

# DARTS-GT: Differentiable Architecture Search for Graph Transformers with Quantifiable Instance-Specific Interpretability Analysis

Shruti Sarika Chakraborty Peter Minary

**Abstract**—Graph Transformers (GTs) have emerged as powerful architectures for graph-structured data, yet remain constrained by rigid designs and lack quantifiable interpretability methods. Current state-of-the-art GTs commit to fixed GNN types across all layers, missing potential benefits of depth-specific component selection, while their increasingly complex architectures become opaque black boxes where performance gains cannot be distinguished between meaningful structural patterns and spurious correlations. We redesign the GT attention mechanism through asymmetry, decoupling structural encoding from feature representation. Queries derive directly from node features, while keys and values come from graph neural network (GNN) transformations, separating how the model learns features from how it encodes graph structure. Within this asymmetric framework, we use Differentiable ARchiTecture Search (DARTS) to select optimal GNN operators at each layer, enabling depth-wise heterogeneity inside the transformer attention itself, hence the name DARTS-GT. To understand these discovered architectures, we develop the first quantitative interpretability framework for GTs through causal ablation that identifies which heads and nodes actually drive predictions. Our metrics: Head-deviation, Specialization, and Focus, reveal the specific components responsible for each prediction while enabling broader model comparison. Experiments across eight benchmarks demonstrate that DARTS-GT achieves state-of-the-art performance on four datasets while remaining competitive on others, with discovered architectures revealing dataset-specific patterns ranging from highly specialized to balanced GNN distributions. Our interpretability analysis reveals that visual attention salience and causal importance do not necessarily correlate, indicating that widely used visualization approaches may miss the components that actually matter for predictions. Crucially, the heterogeneous architectures found by DARTS-GT consistently produced more interpretable models than baseline GTs, establishing that Graph Transformers do not need to choose between performance and interpretability.

**Index Terms**—Graph Transformers, Neural Architecture Search, Interpretability, Causal Ablation, Graph Neural Networks

## I. INTRODUCTION

For graph-structured data, Graph Transformers (GTs) have become a dominant architectural choice, combining attention mechanisms with graph-awareness [1], [2]. Their success spans protein structure-to-function prediction [3], drug discovery [4], and materials design [5], where understanding complex structural patterns is crucial.

Current state-of-the-art GTs incorporate graph structure through GNN-Transformer combinations [2], [6], specialized positional encodings [7], and attention augmentation with structural biases [8]. However, these architectures commit to a fixed GNN type throughout all layers or a predetermined encoding scheme. This rigidity prompts an investigation into whether GTs benefit from depth-specific component selection.

Depth-specific, task-adaptive GNN selection is motivated by recent neural-architecture-search (NAS) successes in GNNs. Methods like GraphNAS [9], SANE [10], and PAS [11] show that heterogeneous architectures employing different GNN operations across depths outperform homogeneous models. These works reveal that optimal architectures develop dataset-adaptive depth-wise specialization, learning complementary patterns at different scales. In GTs, where multiple attention heads already capture diverse node-node interactions, depth-wise architectural diversity becomes even more compelling, yet remains unexplored.

Relaxing architectural constraints introduces a broader challenge. As Graph Transformer variants become more flexible, their internal decision processes become increasingly opaque. Without interpretability methods, it is difficult to determine whether reported gains reflect meaningful structural patterns or spurious correlations. This, for example, forms a crucial distinction in scientific domains such as drug discovery, where predictions must be justifiable. However, this issue is not specific to a particular GT design: for instance, in GraphGPS (also referred to as GPS) [6], which has a hybrid-GT architecture by combining message-passing-neural-networks (MPNN) in parallel with transformer attention, it is unclear whether improvements arise from the auxiliary MPNN or the Transformer path. Similarly, in Graph-Trans [12] and SAT [2], the contribution of the GNN pre-encoding or k-subtree extraction, respectively, is similarly entangled. In short, regardless of the GT approach, interpretability is essential to clarify which components and structures truly drive model performance.

Recent NAS efforts for GTs include EGTAS [13], AutoGT [14], UNIFIEDGT [15] and UGAS [16]. EGTAS uses evolutionary search to explore topologies and graph-aware strategies by searching through established architectural paradigms from state-of-the-art designs. This essentially navigates between existing construction patterns, selecting in this process a single GNN block type, applied uniformly across depth. AutoGT employs evolutionary strategies over established GT templates and encodings, without depth-wise GNN selection. UNIFIEDGT exposes modular search over

Department of Computer Science, University of Oxford, Oxford OX1 3QG, UK. Emails: shruti.chakraborty@cs.ox.ac.uk, peter.minary@cs.ox.ac.uk.

components and selects an overall configuration rather than layer-wise GNN selection within attention blocks. UGAS uses Differentiable ARchiTecture Search (DARTS) or Genetic Algorithms (GA) to search over subsets of MPNN modules (e.g., GCN [17], GAT [18]) and GT modules (Transformer [19], Performer [20]) deployed in parallel within each layer. Each layer may contain only MPNNs, only GTs, or any mixture, with GTs treated as optional candidates rather than architectural foundations.

The selection of depth-specific GNN presents a formidable optimization problem: with  $n$  GNN variants across  $L$  layers, the search space contains  $n^L$  configurations. Thus, systematic exploration of this search space is infeasible in most practical applications. An effective search must intelligently adapt to datasets and tasks, determining which GNN types to deploy and their optimal layer positions (depth). This requires automated search techniques, specifically NAS, capable of navigating this vast configuration space.

These discovered architectures must also be interpretable for trustworthiness and scientific validity. Unlike well-developed GNN explainability methods [21] that identify prediction-critical subgraphs, GT interpretation, widely regarded as a blackbox [22], remains dominated by attention visualization [23] lacking quantitative metrics and the ability to differentiate visually prominent but functionally irrelevant patterns from subtle yet critical ones. More recently, noteworthy efforts in GT interpretability have focused on analyzing attention graphs (i.e. aggregating head/layer attention into an attention graph) [24]. However, none of the prior methods test faithfulness (whether heads/nodes causally affect predictions) nor enable cross-model comparison via standardized scores. Task-specific self-explanations exist but do not generalize to arbitrary GTs or provide head attributions [25]. We seek a GT-agnostic instance-level interpretability framework grounded in perturbation/ablation that delivers model-level scores for comparing GT interpretability.

Hence, our investigation addresses two complementary research questions: *Q1: Can Graph Transformers achieve improved performance and adaptability by dynamically selecting GNN types per layer (depthwise) and creating dataset-specific architectures ranging from uniform to heterogeneous? Q2: How can we quantitatively assess the interpretability of a GT and identify which regions of the graph and architectural components drive specific predictions?*

These questions are inter-linked: Q1 explores performance potential, while Q2 ensures interpretability and trustworthiness, essential for high-stakes domains.

To address these challenges, we introduce a comprehensive framework with two synergistic components:

First, we employ DARTS to select GNN operators within the attention blocks at each layer. In doing so, we reimagine the asymmetric attention mechanism initially proposed by GMTPool [26], but in a more computationally efficient form: queries are derived directly from node features, while keys and values are obtained through GNN transformations of those features. This design explicitly separates structural encoding from feature representation and enables depth-wise heterogeneity across layers, embedding it directly inside the attention

mechanism rather than through external augmentation, thereby rewiring how GTs capture graph structure. The search procedure is powered by DARTS [27], which leverages gradient-based optimization of the continuously relaxed architecture space, avoiding the expense of training discrete candidates from scratch. DARTS has been extensively validated in prior graph NAS studies (e.g., SANE [10], GASSO [28], PAS [11], UGAS), underscoring its computational efficiency and robustness in navigating combinatorial design spaces.

Second, we introduce a Fidelity like [29], [30], instance-level, ablation-grounded score: **Head-deviation**, quantifying prediction changes when specific heads are masked. From per-head effects on each instance, we compute two complementary metrics: **Specialization** captures *head distinguishability* as dispersion of head impacts (high = few heads carry causal load; low = heads act interchangeably); **Focus** captures *consensus* over graph regions by measuring top-head agreement on top-k nodes (high = heads converge on same salient substructures or nodes). Aggregating per-instance values by dataset median yields robust model-level summaries. Together, these form our interpretability framework that we propose to analyse our model along with existing models for comparison. Head-deviation provides head and graph-region attributions per input, while median Specialization and median Focus enable dataset-level GT comparison indicating which model is easier to interpret at matched accuracy.

This interpretation is particularly valuable in applications like drug toxicity prediction. When identifying toxic molecules, practitioners must verify the focus on actual toxicophores versus spurious correlations critical for regulatory approval and scientific validity. Also, without methods to quantify the difficulty of interpretation, meaningful architecture comparison remains impossible. Our framework provides quantifiable metrics representing the first quantitative framework for both globally evaluating and comparing GT interpretability while providing instance-specific analysis that traces predictions from architectural components to graph regions.

Our specific contributions:

- 1) The first DARTS-based architecture search for depth-wise GNN selection within GT layers, exploring heterogeneity through layer-specific GNN operators within attention blocks. Our approach maintains computational efficiency to find depth-wise optimal GNN selections among the possible configurations ( $n$  GNN variants,  $L$  layers). This enables automated discovery of task optimal architectures.
- 2) Strong empirical performance achieving competitive or SOTA results on multiple benchmarks. Our architecture consistently outperforms standard GTs [19] across dense and sparse attention mechanisms, while matching or exceeding established baselines: Recent GNNs (e.g. GCN [17], GIN [31], GatedGCN [32]), recent GT architectures (e.g. GPS [6], Graphormer [8]), and NAS-based approaches (e.g. EGTAS [13], AutoGT [14], UGAS [16]).
- 3) To our knowledge, we also introduce the first GT-agnostic interpretability framework that is both instance-

level and quantitative. We compute an ablation-grounded Head-deviation metric per input, then derive Specialization (head distinguishability) and Focus (consensus of top-head node sets). Dataset-level medians provide robust model-level summaries, enabling both local evaluation (which heads/nodes mattered) and global evaluation (which model is easier to interpret). Our analyses reveal that visually broad attention can emerge as functionally irrelevant, while sparse patterns can be decisively predictive: underscoring the limits of visualization-based approaches.

- 4) We apply the interpretability framework across three architectures (DARTS-GT, GPS [6] and standard (Vanilla) GT [19]) and show that the DARTS-GT derived model is more consistent overall in terms of overall consensus (Focus) and head-distinguishability (Specialization).

In summary, we (i) propose a differentiable NAS framework that builds depth-wise heterogeneous Graph Transformers through direct redesigning of the attention mechanism rather than auxiliary GNN augmentation and (ii) develop quantitative interpretability metrics at head- and node-level per instance. Our code is available at: [https://github.com/shrutiOx/DARTS\\_GT](https://github.com/shrutiOx/DARTS_GT).

## II. RELATED WORK

We discuss three interconnected themes related to this work in separate paragraphs.

**Graph Transformer** : A broad line of recent work combines global self-attention with graph inductive bias (GNNs) [12]. GraphGPS [6] interleaves local message passing with global attention and aggregates rich positional/structural encodings, effectively in a parallel MPNN and transformer attention design, setting a strong template for hybrid GTs. SAN [7] injects spectral positional information and runs a fully connected Transformer, showing that learned spectral PEs can offset oversquashing while preserving structure. SAT [2] goes further and makes attention itself structure-aware by extracting k-subtree/k-subgraph features with a base GNN before computing attention, consistently improving performance over baseline GTs. Graphormer [8] augments attention with centrality, shortest-path and edge encodings, and demonstrated state-of-the-art molecular property prediction at scale. Variants in other modalities (e.g. Mesh Graphormer [11]) reinforce the pattern of mixing graph convolution with attention to capture local+global interactions. Graph-Trans [12] integrates a preliminary stack of GNN layers before Transformer blocks, using the GNN to encode local structural patterns that the subsequent attention layers can then exploit. Our work sits in this stream but differs in how the local component is chosen: instead of a fixed GNN extractor or a single hybrid template, we dynamically select the GNN type per Transformer layer to yield dataset-specific, potentially heterogeneous stacks.

A related pooling-centric line is Graph Multiset Transformer (GMT) [26], which builds graph-level representations with attention-based pooling. GMT computes K/V via GNNs to explicitly encode structure while forming queries from node embeddings, improving over linear projections used in standard multi-head-attention (MHA). Yet GMT’s implementation

suffers from computational inefficiency: each attention head requires independent GNN operations for its keys and values, scaling complexity linearly with head count. Moreover, their framework remains confined to graph-level pooling rather than general transformation layers. We borrow the idea of asymmetric query–key/value sources, but reimagine it within full GT layers with a computationally efficient approach: queries come directly from node features (instead of node embeddings in GMT that undergo another layer of GNN), while keys/values are produced by layer-specific GNN operators, shared across heads for efficiency and generalization beyond graph-level pooling.

**NAS for Graphs and Graph Transformers:** Early Graph neural-architecture-search (GNAS) studies such as GASSO [28], SANE [10], GraphNAS [9] and PAS [11] are searched over message-passing cells on pure GNN based backbone supernet. However, for GTs, AutoGT [14] defines a unified GT formulation and searches Transformer hyperparameters plus structural/positional encodings, but the resulting architecture is globally tied rather than layer-wise heterogeneous. EGTAS [13] expands the GT search space with macro- (topology) and micro- (graph-aware strategies) design and uses evolutionary search to assemble end-to-end GTs. UGAS [16] proposes a Unified Graph Neural Architecture Search that explores which subsets of MPNN modules and GT modules to activate in parallel per layer, with GTs treated as optional candidates rather than foundational components. Complementary frameworks such as UNIFIEDGT [15] modularize GT components (sampling, structural priors, attention, local/global mixing, FFNs) to instantiate many GT variants. Our work is orthogonal. We keep the GT spine and search for the local operator (GNN) per layer with DARTS, yielding adaptive uniform/heterogeneous architectures while preserving global attention. This enables direct analysis of how structural encoding choices impact performance within a consistent computational framework.

**Interpretability of (Graph) Transformers** Existing transformer studies investigate head importance for pruning [33], i.e., measuring global accuracy drops to compress models, or through head ablation to assess model performance under masking [34], [35]. We instead quantify instance-specific importance for interpretability, identifying which heads and nodes drive individual predictions rather than finding globally redundant components. This connects to the broader line of causal interventions known as activation patching, where internal components are ablated or replaced, and the output change is measured [36] [37]. Zero-ablation, as we employ, is a popular variant [36] amongst other causal-tracing techniques. Our contribution adapts this causal, instance-level paradigm to Graph Transformers, introducing head-specific ablations with novel metrics. Although GNN explainability has matured through tools like GNNExplainer [21] that identify prediction-critical subgraphs, graph transformers lack equivalent methodologies. Most GT interpretability either visualizes attention [23] or aggregates it across layers. Methods based on attention flow track information paths through attention, but remain correlational [24]. Lastly, some protein structure to function prediction papers [3] use GradCAM in GT attention

to identify ‘important’ residues, but this only shows where gradients flow, not whether those regions actually help correct prediction i.e. if it truly contributes to accuracy. Our study complements this space with causal ablations at two levels: (i) Specialization (indicates whether heads are easily distinguishable by their causal impact) and (ii) Focus (are top-performing head agree on node consensus). The proposed metrics align with established interpretability principles for GNNs (e.g. sparsity, consistency, fidelity) [29] while tying interpretations directly to measured performance changes under controlled deletions.

### III. METHODS

#### A. Preliminaries

We consider a graph  $G = (V, E)$  with node features  $\mathbf{F} \in \mathbb{R}^{n \times d}$ , where  $n = |V|$  and  $d$  is the feature dimension. A Graph Transformer consists of  $L$  layers, each containing multi-head attention followed by feed-forward modules. Let  $\mathbf{Z}^{(\ell)} \in \mathbb{R}^{n \times d}$  denote the node embeddings in layer  $\ell$ .

For brevity, we omit the layer index  $\ell$  when the context is clear. With  $M$  attention heads where  $m \in [1, 2, \dots, M]$ , the attention components are computed as:

$$\mathbf{Q}_m = \mathbf{Z}\mathbf{W}_{Q,m}, \quad \mathbf{K}_m = \mathbf{Z}\mathbf{W}_{K,m}, \quad \mathbf{V}_m = \mathbf{Z}\mathbf{W}_{V,m} \quad (1)$$

where  $\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m \in \mathbb{R}^{n \times d_m}$ ,  $\mathbf{W}_{Q,m}, \mathbf{W}_{K,m}, \mathbf{W}_{V,m} \in \mathbb{R}^{d \times d_m}$  and  $d_m = d/M$ . We assume that  $d_m \in \mathbb{Z}_{>0}$  (i.e.  $d$  is divisible by  $M$ ).

1) *Attention Mechanisms*: In our study, we explore dense and sparse attention.

a) *Full Attention*:: Each node attends to all others:

$$\mathbf{S}_m = \text{softmax}\left(\frac{\mathbf{Q}_m \mathbf{K}_m^\top}{\sqrt{d_m}}\right), \quad \mathbf{Y}_m = \mathbf{S}_m \mathbf{V}_m \quad (2)$$

where  $\mathbf{S}_m \in \mathbb{R}^{n \times n}$ ,  $\mathbf{Y}_m \in \mathbb{R}^{n \times d_m}$  and  $\text{softmax}$  is applied row-wise for full attention (per-query normalization). This leads to a complexity of  $\mathcal{O}(n^2 d_m)$  per head and  $\mathcal{O}(n^2 \sum_m d_m) = \mathcal{O}(n^2 d)$  across  $M$  heads.

b) *Sparse Attention*: In this formulation, nodes attend only to graph neighbors. This leads to a complexity of  $\mathcal{O}(|E| d_m)$  per head and  $\mathcal{O}(|E| \sum_m d_m) = \mathcal{O}(|E| d)$  across  $M$  heads. For the edge set  $E = \{(i, j)\}$  where  $i$  and  $j$  denote source and target nodes. For each edge  $(i, j)$ , we compute the attention coefficient between source node  $i$  and target node  $j$ . Thus for edge  $(i, j) \in E$ :

$$\beta_{m,ij} = \frac{\langle [\mathbf{K}_m]_i, [\mathbf{Q}_m]_j \rangle}{\sqrt{d_m}} \quad (3)$$

$$s_{m,ij} = \frac{\exp(\beta_{m,ij})}{\sum_{i':(i',j) \in E} \exp(\beta_{m,i'j})} \quad (4)$$

$$[\mathbf{Y}_m]_j = \sum_{i:(i,j) \in E} s_{m,ij} [\mathbf{V}_m]_i \quad (5)$$

where  $[\mathbf{Y}_m]_j$ ,  $[\mathbf{Q}_m]_j$ ,  $[\mathbf{K}_m]_i$ , and  $[\mathbf{V}_m]_i$  denote the  $j$ -th and  $i$ -th rows of the respective matrices.

2) *Multi-Head Concatenation*: Each head outputs  $\mathbf{Y}_m \in \mathbb{R}^{n \times d_m}$ , with  $d_m = d/M$ . We concatenate the head outputs along the feature dimension and project back to the model dimension  $d$ :

$$\mathbf{Y} = [\mathbf{Y}_1 \parallel \mathbf{Y}_2 \parallel \dots \parallel \mathbf{Y}_M] \mathbf{W}_{\text{out}}, \quad (6)$$

where  $\mathbf{W}_{\text{out}} \in \mathbb{R}^{(Md_m) \times d} = \mathbb{R}^{d \times d}$ .

3) *Feed-Forward Layers and Skip Connections*: Following the attention module, each block applies a two-layer position-wise feed-forward network with residual connections and normalization applied after each sublayer:

$$\hat{\mathbf{Z}}^{(\ell)} = \text{Norm}(\mathbf{Z}^{(\ell)} + \mathbf{Y}^{(\ell)}), \quad (7)$$

$$\tilde{\mathbf{Z}}^{(\ell)} = \sigma(\hat{\mathbf{Z}}^{(\ell)} \mathbf{W}_1^{(\ell)} + \mathbf{b}_1^{(\ell)}) \mathbf{W}_2^{(\ell)} + \mathbf{b}_2^{(\ell)}, \quad (8)$$

$$\mathbf{Z}^{(\ell+1)} = \text{Norm}(\tilde{\mathbf{Z}}^{(\ell)} + \tilde{\mathbf{Z}}^{(\ell)}), \quad (9)$$

where  $\sigma$  is the nonlinearity (we use RELU in our implementation),  $\mathbf{W}_1^{(\ell)} \in \mathbb{R}^{d \times rd}$ ,  $\mathbf{W}_2^{(\ell)} \in \mathbb{R}^{rd \times d}$ , and  $\mathbf{b}_1^{(\ell)}, \mathbf{b}_2^{(\ell)}$  are biases. We use  $r=2$  in our implementation, and Norm denotes either LayerNorm or BatchNorm depending on configuration.

#### B. DARTS-GT

##### Problem. Layer-wise Operation Selection

For a Graph Transformer with  $L$  layers and a set of candidate operations  $\mathcal{O} = \{o_1, o_2, \dots, o_n\}$ , where  $|\mathcal{O}| = n$ , the goal is to find the optimal sequence of layer-wise operations from the search space  $\mathcal{S}$  of  $n^L$  possible configurations leading to the optimal architecture  $\mathcal{A}^*$ , an *ordered sequence of layer-wise operations* (one per layer):

$$\mathcal{A}^* = \{o^{(1)*}, o^{(2)*}, \dots, o^{(L)*}\} \quad (10)$$

Here  $o^{(\ell)*} \in \mathcal{O}$  denotes the selected operation at layer  $\ell$ . Formally, this architecture search is formulated as:

$$\mathcal{A}^* = \arg \min_{\mathcal{A} \in \mathcal{S}} \mathcal{L}_{\text{val}}(w^*(\mathcal{A}), \mathcal{A}) \quad (11)$$

subject to:

$$w^*(\mathcal{A}) = \arg \min_w \mathcal{L}_{\text{train}}(w, \mathcal{A}) \quad (12)$$

where  $\mathcal{S}$  denotes the search space of all  $n^L$  architectures,  $w$  represents the network weights, and  $w^*(\mathcal{A})$  are the optimal weights for architecture  $\mathcal{A}$ . The following sections describe the approach we chose to solve the architecture search problem and the set of candidate operations, and hence, the search space.

1) *Bilevel Optimization via DARTS*: Enumerating all possible configurations in  $\mathcal{S}$  is generally infeasible, so we adopt the Differential Architecture Search (DARTS) [27] framework to solve the optimization problem formulated by equations 11 and 12. DARTS reformulates discrete selection as a continuous, differentiable optimization. Specifically, it introduces a softmax relaxation over candidate operations, casting the architecture search as the following bilevel optimisation problem [38]:

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}_{\text{val}}(w^*(\alpha), \alpha, \mathcal{D}_{\text{DARTSval}}) \\ \text{s.t.} \quad & w^*(\alpha) = \arg \min_w \mathcal{L}_{\text{train}}(w, \alpha, \mathcal{D}_{\text{DARTStrain}}) \end{aligned} \quad (13)$$

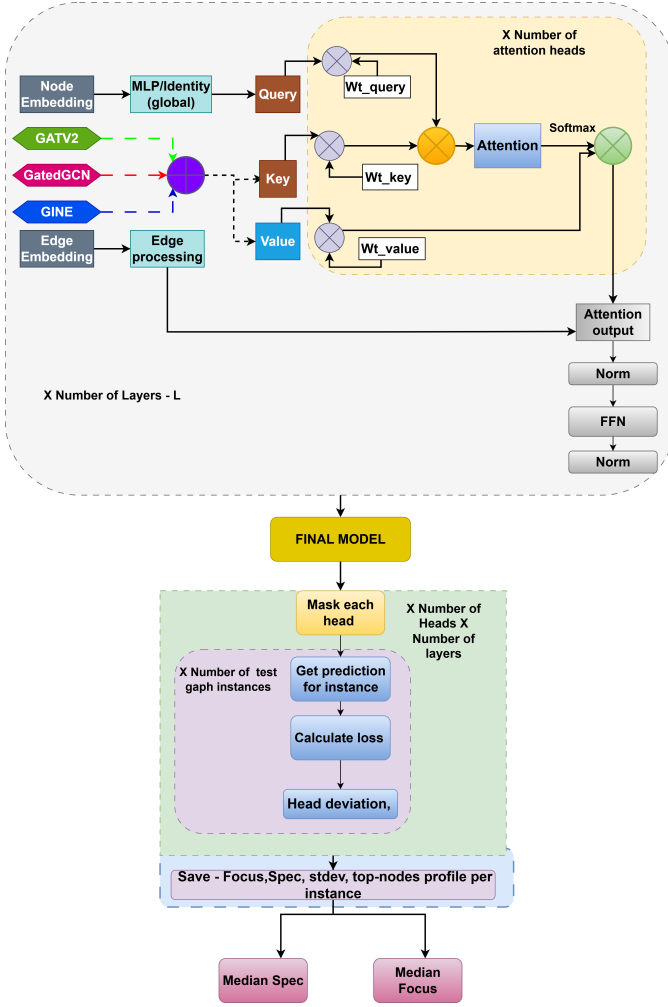


Fig. 1: **DARTS-GT with interpretability framework.** Node features produce *Query* via a global MLP/identity, while *Key, Value* come from a *single* GNN applied once per layer and *shared across all heads*; heads differ only in their per-head linear projections, reducing cost versus per-head GNNs. The violet disc denotes the DARTS *mixed operation* over candidate GNNs (GINE, GATv2, GatedGCN) during search; *after search, exactly one GNN is fixed per layer* (e.g. final searched network in the supplementary). Grey discs indicate the *per-head linear transformations* (matrix multiplications with learned weights). The yellow  $\times$  is the score computation  $QueryKey^T$  (followed by Softmax), and the green  $\times$  is the matrix product of the *Softmax* expression with *Value* yielding the attention output. An *edge pathway* (edge embedding  $\rightarrow$  edge processing) supplies an optional gated residual added before *Norm-FFN-Norm*. **Interpretability (bottom):** on each test graph, mask one head at a time and recompute loss to obtain *Head Deviation*; from these, compute per-instance *Specialization* (spread across heads) and *Focus* (agreement of top- $k$  heads on top- $r\%$  nodes), then report dataset medians.

where  $w$  represents the network weights and  $\alpha = \{\alpha_o^{(\ell)} \mid o \in \mathcal{O}, \ell \in \{1, \dots, L\}\}$  are continuous architecture parameters, with  $\alpha_o^{(\ell)} \in \mathbb{R}$  representing the architecture weight (importance factor) for the operation  $o$  at layer  $\ell$ . Rather than solving the full bilevel optimization, we adopt the efficient first-order approximation of DARTS [27], which avoids Hessian-vector products. In practice, we first randomly shuffle the entire training dataset  $\mathcal{D}_{\text{train}}$  and partition it into darts-training split,  $\mathcal{D}_{\text{DARTS}_{\text{train}}}$  for updating weights  $w$  and darts-validation split,  $\mathcal{D}_{\text{DARTS}_{\text{val}}}$  for updating architecture parameters  $\alpha$ , alternating the single gradient steps [27]. More details about this approach can be found in [27]. Although the second-order DARTS update is more accurate, it requires expensive Hessian-vector products and offers limited practical gains in our setting.

2) *Search Space Definition:* In our work, we use the following set of candidate operations:  $\mathcal{O} = \{\text{GINE [39], GATv2 [40], GatedGCN [32]}\}$ . Hence,  $|\mathcal{O}| = n = 3$  leads to a search space of  $3^L$  architectures.

The following operators were selected because they represent very distinct message-passing paradigms that capture complementary aspects of graph structure.

- **GINE:** Extends GIN [31] with edge features via permutation-invariant MLP aggregation,
- **GATv2:** Employs dynamic, edge-aware attention-based aggregation,
- **GatedGCN:** Uses edge-gated message propagation with soft edge selection.

This diversity ensures that our search space covers complementary inductive biases while remaining computationally efficient.

3) *Asymmetric Attention Design:* Departing from the standard transformer formalism outlined in the Preliminaries section, the novelty of our approach includes an *asymmetric design*: queries are derived directly from node features, while keys and values are produced by layer-specific GNN operators. This is initially motivated by the GMH design of GMT [26], where they demonstrated that decoupling the computational sources for the attention components enhances graph-aware representations. In GMT, node features are first transformed by a GNN to obtain  $H$ , with queries taken directly as  $Q = H$  and each head computing its own keys and values via an additional GNN applied to  $H$ . This asymmetry separates structural encoding from feature representation, but at the cost of computational inefficiency, since every key and value inside every head requires an independent GNN. In contrast, our design takes node features directly into  $Q$ , and shares the GNN operator across heads within a layer, producing an efficient formulation that still preserves the benefits of feature-structure decoupling. More broadly, rather than augmenting Graph Transformers externally with auxiliary MPNNs, a widely practiced approach [6], [16], we demonstrate that graph awareness can emerge from an architectural redesign of the attention mechanism itself.

$$\mathbf{Q}_m^{(\ell)} = \phi(\mathbf{Z}^{(\ell)})\mathbf{W}_{Q,m}^{(\ell)} \quad (14)$$

$$\mathbf{K}_m^{(\ell)} = o^{(\ell)}(\mathbf{Z}^{(\ell)}, E)\mathbf{W}_{K,m}^{(\ell)} \quad (15)$$

$$\mathbf{V}_m^{(\ell)} = o^{(\ell)}(\mathbf{Z}^{(\ell)}, E)\mathbf{W}_{V,m}^{(\ell)} \quad (16)$$

where  $\phi$  is a learned transformation (implemented as a two-layer MLP when additional expressiveness is beneficial, or identity otherwise) and  $o^{(\ell)}$  is the operator (GNN variant) at layer  $\ell$  and  $o^{(\ell)}(\mathbf{Z}^{(\ell)}, E) \in \mathbb{R}^{n \times d}$ ;  $\mathbf{W}_{Q,m}^{(\ell)}, \mathbf{W}_{K,m}^{(\ell)}, \mathbf{W}_{V,m}^{(\ell)} \in \mathbb{R}^{d \times d_m}$ . This design decouples the query space from the structural encoding, allowing the attention mechanism to learn optimal feature-structure alignments through the query-key interaction.

4) *Continuous relaxation (DARTS).*: During the search phase, DARTS [27] relaxes the selection of discrete operators into a continuous mixture. At each layer  $\ell$ , instead of selecting a single operator, we compute a weighted combination:

$$\bar{o}^{(\ell)}(\mathbf{Z}^{(\ell)}, E) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(\ell)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(\ell)})} \cdot o(\mathbf{Z}^{(\ell)}, E) \quad (17)$$

where:

- $\bar{o}^{(\ell)}$  denotes the *mixed* (continuous) operator at layer  $\ell$ , distinguished from the final discrete selection,
- $\alpha_o^{(\ell)} \in \mathbb{R}$  is the learnable architecture parameter for operator  $o$  at layer  $\ell$ ,
- $o(\mathbf{Z}^{(\ell)}, E)$  is the output of applying operator  $o$  to the node features  $\mathbf{Z}^{(\ell)}$  and edge set  $E$ .

a) *Use in attention during search.*: During search, the mixed operator output  $\bar{o}^{(\ell)}(\mathbf{Z}^{(\ell)}, E)$  provides the structural path for the keys and values:

$$\mathbf{K}_m^{(\ell)} = \bar{o}^{(\ell)}(\mathbf{Z}^{(\ell)}, E)\mathbf{W}_{K,m}^{(\ell)}, \quad (18a)$$

$$\mathbf{V}_m^{(\ell)} = \bar{o}^{(\ell)}(\mathbf{Z}^{(\ell)}, E)\mathbf{W}_{V,m}^{(\ell)}. \quad (18b)$$

We compute  $\bar{o}^{(\ell)}(\mathbf{Z}^{(\ell)}, E)$  once per layer and share it across all heads to form  $\mathbf{K}_m^{(\ell)}$  and  $\mathbf{V}_m^{(\ell)}$ ,  $m \in [1, \dots, M]$ . After discretization,  $\bar{o}^{(\ell)}$  is replaced by the selected operator  $o^{(\ell)*}$  (see 5) below). Attention may be dense or sparse, as that only changes the softmax normalization (all-pairs vs. edge-softmax) and does not alter the source of  $\mathbf{K}_m, \mathbf{V}_m$ .

5) *Discretization and Final Architecture Training.*: Upon search completion, we derive the discrete architecture by selecting the operator with maximum architecture weight at each layer:

$$o^{(\ell)*} = \arg \max_{o \in \mathcal{O}} \alpha_o^{(\ell)} \quad (19)$$

Following the standard DARTS protocol, the final discretized architecture  $\mathcal{A}^* = \{o^{(1)*}, \dots, o^{(L)*}\}$  is then re-trained from scratch using the full training data. In our implementation, we instantiate a fresh model containing only the selected operations per layer. The algorithm is discussed in Algorithm 1.

---

**Algorithm 1** DARTS-GT: Two-Phase Architecture Search
 

---

**Require:** Graph dataset  $\mathcal{D}$ , candidate operations  $\mathcal{O}$ ,  $L$  layers  
**Ensure:** Optimized architecture  $\mathcal{A}^*$  with trained weights

- 1: **Phase 1: Architecture Search (first-order) with Mixed Operations**
  - 2: Randomly partition  $\mathcal{D}_{\text{train}}$  into  $\mathcal{D}_{\text{DARTS}_{\text{train}}}$  (60%) and  $\mathcal{D}_{\text{DARTS}_{\text{val}}}$  (40%)
  - 3: Initialize architecture parameters to zeroes,  $\alpha = \{\alpha_o^{(\ell)}\}$  for all  $o \in \mathcal{O}$ ,  $\ell \in [L]$
  - 4: Initialize network weights  $w$
  - 5: **for** epoch = 1 to  $E_{\text{search}}$  **do**
  - 6:   Sample mini-batch from  $\mathcal{D}_{\text{DARTS}_{\text{train}}}$  and  $\mathcal{D}_{\text{DARTS}_{\text{val}}}$
  - 7:   Update  $w$  by  $\nabla_w \mathcal{L}_{\text{train}}(w, \alpha)$    ▷ First-order step (no Hessian terms)
  - 8:   Update  $\alpha$  by  $\nabla_{\alpha} \mathcal{L}_{\text{val}}(w, \alpha)$    ▷ First-order step   ▷ Single gradient step
  - 9: **end for**
  - 10: **Extract optimal operations:**
  - 11: **for** each layer  $\ell \in [L]$  **do**
  - 12:    $o^{(\ell)*} \leftarrow \arg \max_{o \in \mathcal{O}} \alpha_o^{(\ell)}$
  - 13: **end for**
  - 14:  $\mathcal{A}^* \leftarrow \{o^{(1)*}, o^{(2)*}, \dots, o^{(L)*}\}$
  - 15: **Phase 2: Discrete Architecture Training**
  - 16: Instantiate fresh model  $\mathcal{M}_{\text{discrete}}$  with architecture  $\mathcal{A}^*$
  - 17: Train  $\mathcal{M}_{\text{discrete}}$  on full  $\mathcal{D}_{\text{train}}$  for  $E_{\text{final}}$  epochs
  - 18: **return** Trained model  $\mathcal{M}_{\text{discrete}}$
- 

6) *Edge Residual Stream.*: We inject a gated edge term into the post-attention residual (before the first normalization). For edge features  $\mathbf{e}_{(i,j)} \in \mathbb{R}^{d_e}$ :

$$\tilde{\mathbf{e}}_{(i,j)} = \text{MLP}_{\text{edge}}(\mathbf{e}_{(i,j)}), \quad (20)$$

$$\mathbf{h}_j^{\text{edge}} = \sum_{(i,j) \in E} \tilde{\mathbf{e}}_{(i,j)}, \quad (21)$$

$$\mathbf{H}^{\text{edge}} = [\mathbf{h}_1^{\text{edge}} \dots \mathbf{h}_n^{\text{edge}}]^\top \in \mathbb{R}^{n \times d}, \quad (22)$$

$$\mathbf{Y}^{(\ell)} \leftarrow \mathbf{Y}^{(\ell)} + \sigma_g(\gamma^{(\ell)}) \mathbf{H}^{\text{edge}} \quad (23)$$

$\text{MLP}_{\text{edge}} : \mathbb{R}^{d_e} \rightarrow \mathbb{R}^d$  is a two-layer MLP with dropout;  $\sigma_g(x)$  is the sigmoid function, and  $\gamma^{(\ell)} \in \mathbb{R}$  is a per-layer learnable scalar controlling the gate;  $\mathbf{Y}^{(\ell)}$  is the pre-residual attention output.  $\text{MLP}_{\text{edge}}$  is shared across layers and  $\mathbf{H}^{\text{edge}}$  is computed once per forward pass. Please see Supplementary Table I for ablation studies related to the edge residual stream. This edge residual stream operates independently of the selected GNN operations, allowing the model to leverage edge information even when the discretized architecture uses operations with limited utilization of edge features.

7) *Computational Cost.*: Our proposed asymmetric block applies a *single* GNN per layer to produce the structural representations used in both  $K$  and  $V$ . In contrast, GMT's Graph Multi-head Attention (GMH) [26] instantiates *per-head* GNNs for both keys and values, i.e.  $\{\text{GNN}_i^K, \text{GNN}_i^V\}_{i=1}^M$ . Consequently, the GNN forward cost in GMH scales as  $\mathcal{O}(2M \cdot \text{cost}_{\text{GNN}})$ , while in our design it is  $\mathcal{O}(1 \cdot \text{cost}_{\text{GNN}})$  per layer, where  $\text{cost}_{\text{GNN}}$  denotes the computational cost of a single GNN forward pass per layer. Thus, we preserve the fea-

ture–structure interaction ( $Q$  from features;  $K, V$  from GNN) while reducing the structural computation across heads. From the perspective of architecture search overhead, DARTS [27] uses a first-order approximation that avoids Hessian–vector product overhead.

### C. Interpretability Framework

1) *Framework Overview*: We propose a systematic framework to quantify Graph Transformer interpretability through head-level and node-level analysis. Our approach goes beyond visualization and yields *quantitative, instance-specific* evidence of which components *causally* drive the predictions. We introduce *Head-Deviation*, *Specialization*, and *Focus*, aligned with established interpretability principles where *fidelity*, *sparsity*, and *consensus* are standard evaluation criteria [29], [30]. Because these metrics are ablation-based, they measure causal contribution rather than correlation, allowing robust, fair comparisons of interpretability across architectures.

a) *Masking protocol (causal ablation)*: For head  $m$  in layer  $\ell$ , we implement a weight-level ablation that removes its contribution at inference. Concretely, we *silence head  $m$  in layer  $\ell$* , by zeroing its pre-concatenation output  $Y_m^{(\ell)}$  (equivalently, zeroing the corresponding slice of  $\mathbf{W}_{\text{out}}$  in Eq. 6). All other parameters, including the outputs of other heads, layer norms, and feed-forward sublayers, are left unchanged. The models are evaluated in `eval` mode (no dropout), with a fixed seed and without any retraining or adaptation.

2) *Head Deviation*: To quantify the importance of individual attention heads, we measure how their removal affects the task loss. For a graph  $G_i$  with the true label  $y_i$ , let  $f(G_i)$  denote the prediction of the model and  $f^{-m,\ell}(G_i)$  denote the prediction with head  $m$  in layer  $\ell$  masked.

**Definition 1** (Head Deviation). *The head deviation measures the change in task loss when head  $m$  in layer  $\ell$  is masked:*

$$\delta_{m,\ell,i} = \text{loss}(f^{-m,\ell}(G_i), y_i) - \text{loss}(f(G_i), y_i), \quad (24)$$

Here,  $\text{loss}(\cdot, \cdot)$  denotes the loss appropriate to the machine learning task (e.g., MAE for regression, binary cross-entropy for binary classification, cross-entropy for multi-class classification). We quantify head importance through *causal intervention*: the deviation  $\delta_{m,\ell,i}$  measures how the prediction changes when head  $m$  in layer  $\ell$  is ablated, providing direct evidence of its functional necessity rather than mere correlational patterns. This approach asks “*what happens without this head?*”, distinguishes truly important components from those that merely appear salient in visualizations.

a) *Head ranking*: For each instance, we rank the heads by their deviation  $\delta_{m,\ell,i}$  in descending order. *Beneficial heads* ( $\delta_{m,\ell,i} > 0$ , masking increases loss) thus appear first, while *harmful heads* ( $\delta_{m,\ell,i} < 0$ ) appear last. When we need a sign-agnostic view, we report the magnitude  $|\delta_{m,\ell,i}|$  as a measure of head impact.

3) *Global Interpretability Metrics*: Beyond individual head importance, we summarize *how interpretable* the model is for a given instance across all heads. We define two complementary metrics: *specialization* (importance concentration or head-distinguishability) and *focus* (consensus of important heads on nodes).

**Definition 2** (Specialization). *For graph  $G_i$ ,*

$$\text{Spec}(G_i) = \text{std}\left(\{\delta_{m,\ell,i}\}_{m \in \mathcal{M}, \ell \in \mathcal{L}}\right), \quad (25)$$

where  $\text{std}(\cdot)$  denotes the standard deviation across all heads’ deviations with  $\mathcal{M} = \{1, \dots, M\}$ ,  $\mathcal{L} = \{1, \dots, L\}$  where  $M$  is the number of attention-heads and  $L$  total number of layers, defined earlier.

High specialization indicates that heads are easily distinguishable by their causal impact, whereas low specialization suggests similar head impacts, making it harder to isolate drivers unless complemented by the focus metric.

**Definition 3** (Focus). *Let  $\mathcal{P}_i^* = \{p_1, \dots, p_k\}$  be the top- $k$  head-layer pairs for instance  $G_i$ , ranked according to the head-ranking rule defined above, where  $p_j = (i'_j, i''_j)$ ,  $j = 1, \dots, k$ ,  $i'_j \in \mathcal{M}$ ,  $i''_j \in \mathcal{L}$  and  $i'_j, i''_j$  are index variables. For a pair  $p$ , we define  $N_p$  as the set of top- $r\%$  nodes by incoming attention mass for that head on  $G_i$ .  $N_p$  contains the top- $r\%$  nodes by the following values:*

- **Dense attention**: with head softmax matrix  $\mathbf{S}_m^{(\ell)}$ , incoming mass for node  $j$  is  $\sum_i [\mathbf{S}_m^{(\ell)}]_{ij}$ ;
- **Sparse attention**: with edge-softmax  $s_{m,ij}$  on  $(i, j) \in E$ , incoming mass for node  $j$  is  $\sum_{i:(i,j) \in E} s_{m,ij}$ .

The Focus metric computes the average Jaccard similarity between all unique pairs:

$$\text{Focus}(G_i) = \frac{1}{\binom{k}{2}} \sum_{j=1}^{k-1} \sum_{m=j+1}^k \frac{|N_{p_j} \cap N_{p_m}|}{|N_{p_j} \cup N_{p_m}|}, \quad (26)$$

where  $\binom{k}{2} = \frac{k(k-1)}{2}$  is the number of unique head pairs.

Unless stated otherwise, in our implementation, we use  $k=5$  and a  $r = 10\%$  node fraction. This metric quantifies the degree of consensus among the most causally important heads, i.e. measure top- $k$ -head agreement on top- $r\%$  nodes. A high Focus score indicates that the model’s key components agree on the same salient graph substructures, simplifying interpretation.

4) *Dataset-Level Aggregation*: For dataset  $\mathcal{D}$  with  $N$  graphs, we use the median for robustness to instance variability and outliers:

$$\overline{\text{Spec}} = \text{median}_{i=1}^N \{\text{Spec}(G_i)\}, \quad (27)$$

$$\overline{\text{Focus}} = \text{median}_{i=1}^N \{\text{Focus}(G_i)\}. \quad (28)$$

5) *Interpretation Guidelines*: The combination of specialization and focus determines how interpretable an instance-level prediction is.

- **High Spec + High Focus**: Most interpretable. Few heads dominate and agree on important nodes.
- **High Spec + Low Focus**: Important heads are identifiable but attend to different nodes, suggesting complementary strategies.
- **Low Spec + High Focus**: Uniform head importance but consensus on nodes. Clear node-level attribution despite unclear head roles.
- **Low Spec + Low Focus**: Least interpretable. Uniform importance with disparate attention patterns.

**Algorithm 2** Graph Transformer Interpretability Analysis

---

**Require:** Model  $f$ , Test set  $\mathcal{D}_{\text{test}}$ , Top heads  $k$   
**Ensure:**  $\{\text{Spec}(G_i), \text{Focus}(G_i)\}$  for each  $G_i$

```

1: Phase 1: Compute Head Deviations
2: for  $\ell = 1$  to  $L$  do
3:   for  $m = 1$  to  $M$  do
4:      $f^{-m,\ell} \leftarrow$  model with head  $m$  in layer  $\ell$  masked
5:     for  $G_i \in \mathcal{D}_{\text{test}}$  do
6:        $e_{\text{base}} \leftarrow \text{loss}(f(G_i), y_i)$ 
7:        $e_{\text{masked}} \leftarrow \text{loss}(f^{-m,\ell}(G_i), y_i)$ 
8:        $\delta_{m,\ell,i} \leftarrow e_{\text{masked}} - e_{\text{base}}$ 
9:     end for
10:   end for
11: end for
12: Phase 2: Compute Metrics
13: for  $G_i \in \mathcal{D}_{\text{test}}$  do
14:    $\Delta_i \leftarrow \{\delta_{m,\ell,i} : m = 1..M, \ell = 1..L\}$ 
15:    $\text{Spec}(G_i) \leftarrow \text{std}(\Delta_i)$ 
16:    $\mathcal{P}_i^* \leftarrow$  top- $k$  pairs of head  $m$  in layer  $\ell$ :  $(m, \ell)$  ranked
    by  $\delta_{m,\ell,i}$  (descending)
17:   for  $p \in \mathcal{P}_i^*$  do
18:      $N_{p_i} \leftarrow$  top- $r\%$  nodes by incoming attention mass
    for  $p$  on  $G_i$ 
19:   end for
20:    $\text{Focus}(G_i) \leftarrow \frac{1}{\binom{k}{2}} \sum_{j=1}^{k-1} \sum_{m=j+1}^k \frac{|N_{p_j} \cap N_{p_m}|}{|N_{p_j} \cup N_{p_m}|}$ 
21: end for
22: return  $\{(\text{Spec}(G_i), \text{Focus}(G_i)) : G_i \in \mathcal{D}_{\text{test}}\}$ 

```

---

$\text{Focus} \in [0, 1]$  measures consensus on node attribution.  $\text{Spec} \in [0, \infty)$  measures concentration of head importance or head-distinguishability. Together, they reveal which heads matter (specialization) and what they attend to (focus), providing faithful, comparable, instance-level interpretability and a principled route to model-level comparison. The complete process is depicted in Algorithm 2.

## IV. EXPERIMENTAL RESULTS

In this section, we comprehensively evaluate DARTS-GT in multiple aspects to establish both its performance advantages and its interpretability characteristics. Our experimental analysis comprises five key components:

- 1) Direct comparison with standard (vanilla) Graph Transformer [19] to isolate the impact of our architectural innovations;
- 2) Comparison against state-of-the-art baselines including hand-designed GNNs, recent Graph Transformers, and neural architecture search methods;
- 3) Analysis of heterogeneous versus uniform architectures and random-search obtained architectures, to validate the benefits of layer-wise GNN diversity through DARTS search;
- 4) Ablation studies examining the role of asymmetric design while understanding edge features role is given in the supplementary;

- 5) Quantitative interpretability analysis using our novel framework to assess model transparency. Our evaluation of the interpretability framework operates on two levels.

- a) At the population level, we quantify overall model interpretability through metrics (Specialization, Focus) that characterize potentially how easily practitioners can identify important heads and nodes driving predictions.
- b) At the instance level, we demonstrate how our framework reveals specific graph components critical for individual predictions.

To ensure a robust evaluation, we perform experiments with 4 random seeds for SOTA baseline comparisons (Tables III-IV) and vanilla-GT versus DARTS-GT comparisons (Table II) following the benchmarking protocol introduced in [41], while using 3 seeds for architectural analysis experiments (Tables V-VII, VIII) due to computational constraints, which is also widely followed protocol [6].

## A. Dataset and Task Selection

We evaluated DARTS-GT against three categories of baselines: SOTA GNN architectures (e.g. GCN [17], GIN [31], GatedGCN [32], PNA [42]), recent Graph Transformer models (e.g. SAN [7], Graphormer [8], GPS [6]), and neural architecture search methods for Graph Transformers (UGAS [16], EGTAS [13], AutoGT [14]). Our evaluation spans eight diverse datasets that encompasses both graph-level and node-level prediction tasks, as detailed in Table I.

TABLE I: Dataset Statistics and Characteristics

| Dataset         | #Graphs | Avg# Nodes | Avg# Edges | Prediction Level | Task             |
|-----------------|---------|------------|------------|------------------|------------------|
| ZINC            | 12,000  | 23.2       | 24.9       | Graph            | Regression       |
| OGBG-MolHIV     | 41,127  | 25.5       | 27.5       | Graph            | Binary Cls.      |
| OGBG-MolPCBA    | 437,929 | 26.0       | 28.1       | Graph            | Multi-task Cls.  |
| MalNet-Tiny     | 5,000   | 1,410.3    | 2,859.9    | Graph            | Multi-class Cls. |
| PATTERN         | 14,000  | 118.9      | 3,039.3    | Node             | Binary Cls.      |
| CLUSTER         | 12,000  | 117.2      | 2,150.9    | Node             | Multi-class Cls. |
| Peptides-Struct | 15,535  | 150.9      | 307.3      | Graph            | Multi-task Reg.  |
| Peptides-Func   | 15,535  | 150.9      | 307.3      | Graph            | Multi-task Cls.  |

The selected benchmarks [6] provide comprehensive coverage in multiple dimensions. This range accommodates diverse tasks such as graph regression (single and multi-task), graph classification (binary, multi-class, and multi-task), and node-level predictions (binary and multi-class classifications). The datasets exhibit substantial variation in scale, from small molecular graphs in ZINC (averaging 23 nodes) to large-scale graphs in MalNet-Tiny (averaging 1,410 nodes), testing the ability of our architecture to handle varying structural complexities. Dataset sizes range from 5,000 to more than 437,000 graphs, ensuring robust evaluation across different dataset sizes. This diversity, spanning molecular property prediction (OGBG), biological sequences (Peptides), and synthetic benchmarks (PATTERN, CLUSTER), demonstrates that our architectural choices generalize beyond domain-specific patterns. Notably, the inclusion of two Long-Range Graph Benchmark (LRGB) datasets, Peptides-Struct and Peptides-Func, facilitated evaluation of the capacity of our model to capture long-range dependencies, a critical challenge for Graph Transformers.

### B. Comparison Against Standard Graph Transformers

Table II presents the performance comparison of DARTS-GT with the standard Graph Transformer (Vanilla-GT) introduced by Dwivedi et al. [19], which forms the foundation of many subsequent works [16]. Unlike recent approaches that enhance graph transformers through parallel MPNN paths [6], novel positional encoding [7], or structural encoding-augmented attention [8], DARTS-GT alters the core attention mechanism through asymmetric query-key value computation and heterogeneous per-layer GNN selection. This is a significant departure from standard GT where Q, K, V share the same computational source without GNNs. This experiment quantifies the improvement of DARTS-GT over the baseline it extends. For rigorous comparison, we standardized most experimental conditions using the GPS configuration [6]. We extend our implementation from their opensource repository, adopting their fixed positional encoding, hidden dimensions, attention heads, dropout rates, and pre/post-processing components for both models. Only the layer count was adjusted to maintain a comparable parameter budget range. This controlled setup (Supplementary-Tables II,III,IV), prioritizing methodological clarity over peak performance, ensures that observed improvements stem from design changes rather than hyperparameter tuning or auxiliary components.

TABLE II: Performance Comparison: Vanilla-GT vs DARTS-GT across diverse datasets with Dense and Sparse Attention Mechanisms.  $\Delta$  represents the relative improvement of DARTS-GT over Vanilla-GT, calculated as  $\frac{|\text{DARTS-GT} - \text{Vanilla-GT}|}{|\text{Vanilla-GT}|} \times 100\%$ , where improvement direction accounts for metric type (decrease for MAE, increase for others). Shown is the mean  $\pm$  std across 4 seeds. Best performances are highlighted in **bold**.

| Attn.  | Model           | Pep-Struct<br>MAE↓ | Pep-Func<br>AP↑   | PATTERN<br>Acc↑   | MolHIV<br>AUROC↑  | MolPCBA<br>AP↑    |
|--------|-----------------|--------------------|-------------------|-------------------|-------------------|-------------------|
|        | Vanilla-GT      | 0.254±.002         | 0.623±.009        | 0.678±.164        | 0.761±.009        | 0.048±.007        |
| Dense  | <b>DARTS-GT</b> | <b>0.247±.001</b>  | <b>0.669±.004</b> | <b>0.852±.001</b> | <b>0.766±.016</b> | <b>0.201±.008</b> |
|        | $\Delta(\%)$    | 2.8%               | 7.4%              | 25.7%             | 0.7%              | 318%              |
|        | Vanilla-GT      | 0.268±.001         | 0.557±.005        | 0.681±.199        | 0.758±.009        | 0.241±.032        |
| Sparse | <b>DARTS-GT</b> | <b>0.263±.001</b>  | <b>0.609±.014</b> | <b>0.867±.000</b> | <b>0.776±.006</b> | <b>0.258±.004</b> |
|        | $\Delta(\%)$    | 1.9%               | 9.3%              | 27.3%             | 2.4%              | 7.1%              |

We tested both dense (full attention) and sparse attention mechanisms to understand how architectural improvements translate across different computational settings. The results reveal consistent gains across all benchmarks, with particularly striking improvements on PATTERN (25-27% relative accuracy increase) and OGBG-MolPCBA (318% relative improvement in average precision for dense attention). Even on long-range benchmarks (Peptides), where standard transformers already perform reasonably, DARTS-GT achieves meaningful non-trivial improvements over the baseline. Our comparison reveals that standard Graph Transformers, when isolated from auxiliary components, exhibit significant limitations, particularly evident in MolPCBA dense attention. This raises a critical question about hybrid architectures like GPS (which uses dense attention with parallel MPNN): do performance

TABLE III: Test Performance on Standard Benchmarks. Shown is the mean  $\pm$  std across 4 seeds. Best performance is **bolded** while second best is underlined. Datasets where we have outperformed GPS (but not best) is marked with asterisks

| Model              | ZINC<br>MAE ↓      | MALNET<br>Acc ↑     | PATTERN<br>Acc ↑    | CLUSTER<br>Acc ↑    |
|--------------------|--------------------|---------------------|---------------------|---------------------|
| GCN [17]           | 0.367±0.011        | NA                  | 71.892±0.334        | 68.498±0.976        |
| GIN [31]           | 0.526±0.051        | NA                  | 85.387±0.136        | 64.716±1.553        |
| GAT [18]           | 0.384±0.007        | NA                  | 78.271±0.186        | 70.587±0.447        |
| GatedGCN [12]      | 0.282±0.015        | 0.9223±0.65         | 85.568±0.088        | 73.840±0.326        |
| GatedGCN-LSPE [43] | 0.090±0.001        | NA                  | NA                  | NA                  |
| PNA [42]           | 0.188±0.004        | NA                  | NA                  | NA                  |
| DGN [44]           | 0.168±0.003        | NA                  | 86.680±0.034        | NA                  |
| GSN [45]           | 0.101±0.010        | NA                  | NA                  | NA                  |
| CIN [46]           | 0.079±0.006        | NA                  | NA                  | NA                  |
| CRaW1 [47]         | 0.085±0.004        | NA                  | NA                  | NA                  |
| GIN-AK+ [48]       | 0.080±0.001        | NA                  | <u>86.850±0.057</u> | NA                  |
| EGTAS [13]         | 0.075±0.073        | NA                  | 86.742±0.053        | <b>79.236±0.215</b> |
| UGAS [16]          | NA                 | NA                  | <b>86.89±0.02</b>   | 78.14±0.21          |
| SAN [7]            | 0.139±0.006        | NA                  | 86.581±0.037        | 76.691±0.65         |
| Graphormer [8]     | 0.122±0.006        | NA                  | NA                  | NA                  |
| K-Subgraph SAT [2] | 0.094±0.008        | NA                  | 86.848±0.037        | 77.856±0.104        |
| EGT [23]           | 0.108±0.009        | NA                  | 86.821±0.020        | <u>79.232±0.348</u> |
| GPS [6]            | <u>0.070±0.004</u> | <u>0.9264±0.78</u>  | 86.685±0.059        | 78.016±0.180        |
| <b>DARTS-GT</b>    | <b>0.066±0.003</b> | <b>0.9325±0.006</b> | 86.68±0.036         | 78.299±0.07*        |

gains stem primarily from the MPNN path itself, or from the synergistic interaction where MPNN features and transformer attention are jointly fed into subsequent layers? The DARTS-GT design removes this ambiguity by demonstrating competitive performance using only enhanced attention mechanisms without any parallel MPNN component. This isolation proves that the Graph Transformer itself can be made structurally aware through architectural innovation in the attention mechanism, rather than relying on auxiliary components or cross-pathway synergies.

### C. Comparison Against SOTA leaderboard

Table III and Table IV benchmark DARTS-GT against three categories of methods: hand-designed GNNs (e.g., GCN, GIN, GatedGCN, PNA), Graph Transformers (e.g., SAN, Graphormer, GPS), and NAS-based GT approaches (EGTAS, AutoGT, UGAS). Each method is reported under its validation-selected configuration (dense or sparse). Full settings are available in supplementary. DARTS-GT achieves state-of-the-art results on four datasets: ZINC (0.066), MalNet-Tiny (93.25%), and both LRGB benchmarks: Peptides-Func (0.669) and Peptides-Struc (0.246). Most notably, for the other four datasets, PATTERN, CLUSTER, MolHIV, MolPCBA, there are *distinct* best performing methods, namely EGTAS (for CLUSTER), UGAS (for PATTERN), CIN (for MolHIV) and CRAW1 (for MolPCBA). *Therefore, DARTS-GT substantially stands out as outperforming all methods in four datasets, while none of the other four highlighted methods or any other methods outperformed all methods for more than one dataset.*

For LRGB [49] benchmarks, DARTS-GT advances beyond both GPS and the recent UGAS method, with considerably

TABLE IV: Test Performance on OGB and LRGB Benchmarks. Shown is the mean  $\pm$  std across 4 seeds. Best performance is **bolded** while second best is underlined.

| Model               | MolHIV<br>AUROC $\uparrow$          | MOLPCBA<br>AP $\uparrow$            | PEP-FUNC<br>AP $\uparrow$         | PEP-STRUC<br>MAE $\downarrow$      |
|---------------------|-------------------------------------|-------------------------------------|-----------------------------------|------------------------------------|
| GCN                 | NA                                  | NA                                  | 0.5930 $\pm$ 0.002                | 0.3496 $\pm$ 0.0013                |
| GINE [39]           | NA                                  | NA                                  | 0.5498 $\pm$ 0.0079               | 0.3547 $\pm$ 0.0045                |
| GCN+virtualnode [6] | 0.7599 $\pm$ 0.0119                 | 0.2424 $\pm$ 0.0034                 | NA                                | NA                                 |
| GIN+virtualnode [6] | 0.7707 $\pm$ 0.0149                 | 0.2703 $\pm$ 0.0023                 | NA                                | NA                                 |
| GatedGCN            | NA                                  | NA                                  | 0.5864 $\pm$ 0.0077               | 0.3420 $\pm$ 0.0013                |
| GatedGCN-LSPE       | NA                                  | 0.267 $\pm$ 0.002                   | NA                                | NA                                 |
| PNA                 | 0.7905 $\pm$ 0.0132                 | 0.2838 $\pm$ 0.0035                 | NA                                | NA                                 |
| DeeperGCN [6]       | 0.7858 $\pm$ 0.0117                 | 0.2781 $\pm$ 0.0038                 | NA                                | NA                                 |
| DGN                 | 0.7970 $\pm$ 0.0097                 | 0.2885 $\pm$ 0.0030                 | NA                                | NA                                 |
| GSN(directional)    | 0.8039 $\pm$ 0.0090                 | NA                                  | NA                                | NA                                 |
| CIN                 | <b>0.8094<math>\pm</math>0.0057</b> | NA                                  | NA                                | NA                                 |
| CRaWl               | NA                                  | <b>0.2986<math>\pm</math>0.0025</b> | NA                                | NA                                 |
| GIN-AK+             | 0.7961 $\pm$ 0.0119                 | <u>0.2930<math>\pm</math>0.0044</u> | NA                                | NA                                 |
| ExpC [50]           | 0.7799 $\pm$ 0.0082                 | 0.2342 $\pm$ 0.0029                 | NA                                | NA                                 |
| AUTOGT              | 0.7495 $\pm$ 0.0102                 | NA                                  | NA                                | NA                                 |
| EGTAS               | 0.7981 $\pm$ 0.0117                 | NA                                  | NA                                | NA                                 |
| UGAS                | <u>0.7997<math>\pm</math>0.51</u>   | NA                                  | <u>0.667<math>\pm</math>0.007</u> | <u>0.247<math>\pm</math>0.002</u>  |
| SAN                 | 0.7785 $\pm$ 0.2470                 | 0.2765 $\pm$ 0.0042                 | NA                                | NA                                 |
| SAN+LAPPE [6]       | NA                                  | NA                                  | 0.6384 $\pm$ 0.0121               | 0.2683 $\pm$ 0.0043                |
| GraphTrans [12]     | NA                                  | 0.2761 $\pm$ 0.0029                 | NA                                | NA                                 |
| GPS [6]             | 0.7880 $\pm$ 0.0101                 | 0.2907 $\pm$ 0.0028                 | 0.6535 $\pm$ 0.0041               | 0.2500 $\pm$ 0.0005                |
| <b>DARTS-GT</b>     | 0.7766 $\pm$ 0.006                  | 0.2584 $\pm$ 0.004                  | <b>0.669<math>\pm</math>0.004</b> | <b>0.246<math>\pm</math>0.0006</b> |

lower variance on Peptides-Struc than the latter ( $\pm 0.0006$  vs UGAS's  $\pm 0.002$ ), indicating more stable solutions. The consistent gains across diverse tasks validate that differentiable architecture search discovers effective designs within the transformer framework. In addition to the SOTA performance achieved by the DARTS-GT discovered architectures, they exhibit enhanced interpretability characteristics, as demonstrated in sections IV-F and IV-G.

#### D. Dissecting DARTS-GT's Architecture Discovery Patterns

Figure 2 visualizes the GNN distributions discovered by DARTS across eight datasets, revealing prominent diversity in architectural choices. The selection patterns range from highly specialized (MolPCBA with 81% GINE dominance) to nearly uniform distributions (ZINC, MolHIV with balanced GNN usage). This demonstrates that DARTS adapts to dataset-specific requirements rather than converging to a universal solution, as expected from a neural architecture search (NAS) algorithm. Interestingly, node-level tasks (PATTERN, CLUSTER) favored GatedGCN. We could also observe the complete absence of GATV2 in Peptides datasets, coupled with GatedGCN's dominance (87.5% in Peptides-Func). However, these proportions only show that the GNN selection is utilised, but the layer-wise positioning of each GNN, which DARTS optimizes through gradient-based search, is equally critical. A dataset might use 50% GINE, but whether these appear in early or late layers inherently impacts feature extraction and

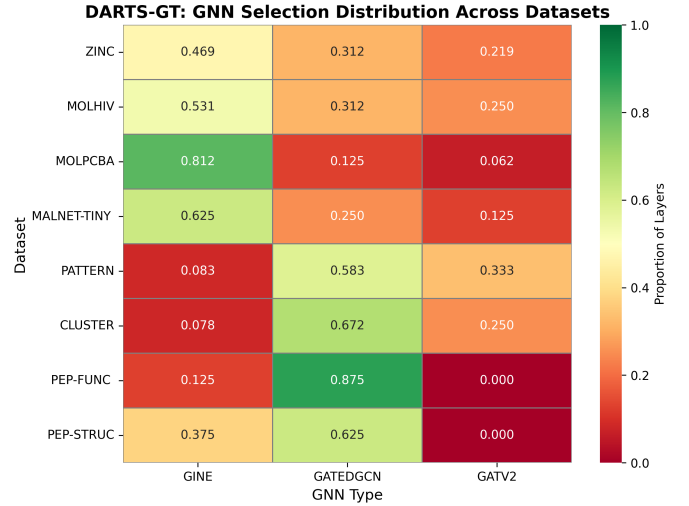


Fig. 2: Heatmap showing the proportion of each GNN type selected by DARTS-GT across different datasets. The values indicate fraction of total layers using each GNN operation, that is averaged over 4 seeds. Color intensity reflects the usage frequency as indicated

TABLE V: Comparison of DARTS-GT against uniform (homogeneous) and random-searched architectures across different attention types and dataset variety. Shown is the mean  $\pm$  std across 3 seeds. Best results are in **bold**.

| Dataset       | Attn-Type | Model Type             | Perf                                |
|---------------|-----------|------------------------|-------------------------------------|
| CLUSTER       | Sparse    | Random                 | 0.7388 $\pm$ 0.0741                 |
|               |           | Uniform-GINE           | 0.7688 $\pm$ 0.0018                 |
|               |           | Uniform-GATEDGCN       | 0.7047 $\pm$ 0.069                  |
|               |           | Uniform-GATV2          | 0.7805 $\pm$ 0.00213                |
|               |           | <b>DARTS-GT (ours)</b> | <b>0.7835<math>\pm</math>0.0004</b> |
| Peptides-Func | Dense     | Random                 | 0.646 $\pm$ 0.0048                  |
|               |           | Uniform-GINE           | 0.6555 $\pm$ 0.0093                 |
|               |           | Uniform-GATEDGCN       | 0.6718 $\pm$ 0.0035                 |
|               |           | Uniform-GATV2          | 0.6530 $\pm$ 0.0081                 |
|               |           | <b>DARTS-GT (ours)</b> | <b>0.6730<math>\pm</math>0.0056</b> |
| Malnet-Tiny   | Sparse    | Random                 | 0.919 $\pm$ 0.0081                  |
|               |           | Uniform-GINE           | 0.919 $\pm$ 0.0081                  |
|               |           | Uniform-GATEDGCN       | 0.929 $\pm$ 0.0060                  |
|               |           | Uniform-GATV2          | 0.922 $\pm$ 0.0098                  |
|               |           | <b>DARTS-GT (ours)</b> | <b>0.934<math>\pm</math>0.0057</b>  |

performance (examples of a few searched architectures are given in Supplementary Figure 2).

Although Figure 2 reveals that GatedGCN dominates in CLUSTER (67.2%), this prevalence does not indicate superiority as a standalone operation. Table V exposes a counter-intuitive phenomenon: *The most-used GNN in a heterogeneous architecture may not be the best performing uniform architecture*.

Table V compares DARTS-GT against uniform architec-

tures and random heterogeneous selection (mean of 3 different random configurations). Despite dominant operations in DARTS-GT architectures (e.g., GatedGCN at 67.2% in CLUSTER, 87.5% in Peptides-Func), uniform deployment of these same operations underperforms: uniform GatedGCN achieves 0.7047 vs DARTS-GT’s 0.7835 on CLUSTER and 0.6718 vs 0.6730 on Peptides-Func. On MalNet-Tiny, uniform GatedGCN outperforms uniform GINE despite GINE dominating (62.5%) the DARTS architecture. This demonstrates that no single operation is universally optimal, and “popular” operations in DARTS-GT architecture can underperform when deployed homogeneously. Although DARTS-GT’s gains appear modest, they align with typical graph benchmark improvements where SOTA advances are often under 1% (as seen in SAN, GPS, UGAS and our Tables III-IV). These gains are consistent across datasets/seeds, not isolated cases. Moreover, finding even the best uniform architecture requires testing  $n$  separate models, while the heterogeneous space grows exponentially ( $3^4 = 81$  for 4 layers;  $3^{10} = 59,049$  for 10 layers). DARTS-GT efficiently navigates this space without exhaustive enumeration.

Three key insights emerge: (1) DARTS discovers synergistic layer-wise combinations rather than repeating the strongest operation; (2) DARTS-GT consistently outperforms random heterogeneous architectures, confirming principled optimization beats chance; (3) DARTS-GT’s value lies in removing the rigid commitment to uniformity or heterogeneity in advance. It adaptively recovers strong uniform solutions when sufficient and heterogeneous mixtures when advantageous. Our contribution extends beyond numerical improvements. While prior GTs relied on auxiliary GNN paths or hand-designed uniform backbones, we show that the core transformer architecture itself can be systematically optimized. This reframes GT design as an open direction for principled exploration, where the demonstrated consistent improvements evidence yet unexplored directions in Graph Transformers.

#### E. Ablation Studies: Assymetric vs Symmetric attention

Although Table II demonstrated DARTS-GT’s superiority over Vanilla-GT (standard GT), a critical question remains: Is asymmetric attention design essential, or would a symmetric variant where  $Q_m$ ,  $K_m$  and  $V_m$  all derive from GNN outputs perform better?

To address this, we conducted controlled experiments across 3 seeds comparing our asymmetric design against its symmetric counterpart. All other experimental conditions remained identical between models.

The results obtained consistently favored the DARTS-GT asymmetric design over the symmetric variant in all configurations tested. This validates our architectural choice: decoupling queries from structural encoding ( $Q_m$  from features,  $K_m/V_m$  from GNNs) provides superior performance compared to the seemingly natural approach of deriving all attention components from the same GNN source.

#### F. Interpretability Analysis

Although performance gains demonstrate DARTS-GT’s effectiveness, modern applications increasingly demand inter-

TABLE VI: Ablation Study: Symmetric ( $Q_m$ ,  $K_m$  and  $V_m$ ) vs Asymmetric Attention Design (DARTS-GT design). Shown is the mean  $\pm$  std across 3 seeds. Best performance is **bolded**.

| Dataset       | Attn-Type | Model Type                   | Performance                         |
|---------------|-----------|------------------------------|-------------------------------------|
| CLUSTER       | Sparse    | Symmetric                    | 78.2 $\pm$ 0.0010                   |
|               |           | <b>DARTS-GT (Asymmetric)</b> | <b>78.35<math>\pm</math>0.0004</b>  |
| Peptides-Func | Dense     | Symmetric                    | 0.667 $\pm$ 0.009                   |
|               |           | <b>DARTS-GT (Asymmetric)</b> | <b>0.6730<math>\pm</math>0.0056</b> |
| MalNet-Tiny   | Sparse    | Symmetric                    | 0.923 $\pm$ 0.008                   |
|               |           | <b>DARTS-GT (Asymmetric)</b> | <b>0.9334<math>\pm</math>0.0057</b> |

pretable decisions. Having established DARTS-GT’s competitive performance, in this section, we examine whether these heterogeneous architectures offer advantages in model transparency using the framework introduced in Section III-C.

1) *Quantitative Interpretability Metrics*: Table VII presents the interpretability metrics derived from our causal ablation framework (Section III-C). Through systematic head masking on each test instance, we compute head deviations that quantify causal importance, then aggregate using median values for robustness to outliers. The median Specialization captures the typical concentration of head importance across instances, while median Focus measures consensus among important heads on critical nodes. Supplementary Figures 4 and 5 show population-level distributions as scatter plots, with median values indicated, for all tests under consideration. We evaluated on two diverse benchmarks: MolHIV (OGB, molecular graphs, classification) and Pep-Struc (LRGB, peptide graphs, regression), representing different domains, graph sizes, and task types. For transparency, we report GPS [6] with both sparse attention (matching our implementation) and dense attention (authors’ original) on MolHIV, revealing how attention mechanism choices significantly impact interpretability (Focus: 0.5734 sparse vs 0.111 dense).

Higher Focus facilitates identifying critical nodes as important heads converge on common graph regions, while higher Specialization enables distinguishing which heads drive predictions. The combination of these metrics determines the practical effort required to understand model decisions.

DARTS-GT demonstrates consistent interpretability across both datasets, achieving best Specialization on MolHIV (0.00124) and best Focus on Pep-Struc (0.129), while securing second-best performance on the complementary metrics. This consistency across diverse domains suggests that DARTS-GT’s produces reliably interpretable models. In contrast, GPS and Vanilla-GT show more variable patterns: GPS (sparse) achieves the best Focus but minimal Specialization on MolHIV, making head importance indistinguishable, while Vanilla-GT shows the best Specialization but lowest Focus on Pep-Struc, requiring reconciliation of divergent attention patterns.

These metrics characterize the complexity of the interpretation rather than establishing absolute superiority. Our evaluation on two datasets provides initial empirical evidence on real-world applications but, like any empirical study, cannot guarantee trends generalize across all benchmarks. However, one

TABLE VII: Quantitative interpretability metrics across architectures. Median Specialization: concentration of importance in fewer heads; Median Focus: consensus on important nodes. Best values in **bold** while second-best are underlined.

| Dataset   | Model                    | Specialization  | Focus         |
|-----------|--------------------------|-----------------|---------------|
| MolHIV    | Vanilla-GT (sparse)      | 0.000634        | 0.533         |
|           | GPS (sparse)             | 0.000044        | <b>0.5734</b> |
|           | GPS (dense)              | <u>0.000091</u> | 0.111         |
|           | <b>DARTS-GT</b> (sparse) | <b>0.00124</b>  | <u>0.555</u>  |
| Pep-Struc | Vanilla-GT (dense)       | <b>0.055</b>    | 0.067         |
|           | GPS (dense)              | 0.038           | <u>0.085</u>  |
|           | <b>DARTS-GT</b> (dense)  | <u>0.044</u>    | <b>0.129</b>  |

of our primary contributions, the interpretability framework itself, enables practitioners to quantify the difficulty of interpretation. Instance-level analysis (Supplementary Figures 4 and 5) reveals considerable variation across graphs, reminding us that specific instances may present unique interpretability challenges regardless of model-level trends.

In our next section, we demonstrate how our framework enables granular, instance-specific interpretation grounded in causal head ablation, moving beyond aggregate statistics to understand individual prediction drivers.

### G. Instance-Specific Interpretation Analysis

Having established the interpretability characteristics at the dataset level, we now demonstrate how our framework provides actionable, instance-specific insights to understand individual predictions. Unlike aggregate metrics that characterize general model behavior, instance-level analysis reveals which specific graph components causally drive predictions for particular test cases through systematic head ablation. This approach is analogous to how GNNExplainer [21] identifies critical subgraphs for GNNs but is adapted for Graph Transformer attention mechanisms. It grounds interpretation in causal evidence rather than correlational attention patterns, revealing not just where models attend but which attention heads causally influence outcomes.

Our framework automatically generates per-instance JSON outputs containing head deviations, top-k important heads (we keep  $k = 4$  for MolHIV and  $k = 3$  for Peptides-struct for our experiments), their attended nodes, attention distribution statistics, and interpretability metrics (Specialization, Focus). The higher absolute deviation ( $|\delta|$ ) indicates greater head significance, i.e. removing such heads causes larger prediction errors, providing causal evidence of their contribution.

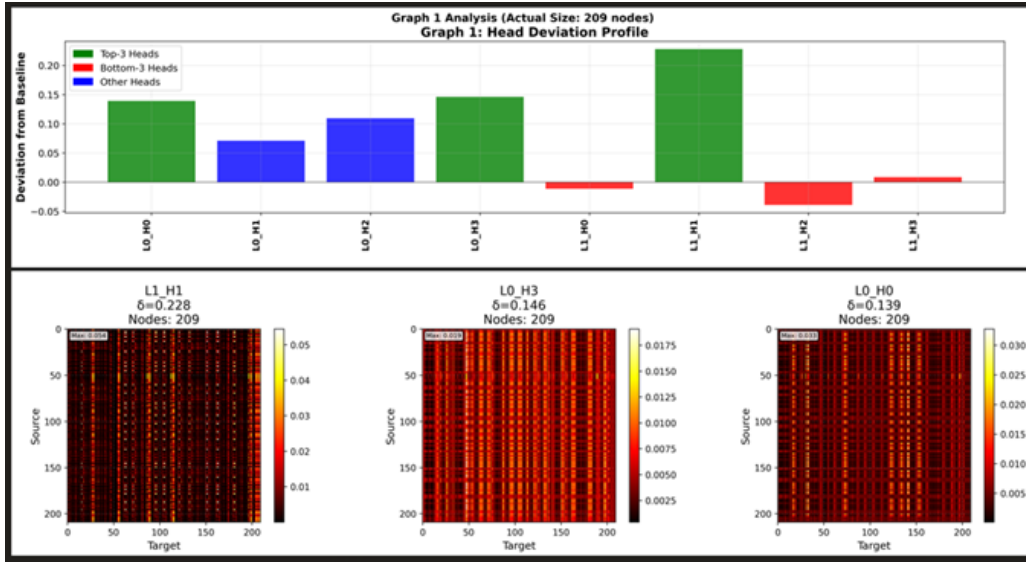
Additionally, we computed the standard deviation of each head’s attention distribution to characterize its focus pattern. Although not a central metric, this simple statistic provides useful context: a low standard deviation indicates that a head distributes attention uniformly across nodes (broad emphasis), while a high standard deviation suggests concentrated attention on specific nodes (selective emphasis).

Figure 3 presents a representative instance from the Peptides-Struc datasets, where deviation profiles identify truly important heads regardless of visual prominence, and attention heatmaps of these top-k heads reveal critical graph regions. Each attention head is uniquely identified by its layer and head index, following the format L<layer\_index>\_H<head\_index>. For example, L0\_H0 refers to the first head in the first layer. While we demonstrate Peptides-struct example here, MolHIV is moved to supplementary (Supplementary Figure 3) for space constraints.

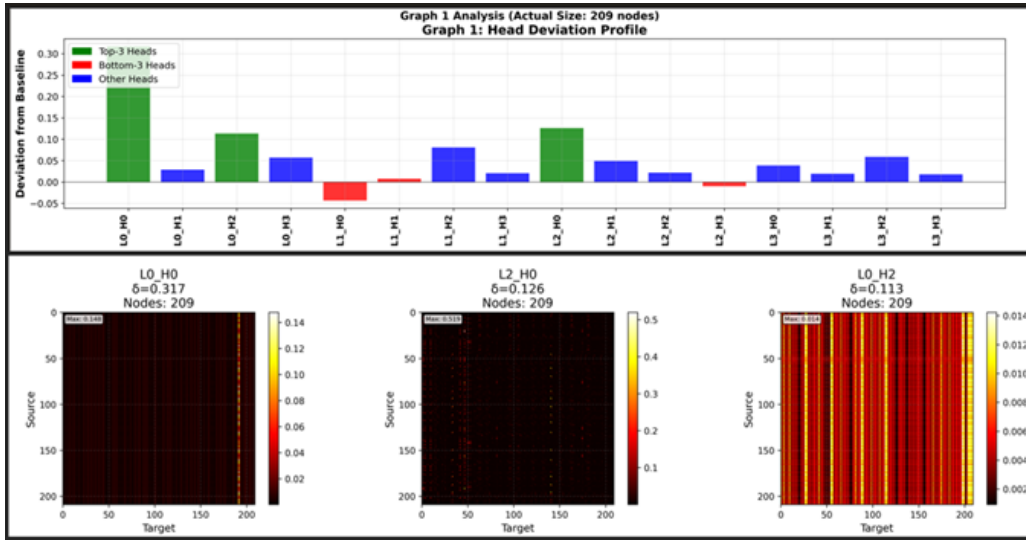
Graph 1 (Figure 3, Peptides-Struct, 209 nodes) demonstrates the attention visualization paradox where visual prominence does not always correlate with importance. (a) *DARTS-GT*: Top heads L1\_H1 ( $\delta = 0.228$ ), L0\_H3 ( $\delta = 0.146$ ), L0\_H0 ( $\delta = 0.139$ ) all exhibit low standard deviation of their attention distributions (stdev) (0.0076, 0.0029, 0.0038) producing uniformly colorful attention patterns, while targeting different nodes as L1\_H1 focuses most on nodes 27-28, L0\_H3 on node 47,197 and L0\_H0 on nodes 32,141. Moderate specialization (Spec=0.085) with low Focus (0.059) indicates distinguishable heads attending to disparate regions. (b) *GPS*: Demonstrates the paradox clearly—the most important head L0\_H0 ( $\delta = 0.317$ , stdev = 0.0063) appears dark/sparse concentrating most on node 191 (attention score 17.93) followed by 202 (attention score 2.20), while visually prominent colorful head with overall uniform attention, such as L0\_H2 ( $\delta = 0.113$ , stdev = 0.0026) have minimal impact with a much weaker attention on node 197 (attention score 2.657). Very low Focus (0.036) further confirms disparate mechanisms. (c) *Vanilla-GT*: Inverts the GPS pattern—highest impact head L0\_H0 ( $\delta = 0.262$ , stdev = 0.0043) shows uniform/colorful attention across nodes 54,203,26,113 and more (attn score around 4.0) while L2\_H1 ( $\delta = 0.188$ , stdev = 0.0179) displays a concentrated dark pattern on node 46 (attn score 13.07) despite lower importance. Highest specialization (Spec=0.127) with low Focus (0.064) suggests strongly differentiated but uncoordinated heads for this particular instance under this method.

To our knowledge, the above is the first instance-specific quantitative method for Graph Transformers to identify which heads and nodes causally drive predictions. We have quantitatively shown further (Supplementary Figure 1, Supplementary Section I), that entropy heuristics cannot replace our head-deviation based causal ablation method. Although space constraints limit us to two representative examples (additional instances available in our GitHub repository), these were strategically selected: Graph 1 (Peptides-struct) showcases a deliberate paradox where visual prominence do not always correlate with causal importance while Graph 50 (MolHIV) presented in supplementary section III, demonstrates our framework’s standard operation and how these metrics varies based on attention types.

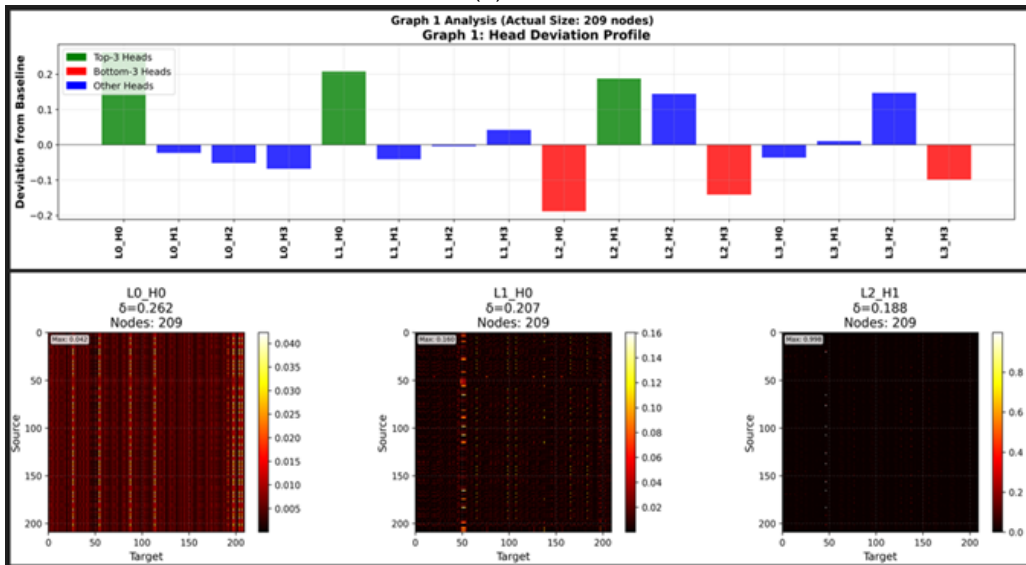
Across instances, we did not observe a reliable correlation between attention salience and causal head importance. In GPS, the most critical head (L0\_H0,  $\delta = 0.317$ ) appears sparse/dark while colorful heads contribute minimally. In contrast, in Vanilla-GT, a visually prominent dark pattern (L2\_H1) proves causally less important than a uniform/colorful head



(a) DARTS-GT



(b) GPS



(c) Vanilla-GT

Fig. 3: Instance analysis of Graph 1 (Peptides-Struc, 209 nodes) demonstrating deviation profiles and top-3 attention heatmaps. In the attention heatmaps, columns represent target nodes that receive attention.

(L0\_H0,  $\delta = 0.262$ ). *This instance-specific variability, where concentrated attention may be causally critical, irrelevant, or anywhere between, demonstrates that practitioners cannot determine head importance from visualization alone. The quantitative deviation analysis of our framework resolves this ambiguity by directly measuring the casual contribution of each head through ablation, regardless of visual appearance.* This enables reliable identification of functionally significant components that attention visualization would miss or misrepresent. Users can generate similar analyses for any instance using our code provided, extending beyond these illustrative examples to their specific use cases.

## V. CONCLUSION AND FUTURE DIRECTIONS

This work introduced DARTS-GT, a differentiable architecture search framework for Graph Transformers that discovers optimal depth-wise GNN configurations while providing quantifiable interpretability. We now revisit the research questions posed at the outset.

**Addressing Q1 - Architectural Performance and Adaptability:** Our experiments demonstrate that Graph Transformers can achieve superior performance through dynamic, layer-wise GNN selection. DARTS-GT consistently outperformed vanilla Graph Transformers on all benchmarks, with relative improvements ranging from 0.7% to 318% depending on the complexity of the dataset. Against state-of-the-art methods, DARTS-GT achieved the best new results on 4 of 8 datasets (ZINC, MalNet-Tiny, Peptides-Func, and Peptides-Struc), while remaining competitive on others.

The architectures discovered reveal DARTS' adaptive nature: selection patterns range from highly specialized (81% GINE on MolPCBA) to nearly uniform distributions (balanced usage on ZINC). This dataset-specific optimization validates that no universal architecture exists; rather, optimal configurations emerge from the interplay between task requirements and graph properties. Our ablation studies further confirmed that the asymmetric attention design (Q from features, K/V from GNNs) consistently outperforms symmetric variants, demonstrating that decoupling structural encoding from feature queries improves graph understanding. This suggests that without auxiliary MPNNs and by redesigning the attention mechanism itself, comparable SOTA performance can be achieved.

**Addressing Q2 - Quantifiable Interpretability:** Our interpretability framework provides the first quantitative method to understand Graph Transformer decisions at both the population and the instance levels. The median-based aggregation of Specialization and Focus metrics offers robust dataset-level characterization. Practitioners can quickly assess whether the prediction of a model will be easily interpretable (high Specialization, high Focus) or require extensive analysis (low values on both). These population-level metrics serve as practical guidelines for model selection in deployment scenarios where interpretability matters.

At the instance level, our framework reveals a critical finding: visual attention salience does not reliably indicate functional importance. Through systematic ablation, we demonstrated cases where sparse, nearly invisible attention patterns

drive predictions, while visually prominent patterns prove dispensable. This invalidates current visualization-based interpretation practices [23], [24] and establishes causal deviation analysis as essential to understand Graph Transformer decisions.

**Future Directions:** The convergence of improved performance and quantifiable interpretability opens exciting possibilities for scientific discovery. An appropriate real-world impactful domain application to test our DARTS-GT with the interpretability framework would be the protein structure-to-function prediction problem, where given a protein with known functions  $[F1, F2, F3, F4]$ , DARTS-GT could not only predict these functions accurately, but also identify which structural motifs causally contribute to each function. When the model makes a confident prediction on a novel protein, our interpretability framework can distinguish whether the prediction relies on established structural patterns (suggesting genuine functional similarity) or spurious correlations (indicating potential overfitting).

In conclusion, DARTS-GT demonstrates that Graph Transformers do not need to choose between performance and interpretability. By redesigning the attention mechanism through differentiable architecture search and grounding interpretation in causal analysis, we provide a path toward models that are both more capable and more trustworthy: essential qualities for deployment in high-stakes scientific and medical applications.

## VI. ACKNOWLEDGMENTS

This research was funded by Google DeepMind, with S.C. being supported by Google DeepMind Scholarship for her DPhil studies. The authors also thank Dr. Petar Veličković for helpful discussions. For the purpose of Open Access, the authors have applied a CC BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission. The authors acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work. <https://doi.org/10.5281/zenodo.22558>.

## REFERENCES

- [1] K. Lin, L. Wang, and Z. Liu, "Mesh Graphormer," Aug. 2021, arXiv:2104.00272 [cs]. [Online]. Available: <http://arxiv.org/abs/2104.00272>
- [2] D. Chen, L. O'Bray, and K. Borgwardt, "Structure-Aware Transformer for Graph Representation Learning," Jun. 2022, arXiv:2202.03036 [stat]. [Online]. Available: <http://arxiv.org/abs/2202.03036>
- [3] Z. Gu, X. Luo, J. Chen, M. Deng, and L. Lai, "Hierarchical graph transformer with contrastive learning for protein function prediction," *Bioinformatics*, vol. 39, no. 7, p. btad410, Jul. 2023. [Online]. Available: <https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btad410/7208864>
- [4] M. Gao, D. Zhang, Y. Chen, Y. Zhang, Z. Wang, X. Wang, S. Li, Y. Guo, G. I. Webb, A. T. N. Nguyen, L. May, and J. Song, "GraphormerDTI: A graph transformer-based approach for drug-target interaction prediction," *Computers in Biology and Medicine*, vol. 173, p. 108339, May 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482524004232>
- [5] C. Yuan, K. Zhao, E. E. Kuruoglu, L. Wang, T. Xu, W. Huang, D. Zhao, H. Cheng, and Y. Rong, "A Survey of Graph Transformers: Architectures, Theories and Applications," Feb. 2025, arXiv:2502.16533 [cs] version: 2. [Online]. Available: <http://arxiv.org/abs/2502.16533>
- [6] L. Rampá, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, "Recipe for a General, Powerful, Scalable Graph Transformer."

- [7] D. Kreuzer, D. Beaini, W. L. Hamilton, V. Létourneau, and P. Tossou, “Rethinking Graph Transformers with Spectral Attention,” Oct. 2021, arXiv:2106.03893 [cs]. [Online]. Available: <http://arxiv.org/abs/2106.03893>
- [8] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, “Do Transformers Really Perform Bad for Graph Representation?” Nov. 2021, arXiv:2106.05234 [cs]. [Online]. Available: <http://arxiv.org/abs/2106.05234>
- [9] Y. Gao, H. Yang, P. Zhang, C. Zhou, and Y. Hu, “GraphNAS: Graph Neural Architecture Search with Reinforcement Learning,” Aug. 2019, arXiv:1904.09981 [cs]. [Online]. Available: <http://arxiv.org/abs/1904.09981>
- [10] H. Zhao, Q. Yao, and W. Tu, “Search to aggregate neighborhood for graph neural network,” Apr. 2021, arXiv:2104.06608 [cs]. [Online]. Available: <http://arxiv.org/abs/2104.06608>
- [11] L. Wei, H. Zhao, Z. He, and Q. Yao, “Neural Architecture Search for GNN-Based Graph Classification,” *ACM Transactions on Information Systems*, vol. 42, no. 1, pp. 1–29, Jan. 2024. [Online]. Available: <https://dl.acm.org/doi/10.1145/3584945>
- [12] Z. Wu, P. Jain, M. A. Wright, A. Mirhoseini, J. E. Gonzalez, and I. Stoica, “Representing Long-Range Context for Graph Neural Networks with Global Attention,” Jan. 2022, arXiv:2201.08821 [cs]. [Online]. Available: <http://arxiv.org/abs/2201.08821>
- [13] C. Wang, J. Zhao, L. Li, L. Jiao, F. Liu, and S. Yang, “Automatic Graph Topology-Aware Transformer,” Aug. 2024, arXiv:2405.19779 [cs]. [Online]. Available: <http://arxiv.org/abs/2405.19779>
- [14] Z. Zhang, X. Wang, C. Guan, Z. Zhang, H. Li, and W. Zhu, “AUTOGT: AUTOMATED GRAPH TRANSFORMER ARCHITECTURE SEARCH,” 2023.
- [15] J. Lin, X. Guo, S. Zhang, D. Zhou, Y. Zhu, and J. Shun, “UnifiedGT: Towards a Universal Framework of Transformers in Large-Scale Graph Learning,” in *2024 IEEE International Conference on Big Data (BigData)*. Washington, DC, USA: IEEE, Dec. 2024, pp. 1057–1066. [Online]. Available: <https://ieeexplore.ieee.org/document/10825327/>
- [16] W. Song, X. Wu, B. Yang, Y. Zhou, Y. Xiao, Y. Liang, H. Ge, H. P. Lee, and C. Wu, “Towards Efficient Few-shot Graph Neural Architecture Search via Partitioning Gradient Contribution,” Jun. 2025, arXiv:2506.01231 [cs]. [Online]. Available: <http://arxiv.org/abs/2506.01231>
- [17] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” Feb. 2017, arXiv:1609.02907 [cs]. [Online]. Available: <http://arxiv.org/abs/1609.02907>
- [18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” Feb. 2018, arXiv:1710.10903 [stat]. [Online]. Available: <http://arxiv.org/abs/1710.10903>
- [19] V. P. Dwivedi and X. Bresson, “A Generalization of Transformer Networks to Graphs,” Jan. 2021, arXiv:2012.09699 [cs]. [Online]. Available: <http://arxiv.org/abs/2012.09699>
- [20] K. Choromanski, V. Likhoshesterov, D. Dohan, X. Song, A. Kane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller, “Rethinking Attention with Performers,” Nov. 2022, arXiv:2009.14794 [cs]. [Online]. Available: <http://arxiv.org/abs/2009.14794>
- [21] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, “GNNExplainer: Generating Explanations for Graph Neural Networks,” Nov. 2019, arXiv:1903.03894 [cs]. [Online]. Available: <http://arxiv.org/abs/1903.03894>
- [22] A. Shehzad, F. Xia, S. Abid, C. Peng, S. Yu, D. Zhang, and K. Verspoor, “Graph Transformers: A Survey,” Jul. 2024, arXiv:2407.09777 [cs]. [Online]. Available: <http://arxiv.org/abs/2407.09777>
- [23] M. S. Hussain, M. J. Zaki, and D. Subramanian, “Global Self-Attention as a Replacement for Graph Convolution,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2022, pp. 655–665, arXiv:2108.03348 [cs]. [Online]. Available: <http://arxiv.org/abs/2108.03348>
- [24] B. El, D. Choudhury, P. Liò, and C. K. Joshi, “Towards Mechanistic Interpretability of Graph Transformers via Attention Graphs,” Feb. 2025, arXiv:2502.12352 [cs]. [Online]. Available: <http://arxiv.org/abs/2502.12352>
- [25] L. Li, J. Liu, X. Ji, M. Wang, and Z. Zhang, “Self-Explainable Graph Transformer for Link Sign Prediction,” Dec. 2024, arXiv:2408.08754 [cs]. [Online]. Available: <http://arxiv.org/abs/2408.08754>
- [26] J. Baek, M. Kang, and S. J. Hwang, “Accurate Learning of Graph Representations with Graph Multiset Pooling,” Jun. 2021, arXiv:2102.11533 [cs]. [Online]. Available: <http://arxiv.org/abs/2102.11533>
- [27] H. Liu, K. Simonyan, and Y. Yang, “DARTS: Differentiable Architecture Search,” Apr. 2019, arXiv:1806.09055 [cs]. [Online]. Available: <http://arxiv.org/abs/1806.09055>
- [28] Y. Qin, X. Wang, Z. Zhang, and W. Zhu, “Graph Differentiable Architecture Search with Structure Learning,” Nov. 2021. [Online]. Available: [https://openreview.net/forum?id=kSv\\_AMdehh3](https://openreview.net/forum?id=kSv_AMdehh3)
- [29] K. Lukyanov, G. Sazonov, S. Boyarsky, and I. Makarov, “Robustness questions the interpretability of graph neural networks: what to do?” May 2025, arXiv:2505.02566 [cs]. [Online]. Available: <http://arxiv.org/abs/2505.02566>
- [30] K. Amara, R. Ying, Z. Zhang, Z. Han, Y. Shan, U. Brandes, S. Schemm, and C. Zhang, “GraphFramEx: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks,” May 2024, arXiv:2206.09677 [cs]. [Online]. Available: <http://arxiv.org/abs/2206.09677>
- [31] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?” Feb. 2019, arXiv:1810.00826 [cs]. [Online]. Available: <http://arxiv.org/abs/1810.00826>
- [32] X. Bresson and T. Laurent, “Residual Gated Graph ConvNets,” Apr. 2018, arXiv:1711.07553 [cs]. [Online]. Available: <http://arxiv.org/abs/1711.07553>
- [33] P. Michel, O. Levy, and G. Neubig, “Are Sixteen Heads Really Better than One?” Nov. 2019, arXiv:1905.10650 [cs]. [Online]. Available: <http://arxiv.org/abs/1905.10650>
- [34] N. Pochinkov, B. Pasero, and S. Shibayama, “Investigating Neuron Ablation in Attention Heads: The Case for Peak Activation Centering,”
- [35] N. Bahador, “Mechanistic Interpretability of Fine-Tuned Vision Transformers on Distorted Images: Decoding Attention Head Behavior for Transparent and Trustworthy AI,”
- [36] S. Heimersheim and N. Nanda, “How to use and interpret activation patching,” Apr. 2024, arXiv:2404.15255 [cs]. [Online]. Available: <http://arxiv.org/abs/2404.15255>
- [37] F. Zhang and N. Nanda, “Towards Best Practices of Activation Patching in Language Models: Metrics and Methods,” Jan. 2024, arXiv:2309.16042 [cs]. [Online]. Available: <http://arxiv.org/abs/2309.16042>
- [38] G. Anandalingam and T. L. Friesz, “Hierarchical optimization: An introduction,” *Annals of Operations Research*, vol. 34, no. 1, pp. 1–11, Dec. 1992. [Online]. Available: <http://www.scopus.com/inward/record.url?scp=0002633543&partnerID=8YFLogXK>
- [39] H. Wang, H. Yin, M. Zhang, and P. Li, “EQUIVARIANT AND STABLE POSITIONAL ENCODING FOR MORE POWERFUL GRAPH NEURAL NETWORKS,” 2022.
- [40] S. Brody, U. Alon, and E. Yahav, “How Attentive are Graph Attention Networks?” Jan. 2022, arXiv:2105.14491 [cs]. [Online]. Available: <http://arxiv.org/abs/2105.14491>
- [41] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, “Benchmarking Graph Neural Networks,” Dec. 2022, arXiv:2003.00982 [cs]. [Online]. Available: <http://arxiv.org/abs/2003.00982>
- [42] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, “Principal Neighbourhood Aggregation for Graph Nets,” Dec. 2020, arXiv:2004.05718 [cs]. [Online]. Available: <http://arxiv.org/abs/2004.05718>
- [43] V. P. Dwivedi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, “Graph Neural Networks with Learnable Structural and Positional Representations,” Feb. 2022, arXiv:2110.07875 [cs]. [Online]. Available: <http://arxiv.org/abs/2110.07875>
- [44] D. Beaini, S. Passaro, V. Létourneau, W. L. Hamilton, G. Corso, and P. Liò, “Directional Graph Networks,” Apr. 2021, arXiv:2010.02863 [cs]. [Online]. Available: <http://arxiv.org/abs/2010.02863>
- [45] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein, “Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 657–668, Jan. 2023, arXiv:2006.09252 [cs]. [Online]. Available: <http://arxiv.org/abs/2006.09252>
- [46] C. Bodnar, F. Frasca, N. Otter, Y. G. Wang, P. Liò, G. Montúfar, and M. Bronstein, “Weisfeiler and Lehman Go Cellular: CW Networks,” Jan. 2022, arXiv:2106.12575 [cs]. [Online]. Available: <http://arxiv.org/abs/2106.12575>
- [47] J. Tönshoff, M. Ritzert, H. Wolf, and M. Grohe, “Walking Out of the Weisfeiler Leman Hierarchy: Graph Learning Beyond Message Passing,” Aug. 2023, arXiv:2102.08786 [cs]. [Online]. Available: <http://arxiv.org/abs/2102.08786>
- [48] L. Zhao, W. Jin, L. Akoglu, and N. Shah, “From Stars to Subgraphs: Uplifting Any GNN with Local Structure Awareness,” Apr. 2022,

- arXiv:2110.03753 [cs]. [Online]. Available: <http://arxiv.org/abs/2110.03753>
- [49] V. P. Dwivedi, L. Rampášek, M. Galkin, A. Parviz, G. Wolf, A. T. Luu, and D. Beaini, “Long Range Graph Benchmark,” Nov. 2023, arXiv:2206.08164 [cs]. [Online]. Available: <http://arxiv.org/abs/2206.08164>
- [50] M. Yang, R. Wang, Y. Shen, H. Qi, and B. Yin, “Breaking the Expression Bottleneck of Graph Neural Networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5652–5664, Jun. 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/9759979/>

# Supplementary Materials - DARTS-GT: Differentiable Architecture Search for Graph Transformers with Quantifiable Instance-Specific Interpretability Analysis

Shruti Sarika Chakraborty Peter Minary

## I. RELATIONSHIP BETWEEN ATTENTION CONCENTRATION AND FUNCTIONAL IMPORTANCE

We investigate if heads with concentrated attention patterns exhibit higher functional importance. We test this systematically by correlating attention entropy with head deviation across all test instances for one representative dataset : Peptides-Struc for one method - DARTS-GT.

a) *Methodology.*: For each graph  $G_i$  with  $n$  nodes, head  $m$  in layer  $\ell$ , we compute attention entropy from the attention matrix  $A_m^{(\ell)} \in \mathbb{R}^{n \times n}$ . First, we sum incoming attention for each target node:

$$S_m^{(\ell)}[j] = \sum_{k=1}^n A_m^{(\ell)}[k, j] \quad (1)$$

Then normalize to obtain a probability distribution:

$$P_m^{(\ell)}[j] = \frac{S_m^{(\ell)}[j]}{\sum_{j'=1}^n S_m^{(\ell)}[j']} \quad (2)$$

Finally, compute Shannon entropy:

$$H(m, \ell, i) = - \sum_{j=1}^n P_m^{(\ell)}[j] \log P_m^{(\ell)}[j] \quad (3)$$

where low entropy indicates concentrated attention on few nodes and high entropy indicates diffuse attention. For each instance  $i$ , we compute the Spearman rank correlation between  $\{H(m, \ell, i)\}_{m \in \mathcal{M}, \ell \in \mathcal{L}}$  and  $\{\delta_{m, \ell, i}\}_{m \in \mathcal{M}, \ell \in \mathcal{L}}$  across all heads and layers.

b) *Results on Peptides-Struct*: We apply this analysis to DARTS-GT on the Peptides-Struct dataset. Supplementary Figure 1 shows the distribution of per-instance correlations. The left histogram reveals correlations spanning from -1.0 to +1.0, with a median of only 0.167 and standard deviation of 0.437. Notably, the standard deviation exceeds twice the median value. The right boxplot confirms this inconsistency: the interquartile range crosses zero, and whiskers extend across nearly the full correlation range. Substantial portions of instances exhibit negative correlations (where concentrated attention corresponds to low importance), near-zero correlations (no relationship), and positive correlations (concentrated attention corresponds to high importance).

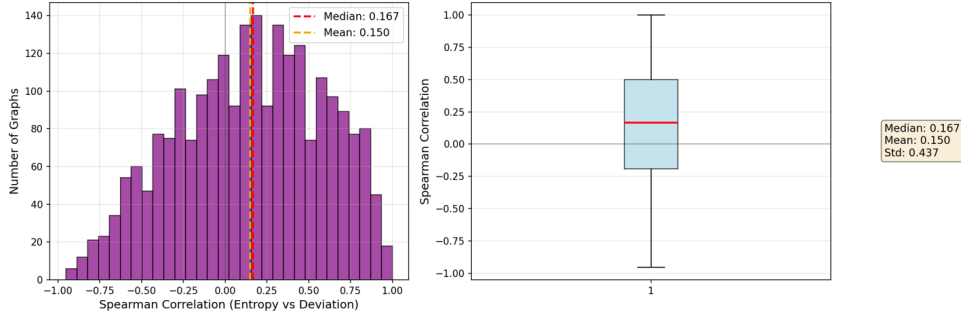
The weak median correlation (0.167) combined with high variance (0.437) demonstrates that attention concentration (entropy) is not a reliable predictor of functional importance. The relationship varies fundamentally across instances i.e. concentrated attention may indicate critical heads, dispensable heads, or have no relationship to importance at all. *This proves that attention-based visualization or entropy heuristics cannot substitute for causal ablation methods.* Our head deviation framework remains necessary for faithful interpretation, as it directly measures functional contribution rather than relying on correlational attention patterns that may be misleading.

## II. ABLATION STUDIES: DATASET-DEPENDENT ROLE OF EDGE INFORMATION

Here we examine the impact of edge residual connections across different attention mechanisms and datasets. While our GNN experts (GINE, GatedGCN, GATV2) inherently utilize edge features, these ablation studies demonstrate whether adding explicit edge residual connections to the attention mechanism provides additional benefits. Supplementary Table I presents results across four datasets comparing dense and sparse attention with and without edge residual connections. For computational efficiency, ZINC experiments use 1000 epochs, sufficient to demonstrate relative performance trends. The results reveal dataset-specific utility of edge residuals. These findings demonstrate that edge residual connections are not universally beneficial, validating our design choice of making them optional.

## III. INSTANCE-SPECIFIC INTERPRETABILITY ANALYSIS FOR MOLHIV :

Here we demonstrate instance-specific interpretability analysis for MolHIV's graph number: 50. For Graph-50 (17 nodes) of MolHIV, head deviation profiles and attention patterns (Supplementary Figure 3) reveal distinct interpretability characteristics. In the attention heatmaps, columns represent target nodes that receive attention. (a) *DARTS-GT*: Clear head specialization (Spec= 0.00288) with top heads forming two consensus pairs - L0\_H3 ( $\delta = 0.016$  - highest) /L1\_H1 attending to target node 1, L2\_H0/L1\_H0 attending to target node 9. The Focus for this instance is 0.333. (b) *GPS with dense attention*: Minimal head specialization (Spec=  $1.01 \times 10^{-9}$ ) with nearly imperceptible deviations (highest  $\delta = 4.87 \times 10^{-9}$  at L5\_H1),



Supplementary Figure 1: Distribution of per-instance Spearman correlations between attention entropy and head deviation on Peptides-Struct.

Supplementary Table I: Effect of edge features on Graph Transformer performance across multiple datasets and attention types. Shown is the mean  $\pm$  std across 3 seeds. Best results are in **bold**.

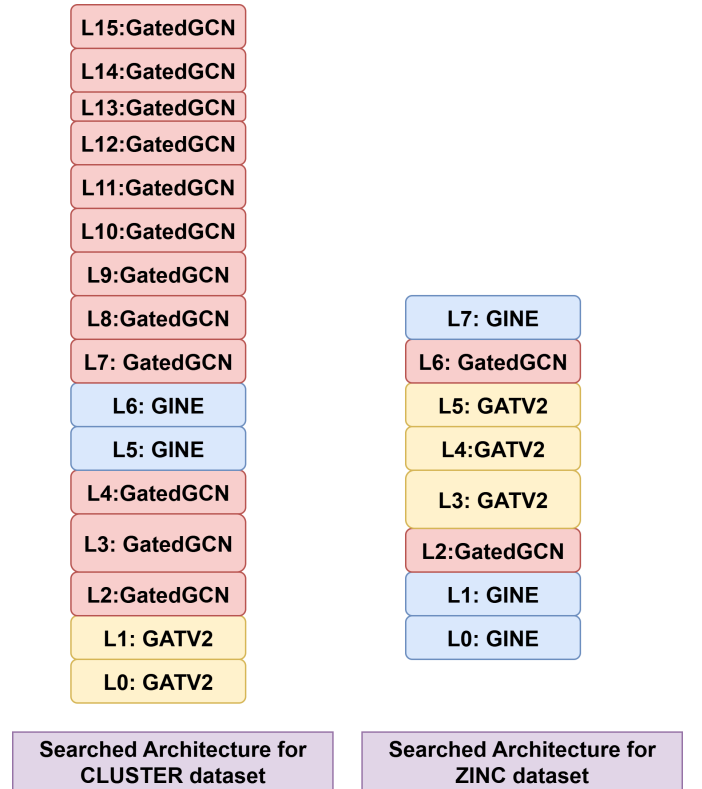
| Dataset        | Attn Type | Mode            | Perf                                                       |
|----------------|-----------|-----------------|------------------------------------------------------------|
| ZINC*          | Dense     | No edge<br>Edge | 0.1054 $\pm$ 0.0055<br><b>0.098<math>\pm</math>0.0082</b>  |
|                | Sparse    | No edge<br>Edge | 0.0812 $\pm$ 0.0043<br><b>0.0782<math>\pm</math>0.006</b>  |
| MOLPCBA        | Dense     | No edge<br>Edge | 0.2011 $\pm$ 0.0101<br><b>0.2224<math>\pm</math>0.004</b>  |
|                | Sparse    | No edge<br>Edge | <b>0.258<math>\pm</math>0.0043</b><br>0.2545 $\pm$ 0.003   |
| CLUSTER        | Dense     | No edge<br>Edge | 0.2693 $\pm$ 0.0013<br><b>0.3107<math>\pm</math>0.0122</b> |
|                | Sparse    | No edge<br>Edge | 0.7773 $\pm$ 0.0020<br><b>0.7835<math>\pm</math>0.0004</b> |
| Peptides-Struc | Dense     | No edge<br>Edge | <b>0.2466<math>\pm</math>0.0003</b><br>0.248 $\pm$ 0.0025  |
|                | Sparse    | No edge<br>Edge | 0.263 $\pm$ 0.0004<br>0.263 $\pm$ 0.0008                   |

suggests uniform head importance. Partial consensus with L5\_H1/L0\_H1 attending to target node 10 while others target nodes 7,11. The overall Focus attained is 0.167. (c) *GPS with sparse attention*: Low head specialization (Spec =  $7.39 \times 10^{-5}$ ) with low deviations (highest  $\delta = 0.0003$  at L4\_H3). High consensus with L4\_H3/L6\_H0/L5\_H3 attending to target node 9 while L6\_H2 target node 14. The overall Focus attained is very high 0.5. (d) *Vanilla-GT*: Low specialization (Spec =  $2.47 \times 10^{-5}$ ). Similar partial consensus with L14\_H1/L3\_H2 attending to target node 14, others targeting nodes 8,5 and overall Focus is same as GPS (dense attention) with 0.167. Despite similar Focus values in GPS (dense attn) and Vanilla-GT, their vastly different Specialization values indicate distinct interpretation challenges.

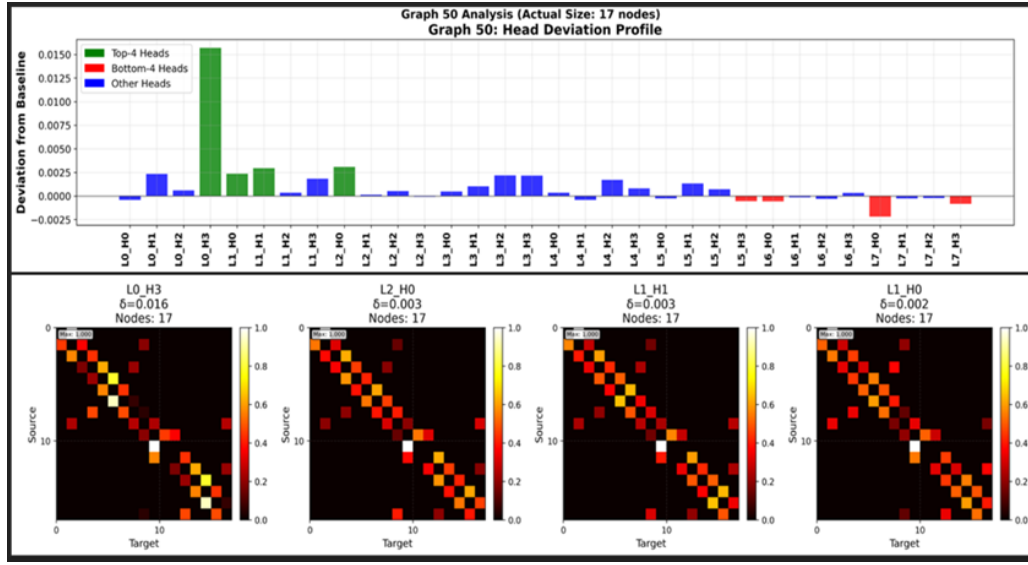
#### IV. LAYER-WISE ARCHITECTURE DISCOVERY VIA DARTS

The DARTS search process automatically discovers optimal GNN operators for each layer, as illustrated in Supplementary Figure 2 for CLUSTER and ZINC datasets. These results demonstrate that different layers require different inductive

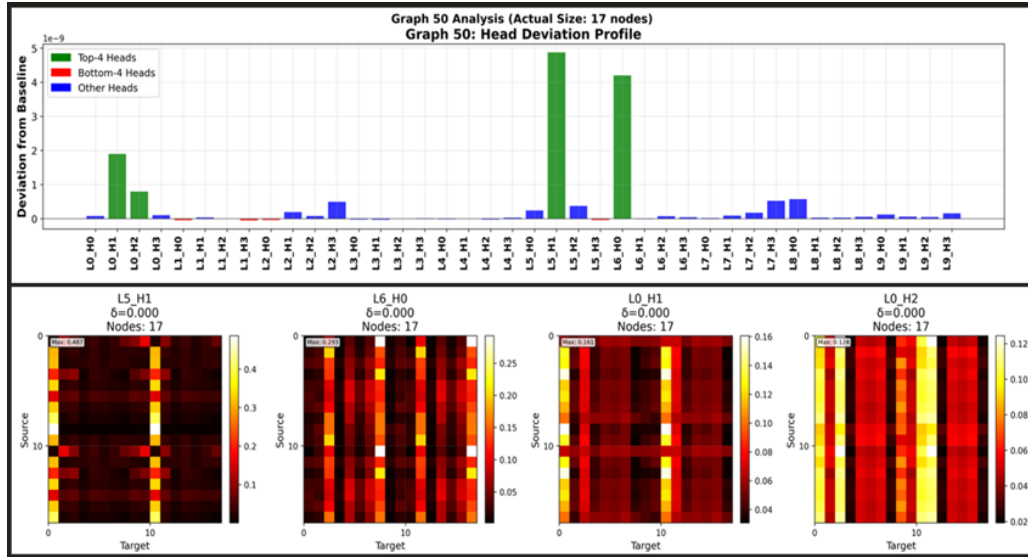
biases, with the search adapting operator selection based on both network depth and dataset characteristics.



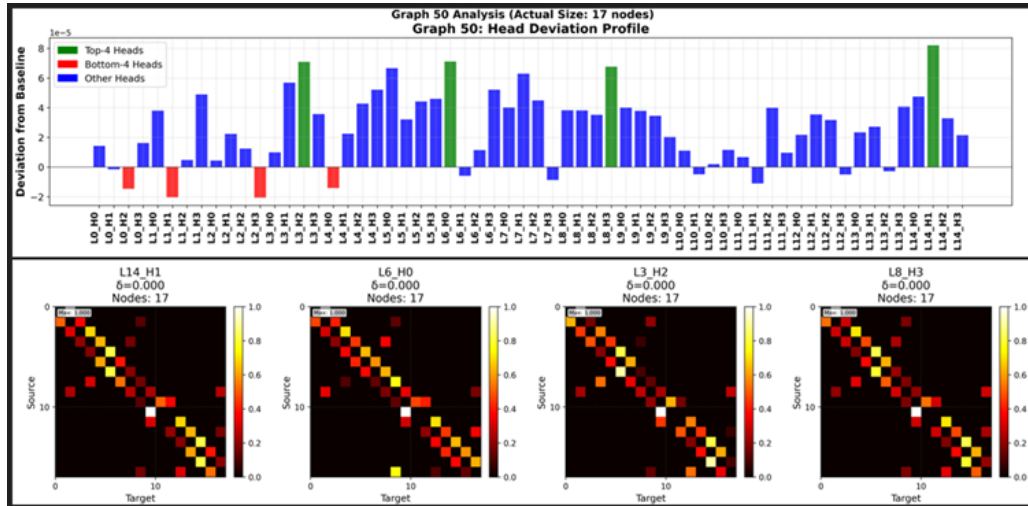
Supplementary Figure 2: **Discovered layer-wise operators after DARTS**. For each layer, the search selects exactly one operator from {GINE, GATv2, GatedGCN}. *CLUSTER* (left, L0–L15): the architecture is dominated by **GatedGCN** across depth, with brief **GINE** selections around L5–L6 and **GATv2** at L0–L1. *ZINC* (right, L0–L7): shallow layers favor **GINE** (L0–L1, L7), the mid-depth region prefers **GATv2** (L3–L5), and **GatedGCN** appears at L2 and L6. These dataset-specific, depth-varying patterns indicate that the search adapts inductive biases per layer, rather than converging to a uniform choice. After search, the selected per-layer GNN is used once to produce  $K, V$  and is shared across heads.



(a) DARTS-GT

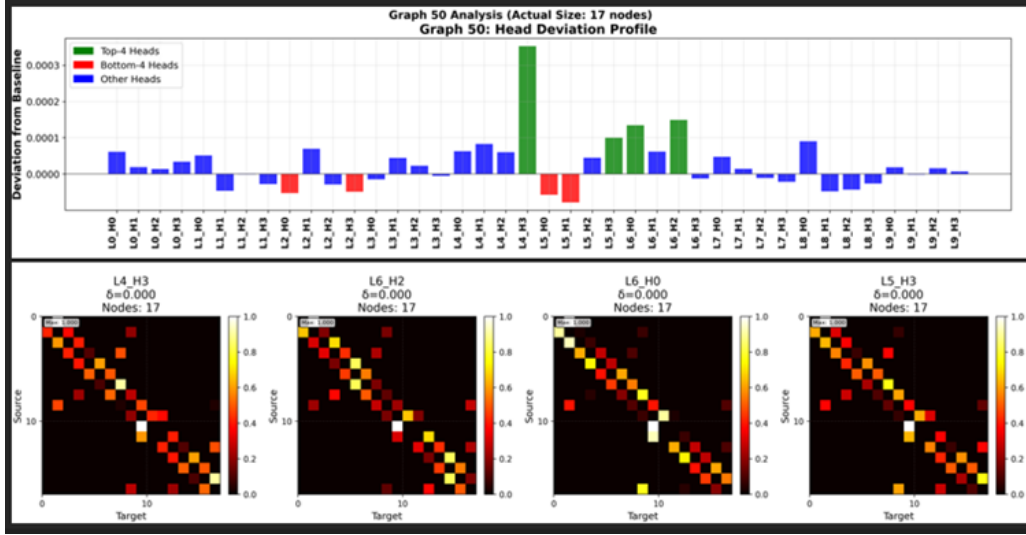


(b) GPS



(c) Vanilla-GT

Supplementary Figure 3: Instance analysis of Graph 50 (MolHIV, 17 nodes). Deviation profiles and top-4 attention heatmaps for each model, with columns representing target nodes. GPS with dense attention is given here.

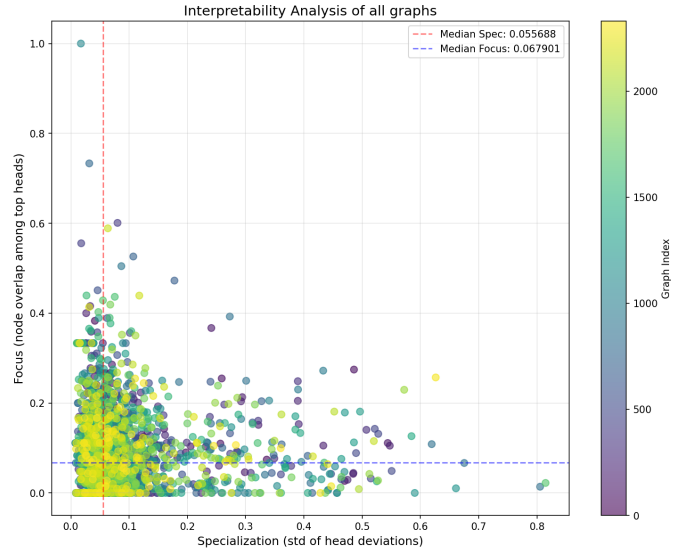


(d) GPS (sparse)

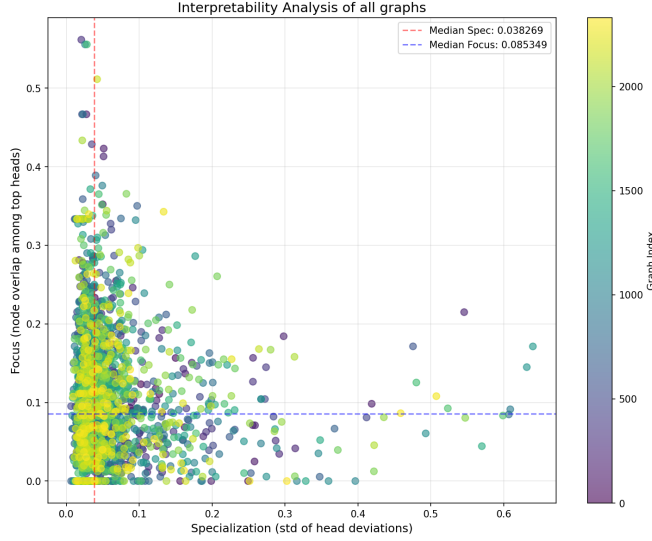
Supplementary Figure 3: (continued) Instance analysis of Graph 50 - GPS with sparse attention.

## V. POPULATION LEVEL INTERPRETABILITY ANALYSIS

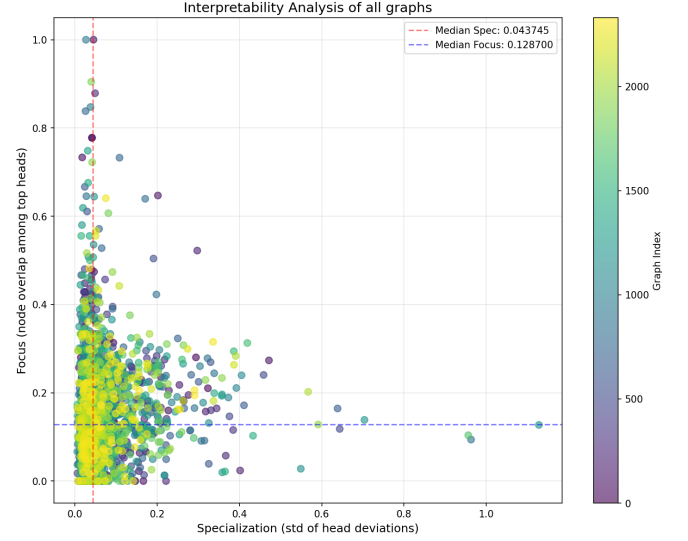
Supplementary figures 4 and 5 visualize population-level interpretability distributions across two test sets : Peptide-Struc and MolHIV respectively, revealing distinct patterns between architectures and datasets. The plots demonstrate that interpretability is inherently dataset-dependent and varies significantly at the instance level. While we report median values for dataset-level comparison, each point represents an individual graph with its own Specialization and Focus values derived from our causal ablation framework. This instance-level variation, visible as the scatter around median values, underscores that some graphs remain challenging to interpret regardless of model choice, while others yield clear interpretability patterns.



Supplementary Figure 4: Instance-level interpretability distributions on Peptides-Struc test set. (a) Vanilla-GT model (dense attention). Each point represents a single test graph positioned by its Specialization (head importance concentration) and Focus (consensus among important heads).



(a) GPS (Pep-Struc, dense)



(b) DARTS-GT (Pep-Struc, dense)

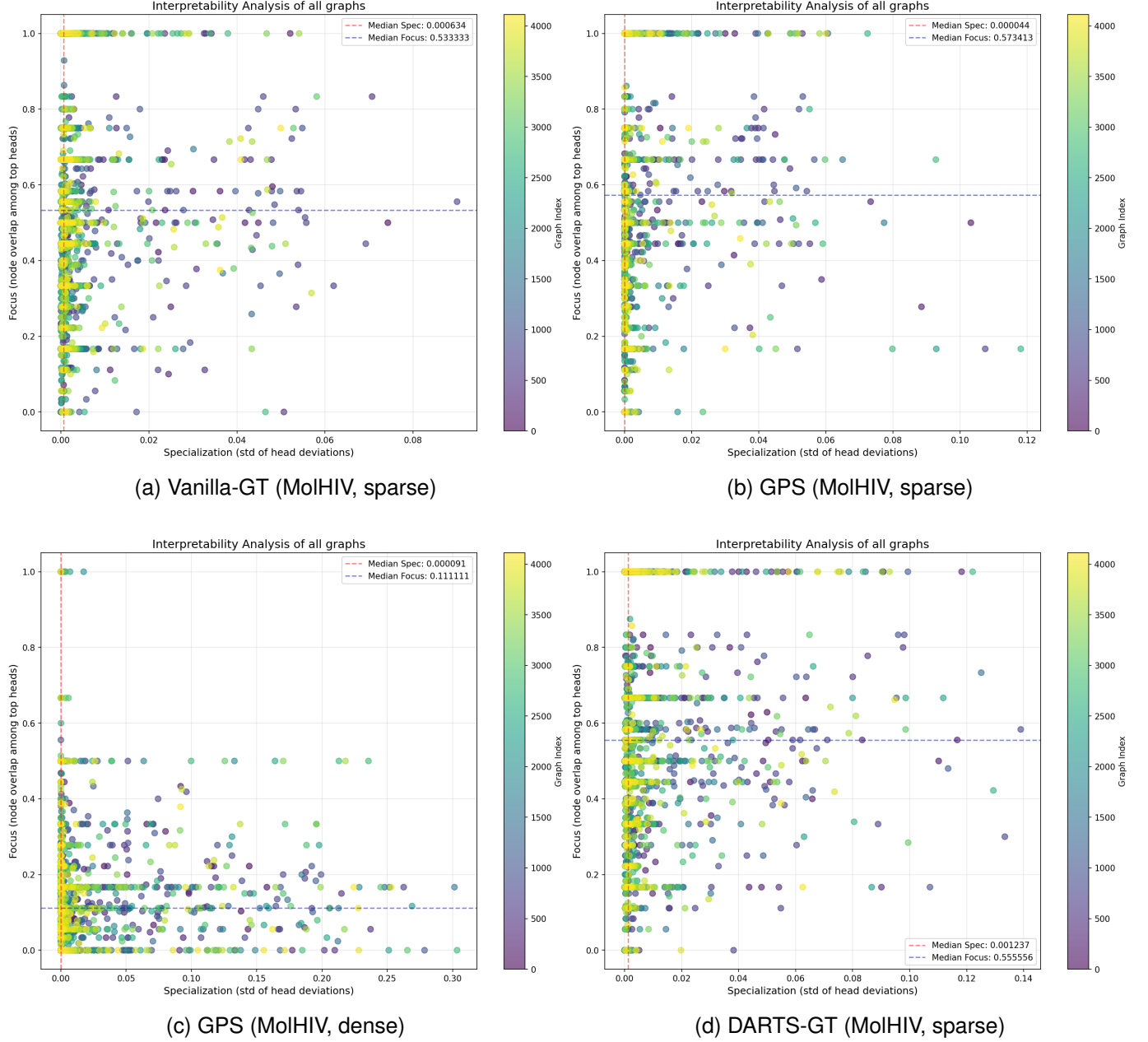
Supplementary Figure 4: (continued) (b) GPS model shows different clustering patterns compared to Vanilla-GT. (c) DARTS-GT achieves the highest median Focus (0.129), indicating stronger node consensus. All models use dense attention. Note the different scale compared to MolHIV (Supplementary Figure 5), with Specialization value, suggesting dataset-specific interpretability characteristics.

Supplementary Table II: Hyperparameters across all datasets (Part 1)

| Hyperparameters                  | ZINC        | MolHIV      | MOLPCBA     | MALNET       |
|----------------------------------|-------------|-------------|-------------|--------------|
| <i>Architecture Parameters</i>   |             |             |             |              |
| Number of Layers                 | 8           | 8           | 4           | 4            |
| Hidden dim                       | 64          | 64          | 384         | 64           |
| Attention type                   | Sparse      | Sparse      | Sparse      | Sparse       |
| Heads                            | 4           | 4           | 4           | 4            |
| Dropout                          | 0           | 0.05        | 0.2         | 0            |
| Attention dropout                | 0.5         | 0.5         | 0.5         | 0.5          |
| Edge residual weight             | 0.1         | NA          | NA          | 0.1          |
| Q-projection                     | Identity    | Identity    | Identity    | Identity     |
| <i>DARTS Parameters</i>          |             |             |             |              |
| DARTS epochs                     | 120         | 30          | 30          | 30           |
| DARTS splits ratio               | 0.6         | 0.6         | 0.6         | 0.6          |
| Architecture LR                  | 4.00E-04    | 4.00E-04    | 4.00E-04    | 4.00E-04     |
| Grad clip                        | 5           | 5           | 5           | 5            |
| <i>Training Parameters</i>       |             |             |             |              |
| LR reduce factor                 | 0.5         | 0.5         | 0.5         | 0.5          |
| LR schedule patience             | 10          | 10          | 10          | 10           |
| Min learning rate                | 1.00E-06    | 1.00E-06    | 1.00E-06    | 1.00E-06     |
| Initial learning rate            | 0.0025      | 0.0025      | 0.0025      | 0.0025       |
| Weight decay                     | 3.00E-04    | 3.00E-04    | 3.00E-04    | 3.00E-04     |
| Epochs                           | 2000        | 100         | 100         | 100          |
| Warmup epochs                    | 50          | 5           | 5           | 10           |
| <i>Graph-specific Parameters</i> |             |             |             |              |
| Graph pooling                    | sum         | mean        | mean        | max          |
| Positional Encoding              | RWSE-20     | RWSE-16     | RWSE-16     | None         |
| PE dim                           | 28          | 16          | 20          | NA           |
| PE encoder                       | linear      | linear      | linear      | NA           |
| <i>Computational Resources</i>   |             |             |             |              |
| DARTS-Parameters                 | 641989      | 583017      | 10186472    | 327893       |
| Discrete-Parameters              | 412397      | 344929      | 5707484     | 208329       |
| Total DARTS time                 | 4490.0s     | 2777.0s     | 4441.0s     | 630.0s       |
| Discrete phase time              | 26.0s/15.0h | 63.0s/2.37h | 158.0s/5.0h | 16.0s/0.645h |

Supplementary Table III: Hyperparameters across all datasets (Part 2)

| Hyperparameters                  | PATTERN     | CLUSTER     | PEP-FUNC      | PEP-STRUC    |
|----------------------------------|-------------|-------------|---------------|--------------|
| <i>Architecture Parameters</i>   |             |             |               |              |
| Number of Layers                 | 6           | 16          | 2             | 2            |
| Hidden dim                       | 64          | 48          | 120           | 120          |
| Attention type                   | Sparse      | Sparse      | Dense         | Dense        |
| Heads                            | 4           | 8           | 4             | 4            |
| Dropout                          | 0           | 0           | 0             | 0            |
| Attention dropout                | 0.5         | 0.5         | 0.5           | 0.5          |
| Edge residual weight             | 0.1         | 0.1         | NA            | NA           |
| Q-projection                     | Identity    | Identity    | MLP           | MLP          |
| <i>DARTS Parameters</i>          |             |             |               |              |
| DARTS epochs                     | 20          | 25          | 50            | 50           |
| DARTS splits ratio               | 0.6         | 0.6         | 0.6           | 0.6          |
| Architecture LR                  | 4.00E-04    | 4.00E-04    | 4.00E-04      | 4.00E-04     |
| Grad clip                        | 5           | 5           | 5             | 5            |
| <i>Training Parameters</i>       |             |             |               |              |
| LR reduce factor                 | 0.5         | 0.5         | 0.5           | 0.5          |
| LR schedule patience             | 10          | 10          | 10            | 10           |
| Min learning rate                | 1.00E-06    | 1.00E-06    | 1.00E-06      | 1.00E-06     |
| Initial learning rate            | 0.0025      | 0.0025      | 0.0025        | 0.0025       |
| Weight decay                     | 3.00E-04    | 3.00E-04    | 3.00E-04      | 3.00E-04     |
| Epochs                           | 100         | 100         | 200           | 200          |
| Warmup epochs                    | 5           | 5           | 0             | 0            |
| <i>Graph-specific Parameters</i> |             |             |               |              |
| Graph pooling                    | NA          | NA          | mean          | mean         |
| Positional Encoding              | LapPE-16    | LapPE-10    | LapPE-10      | LapPE-10     |
| PE dim                           | 16          | 16          | 16            | 16           |
| PE encoder                       | DeepSet     | DeepSet     | DeepSet       | DeepSet      |
| <i>Computational Resources</i>   |             |             |               |              |
| DARTS-Parameters                 | 487369      | 728678      | 604817        | 604696       |
| Discrete-Parameters              | 338775      | 511250      | 426341        | 448240       |
| Total DARTS time                 | 1320.45s    | 3454.26s    | 409.0s        | 411.54s      |
| Discrete phase time              | 58.0s/1.87h | 97.4s/3.88h | 12.49s/0.775h | 12.5s/0.845h |



Supplementary Figure 5: Instance-level interpretability distributions on MolHIV test set. Each point represents a single test graph positioned by its Specialization (head importance concentration) and Focus (consensus among important heads). GPS shown with both sparse and dense attention to demonstrate architectural impact on interpretability.

Supplementary Table IV: Comprehensive parameter comparison across all models and datasets. Best (lowest) parameter count is highlighted in **bold**

| Model                   | Attention | ZINC            | MolHIV          | MOLPCBA          | MALNET   | PATTERN         | CLUSTER         | PEP-STRUC       | PEP-FUNC        |
|-------------------------|-----------|-----------------|-----------------|------------------|----------|-----------------|-----------------|-----------------|-----------------|
| <i>Dense Attention</i>  |           |                 |                 |                  |          |                 |                 |                 |                 |
| Vanilla-GT              | Dense     | <b>3,56,261</b> | 4,51,793        | <b>84,06,236</b> | 5,03,749 | <b>3,02,385</b> | <b>3,85,126</b> | 4,29,611        | 4,29,490        |
| GPS                     | Dense     | 4,23,717        | 5,58,625        | 97,44,496        | 5,27,237 | 3,37,201        | 5,02,054        | 5,04,459        | 5,04,362        |
| DARTS-GT                | Dense     | 4,41,751        | <b>3,98,289</b> | 85,57,152        | NA       | 3,61,591        | 5,07,206        | <b>4,26,341</b> | <b>4,48,240</b> |
| <i>Sparse Attention</i> |           |                 |                 |                  |          |                 |                 |                 |                 |
| Vanilla-GT              | Sparse    | <b>3,56,261</b> | 4,51,793        | 84,06,236        | NA       | <b>3,02,385</b> | <b>3,38,086</b> | 4,12,397        | 4,29,490        |
| DARTS-GT                | Sparse    | 4,28,333        | <b>3,44,929</b> | <b>57,07,484</b> | 2,08,329 | 3,38,775        | 5,11,250        | <b>3,97,301</b> | <b>4,08,160</b> |