], we evaluate our model under the transductive semi-supervised setting on five widely used hypergraph vertex classification datasets, including Cora [58], Pubmed [58], Cora-CA [58], Citeseer [58], and Zoo [62]. For fair comparison, we adopt the same train/validation/test split ratio of 50%/25%/25%, and report the mean accuracy together with the standard deviation over 20 runs with different random splits and initializations.

Table 16: Results on hypergraph benchmarks (mean accuracy (%) $\pm$ standard deviation).

Method	**Cora**	**Pubmed**	**Cora-CA**	**Citeseer**	**Zoo**
MLP	75.17 ± 1.21	87.47 ± 0.51	74.31 ± 1.89	72.67 ± 1.56	87.18 ± 4.44
HGNN	79.39 ± 1.36	86.44 ± 0.44	82.64 ± 1.65	72.45 ± 1.16	92.50 ± 4.58
HNHN	76.36 ± 1.92	86.90 ± 0.30	77.19 ± 1.49	72.64 ± 1.57	93.59 ± 5.88
HyperAttn	79.14 ± 1.02	86.41 ± 0.36	82.55 ± 0.97	72.42 ± 1.42	93.65 ± 6.15
UniGCNII	78.81 ± 1.05	88.25 ± 0.40	83.60 ± 1.14	73.05 ± 2.21	93.65 ± 4.37
HyperGCN	78.45 ± 1.26	82.84 ± 8.67	79.48 ± 2.08	71.28 ± 0.82	N/A
HAN (mini batch)	79.70 ± 1.77	85.32 ± 2.25	81.71 ± 1.73	74.12 ± 1.52	75.77 ± 7.10
AllSetTransformer	78.59 ± 1.47	88.72 ± 0.37	83.63 ± 1.47	73.08 ± 1.20	97.50 ± 3.59
Correlator (Ours)	$\mathbf{83.76 \pm 1.15}$	$\mathbf{89.81 \pm 0.44}$	$\mathbf{84.73 \pm 1.36}$	$\mathbf{75.15 \pm 1.62}$	$\mathbf{98.46 \pm 3.08}$

As shown in Table 16, our Correlator consistently achieves superior performance on all five benchmarks. Compared with representative hypergraph baselines, including HGNN [20], HNHN [63], HCHA [59], UniGCNII [41], HyperGCN [58], HAN [64], and AllSetTransformer [61], our method improves the accuracy from 79.70% to 83.76% on Cora, from 88.72% to 89.81% on Pubmed, from 83.63% to 84.73% on Cora-CA, from 74.12% to 75.15% on Citeseer, and from 97.50% to 98.46% on Zoo, respectively. The results indicate that the effectiveness of the proposed Correlator is not limited to visual backbone design, but also generalizes well to standard hypergraph representation learning tasks.

We attribute this advantage to its correlation-driven propagation mechanism. Different from conventional hypergraph operators that mainly rely on hyperedge-level smoothing, the proposed Correlator emphasizes differential correlation aggregation, which is more effective in preserving informative feature variations while still exploiting high-order dependencies. The consistent improvements across these datasets further suggest that the proposed Correlator provides a general and robust solution for high-order correlation learning.

H Discussion

Advantages. This work aims to move visual backbone design from pairwise graph message passing toward a broader hypergraph-based correlation modeling paradigm. Its main advantage lies in separating visual understanding into Correlation Induction and Correlation Propagation. This formulation makes the correlation structures to be modeled explicit: token anchors preserve local details, while semantic anchors adaptively induce global high-order hyperedges for cross-region and cross-part semantic coordination. Compared with using hypergraph computation as an auxiliary module, ViC instantiates the framework at the backbone level and can be deployed in both isotropic and pyramid architectures. The proposed Differential Mixed Aggregation further combines local neighbor differences and global hyperedge-level semantic differences within a single vertex update, which helps preserve discriminative token variations while exploiting high-order semantics. ViC provides a practical balance among expressive correlation coverage, computational efficiency, and structural robustness.

Scope and limitations. The theoretical result establishes a strict expressive separation under a controlled setting: a fixed vertex set, single-layer pairwise graph message passing, no auxiliary vertices, and no extra edge attributes encoding hyperedges. It is therefore intended to justify the necessity of high-order correlation modeling, rather than to claim that every hypergraph instantiation universally dominates every graph-based architecture. Empirically, our validation covers image classification, object detection, instance segmentation, and standard hypergraph benchmarks. Further evaluation on semantic segmentation, video understanding, and multimodal perception would help assess the applicability of ViC to a broader range of tasks. Extending the framework to segmentation, video understanding, and multimodal perception is a promising direction. In addition, ViC uses several structural hyperparameters, such as the numbers of semantic anchors and local/global support sizes. Although the sparse top-k design and ablation studies suggest stable behavior under practical settings, automatically adapting these structures to different tasks and resolutions may further improve flexibility.