MUST GRAPH STRUCTURE REMAIN IMPLICIT? EXPLICIT TOPOLOGICAL REPRESENTATIONS VIA SPARSE DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks and graph transformers achieve strong predictive performance but typically encode topology in latent coordinates without directly identifiable structural meaning. Classical graphlet representations expose structural patterns through predefined families. This raises a broader question: can an explicit, data-driven representation support competitive graph learning while keeping topology and attributes separately accessible? Building on sparse network decomposition, we show that explicit, learnable structural coordinates can support competitive prediction while making structural evidence independently auditable and traceable. Our framework combines shared-dictionary encoding of rooted patches with permutation-invariant readout, preserving occurrence-level links and supporting separate topology, attribute, and binding channels. The exact radius-r rooted-patch profile refines r rounds of 1-WL, while radius-one profiles distinguish some graph pairs unresolved by stable 1-WL. On BREC, the exact pair-rooted profile distinguishes 300 of 400 graph pairs, and a compact 128-atom representation retains nearly all of these distinctions. Across node- and graph-level benchmarks, the framework achieves competitive performance against recent GNN and graph Transformer baselines using conventional predictors. Coalition retraining quantifies standalone and conditional information-source contributions, while molecular case studies demonstrate how structural tracing supports testable chemical hypotheses and fixed-model interventions that identify split-specific structural shortcuts.

1 INTRODUCTION

Graph neural networks (GNNs) and graph transformers (GTs)(Kipf & Welling, 2017; Xu et al., 2019; Rampášek et al., 2022; Zhang et al., 2023) have become central tools for learning from relational data. Through neighborhood aggregation, attention, and structural encodings, they turn graph connectivity into representations that support a wide range of prediction tasks. Yet the structural content of these representations is often difficult to inspect: individual latent coordinates generally do not correspond directly to identifiable connectivity patterns. When topology and attributes are jointly encoded, it is also difficult to determine what structural evidence is retained and how much predictive value it adds beyond the attributes. Classical graphlet representations expose topology through counts of predefined subgraph types (Shervashidze et al., 2009; 2011). Although these features can be combined with attributes, their structural coordinates are specified in advance rather than learned from observed connectivity patterns. This raises a broader question: can an explicit, data-driven structural representation provide a shared basis for competitive graph learning while keeping topology, attributes, and their interactions separately accessible?

Sparse network decomposition provides a starting point. Prior work showed that network topology can be decomposed, compressed, and reconstructed through dictionaries and sparse coefficients, while related graph dictionary methods explored graph embeddings and downstream learning (Zhai et al., 2020; Vincent-Cuaz et al., 2021; Lyu et al., 2024). Building on these foundations, we organize local connectivity patterns into shared, learnable structural coordinates. The central idea is to keep structural evidence explicitly accessible before prediction, allowing the same representation to support predictive learning, information-source auditing, and occurrence-level tracing.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We instantiate the framework with node- or node-pair-rooted induced patches. Dataset-adapted, root-preserving canonicalization makes local observations comparable under node relabeling; a shared dictionary encodes them as sparse activations; and permutation-invariant readout produces fixed-dimensional representations for node- and graph-level tasks. Graph statistics and supplied attributes remain separately accessible, with optional topology–attribute binding capturing their associations. Retaining the correspondence between activations and patch occurrences allows learned components to be traced to concrete graph patterns.

Our results show that graph structure can form an explicit, learnable coordinate system for prediction and analysis. Under information-preservation conditions, the exact radius-r rooted-patch profile retains every distinction made by r rounds of 1-WL and also distinguishes explicit graph pairs unresolved by stable 1-WL. With conventional downstream predictor, the resulting coordinates achieve competitive or superior performance against recent graph learning baselines across node- and graph-level benchmarks. Because structural and attribute blocks remain independently accessible, their standalone and complementary predictive contributions can be audited directly. Learned components can further be traced to their activating subgraphs, supporting analyses ranging from testable structural hypotheses to the diagnosis of generalization failures. Our main contributions are:

- We develop an explicit, learnable graph representation framework based on sparse network decomposition. Rooted induced patches are encoded over a shared dictionary and pooled into common structural coordinates for node- and graph-level learning. Separately accessible statistics, topology, and attributes support controlled combination and explicit topology–attribute binding.
- We establish permutation invariance and characterize the discriminative power of exact rooted-patch profiles. The radius-r profile refines r rounds of 1-WL, while radius-one profiles distinguish some graph pairs unresolved by stable 1-WL. The exact radius-two adjacent-pair-rooted profile is incomparable with the pair-based higher-order refinement defined in our analysis. Compact sparse representations retain nearly all observed exact-profile distinctions on BREC. Across node- and graph-level benchmarks, the framework achieves competitive performance against recent GNNs and graph transformers.
- We demonstrate how explicit structural coordinates support information-source auditing and deeper analysis of graph-learning outcomes. Controlled coalition retraining quantifies the standalone and conditional predictive utility of statistics, topology, and attributes. Occurrence-level tracing connects learned components to concrete subgraphs, supporting the formulation and statistical examination of a chemical hypothesis on MolHIV and fixed-model interventions that identify a split-specific structural shortcut on MolBACE.

2 Related Work

Explicit and learned structural representations. Graph kernels and graphlet descriptors represent graphs through identifiable structural patterns (Shervashidze et al., 2009; Zhang & Li, 2021). Data-driven approaches extend this idea through discriminative subgraph selection, learned kernel features, and graph dictionaries (Zhai et al., 2020; Vincent-Cuaz et al., 2021; Lyu et al., 2024). Some previous work like MixSL incorporates precomputed substructure features into GNN inputs and learns to combine them for graph classification (Bouritsas et al., 2023; Wu et al., 2026). Network dictionary learning approximates sampled connectivity patches using learned latent motifs, while convex extensions associate dictionary elements with observed subnetworks (Lyu et al., 2024; Rana et al., 2024). These methods establish foundations for explicit structural learning and reconstruction. Building on this line, we organize canonical rooted observations and sparse activations into shared coordinates for node- and graph-level prediction.

Structure–attribute separation and contribution analysis. Structure–attribute disentanglement is an established objective in graph representation learning (Xu et al., 2021; Wang et al., 2025). Structure-agnostic baselines also assess the predictive value of connectivity beyond node features (Errica et al., 2020), while post-hoc methods such as GraphSVX attribute predictions of a fixed model to structural and feature inputs (Duval & Malliaros, 2021). Our framework specifies the inputs available to topology, attribute-composition, and binding channels, and evaluates their predictive utility by retraining predictors for each channel coalition.

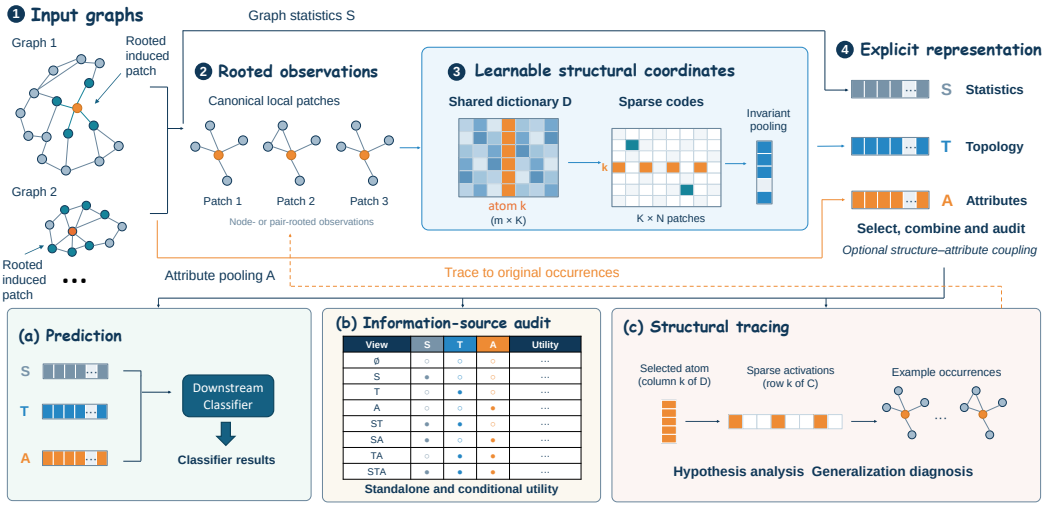

Figure 1: Overview of the proposed framework. Rooted graph observations are sparsely encoded over a shared learned dictionary and pooled into explicit structural coordinates. It supports prediction, information-source auditing, and tracing learned components to original subgraph occurrences.

Positioning of our work. We build on sparse structural decomposition to study an explicit representation framework combining shared learned coordinates, invariant aggregation, separately accessible information channels, and occurrence-level tracing. Our evaluation connects these design choices to exact-profile discrimination, information retention after compression, competitive prediction, and the formulation of testable domain hypotheses.

3 Sparse Structural Representation

Figure 1 illustrates the proposed framework. We first decompose each graph into rooted local observations, encode these observations as sparse activations over a dictionary shared across graphs, and aggregate the activations into fixed-dimensional node- or graph-level coordinates. The resulting representation makes learned topological components explicitly accessible before downstream prediction and allows topology and supplied attributes to be analyzed separately or jointly.

3.1 Rooted Structural Observations

Let $G = (V, E, \mathbf{X}, \mathbf{E}_a)$ denote a graph with optional node attributes $\mathbf{X}$ and edge attributes $\mathbf{E}_a$. Rather than encoding G as a single monolithic object, we describe it through local patches associated with a collection of root sets $\mathcal{R}(G)$. For a root set $R \subseteq V$, its radius-r induced patch and corresponding structural observation are

$$P_r(G, R) = G\left[\left\{u \in V : \min_{v \in R} d_G(u, v) \le r\right\}\right], \qquad \mathbf{y}_{G,R} = \psi_\tau(P_r(G, R), R) \in \mathbb{R}^{d_\tau}, \tag{1}$$

where ψ_τ places the patch into a fixed-dimensional, root-aware local coordinate system, and τ specifies the root type and observation channels. A singleton $R = \{v\}$ produces a node-rooted observation, whereas $R = \{u, v\}$ captures the joint environment of a node pair. We use node-rooted observations for downstream prediction and exact observations rooted at unordered adjacent pairs for BREC.

The observation operator is adapted to the graph domain. Topology-centered benchmarks use deterministically ordered adjacency observations; molecular graphs use typed local coordinates containing selected topology, atom, bond, and shell channels; and BREC uses exact root-preserving canonicalization. These variants change only the construction of $\mathbf{y}_{G,R}$; the shared dictionary and readout below remain unchanged. For information-source audits, topology-only observations define

the T block, while supplied node and edge attributes are retained in a separately removable A block. Complete canonicalization, ordering, padding, and truncation procedures are given in Appendix A.1.

3.2 Shared Dictionary and Sparse Coding

Observations extracted from different graphs are placed in a common coordinate system by learning a shared structural dictionary. Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ collect the normalized observations from the reference graphs prescribed by the evaluation protocol. We learn a dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{d_T \times K}$ and sparse codes $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ through

$$\min_{\mathbf{D},\mathbf{A}} \sum_{i=1}^{N} w_i \left\| \mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i \right\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\alpha}_i\|_0 \le s, \qquad \|\mathbf{d}_k\|_2 = 1, \tag{2}$$

where K is the number of dictionary atoms, s is the sparsity level, and w_i optionally records the multiplicity of repeated observations.

Once fitted, $\mathbf{D}$ is fixed across graphs. Coordinate k therefore refers to the same learned structural component throughout an experiment, and each patch is represented by its sparse activation $\boldsymbol{\alpha}_{G,R}$. We use K-SVD to fit the dictionary and orthogonal matching pursuit for sparse coding (Pati et al., 1993; Aharon et al., 2006). Optimization, normalization, weighting, and large-scale sampling details are provided in Appendix A.3.

3.3 Invariant Readout and Information Organization

A graph produces a variable-size multiset of patch activations. We convert it into a fixed-dimensional topological representation using a permutation-invariant readout:

$$\mathbf{z}_G^T = \rho(\{\{\boldsymbol{\alpha}_{G,R} : R \in \mathcal{R}(G)\}\}) \,. \tag{3}$$

The readout applies symmetric coordinate-wise summaries to the sparse activations, so the resulting representation is independent of patch enumeration order while retaining the prevalence and activation strength of each learned component. Node-level representations use the same structural coordinates, with aggregation restricted to patches associated with the target node's context. The complete readout statistics are listed in Appendix A.4.

For controlled information-source analysis, we organize graph features into separately constructed and independently removable blocks: S contains global graph statistics, T contains the dictionary-based topological coordinates in equation 3, and A summarizes the supplied node and edge attributes without explicitly preserving their structural placement. Concatenations such as ST, TA, and STA support direct coalition-based audits. When finer structure–semantic interactions are useful, explicit coupling features associate local attributes with dictionary activations before readout. *Cross* additionally couples patch activations with root-, shell-, or occurrence-level attributes and therefore represents explicit structure–semantic binding rather than ordinary concatenation. Dataset-specific attribute summaries and coupling operators are provided in Appendix A.5.

4 Theoretical Analysis

We analyze the exact rooted-patch profile before sparse compression. For a graph G and radius r, let

$$\Phi_r(G) = \{\{\kappa(G[B_r(v)], v) \mid v \in V(G)\}\} \,, \tag{4}$$

where κ is an information-preserving, root-preserving canonical representation. In particular,

$$\kappa(P, v) = \kappa(P', v') \quad \Longleftrightarrow \quad (P, v) \simeq (P', v').$$

The profile retains both rooted-patch types and their multiplicities.

Proposition 1 (Invariance of the exact profile). For every permutation π of the node labels,

$$\Phi_r(\pi(G)) = \Phi_r(G). \tag{5}$$

A relabeling maps each root v to $\pi(v)$ and induces a root-preserving isomorphism between the corresponding radius-r patches. Their canonical observations are therefore identical, while the bijection between roots preserves multiplicities. The same argument applies to pair-rooted profiles whenever the root-selection rule is permutation equivariant. This invariance is preserved by a fixed deterministic sparse encoder followed by a permutation-invariant readout, provided that the local observation mapping is invariant under rooted isomorphisms.

Theorem 1 (Equal-depth refinement of 1-WL). For every $r \geq 0$, the color assigned to a node v after r iterations of 1-WL (Weisfeiler & Lehman, 1968; Morris et al., 2019) is determined by its exact radius-r rooted induced patch. Consequently,

$$C_r^{1\text{-WL}}(G) \neq C_r^{1\text{-WL}}(H) \quad \Longrightarrow \quad \Phi_r(G) \neq \Phi_r(H), \tag{6}$$

where $C_r^{1\text{-WL}}(G)$ denotes the graph-level color multiset after r refinement steps.

Proof sketch. The result follows by induction on r. A root-preserving isomorphism between two radius-r patches maps their roots and neighbors bijectively. By the induction hypothesis, corresponding neighbors receive identical $(r-1)$-step WL colors. The two roots therefore receive the same multiset of neighbor colors and hence the same color at step r. Applying this argument to all roots establishes the graph-level implication. The complete proof is provided in Appendix B.3.

Proposition 2 (Strict separation from stable 1-WL). There exist graphs that are indistinguishable to stable 1-WL but have different radius-one exact rooted-patch profiles.

Witness. Let $G = C_6$ and $H = 2C_3$, with uniform initial node labels. Both graphs contain six degree-two vertices, so stable 1-WL assigns them identical graph-level color multisets. However, every radius-one rooted induced patch in C_6 is a rooted path P_3, whereas every such patch in $2C_3$ is a rooted triangle K_3. Hence,

$$\Phi_1(C_6) \neq \Phi_1(2C_3). \tag{7}$$

Together, Theorem 1 and Proposition 2 show that the exact profile retains all equal-depth 1-WL distinctions and can additionally resolve graph pairs that stable 1-WL cannot distinguish. They do not imply that a fixed-radius profile uniformly dominates stable 1-WL on every graph pair.

Relation to higher-order refinement. For the adjacent-pair-rooted construction used in BREC, we further establish that its exact radius-2 profile is incomparable with the pair-based higher-order refinement defined in Appendix B.4. Separating examples in both directions are provided there.

5 Evaluation

5.1 Experimental setup

Our experiments evaluate whether an explicit, learnable graph representation can combine structural distinguishability and competitive prediction with information-source auditing and occurrence-level interpretation. We examine four complementary capabilities: **(i) Structural distinguishability and its limits.** On BREC, we compare exact rooted-patch profiles with sparse representations to assess which graph pairs can be distinguished and whether non-separation arises from the observation scope or subsequent encoding and readout. **(ii) Predictive utility.** Node- and graph-level benchmarks evaluate whether the framework supports competitive prediction with conventional downstream models. **(iii) Information-source contributions.** Controlled coalition experiments quantify the standalone predictive utility and marginal contributions of graph statistics, learned topology, and supplied attributes. **(iv) Structural tracing and model diagnosis.** Molecular case studies trace learned coordinates to concrete patch occurrences, supporting chemical hypothesis testing and fixed-model diagnosis of generalization failures. Together, these experiments assess both the predictive usefulness of explicit structural coordinates and the analyses they enable.

All supervised experiments follow train-only representation fitting: dictionaries, normalization statistics, and other data-dependent transformations are learned without access to test samples. TU benchmarks use stratified outer ten-fold assessment; within each outer fold, the remaining 90% of

Table 1: Pair separation on BREC. Baseline results are taken from Wang & Zhang (2024). The Regular family combines the Regular, 4-Vertex-Condition, and Distance-Regular categories. Complete protocols and robustness checks are provided in Appendix C.

Method	Basic (60)	Regular (140)	Ext. (100)	CFI (100)	Total (400)
3-WL	60	50	100	60	270
N_2 (exact 2-hop types)	60	138	100	0	298
GSN	60	99	95	0	254
I^2-GNN	60	100	100	21	281
Graphormer	16	12	41	10	79
Exact pair-rooted profile	60	140	100	0	300
K-SVD $(128, 16)$	60	139.0 ± 0.8	100	0	299.0 ± 0.8

the data are split 85%/15% for fitting and validation, and the held-out 10% is evaluated only after configuration selection. Airport experiments use transductive 80%/10%/10% label splits, repeated over 20 seeds for Brazil, Europe and USA airports data. OGB and Peptides-func use their official splits with seeds 0–9. BREC follows a separate unlabeled transductive protocol in which the dictionary may use patches from all evaluated graphs, but never pair identities or evaluation outcomes. Unless stated otherwise, XGBoost (Chen & Guestrin, 2016) is used as the downstream predictor. We also evaluate computational efficiency on the largest datasets, measuring runtime and peak memory for the complete pipeline, alongside representative GNN and graph Transformer baselines. Detailed evaluation protocols, configurations, hyperparameters, and efficiency analyses are provided in Appendix D.

Computational accessibility. Unlike conventional end-to-end graph learning pipelines, our method requires no GPU. Once the shared representation is constructed, ten MolHIV downstream seeds require only 26 minutes in total, compared with approximately two hours for a single seed of representative neural baselines. Complete runtime and memory measurements are provided in Appendix D.6.

5.2 Structural Distinguishability

We use BREC to examine which graph distinctions are available at the chosen observation scope and how well sparse dictionary encoding preserves them. Comparing exact profiles with their compressed representations separates limitations of the structural observations from losses introduced by encoding and readout. We construct exact radius-2 *adjacent-pair-rooted* observations. Each unordered adjacent pair $\{u, v\} \in E$ defines an induced patch with two distinguished roots. Its observation contains the canonical adjacency, root-role indicators, and a valid-node mask, with the two root orders canonicalized to the same observation. The exact graph profile records the resulting patch types and their frequencies. Without using attributes or labels, this profile separates 300 of the 400 non-isomorphic pairs (Table 1), including all Basic, Extension, and Regular pairs.

Crucially, the explicit profile also makes the source of non-separation inspectable. The remaining 100 CFI pairs (Cai et al., 1989) have identical radius-2 adjacent-pair-rooted patch multisets, so their distinction is absent from the chosen observations. No subsequent encoding or predictor operating solely on this profile can recover that distinction, establishing a $300/400$ ceiling on BREC under this observation scope. A shared K-SVD dictionary with $(K, s) = (128, 16)$ separates 300, 298, and 299 pairs across three dictionary seeds, retaining nearly all distinctions available in the exact profile. Thus, the unresolved CFI pairs reflect an observation-scope limitation, whereas the small additional shortfall in two seeds reflects limited robust separation after sparse encoding and readout. This comparison distinguishes information absent from the exact profile from distinctions that its compact representation does not resolve under the specified robustness criterion. Table 1 contextualizes these results against representative baselines reported on BREC. The closest exact local baseline, N_2 (Wang & Zhang, 2024), encodes radius-2 node-rooted isomorphism types and separates 298 pairs, while I_2-GNN (Huang et al., 2023), the strongest neural baseline reported in the original BREC study, separates 281. Complete evaluation criteria, tolerance analyses, and relabeling and reliability checks are provided in Appendix C.

Table 2: Predictive performance (%) on graph classification benchmarks. (a) Accuracy on topology-only and attributed TU datasets. (b) ROC–AUC on molecular benchmarks and average precision on Peptides-func.

(a) TU datasets

	Topology-only				Attributed				
Method	IMDB-B	IMDB-M	COLLAB	REDDIT-B	MUTAG	NCI1	PROTEINS	DD	ENZYMES
WL subtree	72.27	50.90	78.26	76.33	86.00	**85.16**	75.66	79.21	53.15
GIN	71.20	48.50	75.60	89.90	84.68	80.51	72.31	72.94	64.60
GSN-v	**74.51**	50.84	81.63	79.45	86.65	81.85	72.70	75.57	56.73
GNN-AK+	72.90	50.90	–	–	86.70	78.90	74.00	76.30	–
GraphMAE	70.80	**52.13**	68.16	86.00	86.32	78.64	72.14	77.46	58.67
Ours	73.87	50.97	80.72	**91.14**	**89.20**	83.09	**77.85**	**79.48**	**69.73**

(b) Molecular and long-range benchmarks

Method	MolHIV ROC–AUC	MolBACE ROC–AUC	MolTox21 ROC–AUC	Peptides-func AP
CIN	80.36	76.28	75.52	61.75
GSN	78.53	76.96	77.44	65.74
GraphGPS	77.31	80.36	74.07	65.74
Subgraphormer	78.77	82.00	76.42	62.91
GPSE	77.89	**82.90**	76.68	66.51
GraphViT	79.02	82.82	**78.15**	69.41
HOD-GNN	81.16	81.03	77.83	68.77
Ours	**81.47**	81.67	76.67	**71.57**

5.3 Downstream Prediction Tasks

We evaluate predictive performance using validation-selected representation configurations within the framework. Section 5.4 separately quantifies the standalone and conditional predictive utility of individual information sources. Our experiments span node- and graph-level tasks, covering topology-only networks, attributed graph benchmarks, molecular property prediction, and long-range graph classification. Baselines span message-passing GNNs (Xu et al., 2019; Zhao et al., 2022; Bouritsas et al., 2023), the protocol-matched Weisfeiler–Lehman subtree kernel or other representing method (Shervashidze et al., 2011; Ribeiro et al., 2017), graph Transformers (Rampášek et al., 2022; Zhang et al., 2023), and several recent methods (Bodnar et al., 2021; He et al., 2023; Bar-Shalom et al., 2024; Cantürk et al., 2024; Bao et al., 2025; Eitan et al., 2026). Full-pipeline runtime and peak-memory comparisons are provided in Appendix D. Baseline sources and evaluation protocols are detailed in Appendix D.5.

Table 2 summarizes graph-level prediction across topology-only TU, attributed TU, molecular, and long-range benchmarks. Across these substantially different domains, metrics, and information settings, best results among evaluated configurations remain competitive with classical kernels, message-passing and higher-order GNNs, and graph Transformers, while attaining the best performance among the compared methods on multiple datasets. Together with the Airport node-classification results reported in Appendix D, these results demonstrate that the same shared-dictionary and invariant-readout framework supports predictive learning across graph domains and task granularities.

5.4 Quantifying the Predictive Contributions of Topology and Attributes

Competitive predictive performance alone does not reveal what information a graph learner relies on. Similar scores may arise from local topology, coarse graph statistics, supplied attributes, or their combinations. Understanding these dependencies is important for interpreting benchmark performance and identifying when structural modeling contributes useful predictive information. Prior benchmark reassessments show that graph-aware models do not uniformly outperform simple or

Table 3: Predictive utility of graph statistics (S), learned topology (T), and supplied attributes (A), together with their normalized Shapley allocation under a common downstream probe. Performance is accuracy (%), except for MolHIV, which reports ROC–AUC (%).

	Predictive performance							Shapley share		
Dataset	S	T	A	ST	SA	TA	STA	ϕ_S	ϕ_T	ϕ_A
NCI1	68.7	75.6	67.3	76.7	72.9	78.0	78.8	26.5	**47.4**	26.1
ENZYMES	37.8	41.6	**68.1**	44.4	**69.7**	63.2	64.1	17.0	14.2	**68.8**
MolHIV	61.97	73.27	73.69	73.10	72.76	75.20	**75.94**	15.6	**42.1**	**42.3**

attribute-based baselines (Errica et al., 2020; Li et al., 2024). These findings indicate that predictive information varies substantially across datasets and that strong classification performance does not necessarily imply strong use of topology. GNNs and graph transformers can investigate such dependencies through input or architectural ablations. However, when topology and attributes are jointly encoded in latent representations, their individual predictive utility and contributions across combinations are not directly readable from the learned coordinates. Establishing these distinctions requires a dedicated information-source analysis.

Our framework supports this analysis through separately constructed and independently removable blocks: graph statistics (S), learned topology (T), and supplied attributes (A). We evaluate the complete coalition lattice $\{\varnothing, S, T, A, ST, SA, TA, STA\}$ using identical data splits and a common downstream evaluation protocol. Single-block performance measures standalone predictive utility, while coalition comparisons quantify how each source contributes in the presence of others. Normalized Shapley allocation summarizes these marginal contributions across coalition contexts. Together, these measurements characterize both which sources are predictive and how their predictive roles change when combined.

Table 3 illustrates three distinct regimes. NCI1 is topology dominated: T is the strongest isolated source and receives the largest Shapley allocation. ENZYMES is predominantly attribute driven, with topology providing limited additional utility once attributes are available. MolHIV exhibits complementary contributions from topology and molecular attributes, with their combination yielding the strongest prediction. These results turn dataset-dependent information reliance into explicit, quantitative comparisons rather than a conclusion inferred from overall accuracy alone. The audit therefore reveals not only whether a source is useful in isolation, but also whether its predictive information remains valuable alongside other sources. Complete coalition scores, uncertainties, interaction effects, and block definitions are provided in Appendix E.

5.5 Tracing Structural Evidence for Hypothesis Testing and Model Diagnosis

Beyond quantifying information-source contributions, we examine which concrete structural patterns underlie predictive signals and how their use affects model behavior. Our representation retains links from graph-level coordinates through sparse activations to their original rooted-patch occurrences. We illustrate this capability through two complementary case studies: MolHIV connects learned coordinates to chemically defined hypotheses, while MolBACE uses targeted interventions to diagnose a structural dependency that fails to transfer across dataset splits.

On MolHIV (Figure 2a), tracing influential activations to their patches and parent molecules reveals an aryl–azo subfamily associated with sulfonate context, motivating the hypothesis that activity enrichment depends on their combination rather than on the aryl–azo structure alone. Molecule-level chemical definitions and statistical analysis show little enrichment for aryl–azo structures without sulfonate groups, but substantial enrichment among labelled active molecules when at least two sulfonate groups are present. This pattern appears across the official training, validation, and test splits, and the interaction remains significant after controlling for molecular size, aromaticity, and split membership. The learned coordinate therefore leads from recurring molecular occurrences to a chemically specified and statistically testable hypothesis.

On MolBACE (Figure 2b), an influential aromatic-root-conditioned coordinate is traced to an aryl-like local carbon environment. Its activation–label association agrees between training and validation but reverses on the official test split, indicating a candidate source of generalization failure. We

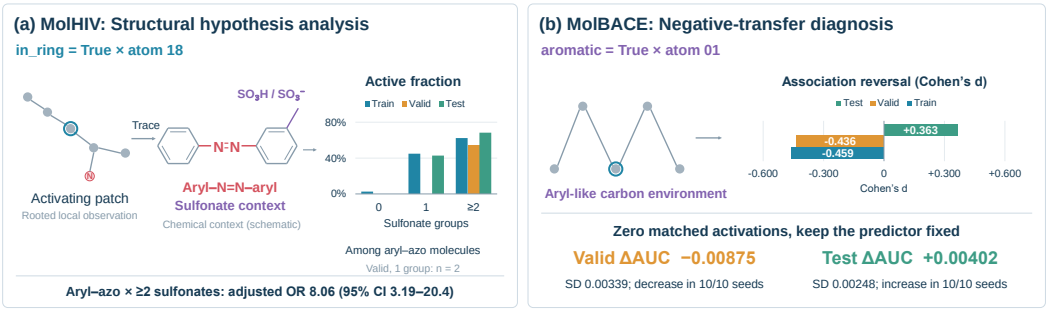

Figure 2: Occurrence-level tracing for chemical hypothesis testing and model diagnosis. (**a**) MolHIV: `root.in_ring=True` $\times$ $\texttt{atom}_{18}$ leads to an aryl–azo subfamily with sulfonate context. The interaction with at least two sulfonate groups has adjusted OR $= 8.06$ (95% CI: 3.19–20.4, $p = 1.03 \times 10^{-5}$); the aryl–azo main effect is not significant (OR $= 0.493$, $p = 0.088$). (**b**) MolBACE: `aromatic root` $\times$ $\texttt{atom}_{01}$ traces to an aryl-like local carbon environment. Its activation–label association reverses from $d = -0.459$ (train) and $d = -0.436$ (validation) to $d = +0.363$ (test). Removing the corresponding patch activations with the predictor fixed worsens validation ranking but improves test ranking across all ten model seeds.

remove only the corresponding patch-level coefficients, recompute the original graph-level readouts, and evaluate the unchanged predictor. This intervention decreases validation ROC–AUC by 0.00875 but increases test ROC–AUC by 0.00402, with the same direction across all ten model seeds, linking the association reversal to a measurable effect on prediction. Further analysis relates this behavior to differences in chemical-series composition and class-conditional difficulty, supporting the diagnosis of a validation-supported but test-fragile structural shortcut.

Together, these cases show how occurrence-level traceability supports both chemical hypothesis testing and targeted diagnosis of model generalization: learned coordinates guide the identification of concrete patterns, whose statistical associations and effects on model predictions can then be examined. Complete definitions, statistical tests, molecular examples, and retrospective analysis and intervention protocols are provided in Appendix F.

6 CONCLUSION

We introduced an explicit, data-driven graph representation framework that organizes rooted structural observations into shared, learnable coordinates. It supports separately accessible information blocks, explicit topology–attribute binding, and occurrence-level tracing. The exact rooted-patch profile refines equal-depth 1-WL and distinguishes some graph pairs unresolved by stable 1-WL, while its compact sparse representation retains nearly all observed BREC distinctions. Across node- and graph-level benchmarks, the framework achieves competitive predictive performance with conventional downstream models. Controlled information-source audits quantify standalone and conditional predictive contributions, while molecular case studies demonstrate how learned components support testable chemical hypotheses and fixed-model diagnosis of structural shortcuts. Together, these results show that explicit structural representations can support competitive graph learning while enabling direct auditing and deeper analysis of prediction and generalization.

Future work will investigate whether dictionaries pretrained on large and heterogeneous graph collections can provide reusable structural vocabularies for transferable graph learning. Our longer-term goal is a graph foundation model that learns transferable structural representations from topology alone, without relying on semantic attributes. We will also explore adaptive multiscale observations, scalable dictionary learning, and the stability and alignment of structural coordinates across datasets and domains. Finally, end-to-end optimization of decomposition and prediction, and integration with GNNs and graph transformers, may improve task adaptation while preserving explicit structural coordinates and occurrence-level traceability.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, code development, and the organization of experimental documentation. All AI-assisted code was reviewed and tested by the authors, and all reported results were independently generated and verified using the disclosed experimental protocols. AI-generated text and research suggestions were critically reviewed against the original experiments and relevant literature. The authors take full responsibility for the final text, claims, code, and experimental results presented in this work.

ETHICS STATEMENT

This work studies general-purpose graph representation methods using public benchmark datasets and does not involve human participants or the collection of personal data. We are not aware of direct ethical risks specific to the proposed method beyond those associated with the downstream use of graph-learning systems. All datasets are used in accordance with their respective licenses and established evaluation protocols.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the representation construction, rooted-patch extraction, dictionary learning, sparse encoding, readout, and evaluation protocols in the main paper and appendices. Complete proofs, hyperparameter settings, data splits, feature definitions, ablation results, and relabeling and reliability checks are included in the supplementary material. Anonymous source code and scripts for reproducing the reported experiments will be made available to reviewers during the review process.

REFERENCES

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. doi: 10.1109/TSP.2006.881199.

Linus Bao, Emily Jin, Michael M. Bronstein, İsmail İlkan Ceylan, and Matthias Lanzinger. Homomorphism counts as structural encodings for graph learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=qFw2RFJS5g.

Guy Bar-Shalom, Beatrice Bevilacqua, and Haggai Maron. Subgraphormer: Unifying subgraph GNNs and graph transformers via graph products. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=6djDWVTUEq.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2625–2640. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/157792e4abb490f99dbd738483e0d2d4-Paper.pdf.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2023. doi: 10.1109/TPAMI.2022.3154319.

J.-Y. Cai, M. Furer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. In *30th Annual Symposium on Foundations of Computer Science*, pp. 612–617, 1989. doi: 10.1109/SFCS.1989.63543.

Semih Cantürk, Renming Liu, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. ICML'24. JMLR.org, 2024.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. KDD '16, pp. 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Alexandre Duval and Fragkiskos D. Malliaros. Graphsvx: Shapley value explanations for graph neural networks. pp. 302–318, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-86519-1. doi: 10.1007/978-3-030-86520-7_19. URL https://doi.org/10.1007/978-3-030-86520-7_19.

Yam Eitan, Moshe Eliasof, Yoav Gelberg, Fabrizio Frasca, Guy Bar-Shalom, and Haggai Maron. On the expressive power of GNN derivatives. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Tvat33IDmK.

Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HygDF6NFPB.

Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. A generalization of vit/mlp-mixer to graphs. ICML'23. JMLR.org, 2023.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. KDD '22, pp. 594–604, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539321. URL https://doi.org/10.1145/3534678.3539321.

Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. Boosting the cycle counting power of graph neural networks with i^2-GNNs. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kDSmxOspsXQ.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Lecheng Kong, Yixin Chen, and Muhan Zhang. Geodesic graph neural network for efficient graph representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5896–5909, 2022.

Zhengdao Li, Yong Cao, Kefan Shuai, Yiming Miao, and Kai Hwang. Rethinking the effectiveness of graph classification datasets in benchmarks for assessing gnns. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 2144–2152, 2024. doi: 10.24963/ijcai.2024/237.

Hanbaek Lyu, Yacoub H. Kureh, Joshua Vendrow, and Mason A. Porter. Learning low-rank latent mesoscale structures in networks. *Nature Communications*, 15, 2024. doi: 10.1038/s41467-023-42859-2.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. *Provably Powerful Graph Networks*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: higher-order graph neural networks. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33014602. URL https://doi.org/10.1609/aaai.v33i01.33014602.

Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 vol.1, 1993. doi: 10.1109/ACSSC.1993.342465.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Vishal Rana, Jianhao Peng, Chao Pan, Hanbaek Lyu, Albert Cheng, Minji Kim, and Olgica Milenkovic. Interpretable online network dictionary learning for inferring long-range chromatin interactions. *PLoS Computational Biology*, 20(5):e1012095, 2024. doi: 10.1371/journal.pcbi.1012095.

Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo. struc2vec: Learning node representations from structural identity. KDD '17, pp. 385–394, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098061. URL https://doi.org/10.1145/3097983.3098061.

Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In David van Dyk and Max Welling (eds.), *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 488–495, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, 12(null):2539–2561, November 2011. ISSN 1532-4435.

Cédric Vincent-Cuaz, Titouan Vayer, Rémi Flamary, Marco Corneli, and Nicolas Courty. Online graph dictionary learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10564–10574. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/vincent-cuaz21a.html.

Jinlu Wang, Jipeng Guo, Yanfeng Sun, Junbin Gao, Shaofan Wang, Yachao Yang, and Baocai Yin. Dgnn: Decoupled graph neural networks with structural consistency between attribute and graph embedding representations. *IEEE Transactions on Big Data*, 11(4):1813–1827, 2025. doi: 10.1109/TBDATA.2024.3489420.

Yanbo Wang and Muhan Zhang. An empirical study of realized gnn expressiveness. ICML'24. JMLR.org, 2024.

Boris Weisfeiler and A. A. Lehman. A Reduction of a Graph to a Canonical Form and an Algebra Arising During This Reduction. *Nauchno-Technicheskaya Informatsia*, Ser. 2(N9):12–16, 1968.

Zhenpeng Wu, Jiamin Chen, and Jianliang Gao. Improving the representational power of graph neural networks via mixed substructure learning. *Inf. Fusion*, 126(PA), June 2026. ISSN 1566-2535. doi: 10.1016/j.inffus.2025.103558. URL https://doi.org/10.1016/j.inffus.2025.103558.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Minghao Xu, Hang Wang, Bingbing Ni, Wenjun Zhang, and Jian Tang. GraphSAD: Learning graph representations with structure-attribute disentanglement, 2021. URL https://openreview.net/forum?id=ZS-9XoX20AV.

Xuemeng Zhai, Wanlei Zhou, Gaolei Fei, Cai Lu, and Guangmin Hu. Network sparse representation: Decomposition, dimensionality-reduction and reconstruction. *Information Sciences*, 521:307–325, 2020. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2020.02.022. URL https://www.sciencedirect.com/science/article/pii/S0020025520300979.

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power of GNNs via graph biconnectivity. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=r9hNv76KoT3.

Muhan Zhang and Pan Li. Nested graph neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=7_eLEvFjCi3.

Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any GNN with local structure awareness. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Mspk_WYKoEH.

A Additional Methodological Details

A.1 Rooted Structural Observations

Let $G = (V, E, \mathbf{X}, \mathbf{E}_a)$ denote a graph, where V and E are its node and edge sets, and $\mathbf{X}$ and $\mathbf{E}_a$ denote optional node and edge attributes. Our representation begins by decomposing G into a multiset of local observations, each defined relative to a structural root. Depending on the task, the root is either a single node or an unordered pair of nodes:

$$\mathcal{R}(G) = \begin{cases} \big\{\{v\} : v \in V\big\}, & \text{node-rooted setting,} \\ \big\{\{u, v\} : (u, v) \in E\big\}, & \text{adjacent-pair-rooted setting.} \end{cases} \tag{8}$$

The pair-rooted setting treats $\{u, v\}$ as an unordered root; exchanging its two endpoints therefore does not define a different observation.

For a root $R \in \mathcal{R}(G)$, we define the distance from a node w to the root as

$$d_G(w, R) = \min_{v \in R} d_G(w, v), \tag{9}$$

where $d_G(\cdot, \cdot)$ is the shortest-path distance in G. The radius-r rooted patch is the induced subgraph

$$P_r(G, R) = G[\{w \in V : d_G(w, R) \le r\}], \tag{10}$$

together with the identity and role of its root. Because the patch is induced, it retains all edges of G between the selected nodes rather than only the edges traversed during neighborhood expansion.

A rooted patch does not possess a natural matrix coordinate system. We therefore introduce a root-aware structural mapping

$$\phi_r : \big(P_r(G, R), R\big) \longmapsto \mathbf{y}^{(r)}_{G,R} \in \mathbb{R}^{d_r}, \tag{11}$$

which converts the patch into a fixed-dimensional structural observation. The abstract mapping is required to be consistent under rooted isomorphism. Specifically, if $(P, R) \simeq (P', R')$ denotes an isomorphism that preserves adjacency, root roles, and all structural channels included in the observation, then

$$(P, R) \simeq (P', R') \quad \Longrightarrow \quad \phi_r(P, R) = \phi_r(P', R'). \tag{12}$$

For an unordered pair root, the isomorphism may exchange the two root nodes. The resulting observation must therefore be invariant to this exchange.

In practice, ϕ_r may contain the rooted patch adjacency, root and shell indicators, valid-node masks, and optional typed node or edge channels. Patches with different numbers of nodes are embedded into a common capacity using explicit masks, so that padded coordinates remain distinguishable from genuine graph structure. Exact local canonicalization and deterministic typed structural ordering provide two realizations of ϕ_r; their dataset-specific use is described in Appendix A.2.

Collecting the observations over all roots produces an unordered multiset

$$\mathcal{Y}_r(G) = \left\{\!\!\left\{ \mathbf{y}^{(r)}_{G,R} : R \in \mathcal{R}(G) \right\}\!\!\right\}. \tag{13}$$

Thus, no correspondence or ordering is assumed between different patch occurrences. Reordering the roots only permutes the elements of $\mathcal{Y}_r(G)$, which is removed by the permutation-invariant readout described in Appendix A.4. When multiple radii are used, the corresponding multisets $\{\mathcal{Y}_r(G) : r \in \mathcal{H}\}$ are retained as separate structural scales before their graph-level representations are concatenated.

For attributed graphs, each structural observation may additionally be associated with an aligned semantic descriptor

$$\mathbf{p}^{(r)}_{G,R} = \psi_r(P_r(G, R), R; \mathbf{X}, \mathbf{E}_a). \tag{14}$$

The structural observation $\mathbf{y}^{(r)}_{G,R}$ and semantic descriptor $\mathbf{p}^{(r)}_{G,R}$ refer to the same patch occurrence. They can either be aggregated independently or coupled before graph-level aggregation. These two forms of topology–semantic integration are detailed in Appendix A.5.

Table 4: Principal rooted-patch constructions used in our experiments. Dataset-specific radii, capacities, and truncation statistics are reported in Appendix D.1.

Setting	Structural root	Patch scope	Local coordinate construction
BREC	unordered adjacent pair	pair-centered induced patch	exact root-preserving canonicalization
Attributed TU audit	single node	multi-radius induced ego patch	exact root-preserving canonicalization
Molecular benchmarks	single atom	radius-2 induced ego patch	root-first typed-WL shell ordering
Other graph benchmarks	task-specific node or structural root	local induced neighborhood	exact or deterministic structural ordering

A.2 Dataset-Adapted Patch Construction

The formulation in Appendix A.1 does not require all graph domains to use the same local coordinate construction. Instead, each implementation must specify (i) the structural roots, (ii) the scope of the induced patch, and (iii) a root-aware mapping from the unordered patch to a fixed-dimensional observation. We use exact local canonicalization for symmetry-sensitive benchmarks and deterministic typed ordering for attributed graphs where exact canonicalization would be unnecessarily expensive. These variants share the same subsequent dictionary-learning, sparse-coding, and readout pipeline.

Exact pair-rooted construction for BREC. BREC contains highly symmetric and largely unattributed graphs, for which node types and local refinement frequently leave many vertices indistinguishable. We therefore use exact local canonicalization rather than a heuristic node ordering. In the main BREC configuration, every edge defines an unordered root pair $R = \{u, v\}$. The corresponding radius-r node set is

$$V_r(G, \{u, v\}) = \{w \in V : \min(d_G(w, u), d_G(w, v)) \le r\}, \tag{15}$$

and the patch is the induced subgraph $G[V_r(G, \{u, v\})]$.

To canonicalize this patch, the two root nodes are first assigned distinct root colors and all remaining nodes receive a common non-root color. We compute a color-preserving canonical adjacency for both assignments of the two root colors. Let

$$\kappa(P; u \mapsto 1, v \mapsto 2) \quad \text{and} \quad \kappa(P; u \mapsto 2, v \mapsto 1)$$

denote the resulting canonical keys. The representation of the unordered root pair is defined by

$$\kappa_{\text{pair}}(P, \{u, v\}) = \min_{\text{lex}} \{\kappa(P; u \mapsto 1, v \mapsto 2), \kappa(P; u \mapsto 2, v \mapsto 1)\}. \tag{16}$$

Consequently, exchanging u and v, changing the original node labels, or changing the traversal order produces the same local observation.

After canonicalization, the patch is represented by

$$\mathbf{y}^{(r)}_{G,\{u,v\}} = [\text{ut}(\mathbf{A}_{\text{can}}) \parallel \mathbf{q}_1 \parallel \mathbf{q}_2 \parallel \mathbf{m}], \tag{17}$$

where $\text{ut}(\cdot)$ extracts the upper-triangular adjacency, $\mathbf{q}_1$ and $\mathbf{q}_2$ identify the two canonical root roles, and $\mathbf{m}$ is the valid-node mask. Canonicalization is performed independently within each local patch; no canonical ordering of the complete graph is required. The collection of patches is subsequently treated as an unordered multiset.

Exact node-rooted construction for attributed TU audits. The unified $S/T/A$ audit on attributed TU datasets uses node-rooted induced patches at multiple radii. For root v, the root receives a distinguished color and the remaining patch nodes receive the non-root color. Color-preserving canonicalization produces a canonical adjacency matrix and a canonical root marker. The structural observation is

$$\mathbf{y}^{(r)}_{G,v} = [\text{ut}(\mathbf{A}_{\text{can}}) \parallel \mathbf{q}_{\text{root}} \parallel \mathbf{m}]. \tag{18}$$

Node and edge attributes are not used to determine this topological canonical form. They are independently aggregated into the attribute block or aligned with the root, structural shells, and patch edges when explicit topology–semantic coupling is evaluated. This separation prevents semantic attributes from being implicitly absorbed into the topology-only block.

Typed node-rooted construction for molecular graphs. For the molecular benchmarks, every atom v defines an exact radius-2 induced ego patch. Its nodes are assigned to the root, first shell, and second shell:

$$S_j(v) = \{u \in V : d_G(u, v) = j\}, \qquad j \in \{0, 1, 2\}. \tag{19}$$

The root occupies the first coordinate, followed by the nodes in $S_1(v)$ and $S_2(v)$. Within each shell, nodes are ordered using rooted typed-WL refinement. Initial refinement labels include the shell index, atom attributes, and local patch degree; subsequent iterations aggregate neighbor labels together with bond types. The resulting coordinate order is therefore

$$\pi_v = [v \parallel \mathrm{sort}_{\mathrm{tWL}}(S_1(v)) \parallel \mathrm{sort}_{\mathrm{tWL}}(S_2(v))]. \tag{20}$$

Using this order, a molecular patch observation concatenates

$$\mathbf{y}_{G,v} = [\mathbf{y}_{\mathrm{top}} \parallel \mathbf{y}_{\mathrm{bond}} \parallel \mathbf{y}_{\mathrm{atom}} \parallel \mathbf{y}_{\mathrm{shell}} \parallel \mathbf{y}_{\mathrm{mask}}], \tag{21}$$

where the topology and bond channels are stored over upper-triangular node pairs, the atom and shell channels are position-wise, and the mask identifies valid coordinates.

For datasets other than BREC, we first order nodes within each rooted patch using WL refinement keys. If ties remain, we resolve them through exact canonicalization using the full patch connectivity, while preserving root roles and the WL-induced partition. This two-stage procedure ensures relabeling invariance while restricting exact canonicalization to patches with unresolved ties, avoiding its cost for every rooted patch.

Capacity, padding, and truncation. For a prescribed patch capacity M, observations with fewer than M nodes are zero-padded. A valid-node mask is always included, so padding cannot be confused with isolated nodes or absent edges. For downstream datasets, M is determined without using validation or test labels and, where applicable, from the patch-size distribution of the training split. If a patch exceeds M, nodes are retained according to the same root-aware deterministic ordering used to construct its coordinates:

$$\pi_R^{(M)} = \mathrm{prefix}_M(\pi_R). \tag{22}$$

The induced adjacency and aligned semantic channels are then restricted to the retained nodes. Dataset-level truncation frequencies are reported with the experimental configuration.

Other benchmark-specific variants. The remaining topology-only and node-level experiments instantiate the same construction through different choices of roots, radii, capacities, and local ordering rules. These choices alter the structural scope of an observation but not the representation pipeline:

$$G \longrightarrow \left\{\mathbf{y}_{G,R}^{(r)}\right\}_{R,r} \longrightarrow \text{shared dictionary and sparse codes} \longrightarrow \text{permutation-invariant readout}. \tag{23}$$

Their exact configurations are listed in Appendix D.1.

A.3 Shared Dictionary Learning and Sparse Coding

Given the rooted structural observations defined in Appendix A.1, we learn a common structural coordinate system shared across patches and graphs. Let

$$\mathbf{Y} = [\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_M] \in \mathbb{R}^{d_p \times M} \tag{24}$$

denote the matrix containing M preprocessed patch observations from the dictionary-fitting collection. Here, $\widetilde{\mathbf{y}}_i$ is obtained from the raw structural observation $\mathbf{y}_i$ using the scaling and normalization rules described below.

We seek a dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{d_p \times K}$ and sparse codes $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_M] \in \mathbb{R}^{K \times M}$ such that

$$\min_{\mathbf{D},\mathbf{C}} \sum_{i=1}^{M} w_i \|\widetilde{\mathbf{y}}_i - \mathbf{D}\mathbf{c}_i\|_2^2, \quad \text{s.t.} \quad \|\mathbf{c}_i\|_0 \le s, \qquad \|\mathbf{d}_k\|_2 = 1. \tag{25}$$

The number of dictionary atoms is denoted by K, and $s \ll K$ is the maximum number of active atoms for each patch. The weights $w_i > 0$ account for the multiplicities of repeated structural observations. If every patch occurrence is stored explicitly, $w_i = 1$.

Frequency-weighted patch types. Many graphs contain repeated rooted patches, and the same canonical patch can also occur across different graphs. For datasets on which exact canonical patch keys are available, we collect the distinct observations $\{\widetilde{\mathbf{y}}_j\}_{j=1}^{M_{\text{type}}}$ and assign each observation its occurrence count

$$w_j = \sum_G \sum_{R \in \mathcal{R}(G)} \mathbb{I}\left[\phi(P_r(G,R),R) = \widetilde{\mathbf{y}}_j\right]. \tag{26}$$

Optimizing equation 25 over these weighted patch types is equivalent to optimizing over the fully expanded patch collection, while avoiding repeated storage and sparse coding of identical observations. This frequency-weighted implementation is used for BREC and the canonical TU experiments.

Observation scaling. A patch observation may concatenate blocks with substantially different dimensions and numerical densities, such as topology, atom attributes, bond attributes, shell indicators, and masks. Without scaling, a high-dimensional block can dominate the reconstruction objective solely because it contains more coordinates. We therefore optionally apply a diagonal block scaler

$$\overline{\mathbf{y}}_i = \mathbf{\Lambda}\mathbf{y}_i, \tag{27}$$

where the entries of $\mathbf{\Lambda}$ are constant within each observation block. For equal-energy scaling, the coefficient for block b is

$$\lambda_b = \left(\frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{y}_i^{(b)}\right\|_2^2 + \epsilon\right)^{-1/2}. \tag{28}$$

The scaler is estimated only from the collection used to fit the dictionary and is subsequently fixed.

For frequency-weighted canonical patch types, observations are additionally normalized to unit length:

$$\widetilde{\mathbf{y}}_i = \frac{\overline{\mathbf{y}}_i}{\max(\|\overline{\mathbf{y}}_i\|_2, \epsilon)}. \tag{29}$$

This prevents patch size or raw vector energy from determining the dictionary fit. In the streaming implementation for heterogeneous molecular patches, we instead retain the scaled magnitudes after equal-energy block normalization. The exact preprocessing adopted by each experiment is listed in Appendix D.1.

Sparse coding by orthogonal matching pursuit. For a fixed dictionary, each patch is encoded by solving

$$\widehat{\mathbf{c}}_i = \arg\min_{\mathbf{c}} \|\widetilde{\mathbf{y}}_i - \mathbf{D}\mathbf{c}\|_2^2, \qquad \text{s.t.} \quad \|\mathbf{c}\|_0 \le s. \tag{30}$$

We approximate this problem using orthogonal matching pursuit (OMP). At each iteration, OMP selects the atom most correlated with the current residual and then refits the coefficients over the selected support. The resulting codes are signed; their signs and magnitudes jointly describe how the learned atoms combine to approximate a patch.

K-SVD dictionary updates. After sparse coding, the dictionary atoms are updated one at a time. For atom k, let

$$\Omega_k = \{i : c_{ki} \neq 0\} \tag{31}$$

be the set of observations that activate this atom. Removing the contribution of atom k gives the restricted residual

$$\mathbf{E}_k = \mathbf{Y}_{\Omega_k} - \sum_{j \neq k} \mathbf{d}_j \mathbf{c}_{j,\Omega_k}. \tag{32}$$

The K-SVD update obtains the best rank-one approximation

$$\mathbf{E}_k \approx \mathbf{u}_1 \sigma_1 \mathbf{v}_1^\top \tag{33}$$

and sets

$$\mathbf{d}_k \leftarrow \mathbf{u}_1, \qquad \mathbf{c}_{k,\Omega_k} \leftarrow \sigma_1 \mathbf{v}_1^\top. \tag{34}$$

For frequency-weighted observations, the rank-one update is applied to the correspondingly weighted residual. The implementation uses either an exact leading singular vector or power iteration, depending on the scale of the experiment.

If an atom has an empty support, it is replaced using an observation with a large weighted reconstruction residual. This prevents permanently inactive atoms while directing unused dictionary capacity toward poorly reconstructed structural patterns. Sparse coding and atom updates alternate for the prescribed number of iterations.

Initialization and coordinate stabilization. Dictionary atoms are initialized from normalized real patch observations, using frequency-weighted random sampling or a K-means++-style selection. Initializing from observed patches provides a stable starting point and makes the initial atoms directly traceable to genuine local structures.

A dictionary atom and its coefficients have an inherent sign ambiguity: $\mathbf{d}_k c_{ki} = (-\mathbf{d}_k)(-c_{ki})$. After fitting, we fix the sign of each atom by requiring its largest-magnitude coordinate to be nonnegative. Atoms are then ordered by their weighted activation magnitude,

$$u_k = \sum_{i=1}^{M} w_i |c_{ki}|, \tag{35}$$

from highest to lowest usage. These conventions do not change the reconstruction but provide a deterministic coordinate convention within a fitted dictionary.

Streaming dictionary learning. For large collections that cannot be materialized as a single patch matrix, we use a graph-balanced mini-batch implementation. Each update batch contains patches sampled from the current set of training graphs together with patches from a replay reservoir. OMP and one or more K-SVD sweeps are then applied to this batch. Graph-balanced sampling prevents large graphs from dominating solely through their number of rooted patches, while the replay reservoir reduces forgetting of patterns observed in earlier batches. This procedure approximates the same sparse reconstruction objective in equation 25 without storing all patch occurrences simultaneously.

Fixed-dictionary encoding. Once a dictionary has been fitted under the protocol of an experiment, its scaler, atom order, and atom signs are frozen. Every patch from a new graph is processed using the same coordinate construction and scaler and is encoded by equation 30 without updating the dictionary:

$$\mathbf{C}_G = \left[\widehat{\mathbf{c}}_{G,R_1}, \ldots, \widehat{\mathbf{c}}_{G,R_{N_G}}\right]. \tag{36}$$

Because all patches are expressed in the same K-dimensional atom coordinates, their activations are comparable across patch occurrences and graphs. The graph-level structural representation is obtained by aggregating these codes with the permutation-invariant readout described in Appendix A.4.

The dictionary objective uses structural observations and their occurrence frequencies but no task labels. The precise fitting scope—including the train-only downstream protocol and the transductive unlabeled BREC protocol—is specified in Appendix D.1.

A.4 Permutation-Invariant Readout

After fixed-dictionary encoding, a graph G with N_G rooted patches is represented by the sparse-code matrix

$$\mathbf{C}_G = [\mathbf{c}_1, \ldots, \mathbf{c}_{N_G}] \in \mathbb{R}^{K \times N_G}. \tag{37}$$

Because the columns correspond to an unordered collection of patch occurrences, we convert $\mathbf{C}_G$ into a fixed-dimensional representation using symmetric aggregation.

For structural atom k, representative readout coordinates include its signed mean activation, absolute activation magnitude, activation energy, and activation frequency:

$$\mu_k(G) = \frac{1}{N_G} \sum_{i=1}^{N_G} c_{ki}, \tag{38}$$

$$a_k(G) = \frac{1}{N_G} \sum_{i=1}^{N_G} |c_{ki}|, \tag{39}$$

$$q_k(G) = \frac{1}{N_G} \sum_{i=1}^{N_G} c_{ki}^2, \tag{40}$$

$$f_k(G) = \frac{1}{N_G} \sum_{i=1}^{N_G} \mathbb{I}[|c_{ki}| > \tau], \tag{41}$$

where τ is a numerical activity threshold. Depending on the experiment, we additionally retain extrema of the signed or absolute coefficients. These statistics respectively describe the direction, magnitude, energy, prevalence, and strongest occurrence of each learned structural atom.

We also summarize the patch reconstruction residuals

$$e_i = \|\widetilde{\mathbf{y}}_i - \mathbf{D}\mathbf{c}_i\|_2 \tag{42}$$

using their mean, standard deviation, and maximum. The resulting topological representation takes the general form

$$\mathbf{h}_T(G) = \rho\left(\mathbf{C}_G, \mathbf{e}_G\right), \tag{43}$$

where ρ concatenates the selected atom-wise and residual statistics. When several patch radii or structural views are used, their readouts are computed separately and concatenated:

$$\mathbf{h}_T(G) = \Big\|_{r \in \mathcal{H}} \rho_r \left(\mathbf{C}_G^{(r)}, \mathbf{e}_G^{(r)}\right). \tag{44}$$

For canonical patch types stored with multiplicities, the same quantities are computed without explicitly expanding repeated occurrences. If patch type j occurs n_j times, each sum is weighted by n_j and normalized by $\sum_j n_j$. This produces exactly the same readout as aggregating the expanded patch multiset.

The readout is invariant to the order in which patches are extracted. A reordering of patch occurrences right-multiplies the code matrix by a permutation matrix $\mathbf{\Pi}$:

$$\mathbf{C}_G \longmapsto \mathbf{C}_G\mathbf{\Pi}. \tag{45}$$

All quantities in equation 41 are symmetric over the columns of $\mathbf{C}_G$, and therefore

$$\rho(\mathbf{C}_G\mathbf{\Pi}, \mathbf{e}_G\mathbf{\Pi}) = \rho(\mathbf{C}_G, \mathbf{e}_G). \tag{46}$$

A complete permutation-invariance argument, including the rooted-observation construction, is provided in Appendix B.2.

Graph size, patch count, and related global descriptors may be retained in the separate statistical block $\mathbf{h}_S(G)$. In the information-block audit, such quantities are excluded from $\mathbf{h}_T(G)$ to prevent overlap between the statistical and topological blocks. For node-level tasks, the same readout is applied to the patch occurrences associated with a target node, producing a node representation $\mathbf{h}_T(v)$. The precise readout coordinates used for each benchmark are listed in Appendix D.1.

A.5 Topology–Semantic Integration

Independent information blocks. For attributed graphs, we distinguish three independently constructed information blocks:

$$\mathbf{h}_S(G), \qquad \mathbf{h}_T(G), \qquad \mathbf{h}_A(G), \tag{47}$$

where $\mathbf{h}_S(G)$ contains graph-level statistical and compositional descriptors, $\mathbf{h}_T(G)$ is the sparse structural readout defined in Appendix A.4, and $\mathbf{h}_A(G)$ contains independently aggregated node and edge attributes. Their direct combination is

$$\mathbf{h}_{\mathrm{STA}}(G) = [\mathbf{h}_S(G) \,\|\, \mathbf{h}_T(G) \,\|\, \mathbf{h}_A(G)]\,. \tag{48}$$

The representations ST, SA, and TA are defined analogously by selecting the corresponding blocks.

Independent concatenation preserves which statistical, structural, and semantic information is present at the graph level. However, because each block is aggregated separately, it does not explicitly preserve where a semantic attribute occurs relative to the learned structural atoms. Two graphs may therefore have similar marginal structural and attribute summaries while differing in how their attributes are distributed over local structural environments.

Patch-aligned structure–semantic coupling. To retain this correspondence, we associate each structural code with semantic descriptors derived from the same rooted patch. Recall that

$$\mathbf{C}_G = [\mathbf{c}_1, \ldots, \mathbf{c}_{N_G}] \in \mathbb{R}^{K \times N_G} \tag{49}$$

contains the sparse codes of the N_G rooted patches in G. We define the non-negative patch-by-atom activation matrix

$$\mathbf{Z}_G = |\mathbf{C}_G|^\top \in \mathbb{R}^{N_G \times K}. \tag{50}$$

Let

$$\mathbf{P}_G \in \mathbb{R}^{N_G \times d_p}, \qquad \mathbf{R}_G \in \mathbb{R}^{N_G \times d_r} \tag{51}$$

denote patch-level and root-level semantic descriptors, respectively. The i-th rows of $\mathbf{P}_G$, $\mathbf{R}_G$, and $\mathbf{Z}_G$ always correspond to the same rooted patch occurrence.

The basic patch-level coupling operator is

$$\mathbf{B}_{\text{patch-mass}}(G) = \frac{1}{\max(N_G, 1)} \mathbf{P}_G^\top \mathbf{Z}_G \in \mathbb{R}^{d_p \times K}. \tag{52}$$

Its (a, k)-th coordinate measures the average co-activation between semantic dimension a and structural atom k. Unlike independent pooling, this matrix records whether a semantic pattern and a learned structural component occur in the same local observations.

Conditional coupling operators. The value in equation 52 depends on both the prevalence of a semantic attribute and the activation of its associated structural atoms. For non-negative semantic channels, we additionally consider frequency-normalized conditional forms:

$$\mathbf{B}_{\text{patch-cond}}(G) = \mathrm{Diag}\left(\mathbf{P}_G^\top \mathbf{1} + \epsilon\right)^{-1} \mathbf{P}_G^\top \mathbf{Z}_G, \tag{53}$$

$$\mathbf{B}_{\text{root-cond}}(G) = \mathrm{Diag}\left(\mathbf{R}_G^\top \mathbf{1} + \epsilon\right)^{-1} \mathbf{R}_G^\top \mathbf{Z}_G, \tag{54}$$

where $\epsilon > 0$ prevents division by zero. The first operator describes the distribution of structural-atom activations conditioned on patch-level semantics, whereas the second conditions the activation distribution on the semantic attributes of the root.

We also consider the reverse conditional direction:

$$\mathbf{B}_{\text{atom-cond}}(G) = \mathrm{Diag}\left(\mathbf{Z}_G^\top \mathbf{1} + \epsilon\right)^{-1} \mathbf{Z}_G^\top \mathbf{P}_G \in \mathbb{R}^{K \times d_p}. \tag{55}$$

Here, the k-th row represents the activation-weighted mean semantic descriptor of patches associated with structural atom k.

Different aspects of the correspondence can be retained by concatenating these matrices. For example,

$$\mathbf{B}_{\text{mass+root}}(G) = \mathbf{B}_{\text{patch-mass}}(G) \,\|\, \mathbf{B}_{\text{root-cond}}(G), \tag{56}$$

while the mass–condition, condition–root, and all-coupling variants use the corresponding subsets of $\mathbf{B}_{\text{patch-mass}}, \mathbf{B}_{\text{patch-cond}}$, and $\mathbf{B}_{\text{root-cond}}$. The matrices are vectorized before being passed to the downstream predictor.

Position-aware binding on attributed TU graphs. For the attributed TU experiments, semantic descriptors are further separated according to their positions within a rooted patch. For scale r, let $\mathbf{q}_{i,v}^{(r)}$ denote the attribute summary of patch occurrence i at semantic view v, where v corresponds to the root, the first shell, the second shell when available, or the edges contained in the patch. The activation of each patch is normalized across atoms:

$$\widetilde{\mathbf{z}}_i^{(r)} = \frac{\mathbf{z}_i^{(r)}}{\|\mathbf{z}_i^{(r)}\|_1 + \epsilon}. \tag{57}$$

The position-aware binding matrix is then

$$\mathbf{B}_v^{(r)}(G) = \frac{1}{\max(N_G^{(r)}, 1)} \sum_{i=1}^{N_G^{(r)}} \mathbf{q}_{i,v}^{(r)} \left(\widetilde{\mathbf{z}}_i^{(r)}\right)^\top. \tag{58}$$

Concatenating these matrices over structural scales and semantic views gives

$$\mathbf{h}_B(G) = \Big\|_{r,v} \operatorname{vec}\left(\mathbf{B}_v^{(r)}(G)\right). \tag{59}$$

The resulting representation is denoted by

$$\mathbf{h}_{\mathrm{STAB}}(G) = [\mathbf{h}_S(G) \,\|\, \mathbf{h}_T(G) \,\|\, \mathbf{h}_A(G) \,\|\, \mathbf{h}_B(G)]. \tag{60}$$

Thus, A records which attributes are present in the graph, whereas B records how those attributes are associated with structural atoms and local structural positions.

Graph-level interaction. When patch-aligned semantic observations are unavailable or when a coarser interaction is desired, we construct a graph-level outer product. Let $\mathbf{f}_T(G) \in \mathbb{R}^K$ denote the structural-atom activation frequency and let $\mathbf{h}_C(G)$ denote a graph-level semantic or compositional descriptor. We define

$$\mathbf{B}_{\mathrm{outer}}(G) = \operatorname{vec}\left(\mathbf{h}_C(G)\mathbf{f}_T(G)^\top\right). \tag{61}$$

This construction is less spatially resolved than the patch-aligned operators, but it still represents explicit composition–structure interactions rather than the independent presence of the two information sources.

Combined representation and permutation invariance. A selected set of coupling blocks is appended to the independently aggregated baseline:

$$\mathbf{h}_{\mathrm{coupled}}(G) = [\mathbf{h}_{\mathrm{base}}(G) \,\|\, \operatorname{vec}(\mathbf{B}_1(G)) \,\|\, \cdots \,\|\, \operatorname{vec}(\mathbf{B}_m(G))]. \tag{62}$$

The baseline and coupling operators used in each experiment are specified in Appendix D.5.

These coupling operators preserve permutation invariance whenever the underlying rooted observations and their aligned semantic descriptors are permutation consistent. A node relabeling induces the same permutation $\mathbf{\Pi}$ over patch occurrences:

$$\mathbf{P}_G \mapsto \mathbf{\Pi}\mathbf{P}_G, \qquad \mathbf{R}_G \mapsto \mathbf{\Pi}\mathbf{R}_G, \qquad \mathbf{Z}_G \mapsto \mathbf{\Pi}\mathbf{Z}_G. \tag{63}$$

Consequently,

$$(\mathbf{\Pi}\mathbf{P}_G)^\top(\mathbf{\Pi}\mathbf{Z}_G) = \mathbf{P}_G^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{Z}_G = \mathbf{P}_G^\top\mathbf{Z}_G, \tag{64}$$

and the same argument applies to the root-conditioned and position-aware operators. A complete invariance argument is provided in Appendix B.2.

B Proofs and Expressiveness Analysis

B.1 Definitions and Exact Structural Profiles

We first define the exact structural objects used in the expressiveness analysis. Unless otherwise stated, graphs are finite and undirected. The definitions apply to attributed graphs by requiring all isomorphisms to preserve the node and edge attributes included in the structural observation.

Rooted graphs and rooted isomorphism. Let $G = (V, E)$ be a graph and let $R \subseteq V$ be a structural root. We consider either a node root, for which $|R| = 1$, or an unordered pair root, for which $|R| = 2$. A rooted graph is denoted by (G, R).

Two rooted graphs (G, R) and (H, R') are rooted-isomorphic, written

$$(G, R) \simeq_{\mathrm{r}} (H, R'), \tag{65}$$

if there exists a graph isomorphism $\pi : V(G) \to V(H)$ such that

$$\pi(R) = R'. \tag{66}$$

For a node root, this requires the distinguished node to be preserved. For an unordered pair root, the two root nodes are preserved setwise and may be exchanged. If node or edge types are included, π must additionally preserve those types.

Admissible root families. A root family $\mathcal{R}(G)$ specifies the patch centers extracted from G. The principal choices are

$$\mathcal{R}_1(G) = \big\{\{v\} : v \in V(G)\big\}, \tag{67}$$

$$\mathcal{R}_2^{(\delta)}(G) = \big\{\{u, v\} : u \neq v,\ d_G(u, v) \leq \delta\big\}. \tag{68}$$

The adjacent-pair setting used in the main BREC experiment corresponds to $\delta = 1$. A root family is called isomorphism-equivariant if every graph isomorphism $\pi : G \to H$ induces the bijection

$$\mathcal{R}(H) = \big\{\pi(R) : R \in \mathcal{R}(G)\big\}. \tag{69}$$

Both root families above satisfy this property.

Rooted induced patches. For $R \in \mathcal{R}(G)$, define the distance from a node w to the root by

$$d_G(w, R) = \min_{v \in R} d_G(w, v). \tag{70}$$

The radius-r rooted patch is

$$P_r(G, R) = G\left[\{w \in V(G) : d_G(w, R) \leq r\}\right], \tag{71}$$

together with its root R. The use of an induced subgraph ensures that all edges among the selected nodes are retained.

The exact rooted type of this patch is its rooted-isomorphism class:

$$\tau_r(G, R) = \left[(P_r(G, R), R)\right]_{\simeq_{\mathrm{r}}}. \tag{72}$$

Thus,

$$\tau_r(G, R) = \tau_r(H, R') \quad \Longleftrightarrow \quad (P_r(G, R), R) \simeq_{\mathrm{r}} (P_r(H, R'), R'). \tag{73}$$

Exact structural profiles. The radius-r exact structural profile of G is the multiset of rooted patch types over its admissible roots:

$$\Phi_r^{\mathcal{R}}(G) = \{\{\tau_r(G, R) : R \in \mathcal{R}(G)\}\}. \tag{74}$$

Equivalently, the profile can be represented by its multiplicity function

$$m_{r,G}^{\mathcal{R}}(\tau) = \left|\{R \in \mathcal{R}(G) : \tau_r(G, R) = \tau\}\right|. \tag{75}$$

Two profiles are equal if and only if their multiplicities agree for every rooted type:

$$\Phi_r^{\mathcal{R}}(G) = \Phi_r^{\mathcal{R}}(H) \quad \Longleftrightarrow \quad m_{r,G}^{\mathcal{R}}(\tau) = m_{r,H}^{\mathcal{R}}(\tau) \quad \text{for all } \tau. \tag{76}$$

When several patch radii are retained, the multiscale profile is the ordered tuple

$$\Phi_{\mathcal{H}}^{\mathcal{R}}(G) = \big(\Phi_r^{\mathcal{R}}(G)\big)_{r \in \mathcal{H}}, \tag{77}$$

where $\mathcal{H}$ is the selected set of radii. The tuple is ordered by radius, while every profile within it remains an unordered multiset.

For brevity, we write

$$\Phi_r^{(1)}(G) = \Phi_r^{\mathcal{R}_1}(G)$$

for the node-rooted profile and

$$\Phi_r^{(2,\delta)}(G) = \Phi_r^{\mathcal{R}_2^{(\delta)}}(G)$$

for the pair-rooted profile.

Exact vector realizations. A structural mapping

$$\phi_r : (P, R) \longmapsto \mathbf{y}_{P,R}$$

is an exact realization of the rooted type if

$$\phi_r(P, R) = \phi_r(P', R') \quad \Longleftrightarrow \quad (P, R) \simeq_{\mathrm{r}} (P', R'). \tag{78}$$

In this case, the vector multiset

$$\Phi^{\mathcal{R}}_{\phi,r}(G) = \{\{\phi_r(P_r(G, R), R) : R \in \mathcal{R}(G)\}\} \tag{79}$$

contains exactly the same information as $\Phi^{\mathcal{R}}_r(G)$. Exact root-preserving canonical adjacency provides such a realization. By contrast, deterministic typed refinement is not necessarily exact when unresolved symmetric ties remain.

Graph distinguishability. A profile distinguishes two graphs G and H if

$$\Phi^{\mathcal{R}}_r(G) \neq \Phi^{\mathcal{R}}_r(H). \tag{80}$$

We write

$$G \equiv_{\Phi^{\mathcal{R}}_r} H \tag{81}$$

when their profiles are equal. This equivalence relation allows the discriminative power of the exact profile to be compared with WL tests and other graph representations.

For two graph representations F_1 and F_2, we say that F_2 is at least as discriminative as F_1, written

$$F_1 \preceq F_2, \tag{82}$$

if

$$F_2(G) = F_2(H) \quad \Longrightarrow \quad F_1(G) = F_1(H) \tag{83}$$

for every pair G, H in the considered graph class. The relation is strict, written $F_1 \prec F_2$, if $F_1 \preceq F_2$ and there exists at least one pair for which

$$F_1(G) = F_1(H), \qquad F_2(G) \neq F_2(H). \tag{84}$$

Exact profiles versus learned sparse representations. The exact profile should be distinguished from its learned finite-dimensional approximation. The practical representation applies a fixed sparse encoder $\mathcal{E}_{\mathbf{D}}$ to each observation and then a symmetric readout ρ:

$$\Phi^{\mathcal{R}}_{\phi,r}(G) \xrightarrow{\mathcal{E}_{\mathbf{D}}} \{\{\mathbf{c}_{G,R} : R \in \mathcal{R}(G)\}\} \xrightarrow{\rho} \mathbf{h}_T(G). \tag{85}$$

The expressiveness results in the following sections concern the exact profile unless an explicit information-preservation condition is stated. Finite dictionaries, sparse approximation, and non-injective readouts may merge distinct exact profiles. Conditions under which exact distinctions are preserved by the sparse representation are discussed in Appendix B.5.

B.2 Permutation Invariance

We establish permutation invariance at three levels: the exact rooted-patch profile, the fixed-dictionary sparse representation, and the topology–semantic coupling operators. Throughout this section, let $\pi : V(G) \to V(G^\pi)$ be a bijective node relabeling, where G^π is obtained from G by consistently relabeling its nodes, edges, and any included node or edge attributes.

Lemma 1 (Equivariance of rooted patches)**.** *Let $\mathcal{R}$ be an isomorphism-equivariant root family. For every $R \in \mathcal{R}(G)$,*

$$(P_r(G, R), R) \simeq_{\mathrm{r}} (P_r(G^\pi, \pi(R)), \pi(R)). \tag{86}$$

Proof. A graph isomorphism preserves shortest-path distances. Therefore, for every $w \in V(G)$,

$$d_G(w, R) = d_{G^\pi}(\pi(w), \pi(R)). \tag{87}$$

It follows that

$$\pi(\{w : d_G(w, R) \leq r\}) = \{w' : d_{G^\pi}(w', \pi(R)) \leq r\}. \tag{88}$$

Because both patches are induced subgraphs, the restriction of π to the selected nodes preserves every edge in the patches. It also maps R to $\pi(R)$ and preserves all node and edge attributes included in the observation. Hence it is a rooted isomorphism between the two patches. □

Theorem B.1 (Invariance of the exact structural profile)**.** *Let $\mathcal{R}$ be an isomorphism-equivariant root family. Then, for every node relabeling π,*

$$\Phi_r^{\mathcal{R}}(G) = \Phi_r^{\mathcal{R}}(G^\pi). \tag{89}$$

The same result holds for the multiscale profile $\Phi_{\mathcal{H}}^{\mathcal{R}}$.

Proof. By Lemma 1,

$$\tau_r(G, R) = \tau_r(G^\pi, \pi(R)) \tag{90}$$

for every $R \in \mathcal{R}(G)$. Since the root family is isomorphism-equivariant, the mapping

$$R \longmapsto \pi(R) \tag{91}$$

is a bijection from $\mathcal{R}(G)$ to $\mathcal{R}(G^\pi)$. It therefore preserves the multiplicity of every rooted patch type:

$$m_{r,G}^{\mathcal{R}}(\tau) = m_{r,G^\pi}^{\mathcal{R}}(\tau) \qquad \text{for every rooted type } \tau. \tag{92}$$

By equation 76, the two exact profiles are equal. Applying the same argument independently at every radius $r \in \mathcal{H}$ proves the multiscale result. □

Invariant vector observations. The preceding theorem concerns rooted-isomorphism types and does not require a particular vectorization. Suppose the structural mapping ϕ_r satisfies rooted-isomorphism consistency:

$$(P, R) \simeq_{\mathrm{r}} (P', R') \quad \Longrightarrow \quad \phi_r(P, R) = \phi_r(P', R'). \tag{93}$$

Lemma 1 then gives

$$\phi_r(P_r(G, R), R) = \phi_r(P_r(G^\pi, \pi(R)), \pi(R)). \tag{94}$$

Consequently, the vector profile in equation 79 is also permutation invariant. Injectivity of ϕ_r is not required for invariance; it is required only when the vector profile is expected to preserve all distinctions of the exact rooted-isomorphism profile.

Proposition 1 (Invariance of fixed-dictionary sparse representations)**.** *Let $\mathcal{E}_{\mathbf{D}}$ be a deterministic patch encoder associated with a fixed dictionary $\mathbf{D}$, and let ρ be invariant to the ordering of patch occurrences. If ϕ_r satisfies equation 93, then*

$$\mathbf{h}_T(G) = \mathbf{h}_T(G^\pi). \tag{95}$$

Proof. Corresponding rooted patches in G and G^π produce identical vector observations by equation 94. Because the dictionary and sparse encoder are fixed,

$$\mathcal{E}_{\mathbf{D}}\left(\phi_r(P_r(G, R), R)\right) = \mathcal{E}_{\mathbf{D}}\left(\phi_r(P_r(G^\pi, \pi(R)), \pi(R))\right). \tag{96}$$

The sparse codes are therefore identical up to the ordering of their patch occurrences.

Let $\mathbf{\Pi} \in \{0, 1\}^{N_G \times N_G}$ be the permutation matrix induced by the bijection $R \mapsto \pi(R)$. With sparse codes stored as columns,

$$\mathbf{C}_{G^\pi} = \mathbf{C}_G \mathbf{\Pi}^\top. \tag{97}$$

The associated reconstruction-error vector is permuted in the same way:

$$\mathbf{e}_{G^\pi} = \mathbf{\Pi} \mathbf{e}_G. \tag{98}$$

Since ρ is symmetric over patch occurrences,

$$\rho\left(\mathbf{C}_G \mathbf{\Pi}^\top, \mathbf{\Pi} \mathbf{e}_G\right) = \rho(\mathbf{C}_G, \mathbf{e}_G), \tag{99}$$

which proves the result. □

Topology–semantic coupling. For attributed graphs, a relabeling also permutes the patch-aligned semantic descriptors. Recall that

$$\mathbf{P}_G \in \mathbb{R}^{N_G \times d_p}, \qquad \mathbf{R}_G \in \mathbb{R}^{N_G \times d_r}, \qquad \mathbf{Z}_G = |\mathbf{C}_G|^\top \in \mathbb{R}^{N_G \times K}.$$

Provided that the semantic descriptors are constructed consistently from the same patch occurrences, relabeling gives

$$\mathbf{P}_{G^\pi} = \mathbf{\Pi}\mathbf{P}_G, \qquad \mathbf{R}_{G^\pi} = \mathbf{\Pi}\mathbf{R}_G, \qquad \mathbf{Z}_{G^\pi} = \mathbf{\Pi}\mathbf{Z}_G. \tag{100}$$

Proposition 2 (Invariance of explicit coupling)**.** *Under equation 100, the patch-mass, patch-conditioned, root-conditioned, atom-conditioned, and position-aware coupling operators defined in Appendix A.5 are permutation invariant.*

Proof. For patch-mass coupling,

$$\mathbf{P}_{G^\pi}^\top \mathbf{Z}_{G^\pi} = (\mathbf{\Pi}\mathbf{P}_G)^\top (\mathbf{\Pi}\mathbf{Z}_G) \tag{101}$$

$$= \mathbf{P}_G^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{Z}_G \tag{102}$$

$$= \mathbf{P}_G^\top \mathbf{Z}_G. \tag{103}$$

A permutation matrix also preserves the all-ones vector:

$$\mathbf{\Pi}^\top \mathbf{1} = \mathbf{1}. \tag{104}$$

Hence

$$\mathbf{P}_{G^\pi}^\top \mathbf{1} = \mathbf{P}_G^\top \mathbf{1}, \qquad \mathbf{R}_{G^\pi}^\top \mathbf{1} = \mathbf{R}_G^\top \mathbf{1}, \tag{105}$$

so the normalization terms in the patch- and root-conditioned operators are also unchanged.

Likewise,

$$\mathbf{Z}_{G^\pi}^\top \mathbf{P}_{G^\pi} = \mathbf{Z}_G^\top \mathbf{P}_G, \qquad \mathbf{Z}_{G^\pi}^\top \mathbf{1} = \mathbf{Z}_G^\top \mathbf{1}, \tag{106}$$

which proves invariance of the atom-conditioned operator.

The position-aware TU operator is a sum over aligned patch occurrences:

$$\sum_i \mathbf{q}_{i,v} \widetilde{\mathbf{z}}_i^\top. \tag{107}$$

A relabeling only changes the order of the summands, leaving the sum unchanged. Finally, graph-level outer-product coupling combines two independently permutation-invariant graph-level vectors and is therefore also invariant. □

Corollary B.2 (Invariance of the complete representation)**.** *If the statistical block $\mathbf{h}_S$, attribute block $\mathbf{h}_A$, structural block $\mathbf{h}_T$, and any selected coupling blocks satisfy the conditions above, then their concatenation is permutation invariant:*

$$\mathbf{h}_{\text{coupled}}(G) = \mathbf{h}_{\text{coupled}}(G^\pi). \tag{108}$$

Scope of the guarantee. The result above is conditional on rooted-isomorphism consistency of the local observation mapping and on the use of a fixed dictionary. Exact root-preserving canonicalization, including the unordered pair-rooted construction used for BREC, satisfies this condition by construction.

Typed-WL ordering is computationally cheaper but may leave symmetric nodes with identical refinement keys. If such ties are resolved using representation-dependent node indices, equation 93 is not guaranteed for those patches. In our implementation, remaining ties are resolved by exact canonicalization using the full patch connectivity while preserving root roles and the WL-induced partition, ensuring rooted-isomorphism consistency even when refinement keys do not uniquely determine the order. Empirical relabeling audits are reported with the experimental results.

The fixed-dictionary condition is also essential. Refitting a non-convex dictionary-learning objective after each relabeling can produce atom permutations or different local optima, even when the underlying observation multiset is unchanged. Our invariance statement concerns encoding and readout under a dictionary that has already been fitted and fixed according to the specified experimental protocol.

B.3 Relation to the 1-WL Test

We compare the exact node-rooted structural profile with the one-dimensional Weisfeiler–Lehman test (1-WL). The comparison in this section concerns the exact rooted-patch profile introduced in Appendix B.1. Its transfer to the learned sparse representation requires the information-preservation conditions stated below.

The 1-WL refinement. Let $\ell_0(v)$ denote the initial color of node v. In the topology-only setting, all nodes receive the same initial color. At refinement round $t \geq 1$, 1-WL updates the color of v according to

$$\ell_t(v) = \mathrm{HASH}\left(\ell_{t-1}(v), \{\{\ell_{t-1}(u) : u \in N_G(v)\}\}\right), \tag{109}$$

where HASH is injective over its arguments. The graph-level color multiset after t rounds is

$$\mathcal{C}_t^{\mathrm{WL}}(G) = \{\{\ell_t(v) : v \in V(G)\}\}. \tag{110}$$

Two graphs are distinguished by 1-WL if these multisets differ at any refinement round.

Initial node colors, when present, are regarded as part of the rooted structural observation. Accordingly, rooted isomorphisms in this section must preserve these colors.

Lemma 2 (A rooted induced patch determines the corresponding WL color)**.** *For every $t \geq 0$, there exists a function ψ_t such that*

$$\ell_t(v) = \psi_t(\tau_t(G, \{v\})), \tag{111}$$

where $\tau_t(G, \{v\})$ is the exact rooted-isomorphism type of the radius-t induced patch centered at v.

Proof. We proceed by induction over t.

For $t = 0$, the radius-zero rooted patch consists of the root together with its initial color. Therefore, $\ell_0(v)$ is determined by $\tau_0(G, \{v\})$.

Assume that the statement holds at round t. Consider the exact radius-$(t+1)$ rooted patch

$$(P_{t+1}(G, \{v\}), \{v\}).$$

This patch determines the root's radius-t patch and hence determines $\ell_t(v)$ by the induction hypothesis. It also identifies every neighbor $u \in N_G(v)$.

For each such neighbor u, every node at distance at most t from u lies at distance at most $t+1$ from v. Consequently, the induced radius-$(t+1)$ patch centered at v contains the complete radius-t induced patch centered at u. It therefore determines $\tau_t(G, \{u\})$, and hence $\ell_t(u)$, for every neighbor u.

Thus, the rooted radius-$(t+1)$ patch determines both

$$\ell_t(v) \quad \text{and} \quad \{\{\ell_t(u) : u \in N_G(v)\}\}.$$

By the update rule in equation 109, it determines $\ell_{t+1}(v)$. This establishes the induction step. □

Theorem B.3 (Round-matched dominance over 1-WL)**.** *For every $t \geq 0$, the exact radius-t node-rooted profile is at least as discriminative as t rounds of 1-WL. In particular,*

$$\Phi_t^{(1)}(G) = \Phi_t^{(1)}(H) \implies \mathcal{C}_t^{\mathrm{WL}}(G) = \mathcal{C}_t^{\mathrm{WL}}(H). \tag{112}$$

Equivalently,

$$\mathcal{C}_t^{\mathrm{WL}}(G) \neq \mathcal{C}_t^{\mathrm{WL}}(H) \implies \Phi_t^{(1)}(G) \neq \Phi_t^{(1)}(H). \tag{113}$$

Proof. Suppose that

$$\Phi_t^{(1)}(G) = \Phi_t^{(1)}(H).$$

By the definition of profile equality, G and H contain the same multiset of exact radius-t rooted types, including the same multiplicities. Applying the same function ψ_t from Lemma 2 to every element of these two multisets produces identical multisets:

$$\{\{\psi_t(\tau_t(G, \{v\})) : v \in V(G)\}\} = \{\{\psi_t(\tau_t(H, \{u\})) : u \in V(H)\}\}. \tag{114}$$

Using equation 111, this is precisely

$$\mathcal{C}_t^{\mathrm{WL}}(G) = \mathcal{C}_t^{\mathrm{WL}}(H).$$

The second implication follows by contraposition. □

Strict separation. The preceding theorem establishes that exact rooted patches retain all information used by the corresponding number of 1-WL refinement rounds. We next show that the converse does not hold.

Proposition 3 (A graph pair separated beyond 1-WL)**.** *There exist two non-isomorphic graphs that remain indistinguishable to 1-WL at every refinement round but have different exact radius-one node-rooted profiles.*

Proof. Let

$$G = C_6, \qquad H = C_3 \sqcup C_3, \tag{115}$$

where C_6 is a cycle on six nodes and $C_3 \sqcup C_3$ is the disjoint union of two triangles. The two graphs are non-isomorphic, but both contain six nodes and every node has degree two.

Under uniform initial colors, all nodes in both graphs have the same color at round zero. Suppose all nodes have the same color at round t. Every node then observes the same multiset consisting of two copies of that color. equation 109 therefore assigns the same new color to every node in both graphs. By induction,

$$\mathcal{C}_t^{\mathrm{WL}}(C_6) = \mathcal{C}_t^{\mathrm{WL}}(C_3 \sqcup C_3) \tag{116}$$

for every $t \geq 0$. Hence 1-WL cannot distinguish the two graphs.

Their radius-one rooted induced patches are nevertheless different. For every $v \in V(C_6)$, the closed one-hop neighborhood induces a path on three nodes:

$$P_1(C_6, \{v\}) \simeq_{\mathrm{r}} (P_3, \{\text{middle node}\}). \tag{117}$$

For every $u \in V(C_3 \sqcup C_3)$, the closed one-hop neighborhood induces a triangle:

$$P_1(C_3 \sqcup C_3, \{u\}) \simeq_{\mathrm{r}} (C_3, \{\text{root}\}). \tag{118}$$

Since P_3 and C_3 are non-isomorphic,

$$\tau_1(C_6, \{v\}) \neq \tau_1(C_3 \sqcup C_3, \{u\}) \tag{119}$$

for all choices of v and u. Therefore,

$$\Phi_1^{(1)}(C_6) \neq \Phi_1^{(1)}(C_3 \sqcup C_3). \tag{120}$$

Thus, the exact radius-one profile separates a graph pair that 1-WL cannot distinguish at any refinement round. □

Dominance over stabilized 1-WL. Theorem B.3 is a round-matched statement: a radius-t exact profile dominates t rounds of 1-WL. To compare with 1-WL run until stabilization, consider the multiscale exact profile

$$\Phi_{\leq T}^{(1)}(G) = \left(\Phi_0^{(1)}(G), \Phi_1^{(1)}(G), \ldots, \Phi_T^{(1)}(G)\right). \tag{121}$$

Corollary B.4 (Strict dominance over stabilized 1-WL)**.** *Let $\mathfrak{G}$ be a finite graph family, and let T be sufficiently large that 1-WL has stabilized on every pair of graphs in $\mathfrak{G}$. Then*

$$\text{1-WL} \prec \Phi_{\leq T}^{(1)} \qquad \textit{on } \mathfrak{G}, \tag{122}$$

provided that $\mathfrak{G}$ contains the pair C_6 and $C_3 \sqcup C_3$, or another pair satisfying Proposition 3.

Proof. If stabilized 1-WL distinguishes $G, H \in \mathfrak{G}$, then there exists some round $t \leq T$ such that

$$\mathcal{C}_t^{\mathrm{WL}}(G) \neq \mathcal{C}_t^{\mathrm{WL}}(H).$$

By Theorem B.3,

$$\Phi_t^{(1)}(G) \neq \Phi_t^{(1)}(H),$$

and hence

$$\Phi_{\leq T}^{(1)}(G) \neq \Phi_{\leq T}^{(1)}(H).$$

Therefore, the multiscale profile is at least as discriminative as stabilized 1-WL on $\mathfrak{G}$.

Proposition 3 supplies a pair that stabilized 1-WL fails to distinguish but whose exact radius-one profiles differ. Since $\Phi_1^{(1)}$ is a component of $\Phi_{\leq T}^{(1)}$, the dominance is strict. □

Transfer to the learned sparse representation. Let $\mathbf{h}_T(G)$ denote the learned sparse structural representation. On a graph family $\mathfrak{G}$, suppose that the complete practical pipeline preserves every distinction made by the selected exact profile:

$$\Phi^{(1)}_{\mathcal{H}}(G) \neq \Phi^{(1)}_{\mathcal{H}}(H) \quad \Longrightarrow \quad \mathbf{h}_T(G) \neq \mathbf{h}_T(H) \qquad \text{for all } G, H \in \mathfrak{G}. \tag{123}$$

Proposition 4 (Sparse representation under information preservation)**.** *Under equation 123, if $\mathcal{H}$ contains every radius required by Theorem B.3, then the learned sparse representation inherits the corresponding discriminative dominance over 1-WL. If the exact profile strictly separates a 1-WL-indistinguishable pair and this distinction is preserved, then the sparse representation is also strictly more discriminative on the considered graph family.*

Proof. If 1-WL distinguishes G and H, then Theorem B.3 implies that the selected exact profile distinguishes them. Equation 123 then gives $\mathbf{h}_T(G) \neq \mathbf{h}_T(H)$. If the exact profile additionally distinguishes a pair that 1-WL cannot, and this distinction is preserved by the sparse pipeline, strictness follows. □

Scope of the result. The strict theoretical conclusion applies directly to the exact multiscale profile. For a practical fixed radius r, the unconditional guarantee is the round-matched statement in Theorem B.3: the exact radius-r profile retains all distinctions available to the first r rounds of 1-WL and may also retain additional induced-edge information.

The learned K-SVD representation inherits this conclusion only under equation 123. Finite dictionary capacity, sparsity constraints, approximate reconstruction, and non-injective readout may merge distinct exact profiles. We therefore treat the theorem as an expressiveness result for the representation principle and evaluate the amount of discrimination retained after sparse compression empirically in the structural distinguishability experiments.

B.4 Pair-Rooted Profiles and Higher-Order WL

Node-rooted profiles describe the induced structure surrounding one distinguished node. Pair-rooted profiles additionally retain the joint environment of two roots, including their relative position, common neighborhood, and connections between the two rooted neighborhoods. We compare the exact adjacent-pair-rooted radius-2 profile with a folklore two-dimensional Weisfeiler–Lehman refinement (2-FWL) (Maron et al., 2019).

Terminology for higher-order WL variants is not uniform across the literature. We therefore state the refinement operator explicitly and restrict all claims in this section to the definition below.

Exact adjacent-pair-rooted profile. Let

$$\mathcal{R}^{(1)}_2(G) = \{\{u, v\} \subseteq V(G) : (u, v) \in E(G)\} \tag{124}$$

be the set of unordered adjacent root pairs. The exact radius-r pair-rooted profile is

$$\Phi^{(2,1)}_r(G) = \left\{\!\!\left\{ \tau_r(G, \{u, v\}) : \{u, v\} \in \mathcal{R}^{(1)}_2(G) \right\}\!\!\right\}, \tag{125}$$

where rooted isomorphisms may exchange the two roots. Thus, the profile is invariant both to graph relabeling and to reversing an edge used as the root.

The 2-FWL refinement. The 2-FWL procedure assigns colors to all ordered node pairs $(u, v) \in V(G)^2$. In the topology-only setting, the initial color records the atomic relation

$$c^G_0(u, v) = \begin{cases} D, & u = v, \\ E, & u \neq v \text{ and } (u, v) \in E(G), \\ N, & u \neq v \text{ and } (u, v) \notin E(G). \end{cases} \tag{126}$$

At round t, define the joint replacement multiset

$$\mathcal{M}^G_t(u, v) = \left\{\!\!\left\{ \left(c^G_t(u, w), c^G_t(w, v)\right) : w \in V(G) \right\}\!\!\right\}. \tag{127}$$

The next color is

$$c_{t+1}^G(u,v) = \text{HASH}\left(c_t^G(u,v), \mathcal{M}_t^G(u,v)\right), \tag{128}$$

where the hash is globally consistent and injective. The graph-level pair-color histogram is

$$\mathcal{H}_t^{2\text{-FWL}}(G) = \left\{\left\{c_t^G(u,v) : (u,v) \in V(G)^2\right\}\right\}. \tag{129}$$

The procedure distinguishes G and H if these histograms differ at some refinement round.

The pair-rooted profile does not dominate 2-FWL. A fixed-radius pair-rooted profile observes only bounded local neighborhoods, whereas 2-FWL can propagate pairwise information across repeated refinement rounds. The following construction makes this difference explicit.

Lemma 3 (2-FWL detects bounded pairwise reachability)**.** *For every $t \geq 0$, the color $c_t^G(u,v)$ determines whether*

$$d_G(u,v) \leq 2^t, \tag{130}$$

where nodes in different connected components have infinite distance.

Proof. At $t = 0$, the initial color distinguishes $u = v$, adjacent pairs, and non-adjacent pairs. It therefore determines whether $d_G(u,v) \leq 1 = 2^0$.

Suppose the statement holds at round t. By the triangle inequality,

$$d_G(u,v) \leq 2^{t+1} \tag{131}$$

if and only if there exists a node w such that

$$d_G(u,w) \leq 2^t \quad \text{and} \quad d_G(w,v) \leq 2^t. \tag{132}$$

By the induction hypothesis, both predicates are determined by $c_t^G(u,w)$ and $c_t^G(w,v)$. Their simultaneous occurrence is therefore determined by the joint multiset $\mathcal{M}_t^G(u,v)$. Since this multiset is included in the injective update in equation 128, the resulting color $c_{t+1}^G(u,v)$ determines whether the required node w exists. □

Proposition 5 (A pair separated by 2-FWL but not by the local pair profile)**.** *There exist non-isomorphic graphs G, H such that*

$$\Phi_2^{(2,1)}(G) = \Phi_2^{(2,1)}(H), \tag{133}$$

whereas 2-FWL distinguishes G and H.

Proof. Let

$$G = C_{16}, \qquad H = C_8 \sqcup C_8. \tag{134}$$

Both graphs have 16 nodes and 16 edges. For every adjacent root pair in either graph, the radius-2 node set contains six consecutive cycle vertices. Because both cycle lengths exceed six, the induced patch is a path on six nodes, with the unordered root pair forming its central edge. Hence every rooted patch in both graphs has the same exact rooted type. Since both graphs contain 16 adjacent root pairs,

$$\Phi_2^{(2,1)}(C_{16}) = \Phi_2^{(2,1)}(C_8 \sqcup C_8). \tag{135}$$

The graph C_{16} is connected and has diameter 8. Therefore, after $t = 3$ rounds, every ordered node pair in C_{16} satisfies

$$d_{C_{16}}(u,v) \leq 2^3.$$

In $C_8 \sqcup C_8$, ordered pairs whose endpoints belong to different components have infinite distance. By Lemma 3, their round-three colors encode a different reachability predicate from every pair color occurring in C_{16}. Consequently,

$$\mathcal{H}_3^{2\text{-FWL}}(C_{16}) \neq \mathcal{H}_3^{2\text{-FWL}}(C_8 \sqcup C_8), \tag{136}$$

and 2-FWL distinguishes the two graphs. □

2-FWL does not dominate the pair-rooted profile. The converse separation follows from strongly regular graphs. Recall that a strongly regular graph with parameters $\mathrm{srg}(n, d, \lambda, \mu)$ has n nodes, is d-regular, and every adjacent or non-adjacent pair has respectively λ or μ common neighbors.

Lemma 4 (2-FWL on strongly regular graphs)**.** *Two strongly regular graphs with the same parameters (n, d, λ, μ) are indistinguishable by the 2-FWL refinement in Eqs. equation 126–equation 129.*

Proof. We show by induction that, at every round t, the color of an ordered pair (x, y) depends only on whether $x = y$, $x \sim y$, or $x \neq y$ and $x \not\sim y$. The statement holds at $t = 0$ by equation 126. Suppose it holds at round t, and denote the three colors by D_t, E_t, N_t.

If $x = y$, the nonzero multiplicities in $\mathcal{M}_t^G(x, x)$ are

$$(D_t, D_t) : 1, \qquad (E_t, E_t) : d, \qquad (N_t, N_t) : n - d - 1. \tag{137}$$

If $x \sim y$, the nonzero multiplicities are

$$\begin{aligned} &(D_t, E_t) : 1, && (E_t, D_t) : 1, && (E_t, E_t) : \lambda, \\ &(E_t, N_t) : d - 1 - \lambda, && (N_t, E_t) : d - 1 - \lambda, && (N_t, N_t) : n - 2d + \lambda. \end{aligned} \tag{138}$$

If $x \neq y$ and $x \not\sim y$, they are

$$\begin{aligned} &(D_t, N_t) : 1, && (N_t, D_t) : 1, && (E_t, E_t) : \mu, \\ &(E_t, N_t) : d - \mu, && (N_t, E_t) : d - \mu, && (N_t, N_t) : n - 2 - 2d + \mu. \end{aligned} \tag{139}$$

Thus, the complete update input depends only on the atomic pair relation and the four strongly regular parameters. The same remains true at round $t + 1$.

Finally, the numbers of ordered diagonal, edge, and non-edge pairs are

$$n, \qquad nd, \qquad n(n - d - 1), \tag{140}$$

respectively. Hence any two strongly regular graphs with the same parameters have identical pair-color histograms at every round. □

Proposition 6 (A pair separated by the exact pair profile but not by 2-FWL)**.** *There exist non-isomorphic graphs R, S such that 2-FWL cannot distinguish them, while*

$$\Phi_2^{(2,1)}(R) \neq \Phi_2^{(2,1)}(S). \tag{141}$$

Proof. Let R be the 4×4 Rook graph and let S be the Shrikhande graph. Both are strongly regular with parameters

$$(n, d, \lambda, \mu) = (16, 6, 2, 2). \tag{142}$$

By Lemma 4, 2-FWL cannot distinguish them.

Both graphs are connected, non-complete strongly regular graphs with $\mu > 0$, and therefore have diameter two. Consequently, for every adjacent root pair R_0 in either graph,

$$P_2(R, R_0) = R, \qquad P_2(S, R_0) = S. \tag{143}$$

If a pair-rooted patch from the Rook graph were rooted-isomorphic to a pair-rooted patch from the Shrikhande graph, forgetting the root markers would give an isomorphism between the two underlying graphs. This contradicts their non-isomorphism. Hence their radius-2 adjacent-pair-rooted profiles are different:

$$\Phi_2^{(2,1)}(R) \neq \Phi_2^{(2,1)}(S). \tag{144}$$

□

Theorem B.5 (Incomparability with 2-FWL)**.** *The exact radius-2 adjacent-pair-rooted profile $\Phi_2^{(2,1)}$ and the 2-FWL refinement defined above are incomparable in graph discriminative power.*

Proof. Proposition 5 provides a graph pair that 2-FWL distinguishes but $\Phi_2^{(2,1)}$ does not. Therefore,

$$\Phi_2^{(2,1)} \not\geq \text{2-FWL}.$$

Proposition 6 provides a graph pair that $\Phi_2^{(2,1)}$ distinguishes but 2-FWL does not. Therefore,

$$\text{2-FWL} \not\geq \Phi_2^{(2,1)}.$$

Neither representation dominates the other. □

Interpretation and scope. The incomparability reflects two different structural mechanisms. Repeated 2-FWL refinement propagates pairwise relations beyond any fixed local radius and can therefore recover global properties such as bounded reachability. The exact pair-rooted profile instead preserves the complete induced connectivity inside each selected neighborhood. It can retain local or pair-centered structural distinctions that the color refinement compresses into the same pair relation.

The Rook–Shrikhande separation above is a statement about the explicitly defined 2-FWL operator. Because higher-order WL terminology and dimensional indexing vary across the literature, it should not be restated as a claim about "3-WL" without first specifying the adopted WL convention and the required equivalence result.

Finally, the theorem concerns the exact pair-rooted profile. The compressed K-SVD representation inherits a particular separation only when its sparse encoder and readout preserve the corresponding exact-profile difference. The BREC experiments evaluate this preservation empirically.

B.5 From Exact Profiles to Sparse Representations

The preceding expressiveness results concern exact rooted-patch profiles. The practical representation replaces the discrete multiset of rooted isomorphism types with a finite-dimensional sparse approximation. We now state the conditions under which distinctions made by an exact profile are preserved by this approximation and clarify which properties are not guaranteed unconditionally.

The practical compression map. Assume that the rooted structural mapping ϕ_r is an exact realization over the considered patch family, as defined in equation 78. For a graph G, its exact vector profile is

$$\Phi^{\mathcal{R}}_{\phi,r}(G) = \{\{\mathbf{y}_{G,R} : R \in \mathcal{R}(G)\}\}. \tag{145}$$

A fixed dictionary $\mathbf{D}$ and sparse encoder $\mathcal{E}_{\mathbf{D}}$ map each patch observation to

$$\mathbf{c}_{G,R} = \mathcal{E}_{\mathbf{D}}(\mathbf{y}_{G,R}), \tag{146}$$

producing the code multiset

$$\mathcal{C}_{\mathbf{D}}(G) = \{\{\mathbf{c}_{G,R} : R \in \mathcal{R}(G)\}\}. \tag{147}$$

A symmetric readout ρ then constructs the finite-dimensional representation

$$\mathbf{h}_T(G) = \rho(\mathcal{C}_{\mathbf{D}}(G), \mathcal{E}_{\mathrm{rec}}(G)), \tag{148}$$

where $\mathcal{E}_{\mathrm{rec}}(G)$ denotes the multiset of patch reconstruction errors when these are included in the readout. The complete compression pipeline can therefore be written as

$$\Phi^{\mathcal{R}}_{r}(G) \xrightarrow{\phi_r} \Phi^{\mathcal{R}}_{\phi,r}(G) \xrightarrow{\mathcal{E}_{\mathbf{D}}} \mathcal{C}_{\mathbf{D}}(G) \xrightarrow{\rho} \mathbf{h}_T(G). \tag{149}$$

Profile-faithful compression. Let $\mathfrak{G}$ be a graph family. We call the complete sparse pipeline profile-faithful on $\mathfrak{G}$ if

$$\Phi^{\mathcal{R}}_{r}(G) \neq \Phi^{\mathcal{R}}_{r}(H) \quad \Longrightarrow \quad \mathbf{h}_T(G) \neq \mathbf{h}_T(H) \tag{150}$$

for every $G, H \in \mathfrak{G}$. This is a property of the complete combination of structural vectorization, dictionary, sparse encoder, and readout, rather than of K-SVD alone.

Proposition 7 (Preservation under injective compression)**.** *Let $\mathfrak{G}$ be a graph family. Suppose that:*

1. *ϕ_r is injective over the rooted-isomorphism types occurring in $\mathfrak{G}$;*
2. *the fixed sparse encoder $\mathcal{E}_{\mathbf{D}}$ is injective over the corresponding structural observations; and*
3. *the readout ρ is injective over the sparse-code multisets generated by graphs in $\mathfrak{G}$.*

Then the sparse representation is profile-faithful on $\mathfrak{G}$. Consequently, it preserves every graph distinction made by the exact rooted-patch profile on this family.

Proof. Let $G, H \in \mathfrak{G}$ satisfy

$$\Phi_r^{\mathcal{R}}(G) \neq \Phi_r^{\mathcal{R}}(H).$$

Injectivity of ϕ_r implies that their exact vector profiles differ:

$$\Phi_{\phi,r}^{\mathcal{R}}(G) \neq \Phi_{\phi,r}^{\mathcal{R}}(H).$$

Injectivity of the fixed sparse encoder over the relevant observation set then gives

$$\mathcal{C}_{\mathbf{D}}(G) \neq \mathcal{C}_{\mathbf{D}}(H).$$

Finally, injectivity of ρ over the resulting code multisets implies

$$\mathbf{h}_T(G) \neq \mathbf{h}_T(H).$$

Thus, the complete sparse pipeline preserves the exact-profile distinction. □

A reconstruction-based sufficient condition. Although sparse coding is generally not guaranteed to be injective, bounded reconstruction error provides a useful sufficient condition over a finite patch collection. Let

$$\delta_{\mathfrak{P}} = \min_{\substack{\mathbf{y},\mathbf{y}' \in \mathfrak{P} \\ \mathbf{y} \neq \mathbf{y}'}} \|\mathbf{y} - \mathbf{y}'\|_2 \tag{151}$$

be the minimum distance between distinct exact patch vectors in a finite set $\mathfrak{P}$.

Proposition 8 (Patch separation under bounded reconstruction)**.** *Suppose that every $\mathbf{y} \in \mathfrak{P}$ is encoded as $\mathbf{c} = \mathcal{E}_{\mathbf{D}}(\mathbf{y})$ and satisfies*

$$\|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2 < \frac{\delta_{\mathfrak{P}}}{2}. \tag{152}$$

Then two distinct observations in $\mathfrak{P}$ cannot receive the same sparse code.

Proof. Assume for contradiction that distinct observations $\mathbf{y}, \mathbf{y}' \in \mathfrak{P}$ receive the same code $\mathbf{c}$. By the triangle inequality,

$$\|\mathbf{y} - \mathbf{y}'\|_2 \leq \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2 + \|\mathbf{y}' - \mathbf{D}\mathbf{c}\|_2 \tag{153}$$

$$< \frac{\delta_{\mathfrak{P}}}{2} + \frac{\delta_{\mathfrak{P}}}{2} = \delta_{\mathfrak{P}}, \tag{154}$$

contradicting the definition of $\delta_{\mathfrak{P}}$. □

This condition is sufficient but not necessary: distinct observations may receive distinct codes even when their reconstruction errors exceed the bound. Moreover, patch-level separation alone does not ensure graph-level separation if the subsequent readout maps different code multisets to the same vector.

Information loss in practical readouts. The readouts used in our experiments aggregate atom-wise moments, activation frequencies, extrema, and reconstruction statistics. These quantities are permutation invariant and preserve substantially more information than a single mean activation, but a finite collection of moments is not generally injective over arbitrary multisets. Two different sparse-code multisets may therefore share the same pooled statistics.

Likewise, finite dictionary capacity and a restrictive sparsity level may cause distinct structural observations to be represented by similar sparse codes. The degree of information preservation depends on the dictionary size K, sparsity s, observation scaling, optimization, and the selected readout. Accordingly, Proposition 7 is a conditional transfer result rather than an unconditional claim about every K-SVD configuration.

Why a shared fixed dictionary is required. Comparability across graphs requires all patches to be represented in the same atom coordinate system. If separate dictionaries were learned independently for different graphs, their atom indices could differ by permutation, sign, or more general changes of basis, and equal code coordinates would not have a common structural meaning. The shared dictionary and the fixed sign and atom-order conventions described in Appendix A.3 establish a common coordinate system:

$$\mathbf{y}_{G,R} \longmapsto \mathbf{c}_{G,R} \in \mathbb{R}^K \tag{155}$$

with the same atom interpretation for every encoded graph.

Connection to the expressiveness results. The exact-profile results establish which structural distinctions are available before compression. The learned sparse representation measures how much of this information can be retained in a compact shared coordinate system. In particular, the 1-WL and 2-FWL comparisons should be interpreted as follows:

$$\text{exact structural distinction} \quad + \quad \text{profile-faithful compression} \quad \Longrightarrow \quad \text{sparse-representation distinction.} \tag{156}$$

Without the profile-faithfulness condition, an exact-profile theorem alone does not guarantee that a particular finite K-SVD model will preserve the corresponding graph pair.

We therefore evaluate this transfer empirically by comparing exact profiles with their learned sparse representations, varying the dictionary capacity and sparsity, and testing whether distinctions survive compression and node relabeling. These experiments quantify the practical information retained by the sparse representation rather than assuming perfect preservation.

C Complete Structural Distinguishability Results

C.1 BREC Evaluation Protocol

We evaluate structural distinguishability on BREC, which contains 400 pairs of non-isomorphic graphs divided into six categories: Basic, Regular, Extension, CFI, 4-Vertex-Condition, and Distance-Regular. Because our representation is a deterministic non-neural descriptor, we follow the deterministic evaluation path of BREC: representations of a non-isomorphic pair should differ, whereas representations of relabeled copies of the same graph should remain identical.

For every graph, we construct radius-2 induced patches rooted at unordered adjacent node pairs. Each patch is represented by its root-preserving canonical adjacency, root-role indicators, and valid-node mask. Exchanging the two roots is included in the canonicalization. No node attributes, predefined motifs, graphlet counts, or supervised labels are used. The complete collection contains $114{,}752$ patch occurrences and $21{,}718$ distinct canonical patch types. After padding to the maximum observed patch size of 64 nodes, each patch is represented by a $2{,}208$-dimensional vector.

The exact representation of a graph is the multiset of its canonical patch keys. For the compressed representation, a single frequency-weighted K-SVD dictionary is fitted to the unlabeled patch frequencies of all evaluated BREC graphs. This transductive protocol is appropriate here because BREC evaluates pair distinguishability rather than out-of-sample prediction. Once fitted, the dictionary and atom ordering are fixed for every graph and every relabeling.

Given the sparse code $\boldsymbol{\alpha}_P \in \mathbb{R}^K$ of patch P, we use the signed-mean readout

$$\mathbf{h}_{\mathrm{mean}}(G) = \frac{1}{|\mathcal{P}(G)|} \sum_{P \in \mathcal{P}(G)} \boldsymbol{\alpha}_P. \tag{157}$$

A compressed pair (G, H) is considered robustly separated when

$$\|\mathbf{h}_{\mathrm{mean}}(G) - \mathbf{h}_{\mathrm{mean}}(H)\|_\infty > 10^{-2}. \tag{158}$$

The threshold in equation 158 is our numerical robustness criterion, rather than an official BREC threshold. The exact-profile result is determined by exact multiset inequality and does not use this tolerance.

C.2 Exact and Compressed Pair Separation

Table 5 reports the complete category-wise results. The exact pair-rooted profile separates 300 of the 400 pairs, including all pairs outside the CFI category. The 100 CFI pairs have identical radius-2 pair-rooted patch multisets and therefore form an explicit ceiling for the selected local observation scope.

With $K = 128$ atoms and sparsity $s = 16$, the learned representation retains 299.0 ± 0.8 of these 300 distinctions using only the signed-mean sparse-code readout. The matched Gaussian dictionaries separate only 252.0 ± 2.4 pairs. The difference is especially pronounced on the Regular and Distance-Regular categories, indicating that the learned atoms organize the patch space more effectively than arbitrary projections.The matched signed-mean comparison in Table 5 provides the more restrictive test.

Table 5: Category-wise pair separation on BREC. Results are numbers of separated pairs; compressed results are mean ± population standard deviation over three seeds.

Category	Exact profile	Learned K-SVD ($K = 128, s = 16$)	Gaussian dictionary ($K = 128, s = 16$)
Basic (60)	60	60.0 ± 0.0	58.0 ± 0.8
Regular (100)	100	99.3 ± 0.5	70.0 ± 0.0
Extension (100)	100	100.0 ± 0.0	94.3 ± 2.6
CFI (100)	0	0.0 ± 0.0	0.0 ± 0.0
4-VC (20)	20	20.0 ± 0.0	18.3 ± 0.5
Distance-Regular (20)	20	19.7 ± 0.5	11.3 ± 0.5
Overall (400)	**300**	$\mathbf{299.0 \pm 0.8}$	252.0 ± 2.4

Table 6: BREC separation as a function of dictionary capacity. The three entries list the results for K-SVD seeds $\{7, 19, 2027\}$.

Atoms K	Sparsity s	Separated pairs by seed	Mean ± std.
16	4	176, 144, 200	173.3 ± 22.9
32	4	261, 263, 259	261.0 ± 1.6
64	8	278, 286, 286	283.3 ± 3.8
128	16	300, 298, 299	$\mathbf{299.0 \pm 0.8}$

C.3 Effect of Dictionary Capacity

Table 6 evaluates the amount of exact-profile distinguishability retained under different dictionary capacities. Increasing the dictionary from 16 to 128 atoms consistently improves pair separation. At $K = 128$, the compressed representation retains approximately 99.7% of the 300 distinctions available to the exact profile.

These results also clarify the role of sparse decomposition. The exact profile defines the structural information made available by the chosen patches, whereas finite K-SVD determines how much of this information remains accessible in a compact shared coordinate system. Increasing K reduces compression loss but cannot distinguish the CFI pairs because their exact input profiles are already identical.

C.4 Relabeling and Reliability Audits

We additionally evaluate the complete representation pipeline under the relabeling and reliability structure of the official BREC data layout. These checks cover patch extraction, local canonicalization, mapping to fixed dictionary atoms, and graph-level readout, rather than testing the individual components in isolation.

The CFI representative remains invariant but indistinguishable, which is the expected outcome rather than a reliability failure. These audits establish invariance for every relabeling configuration evaluated here. We do not claim that the full 400-pair benchmark was exhaustively replayed under all 32 relabelings; the precise evaluated scope is reported in Table 7. Complete per-pair outputs and the scripts for extending the replay to all available relabelings are included in the supplementary code.

D Experimental Setup and Complete Downstream Results

This appendix provides the complete experimental settings and downstream prediction results supporting Section 5.3. We organize the results according to how structural and semantic information is used: prediction from topology-only inputs, additive integration of independently constructed structural and semantic channels, and explicit structure–semantic coupling. Complete coalition-based information audits are reported separately in Appendix E, while the occurrence-level molecular analyses are provided in Appendix F.

Table 7: Scope and outcome of the evaluated BREC invariance and reliability checks. A reliability failure means that relabeled copies of the same graph produced different representations.

Audit	Evaluated scope	Outcome
Complete pair and reliability evaluation	All 400 non-isomorphic pairs, together with the corresponding evaluated reliability copies	Exact: $300/400$ separated; K-SVD: $299/400$; zero reliability failures
Full relabeling audit by category	One representative pair from each of the six BREC categories; all 32 A/B relabelings and all 64 associated reliability copies	Zero exact-profile, K-SVD, or reliability invariance failures
Fixed-dictionary replay	All 800 graphs in the non-isomorphic-pair block, comparing relabeling 1 with the relabeling-0 reference under a fixed shared dictionary	$800/800$ readouts matched exactly; maximum absolute difference 0

Table 8: Downstream benchmark groups and evaluation protocols. "Nested 10-fold" denotes an outer ten-fold test assessment with an internal training/validation split in every outer fold. Molecular and peptide datasets use their prescribed scaffold or official partitions.

Benchmark group	Datasets	Prediction	Metric	Evaluation protocol
Airport networks	Brazil, Europe, USA	Node	Accuracy	$80/10/10$, repeated splits
Topology-only TU	IMDB-B, IMDB-M, COLLAB, REDDIT-B	Graph	Accuracy	Nested 10-fold
Attributed TU	MUTAG, NCI1, PROTEINS, DD, ENZYMES	Graph	Accuracy	Nested 10-fold
Molecular	MolHIV, MolBACE, MolTox21	Graph	ROC–AUC	Prescribed split, 10 seeds
Peptide	Peptides-func	Graph	Average precision	Official split, 10 seeds

Unless otherwise stated, the downstream predictor is XGBoost. The use of a conventional predictor is intentional: the experiments evaluate how much task-relevant information is retained by the proposed representation without requiring its dictionary and sparse codes to be optimized end-to-end with a neural prediction architecture. All model selection is conducted using training and validation data only. Test partitions are reserved for final evaluation and are not used to select the patch construction, dictionary configuration, feature combination, coupling operator, or predictor hyperparameters.

D.1 Datasets, Evaluation Protocols, and Model Selection

Datasets and tasks. Our downstream evaluation covers both node- and graph-level prediction under several complementary regimes. Brazil, Europe, and USA airport networks evaluate topology-driven node classification. IMDB-BINARY, IMDB-MULTI, COLLAB, and REDDIT-BINARY evaluate graph classification without supplied node attributes. MUTAG, NCI1, PROTEINS, DD, and ENZYMES evaluate graph classification when both topology and node or edge information are available. The molecular and peptide experiments include MolHIV, MolBACE, MolTox21, and Peptides-func under their prescribed train/validation/test partitions. Dataset families, prediction levels, metrics, interactions methods and evaluation protocols are summarized in Table 8, Table 9 and Table 10.

Information channels. Throughout the experiments, we distinguish three independently accessible feature blocks:

$$S = \text{graph statistics}, \qquad T = \text{explicit structural coordinates}, \qquad A = \text{attributes}.$$

The S block contains global or distributional graph statistics, whereas T is obtained by sparse coding rooted structural observations over a shared dictionary followed by invariant readout. The A

Table 9: Configurations corresponding to the headline results reported in Tables 2. Here, q, K, s, and I denote patch capacity, dictionary size, OMP sparsity, and K-SVD iterations, respectively. A dash means that the reported configuration does not use a structural dictionary. B_{role}, B_{mass}, and B_{root} denote the coupling operators defined in Appendix A.5.

Dataset	Structural observation	q	(K, s, I)	Reported view	Coupling	Dim.	Result (%)
Airport node classification							
Brazil	node-centered local samples	4–32	$(4, 4, 10)$	S	none	1	79.29 ± 10.33
Europe	node-centered local samples	4–32	$(8, 4, 10)$	ST	none	81	57.57 ± 6.49
USA	node-centered $R1/R2$ samples	8–64	$(16, 4, 10)$ per radius	ST	none	321	66.55 ± 3.39
Topology-only TU graph classification							
IMDB-B	exact canonical $R1/R2$; buckets $\{16, 32, 64, 128, 256\}$	bucketed	$(8, 3, 2)$ per bucket	ST	none	617	73.87 ± 0.66
IMDB-M	exact canonical $R1/R2$; buckets $\{16, 32, 64, 128, 256\}$	bucketed	$(8, 3, 2)$ per bucket	ST	none	617	50.97 ± 0.63
COLLAB	exact canonical $R1/R2$; buckets $\{64, 128, 256, 384, 512\}$	bucketed	$(8, 3, 2)$ per bucket	ST	none	617	80.72 ± 0.30
REDDIT-B	exact canonical $R1/R2$; buckets $\{64, 128, 256, 512, 4096\}$	bucketed	$(8, 3, 2)$ per bucket	ST	none	617	91.14 ± 0.33
Attributed TU graph classification							
MUTAG	exact canonical node-rooted $R1/R2$	16	$(16, 4, 5)$	$STA + B_{\mathrm{role}}$	role-resolved	979	89.20 ± 1.11
NCI1	exact canonical node-rooted $R1/R2$	16	$(16, 4, 5)$	$STA + B_{\mathrm{role}}$	role-resolved	3355	83.09 ± 0.28
PROTEINS	exact canonical node-rooted $R1/R2$	57	$(16, 4, 5)$	SA	none	33	77.85 ± 0.71
DD	exact canonical node-rooted $R1/R2$	98	$(16, 4, 5)$	$STA + B_{\mathrm{role}}$	role-resolved	7723	79.48 ± 0.51
ENZYMES	exact canonical node-rooted $R1/R2$	26	$(16, 4, 5)$	SA	none	101	69.73 ± 0.87
Molecular and long-range graph prediction							
MolHIV	typed node-rooted $R2$	13	$(64, 8, 5)$	`nora`	five context-mass blocks	1018	81.47 ± 0.60
MolBACE	not used by the reported base view	–	–	69-D base	none	69	81.67 ± 0.44
MolTox21	exact binary-topology $R1/R2$	12	$(32, 6, 5)$	base + structural + all-cross	mass, conditional, and root	4838	76.67 ± 0.27
Peptides-func	structurally ordered node-rooted $R2$	16	$(32, 4, 3)$	$STA + B_{\mathrm{mass}} + B_{\mathrm{root}}$	mass-root	3435	71.57 ± 0.13

Table 10: XGBoost configurations used for the headline results. TU configurations are selected independently in each outer fold from the reported eight-candidate grid.

Datasets	Trees	Depth	η	Child wt.	Subsample	Colsample	λ_1	λ_2	Additional settings
TU datasets	300	$\{2, 4\}$	$\{0.03, 0.08\}$	$\{1, 4\}$	0.85	0.85	0.01	2	inner-validation selection
Brazil	500	3	0.05	default	0.80	0.80	0.10	1	early stopping 20
Europe	77	7	0.01026	default	0.7216	0.8622	0.00105	0.00251	early stopping 20
USA	500	5	0.05	default	0.80	0.70	0.50	1	early stopping 20
MolHIV	900	8	0.02521	8	0.8986	0.7715	0.03850	0.4269	$\gamma = 3.56\times10^{-5}$, bin 128
MolBACE (base)	400	4	0.03	5	0.80	0.80	0	10	$\gamma = 0$, bin 256
MolTox21 / Peptides-func	600	6	0.05	5	0.80	0.90	0	3	$\gamma = 0$, bin 256

block summarizes the supplied node and edge attributes without incorporating the learned structural dictionary. Additive combinations are formed by concatenation; for example,

$$ST = [S\|T], \qquad TA = [T\|A], \qquad STA = [S\|T\|A].$$

Explicit coupling features additionally retain the correspondence between semantic conditions and the structural atoms activated at individual patch occurrences. Their construction is described in Appendix A.5. On datasets without supplied attributes, only S, T, and ST are evaluated.

D.2 Implementation and Reproducibility Details

Nested evaluation on TU benchmarks. TU experiments use stratified outer ten-fold model assessment. For each outer fold, 10% of the graphs or labeled instances are held out as test data. The remaining 90% form the outer training pool, of which 85% is used for inner training and 15% for validation. Equivalently, approximately 76.5%, 13.5%, and 10% of the complete dataset are used for inner training, validation, and outer testing in each fold.

The shared dictionary used during model selection is fitted exclusively to patches extracted from the inner-training partition. Patch construction, dictionary size, sparsity, coupling configuration, and XGBoost hyperparameters are selected using the corresponding inner-validation partition. After selection, the dictionary and predictor are refitted using the complete outer training pool, and the held-out outer fold is evaluated once. Consequently, neither test graphs nor their patch frequencies participate in dictionary learning or model selection.

Unless explicitly noted in an individual result table, the complete protocol is repeated with the ten seeds

$$\{0, 42, 123, 1024, 2026, 777, 3407, 999, 111, 888\}.$$

For every seed, results are first averaged over its ten outer test folds. We then report the mean and population standard deviation of these seed-level means. Any dataset for which fewer repetitions are available is marked explicitly rather than pooled with the complete ten-seed results.

GraphMAE. We follow the original GraphMAE setup (Hou et al., 2022) for architecture and masking, with one protocol difference: whereas the original implementation pretrains on all graphs in the dataset (including held-out test graphs), we restrict pretraining strictly to the training portion of each fold, so that no validation or test graph is seen in any form during pretraining. All downstream fine-tuning and evaluation then follow the same split protocol as our method.

Airport node classification. The airport networks are evaluated in the transductive node-classification setting used by the corresponding structural-learning baselines. For each random seed, labeled nodes are divided into 80% training, 10% validation, and 10% test subsets. The graph topology is observable for all nodes, but validation and test labels are never used for fitting the predictor or selecting its configuration. Because the representation is unsupervised with respect to node labels, its structural dictionary may use the observed topology of the single graph, consistently with the transductive protocol. The primary airport results are averaged over 20 prespecified random splits; any supplementary run using a different number of repetitions is identified in its table caption.

Molecular and peptide benchmarks. MolHIV, MolBACE, MolTox21, and Peptides-func use their prescribed train/validation/test partitions without resampling. Dictionaries, normalization statistics, and any data-dependent feature construction are fitted using the official training partition only and are then frozen for validation and test encoding. Predictor and representation configurations are chosen according to validation performance. Once selected, the configuration is locked and evaluated on test without test-based retuning.

MolHIV and MolBACE are single-task binary problems evaluated by ROC–AUC. MolTox21 is a multi-task binary benchmark; ROC–AUC is computed separately for each task with both classes present and then averaged across valid tasks. Peptides-func is evaluated by mean average precision over its ten functional labels. For these datasets, reported means and standard deviations are computed over seeds $0, \ldots, 9$. Missing molecular labels are excluded task-wise from both model fitting and metric computation.

Model and representation selection. For every benchmark, comparisons among S, T, A, their additive combinations, and explicit coupling variants use identical data partitions and evaluation metrics. A result described as the "selected" or "best" configuration always refers to validation-based selection; it does not denote the largest test value chosen retrospectively. Complete candidate results are reported in the following subsections so that the selected configuration can be compared with the alternatives considered during development.

Published baseline values are reported only when their dataset processing, split, and metric are sufficiently compatible with ours, and are identified as published results in the corresponding captions. Baselines reproduced under our splits are labeled separately. We do not combine uncertainty estimates from incompatible protocols: published standard deviations retain the definitions used by their original sources, whereas ours follow the aggregation rules specified above. Exploratory or test-aware diagnostic experiments, when included for completeness, are explicitly marked and are not used as headline downstream results.

D.3 Prediction from Topology-Only Inputs

We first evaluate whether the proposed structural coordinates retain task-relevant information when no node or edge attributes are available. In this setting, both S and T are derived exclusively from topology, but they describe different levels of structural information. The S block contains 17 global or distributional graph statistics, whereas T represents the activations of canonically constructed rooted patches over dictionaries shared across the training graphs. Their concatenation, $ST = [S \| T]$, tests whether coarse graph statistics and the learned structural coordinates provide complementary information. No semantic attribute or explicit structure–semantic coupling feature is used in this subsection.

Topology-only node classification on airport networks. We additionally evaluate topology-driven node classification on the Brazil, Europe, and USA airport networks. Following the transductive protocol used by the corresponding structural-learning baselines, the complete graph topology is observable, while node labels are divided into $80\%/10\%/10\%$ training, validation, and test sub-

Table 11: Node classification accuracy (%) on airport networks. Our Brazil, Europe and USA results are averaged over 20 stratified $80/10/10$ splits, values are mean $\pm$ population standard deviation. Published baseline values are taken from Kong et al. (2022) and Zhang et al. (2023) and retain the uncertainty definitions of their original sources. For USA, T denotes the concatenated radius-1 and radius-2 structural representation.

Method	Brazil	Europe	USA
GCN	64.55 ± 4.18	52.07 ± 2.79	56.58 ± 1.11
Struc2vec	70.88 ± 4.26	57.94 ± 4.01	61.92 ± 2.61
DE-GNN	75.37 ± 3.25	58.41 ± 3.20	64.16 ± 1.70
Graphormer-GD	77.69 ± 6.39	$\mathbf{59.23 \pm 4.05}$	–
Ours (S)	$\mathbf{79.29 \pm 10.33}$	55.88 ± 6.32	55.57 ± 3.74
Ours (T)	78.93 ± 6.58	54.93 ± 6.56	65.98 ± 3.47
Ours (ST)	78.21 ± 7.98	57.57 ± 6.49	$\mathbf{66.55 \pm 3.39}$

Table 12: Graph classification accuracy (%) on topology-only TU datasets. Degree and GIN results are the rigorously reassessed results of Errica et al. (2020); the remaining baseline results are obtained from our reproductions. Bold and underlined entries indicate the best and second-best results, respectively.

Method	IMDB-B	IMDB-M	COLLAB	REDDIT-B
Degree	70.80 ± 5.00	49.10 ± 3.50	70.20 ± 1.50	82.20 ± 3.00
WL subtree	72.27 ± 0.21	50.90 ± 0.09	78.26 ± 0.05	76.33 ± 0.13
GIN	71.20 ± 3.90	48.50 ± 3.30	75.60 ± 2.30	$\underline{89.90 \pm 1.90}$
GraphMAE	70.80 ± 5.15	$\mathbf{52.13 \pm 2.71}$	68.16 ± 3.74	86.00 ± 2.14
GNN-AK+	72.90 ± 5.70	50.90 ± 3.60	–	–
GSN-v	$\mathbf{74.51 \pm 0.74}$	50.84 ± 0.43	$\mathbf{81.63 \pm 0.42}$	79.45 ± 1.10
Ours (S+T)	$\underline{73.87 \pm 0.66}$	$\underline{50.97 \pm 0.63}$	$\underline{80.72 \pm 0.30}$	$\mathbf{91.14 \pm 0.33}$

sets. The S representation contains node degree, whereas T is obtained from K-SVD decomposition of node-centered local neighborhoods. For USA, the selected structural representation concatenates radius-1 and radius-2 views; ST further appends node degree. For these three node classification tasks, each node-centered induced subgraph is treated as a graph sample and encoded using a dictionary shared across target nodes.

The three networks exhibit different dependencies on structural information. On Brazil, degree alone is highly predictive and slightly exceeds the learned structural and fused representations, indicating that the labels are largely aligned with a simple centrality signal. On Europe, combining degree with the structural representation improves upon either of our individual channels, although Graphormer-GD remains strongest among the reported methods. The USA network exhibits the clearest benefit from multiscale structural decomposition: the radius-1 and radius-2 representation substantially outperforms degree alone, and adding degree provides a further improvement. Thus, the learned structural view is particularly useful when node roles cannot be adequately characterized by degree, while the Brazil result also shows that a more elaborate representation need not dominate when the target is already encoded by a simple structural statistic.

Topology-only graph classification. We evaluate IMDB-BINARY, IMDB-MULTI, COLLAB, and REDDIT-BINARY, none of which supplies informative node or edge attributes. Radius-1 and radius-2 node-rooted adjacency patches are canonicalized and grouped into dataset-adapted size buckets. A separate shared dictionary is fitted for each radius–size bucket using only the relevant training partition, after which the sparse activations are aggregated into a 600-dimensional graph-level structural representation. To reduce direct redundancy with S, the explicit patch-count coordinate is excluded from T. Complete dictionary and bucket configurations are reported in Table 16.

Table 19 reports the complete S, T, and ST results. The proposed structural coordinates outperform the statistics block on all four datasets, showing that T retains discriminative local organization not captured by coarse graph summaries. Combining the two views provides a further improvement in

Table 13: Graph classification accuracy (%) on attributed TU datasets. Baseline results are obtained from our reproductions under their released protocols. Our method and GIN use the same ten outer-fold seed set. Bold and underlined entries denote the best and second-best results, respectively. Complete configurations and reproduction details are provided in Appendix D.

Method	MUTAG	NCI1	PROTEINS	DD	ENZYMES
WL subtree	86.00 ± 0.25	$\mathbf{85.16 \pm 0.06}$	$\underline{75.66 \pm 0.11}$	$\underline{79.21 \pm 0.06}$	53.15 ± 0.35
GIN	84.68 ± 2.70	80.51 ± 0.63	72.31 ± 1.17	72.94 ± 1.77	$\underline{64.60 \pm 2.02}$
GraphMAE	86.32 ± 5.37	78.64 ± 1.09	72.14 ± 3.16	77.46 ± 1.75	58.67 ± 4.88
GNN-AK+	86.70 ± 6.80	78.90 ± 1.40	74.00 ± 3.80	76.30 ± 3.90	–
GSN-v	86.65 ± 2.26	81.85 ± 0.19	72.70 ± 1.09	75.57 ± 0.96	56.73 ± 1.46
Ours	$\mathbf{89.20 \pm 1.11}$	$\underline{83.09 \pm 0.28}$	$\mathbf{77.85 \pm 0.71}$	$\mathbf{79.48 \pm 0.51}$	$\mathbf{69.73 \pm 0.87}$

Table 14: Reproduced test performance on molecular and long-range graph benchmarks. MolHIV, MolBACE, and MolTox21 report ROC–AUC; Peptides-func reports AP. All values are percentages given as mean ± standard deviation.

Method	MolHIV	MolBACE	MolTox21	Peptides-func
CIN	80.36 ± 0.95	76.28 ± 4.36	75.52 ± 0.55	61.75 ± 0.75
GraphGPS	77.31 ± 0.72	80.36 ± 3.12	74.07 ± 0.70	65.74 ± 0.14
GSN	78.53 ± 0.62	76.96 ± 4.08	77.44 ± 0.63	65.74 ± 0.14
Subgraphormer	78.77 ± 0.65	82.00 ± 1.36	76.42 ± 0.67	62.91 ± 0.80
GPSE	77.89 ± 2.07	$\mathbf{82.90 \pm 0.88}$	76.68 ± 0.88	66.51 ± 0.77
GraphViT	79.02 ± 0.41	82.82 ± 1.62	$\mathbf{78.15 \pm 0.34}$	69.41 ± 0.77
HOD-GNN	81.16 ± 0.42	81.03 ± 0.71	77.83 ± 0.74	68.77 ± 0.44
Ours	$\mathbf{81.47 \pm 0.60}$	81.67 ± 0.44	76.67 ± 0.27	$\mathbf{71.57 \pm 0.13}$

every case. The benefit is especially clear on COLLAB and REDDIT-BINARY, where T substantially improves upon S and ST achieves the strongest result among the evaluated configurations.

Across these datasets, T improves over S despite excluding supplied attributes and explicit graph-size coordinates. The consistent additional gain from ST indicates that learned patch organization and conventional graph statistics are complementary topology-derived views rather than interchangeable feature blocks. These results also show that the predictive utility of the representation does not depend on topology being entangled with semantic attributes or optimized jointly with an end-to-end graph neural network.

Reproduced molecular and long-range baselines. We further reproduce representative neural baselines on MolHIV, MolBACE, MolTox21, and Peptides-func using their released implementations and the official dataset splits. ROC–AUC is used for the three molecular benchmarks, while average precision is used for Peptides-func. We follow the random seeds specified by the corresponding studies whenever available and retain each method's released model-selection protocol. Table 14 reports the reproduced test results. Our explicit representation achieves the strongest performance on MolHIV and Peptides-func, while remaining competitive with the strongest reproduced neural models on MolBACE and MolTox21. In particular, its advantage on Peptides-func is observed against graph Transformers, positional-encoding methods, and higher-order GNNs, indicating that the learned structural coordinates remain effective on long-range multi-label prediction. Together with the molecular results, these experiments show that the proposed representation supports competitive prediction across different semantic domains without requiring end-to-end neural representation learning.

Among the reproduced methods, our representation attains the highest mean performance on both MolHIV and Peptides-func. On MolHIV, it reaches 81.47 ± 0.60 ROC–AUC, compared with 81.16 ± 0.42 for the strongest reproduced neural baseline, HOD-GNN. The advantage is more pronounced on Peptides-func, where our method achieves 71.57 ± 0.13 AP, exceeding GraphViT's 69.41 ± 0.77 by 2.16 points. It also remains competitive on MolBACE and MolTox21 despite relying on our framework and a conventional downstream predictor rather than end-to-end neural representation

Table 15: MolHIV predictive configurations under the official scaffold split. Panel A compares matched combinations of composition, typed-patch, and dictionary-conditioned context features. Panel B adds explicit coupling features to the 693-dimensional composition–typed-patch backbone. Within each panel, † denotes the configuration selected by mean validation ROC–AUC; test results do not participate in selection.

Configuration	Included blocks	Dim.	Valid	Test
A. Predictive-view ablation				
Composition	molecular composition	69	76.61 ± 0.79	72.65 ± 0.58
Typed patches	reconstructed typed-patch coordinates	624	79.66 ± 0.58	78.73 ± 0.34
Context	five retained context blocks	325	78.75 ± 0.50	77.16 ± 0.52
Composition + context	composition and context	394	80.45 ± 0.63	79.10 ± 0.71
Base	composition and typed patches	693	82.09 ± 0.61	80.49 ± 0.66
Typed patches + context	typed patches and context	949	81.72 ± 0.60	79.63 ± 0.63
Drop-`ring_any`†	composition, typed patches, and context	1018	$\mathbf{83.41 \pm 0.49}$	$\mathbf{81.47 \pm 0.60}$
B. Explicit coupling on the Base representation				
Base	no explicit coupling	693	82.09 ± 0.61	80.49 ± 0.66
Patch-mass coupling	Base + patch-level semantic mass	3765	83.89 ± 0.56	78.26 ± 0.62
Root-conditioned coupling	Base + root-semantic conditioning	3765	83.13 ± 0.68	80.86 ± 0.74
Mass + root coupling†	Base + both coupling blocks	6837	$\mathbf{83.91 \pm 0.83}$	79.23 ± 0.50

learning. These results show that making topology separately accessible does not require sacrificing predictive utility, including on large molecular and long-range graph benchmarks.

D.4 MolHIV Representation and Coupling Variants

MolHIV is used for predictive evaluation, information-source auditing, explicit structure–semantic coupling, and molecular case studies. We therefore distinguish the predictive configurations used in these experiments. All results in Table 15 use the official OGB scaffold split, a dictionary fitted without test labels, a fixed dictionary seed, and ten downstream-predictor seeds. Reported values are mean ROC–AUC $\pm$ sample standard deviation.

The internal implementation name `nora` denotes the *drop-`ring_any`* view, rather than the removal of all ring or aromatic information. This view concatenates molecular composition (69 dimensions), reconstructed typed-patch coordinates (624 dimensions), and five dictionary-conditioned context blocks (325 dimensions): `ring5`, `ring6`, `aromatic_ring`, `multi_ring`, and `ring_boundary`. Only the aggregate `ring_any` block is excluded, giving 1018 dimensions in total.

The view ablation shows that the three information sources are individually predictive and complementary: the validation-selected drop-`ring_any` configuration improves over the composition–typed backbone and produces the reported 81.47 ± 0.60 test ROC–AUC. The coupling comparison exhibits a different pattern. Combining both coupling blocks gives the highest validation score, whereas the root-conditioned variant gives the highest test score. We retain the validation-selected configuration when comparing coupling operators and report all test results without post-hoc reselection. These predictive variants are separate from the fixed $S/T/A$ coalition audit in Appendix E, whose purpose is contribution analysis rather than selection of the strongest predictive view.

D.5 Implementation Details

Structural observations and dictionary learning. Table 16 summarizes the structural observation and sparse-decomposition settings. Unless otherwise stated, we extract node-rooted induced patches at radii 1 and 2. A canonical patch vector contains the upper-triangular adjacency entries, a root indicator, and a valid-node mask. Dictionary learning is weighted by patch frequency and is restricted to the training portion available at the corresponding evaluation stage. In nested cross-validation, the dictionary is fitted on the inner training split for model selection and refitted on the complete outer training–validation split before testing.

Table 16: Structural-observation and dictionary configurations. Here, q is the maximum patch capacity, K is the number of dictionary atoms, s is the OMP sparsity, and I is the number of K-SVD iterations. All supervised dictionaries are fitted without test graphs.

Dataset	Rooted observation and capacity q	K	s	I
IMDB-B / IMDB-M	Exact canonical R1/R2 patches; size buckets $\{16, 32, 64, 128, 256\}$	8^*	3	2
COLLAB	Exact canonical R1/R2 patches; size buckets $\{64, 128, 256, 384, 512\}$	8^*	3	2
REDDIT-B	Exact canonical R1/R2 patches; size buckets $\{64, 128, 256, 512, 4096\}$	8^*	3	2
MUTAG / NCI1	Exact canonical node-rooted R1/R2 patches; $q = 16$	16	4	5
PROTEINS	Exact canonical node-rooted R1/R2 patches; $q = 57$	16	4	5
ENZYMES	Exact canonical node-rooted R1/R2 patches; $q = 26$	16	4	5
DD	Exact canonical node-rooted R1/R2 patches; $q = 98$	16	4	5
MolHIV	Exact canonical binary-topology R1/R2 patches; $q = 13$	64	8	5
MolBACE / MolTox21	Exact canonical binary-topology R1/R2 patches; $q = 12$	32	6	5
Peptides-func	Structurally ordered node-rooted R2 patches; $q = 16$	32	4	3

*Eight atoms are learned separately for each radius–size bucket.

Table 17: Dimensions of the independently audited information blocks. A dash denotes an unavailable attribute block. The last column lists the explicit coupling used in the corresponding coupling experiment; ordinary ST, SA, TA, and STA denote direct concatenation.

Dataset	$\dim(S)$	$\dim(T)$	$\dim(A)$	Coupling
IMDB-B / IMDB-M	17	600	–	–
COLLAB / REDDIT-B	17	600	–	–
MUTAG	17	230	44	B_{role}
NCI1	17	230	148	B_{role}
PROTEINS	17	230	16	B_{role}
ENZYMES	17	230	84	B_{role}
DD	17	230	356	B_{role}
MolHIV	8	902	61	$B_{\mathrm{mass}}, B_{\mathrm{root}}$
MolBACE	8	454	61	–
MolTox21	8	454	61	B_{mass}
Peptides-func	17	160	186	$B_{\mathrm{mass}} + B_{\mathrm{root}}$

For the larger topology-only TU datasets, patch types are divided into size buckets and a separate dictionary is learned for each radius–bucket combination. This avoids padding small patches to the largest observed neighborhood. For MolHIV, MolBACE, and MolTox21, the strict information audits use exactly canonicalized binary-topology patches; radius-2 patches exceeding the prescribed capacity are excluded rather than partially truncated. Peptides-func uses radius-2 patches arranged by a deterministic structural ordering based on degree and neighboring-degree signatures.

For each topology-only TU radius–bucket combination, at most 256 distinct training patch types are used to fit the corresponding dictionary. DD uses the 20,000 most frequent training patch types. Peptides-func samples 30,000 training patches for dictionary fitting; no Peptides-func patch exceeds its capacity. The complete truncation and patch-frequency diagnostics are included in the released experiment records.

Information blocks and coupling operators. The dimensions of the independently accessible information blocks are listed in Table 17. The S block consists only of graph-size and degree-distribution statistics. The T block is obtained by invariant pooling of sparse patch activations and reconstruction statistics. For TU datasets, A contains the mean, standard deviation, minimum, and maximum of each supplied node or edge attribute. For the molecular and peptide datasets, A contains normalized marginal distributions of the supplied categorical atom and bond attributes, without encoding their structural placement.

The role-resolved TU operator B_{role} associates normalized absolute atom activations with root, first-shell, second-shell, and edge attribute summaries. For the molecular and peptide experiments, B_{mass} measures graph-level semantic–atom coactivation, whereas B_{root} conditions atom activation on the

Table 18: Runtime (hours) and peak memory (GiB) on molecular benchmarks. *Separate repr.* denotes the one-time construction of our shared dictionary and graph coordinates; *Downstream/joint* denotes XGBoost probing for our method and complete end-to-end training for the neural baselines. Our representation is reused across tasks and predictor seeds. CPU memory is system-level peak physical memory for ours and process RSS for the reproduced baselines; GPU memory is peak allocated memory. Because the measurements were obtained under different hardware environments, they characterize computational requirements but do not constitute a hardware-normalized speed comparison.

Method	Scope	Sep. repr.	Train/probe	Total	CPU	GPU
MolHIV: single prediction task						
Ours (CPU)	1 seed / 1 probe	1.01	0.04	1.05	6.76	–
Ours (CPU)	10 seeds / 10 probes	1.01	0.44	1.45	6.76	–
GraphGPS (GPU)	1 seed	–	2.80	2.80	2.75	0.74
HOD-GNN (GPU)	1 seed	–	2.16	2.16	29.47	11.24
MolTox21: 12 prediction tasks						
Ours (CPU)	1 seed (12 probes)	0.14	0.49	0.63	2.14	–
Ours (CPU)	10 seeds (120 probes)	0.14	4.88	5.02	2.14	–
GraphGPS (GPU)	1 joint seed	–	0.35	0.35	2.37	0.16
HOD-GNN (GPU)	1 joint seed	–	1.11	1.11	3.68	3.02

semantics of the patch root. The precise operator definitions are given in Appendix A.5. All coupling configurations are chosen using validation data.

Downstream predictors. All reported representations are evaluated with XGBoost. For TU datasets, we select one of eight configurations independently for every information view and outer fold:

$$n_{\text{trees}} = 300, \qquad d_{\max} \in \{2, 4\},$$
$$\eta \in \{0.03, 0.08\}, \quad w_{\min} \in \{1, 4\},$$

with subsampling and column sampling both set to 0.85, ℓ_1 regularization set to 0.01, and ℓ_2 regularization set to 2.0. The configuration maximizing inner-validation accuracy is refitted on the outer training–validation set before evaluation on the held-out outer fold.

The strict MolBACE, MolTox21, and Peptides-func experiments use 600 trees, maximum depth 6, learning rate 0.05, minimum child weight 5, row subsampling 0.8, column subsampling 0.9, ℓ_2 regularization 3, and `max_bin`$= 256$. MolHIV uses the validation-selected configuration

$$(n_{\text{trees}}, d_{\max}, \eta, w_{\min}) = (900, 8, 0.02521, 8),$$
$$(\text{subsample}, \text{colsample}) = (0.8986, 0.7715),$$
$$(\lambda_1, \lambda_2, \gamma) = (0.03850, 0.4269, 3.56 \times 10^{-5}),$$

with `max_bin`$= 128$. Binary tasks use the logistic objective; multiclass TU tasks use the multiclass soft-probability objective. ROC–AUC and average-precision experiments use their corresponding validation metrics. For multilabel datasets, one binary predictor is trained per task, and class weighting is computed from the corresponding training labels.

D.6 Computational Cost

We separately measure one-time representation construction and downstream prediction, since the constructed structural coordinates are reused across label tasks and predictor seeds. All stages of our method were executed on CPU, with XGBoost restricted to two threads.

The single-task MolHIV experiment illustrates the amortization enabled by the explicit representation. Constructing the shared dictionary and graph coordinates requires 60.4 minutes, but each subsequent XGBoost run requires only approximately 2.6 minutes. Consequently, extending the evaluation from one to ten predictor seeds increases the complete runtime from 1.05 to only 1.45 hours. For reference, our reproduced 100-epoch runs require 2.80 hours for GraphGPS and 2.16

hours for HOD-GNN, each for a single seed. The HOD-GNN paper separately reports 53.34 seconds per training epoch on an NVIDIA A100, which corresponds to approximately 1.48 hours for 100 training epochs before evaluation and setup overhead.

MolTox21 exposes a different computational regime. Its 12 partially labeled binary tasks are handled by separate XGBoost classifiers because they have different observed labels and class distributions. Although representation construction requires only 8.5 minutes, ten-seed evaluation entails $12 \times 10 = 120$ classifier fits, increasing downstream prediction to 4.88 hours and the complete runtime to 5.02 hours. GraphGPS instead predicts the 12 tasks jointly and is therefore faster in this setting. Overall, sparse structural decomposition constitutes a one-time and reusable cost: additional single-task evaluations are inexpensive, whereas the current multi-task runtime is governed primarily by the number of independently fitted probes. A joint multi-output predictor could reduce this remaining overhead without changing the structural representation.

E Complete Structural-Information Audit

E.1 Information Blocks and Audit Metrics

A central advantage of an explicit graph representation is that different information sources remain independently accessible before downstream prediction. This allows us to audit how much predictive information is associated with graph structure, rather than inferring structural importance from entangled latent embeddings.

For experiments following the unified factorial design, we partition each graph representation into three blocks:

- $\mathbf{S}$ contains low-order graph statistics, including the logarithmic numbers of nodes and edges, graph density, degree moments and quantiles, and a normalized degree histogram;
- $\mathbf{T}$ is the learned sparse topological representation obtained from rooted structural observations, a training-fitted shared dictionary, sparse coding, and permutation-invariant readout;
- $\mathbf{A}$ contains permutation-invariant moments of the supplied node and edge attributes, without explicitly binding attributes to their structural locations.

The topology readout excludes the number of extracted patches, preventing $\mathbf{T}$ from trivially reproducing the graph-size information already contained in $\mathbf{S}$. We additionally evaluate an optional binding block $\mathbf{B}$, which associates node and edge attributes with the activations of individual dictionary atoms. Since $\mathbf{B}$ describes an interaction between structure and attributes rather than an independent information source, it is reported separately and is not treated as a fourth player in the contribution analysis.

Whenever all three information blocks are available, we evaluate the complete coalition lattice

$$\varnothing, \quad S, \quad T, \quad A, \quad ST, \quad SA, \quad TA, \quad STA$$

using identical data splits, predictor families, and model-selection procedures. Here, $v(C)$ denotes the held-out performance obtained using coalition C. For accuracy-based datasets, $v(\varnothing)$ is the majority-class accuracy.

We quantify the marginal predictive contribution of block $i \in \{S, T, A\}$ using its Shapley value:

$$\phi_i = \sum_{C \subseteq \{S,T,A\} \setminus \{i\}} \frac{|C|!(2 - |C|)!}{3!} \left[v(C \cup \{i\}) - v(C) \right]. \tag{159}$$

The normalized contribution reported in the main paper is

$$\widetilde{\phi}_i = \frac{\phi_i}{\phi_S + \phi_T + \phi_A} = \frac{\phi_i}{v(STA) - v(\varnothing)}. \tag{160}$$

For structure-only datasets, the corresponding two-player values are

Table 19: Complete audit on structure-only graphs. Predictive results are test accuracy in percent, reported as seed-level mean ± population standard deviation. Shapley values are normalized contribution shares, and $I_{S,T}$ is measured in accuracy points.

	Predictive performance			Contribution (%)		
Dataset	S	T	ST	$\widetilde{\phi}_S$	$\widetilde{\phi}_T$	$I_{S,T}$
COLLAB	78.25 ± 0.12	80.46 ± 0.33	$\mathbf{80.72 \pm 0.30}$	46.1	**53.9**	-25.68
IMDB-BINARY	72.33 ± 0.90	73.68 ± 0.61	$\mathbf{73.87 \pm 0.66}$	47.2	**52.8**	-22.14
IMDB-MULTI	50.40 ± 0.55	50.53 ± 0.53	$\mathbf{50.97 \pm 0.63}$	49.6	**50.4**	-16.64
REDDIT-BINARY	88.86 ± 0.16	90.55 ± 0.19	$\mathbf{91.14 \pm 0.33}$	48.0	**52.0**	-38.27

$$\phi_S = \frac{v(S) - v(\varnothing) + v(ST) - v(T)}{2}, \qquad \phi_T = \frac{v(T) - v(\varnothing) + v(ST) - v(S)}{2}. \tag{161}$$

We also measure finite-difference interactions. Pairwise interactions take the form

$$I_{i,j} = v(ij) - v(i) - v(j) + v(\varnothing), \tag{162}$$

and the third-order interaction is

$$\begin{aligned} I_{S,T,A} = v(STA) - v(ST) - v(SA) - v(TA) \\ + v(S) + v(T) + v(A) - v(\varnothing). \end{aligned} \tag{163}$$

A negative pairwise interaction indicates that two blocks contain overlapping predictive information under the selected predictor. It does not imply that either block is causally harmful.

E.2 Structure-Only Graphs

COLLAB, IMDB-BINARY, IMDB-MULTI, and REDDIT-BINARY contain neither supplied node nor edge attributes. Their audits therefore reduce to S, T, and ST. All four datasets use ten random seeds and ten outer folds. Each outer-training partition is further divided into inner training and validation subsets for model selection. The structural dictionary is fitted using inner-training data during model selection and refitted on the corresponding outer train–validation partition before test evaluation.

The learned topology block outperforms conventional graph statistics on all four datasets, although the difference is small on IMDB-MULTI. Combining the two blocks consistently gives the highest accuracy. The normalized contributions remain comparatively balanced, with topology accounting for 50.4%–53.9% of the total predictive gain. The negative interactions show that graph statistics and learned topology contain substantial overlapping information. Nevertheless, the consistent improvement of T over S, and of ST over either isolated block, indicates that the learned structural coordinates cannot be reduced entirely to graph size and degree statistics.

E.3 Attributed TU Graphs

For attributed TU datasets, we evaluate all eight coalitions of S, T, and A. We additionally report $STAB$, which appends explicit topology–attribute binding features to the ordinary STA concatenation.

The attributed datasets exhibit markedly different information regimes. NCI1 is topology dominated: T substantially outperforms either S or A and receives 47.4% of the total audited contribution. Explicit topology–attribute binding further improves STA by 4.32 accuracy points. ENZYMES is instead attribute dominated, with A alone reaching 68.07%. PROTEINS and DD show more balanced regimes in which attributes and graph statistics are strong, while topology provides less independent information.

Table 20: Complete factorial information-block audit on attributed TU datasets. Values are test accuracy in percent. Each entry reports the mean and population standard deviation across ten seed-level results, where each seed-level result is averaged over ten outer test folds. Boldface indicates the best information-block combination for each dataset.

Dataset	$\varnothing$	S	T	A	ST	SA	TA	STA
NCI1	50.05±0.00	68.74±0.27	75.63±0.34	67.26±0.46	76.70±0.49	72.87±0.21	77.95±0.44	**78.77±0.37**
PROTEINS	59.57±0.00	74.14±0.63	72.28±0.53	73.62±0.31	73.52±0.34	**77.85±0.71**	75.53±0.43	76.68±0.68
ENZYMES	16.67±0.00	37.75±1.08	41.62±1.39	68.07±0.65	44.35±1.51	**69.73±0.87**	63.20±1.12	64.12±1.17
DD	58.66±0.00	76.48±0.24	75.79±0.63	76.88±0.32	76.55±0.80	**79.11±0.56**	77.92±0.48	78.56±0.74
MUTAG	66.49±2.28	86.17±5.38	86.64±5.73	81.93±7.48	87.19±4.43	86.70±4.27	85.58±4.95	**87.93±1.49**

Table 21: Contribution and interaction audit on attributed TU datasets. $\widetilde{\phi}$ denotes normalized Shapley contribution (%), while interaction terms and $\Delta_B = STAB - STA$ are measured in accuracy points. Negative pairwise interactions indicate predictive redundancy between blocks.

	Normalized contribution (%)			Interaction effect (accuracy points)				
Dataset	$\widetilde{\phi}_S$	$\widetilde{\phi}_T$	$\widetilde{\phi}_A$	$I_{S,T}$	$I_{S,A}$	$I_{T,A}$	$I_{S,T,A}$	Δ_B
NCI1	26.5	**47.4**	26.1	−17.62	−13.08	−14.89	+12.83	+4.32
PROTEINS	36.0	23.7	**40.3**	−13.33	−10.34	−10.81	+10.25	−0.50
ENZYMES	17.0	14.2	**68.8**	−18.35	−19.42	−29.82	+17.60	−1.87
MUTAG	**38.3**	36.8	24.9	−19.12	−14.91	−16.49	+16.71	+1.27
DD	33.4	28.7	**37.9**	−17.06	−15.59	−16.09	+15.47	+0.93

All five datasets show negative pairwise interactions, demonstrating extensive redundancy among statistics, topology, and attributes. The positive third-order terms do not imply that STA must always be optimal. For example, SA outperforms STA on PROTEINS and ENZYMES. Instead, these interactions show that the marginal value of one information source depends on which other sources are already available.

E.4 Audits on Molecular and Peptide Benchmarks

We conduct complete factorial audits on MolHIV, MolBACE, MolTox21, and Peptides-func using the coalition lattice

$$\{\emptyset, S, T, A, ST, SA, TA, STA\}.$$

For the three molecular datasets, S contains eight topology-only graph statistics, A contains 61 permutation-invariant summaries of the supplied atom and bond attributes, and T contains readouts of radius-one and radius-two rooted structural patches encoded by a training-fitted shared dictionary. For Peptides-func, the corresponding blocks contain 17 graph statistics, 186 node/edge attribute summaries, and 160 sparse structural coordinates. In every experiment, the dictionary and all data-dependent transformations are fitted using the official training partition only. The same fixed XGBoost configuration is applied to every nonempty coalition, isolating the effect of the available information rather than coalition-specific hyperparameter tuning.

For ROC–AUC datasets, $v(\emptyset) = 0.5$. For Peptides-func, the empty coalition predicts a task-wise constant score equal to the positive-class prevalence estimated from the official training partition. Mean AP is then evaluated separately on the validation and test splits, whose slightly different label prevalences yield baseline scores of 16.48 and 16.47, respectively.

Table 23 reports the corresponding normalized Shapley allocations and finite-difference interactions. Shapley values are computed independently on validation and test performance rather than transferring validation allocations to the test split. Consequently, the table can reveal changes in the predictive reliability of an information source across partitions. Interaction values are expressed in percentage points of the corresponding evaluation metric.

The four benchmarks exhibit distinct information regimes. On MolHIV, topology and supplied molecular attributes make nearly equal marginal contributions on test, while graph statistics account

Table 22: Complete factorial audits on molecular and peptide benchmarks. MolHIV, MolBACE, and MolTox21 report ROC–AUC (%), while Peptides-func reports mean average precision (%). Values are mean ± population standard deviation over ten paired downstream-model seeds. Boldface identifies the validation-selected coalition and its paired test result.

Dataset	Split	$\emptyset$	S	T	A	ST	SA	TA	STA
MolHIV	Valid	50.00	64.88±0.27	74.98±0.52	74.01±0.63	74.54±0.76	77.07±0.78	80.31±0.67	**81.78±0.61**
	Test	50.00	61.97±0.19	73.27±0.49	73.69±0.42	73.10±0.49	72.76±0.44	75.20±0.40	**75.94±0.48**
MolBACE	Valid	50.00	**77.48±0.52**	74.87±0.40	59.70±0.62	75.45±0.39	64.77±0.80	72.44±0.48	73.12±0.51
	Test	50.00	**78.69±0.14**	76.52±0.71	78.98±0.59	78.10±0.71	80.80±0.28	79.86±0.75	80.82±0.78
MolTox21	Valid	50.00	62.12±0.15	68.74±0.17	75.37±0.12	69.61±0.25	77.22±0.17	76.30±0.14	**77.38±0.21**
	Test	50.00	62.50±0.12	66.21±0.22	73.98±0.14	66.53±0.39	75.08±0.12	73.02±0.33	**74.05±0.21**
Peptides-func	Valid	16.48	41.59±0.10	64.79±0.17	59.58±0.18	65.96±0.10	63.03±0.11	67.87±0.15	**68.61±0.18**
	Test	16.47	41.56±0.24	64.59±0.15	60.78±0.21	65.56±0.19	63.77±0.11	67.98±0.19	**68.66±0.21**

Table 23: Normalized Shapley allocations and interaction effects for the complete molecular and peptide audits. Shapley values are percentages of $v(STA) - v(\emptyset)$; interaction terms are percentage points. Negative pairwise interactions indicate predictive redundancy rather than causal harm.

Dataset	Split	$\widetilde{\phi}_S$	$\widetilde{\phi}_T$	$\widetilde{\phi}_A$	$I_{S,T}$	$I_{S,A}$	$I_{T,A}$	$I_{S,T,A}$
MolHIV	Valid	18.5	39.5	42.0	-15.32	-11.82	-18.68	13.73
	Test	15.6	42.1	42.3	-12.14	-12.91	-21.75	13.81
MolBACE	Valid	44.7	55.6	-0.3	-26.90	-22.42	-12.14	22.52
	Test	33.9	28.9	37.2	-27.10	-26.87	-25.64	26.24
MolTox21	Valid	17.7	28.1	54.1	-11.25	-10.26	-17.80	10.47
	Test	19.7	23.2	57.1	-12.19	-11.41	-17.18	12.12
Peptides-func	Valid	18.0	44.9	37.1	-23.93	-21.65	-40.01	21.22
	Test	17.7	43.8	38.5	-24.12	-22.10	-40.92	21.82

for a substantially smaller share. Neither T nor A alone matches their joint performance, and both TA and STA improve over the isolated blocks. The result therefore indicates complementary structural and semantic evidence rather than prediction driven primarily by coarse molecular statistics.

MolBACE exhibits a pronounced split-dependent change. On validation, graph statistics provide the strongest isolated performance, while the Shapley allocation assigns substantial marginal utility to both S and T and approximately zero net utility to A. On test, the allocation becomes more balanced and the attribute contribution becomes strongly positive. Moreover, SA and STA outperform the validation-selected S coalition on test, although these test outcomes are not used to revise the selection decision. This redistribution of predictive utility is consistent with the class-conditional and chemical-series shift examined in Appendix F; it shows that the reliability of an information source can change across official partitions.

MolTox21 is predominantly attribute driven. The attribute block is the strongest isolated source on both validation and test and receives the majority of the Shapley allocation. Although topology is predictive in isolation, its incremental benefit becomes limited after the supplied molecular attributes and graph statistics are available: on test, SA exceeds both TA and STA. Peptides-func presents the complementary regime. Its topological block is the strongest isolated source, while attributes provide substantial additional information; TA and STA therefore improve markedly over either isolated block.

All four datasets show negative pairwise interactions and positive third-order interactions. This pattern reflects extensive overlap among graph statistics, learned topology, and supplied attributes, together with predictor-dependent higher-order complementarity. It does not imply that an information source is intrinsically detrimental. The allocations quantify marginal predictive utility for the specified blocks, downstream probe, metric, and data split; they should not be interpreted as causal effects of molecular structure.

E.5 Interpretation and Limitations

Across the unified and dataset-specific audits, no single information source dominates every benchmark. Learned topology is particularly informative on NCI1, MolHIV, Peptides-func, and the structure-only social-network datasets. Attributes dominate ENZYMES and remain important on PROTEINS and DD. Several datasets exhibit strong redundancy: T may be predictive in isolation while adding relatively little once graph statistics or attributes are already available. Other datasets, such as NCI1, benefit substantially from explicitly binding structural atoms to semantic attributes.

The purpose of this audit is therefore not to establish that topology must always receive the largest contribution. Instead, the explicit representation makes it possible to determine whether structural information is independently predictive, complementary to semantic attributes, largely redundant with coarse statistics, or unstable under distribution shift. These distinctions are difficult to recover directly once topology and attributes have been entangled within task-specific latent embeddings.

Finally, the reported contributions depend on the definitions of the information blocks, the downstream predictor, the evaluation metric, and the data distribution. They quantify marginal predictive information at the representation level and should not be interpreted as causal explanations of individual predictions.

F Molecular Case Studies: Hypothesis Validation and Shortcut Auditing

F.1 Traceability and Evaluation Protocol

The explicit representation retains the correspondence between every rooted observation and its sparse encoding. For a graph G, we store each patch occurrence as

$$\big(G, R, \mathbf{y}_{G,R}, \boldsymbol{\alpha}_{G,R}, \mathbf{p}_{G,R}\big),$$

where R identifies the root, $\mathbf{y}_{G,R}$ is the structural observation, $\boldsymbol{\alpha}_{G,R}$ is its sparse code, and $\mathbf{p}_{G,R}$ contains the aligned root or patch semantics when available. Consequently, a graph-level structural coordinate can be traced through its readout statistic and dictionary atom to the real patch occurrences that contribute to it. Each retrieved occurrence retains its dataset index, root identity, structural code, and original atom and bond attributes, allowing the corresponding local environment to be inspected in the input molecule.

A dictionary atom is a learned basis direction rather than a predefined motif or a guaranteed one-to-one chemical substructure. We therefore assign chemical descriptions only after examining the empirical distribution of real training-set patches that strongly activate the atom. When an atom represents multiple local environments, these occurrences are further partitioned according to their rooted atom and bond configurations. Semantic conditions such as root aromaticity are used to identify chemically coherent subsets of an atom's occurrences; they are not interpreted as changing the topology encoded by the atom itself.

The two case studies use this traceability for different purposes and follow different evaluation protocols, summarized in Table 24. On MolHIV, candidate root-conditioned cells are discovered from training-set attribution patterns and retained only when their direction and stability are confirmed on validation. Their activating training patches are then resolved into empirical chemical subfamilies, after which the locked hypothesis is examined across the official splits. Test labels are used only for descriptive reporting within this procedure. Because earlier project-level experiments had already inspected MolHIV test performance, however, we conservatively regard the complete case study as a retrospective analysis rather than a prospective chemical discovery.

The MolBACE study instead analyzes a previously observed validation–test transfer failure. Candidate carriers are searched retrospectively for patch families whose removal impairs validation ranking but improves test ranking. The dictionary is still fitted on the official training partition, and the audited predictor is not retrained or modified during intervention; nevertheless, the carrier-selection criterion explicitly uses both validation and test behavior. MolBACE is therefore presented as a post-hoc failure diagnosis, not as a blind discovery result.

Table 24: Selection and evaluation roles in the two molecular case studies. "Fixed-model intervention" denotes modification of selected patch-level activations followed by recomputation of the original readout and evaluation of the unchanged predictor.

Case	Component identification	Test-set role	Primary evidence
MolHIV	Train discovery followed by validation confirmation	Descriptive examination after locking the analyzed subfamily	Split-stratified enrichment and covariate-adjusted association analysis
MolBACE	Retrospective search for validation-helpful but test-harmful activation families	Explicitly used for post-hoc failure diagnosis	Exact occurrence-level intervention on a fixed representation and predictor

Table 25: MolHIV component cells discovered on the training split and confirmed on the validation split. Mean absolute TreeSHAP values are averaged over ten fitted XGBoost seeds. Patch concentration is measured among the 400 strongest training-patch activations and reports the fraction belonging to the dominant broad family and the dominant exact patch, respectively. The dagger marks the cell selected for subsequent chemical resolution.

Rank	Root condition q	Atom k	Train $\lvert\text{SHAP}\rvert$	Valid $\lvert\text{SHAP}\rvert$	Broad / exact
1	`in_ring=True`	52	0.01190	0.01015	36.8 / 11.5
2	`numH=1`	19	0.00865	0.00904	12.2 / 5.8
3	`hybridization=1`	46	0.00473	0.00534	15.2 / 3.5
4	`formal_charge=0`	3	0.00472	0.00442	31.2 / 12.8
5†	`in_ring=True`	18	0.00468	0.00474	60.0 / 51.0

The evidential scope also differs between the two studies. MolHIV establishes a statistically testable association between a learned structural component, a composite chemotype, and the recorded activity label; it does not establish a biochemical antiviral mechanism. MolBACE provides a causal intervention on the behavior of the fitted representation–predictor pipeline: deleting a specified set of patch activations causes the reported prediction changes. It does not imply that the corresponding chemical environment causally promotes or prevents BACE inhibition. Together, the cases demonstrate occurrence-level traceability for both hypothesis validation and model-generalization auditing.

F.2 MolHIV: From a Learned Component to a Chemical Hypothesis

This case study audits the 3,765-dimensional *Root-conditioned coupling* configuration reported in Table 15, rather than the combined Mass + root variant. Each fitted predictor receives the 69-dimensional composition block, the 624-dimensional reconstructed typed-patch block, and the 3,072-dimensional root-conditioned coupling block. TreeSHAP attributions and subsequent interventions are computed for these same ten fitted XGBoost models.

Candidate component discovery. The structure–semantic representation contains root-conditioned component cells of the form $c = (q, k)$, where q denotes a chemical condition of the root atom and k indexes a learned structural dictionary atom. We used exact TreeSHAP values for the fitted XGBoost models to identify cells that were both used consistently by the predictor and associated with the same prediction direction across random seeds. TreeSHAP values are expressed on the log-odds scale.

Candidate discovery was performed on the training split. A cell was considered train-eligible if its mean signed contribution was positive for active molecules and negative for inactive molecules, both directions were reproduced in at least 8 of the 10 model seeds, and its training mean absolute SHAP value exceeded 10^{-3}. Eligible cells were ranked by their training mean absolute SHAP values. We then applied the same directional criterion to the validation split, requiring agreement in at least 7 of the 10 seeds. Among the 128 train-eligible cells, only five passed this independent validation criterion (Table 25). Test-set statistics were appended only after the candidate set had been fixed and were not used in this selection.

Table 26: Chemical resolution of the principal aryl–N patch family activated by `in_ring=True` × `atom_18`. Each entry reports the number of molecules containing the subtype, its positive rate (%), and its odds ratio relative to molecules without that subtype within the same split. Candidates were discovered on train and confirmed on validation; test results are descriptive.

Chemical subtype	Train $n/\%/\mathrm{OR}$	Valid $n/\%/\mathrm{OR}$	Test $n/\%/\mathrm{OR}$
Secondary aniline-like (Ar–NH)	$673/2.8/0.74$	$59/3.4/1.77$	$35/2.9/0.90$
Azo-like (Ar–N=N)	$\mathbf{402/18.7/6.21}$	$\mathbf{36/19.4/13.06}$	$\mathbf{84/29.8/15.84}$
Nitro-like (Ar–N($^+$)(O)O)	$314/10.8/3.18$	$15/0.0/0.00$	$14/0.0/0.00$
Imine-like (Ar–N=C)	$155/0.0/0.00$	$25/0.0/0.00$	$25/4.0/1.28$

For chemical follow-up, we selected `in_ring=True` × `atom_18`. This choice was not based solely on attribution magnitude: several higher-ranked cells represented heterogeneous mixtures of local environments. In contrast, atom 18 under the ring-root condition exhibited the most concentrated empirical support among its strongest training activations, with 60.0% assigned to one broad patch family and 51.0% corresponding to a single exact rooted-patch type. It also preserved the required attribution direction on the validation split. Therefore, it provided a comparatively coherent starting point for tracing a learned component back to chemically interpretable occurrences. No chemical identity is assigned to the dictionary atom at this stage; its interpretation is derived below from the real molecular patches that activate it.

Occurrence-level tracing and chemical subtype resolution. We next traced the selected cell from its graph-level coordinate back to the individual rooted patches contributing to it. Specifically, for `in_ring=True` × `atom_18`, we retrieved the original molecule, root atom, induced radius-2 patch, and atom-18 activation associated with every nonzero occurrence. This produced 22,329 active rooted patches, which were grouped into 2,050 empirical families according to their rooted element-and-bond topologies. Patch families were screened using training-set label enrichment and an activation AUC above 0.5, followed by confirmation of the same enrichment direction on the validation split. Eleven families passed this procedure; test statistics were used only for descriptive evaluation.

This occurrence-level analysis also illustrates why a dictionary atom should not be interpreted as a one-to-one chemical motif. Although an exact piperazine-like patch was frequent among the strongest activations, its presence was not positively associated with activity: only 2 of 76 training molecules containing this patch were active (2.63%, $\mathrm{OR} = 0.69$), and no positive example occurred among the corresponding validation or test molecules. It therefore served as a negative control rather than an explanation of the parent cell's enrichment. In contrast, the leading train-discovered and validation-confirmed patch family contained an aromatic carbon root connected through the ring to an exocyclic nitrogen environment. This family occurred in 1,516 training molecules ($\mathrm{OR} = 2.46$), 127 validation molecules ($\mathrm{OR} = 3.60$), and 154 test molecules ($\mathrm{OR} = 8.40$).

Because the finite-radius patch did not completely specify the chemical environment beyond the exocyclic nitrogen, we returned each occurrence to its original molecular graph and inspected the complete adjacency of that nitrogen. This resolved the parent patch family into chemically distinct subtypes, including secondary aniline-like, azo-like, nitro-like, and imine-like environments. Their split-wise statistics are summarized in Table 26. The azo-like subtype was the only well-supported subtype exhibiting strong and consistently positive enrichment on both the training and validation splits: its positive rate was 18.7% among 402 training molecules and 19.4% among 36 validation molecules, compared with substantially lower background activity rates. The same pattern was observed descriptively on the test split, where 25 of 84 molecules (29.8%) were active.

Finally, we examined the full molecular environment of the azo-like occurrences rather than treating the local Ar–N=N pattern as a complete chemotype. Requiring both nitrogens of the N=N bond to have an aromatic-carbon neighbor yielded the more specific aryl–N=N–aryl structure used in the subsequent hypothesis test. Thus, atom 18 is not itself an "aryl–azo atom." Instead, its activating occurrences provided a traceable route from a learned structural coordinate to a chemically defined and independently testable subfamily.

Table 27: MolHIV activity rates stratified by aryl–azo status and sulfonate count. Each entry reports active molecules over all molecules in the stratum, followed by the positive rate. The validation split contains only two singly sulfonated aryl–azo molecules.

Split	Chemotype	No sulfonate	One sulfonate	≥ 2 sulfonates
Train	Non-azo	$1083/31921$ (3.39%)	$15/412$ (3.64%)	$41/198$ (20.71%)
	Aryl–azo	$6/225$ (2.67%)	$9/20$ (45.00%)	$\mathbf{78/125}$ (**62.40**%)
Valid	Non-azo	$69/4017$ (1.72%)	$1/36$ (2.78%)	$5/22$ (22.73%)
	Aryl–azo	$0/25$ (0.00%)	$0/2$ (0.00%)	$\mathbf{6/11}$ (**54.55**%)
Test	Non-azo	$89/3916$ (2.27%)	$8/85$ (9.41%)	$4/26$ (15.38%)
	Aryl–azo	$0/41$ (0.00%)	$3/7$ (42.86%)	$\mathbf{26/38}$ (**68.42**%)

Table 28: Size- and split-adjusted logistic analysis of MolHIV activity. Odds ratios are obtained by exponentiating the fitted coefficients.

Term	Adjusted OR	95% CI	p-value
Aryl–azo	0.493	[0.219, 1.11]	0.0884
One sulfonate	1.36	[0.89, 2.06]	0.157
≥ 2 sulfonates	5.83	[4.20, 8.11]	7.52×10^{-26}
Aryl–azo $\times$ one sulfonate	**22.0**	[**6.74**, **72.2**]	$\mathbf{3.16 \times 10^{-7}}$
Aryl–azo $\times \geq 2$ sulfonates	**8.06**	[**3.19**, **20.4**]	$\mathbf{1.03 \times 10^{-5}}$

Testing the composite aryl–azo–sulfonate hypothesis. Inspection of the full molecules containing the aryl–azo subtype revealed frequent co-occurrence with sulfonate groups, suggesting that the activity enrichment might depend on a composite chemotype rather than on the aryl–azo linkage alone. We therefore performed a molecule-level analysis independent of the learned representation. An aryl–azo structure was defined as an N=N bond whose two nitrogen atoms each have an aromatic-carbon neighbor. A sulfonate was defined as a carbon-bound sulfur atom connected to at least two double-bonded oxygens and one single-bonded oxygen, covering both the neutral sulfonic-acid and anionic sulfonate encodings present in the dataset. Molecules were stratified by aryl–azo status and by the number of sulfonate groups (0, 1, or ≥ 2).

Table 27 shows that the aryl–azo linkage alone was not associated with elevated activity. Among molecules without a sulfonate group, aryl–azo compounds had positive rates of 2.67%, 0%, and 0% in the training, validation, and test splits, respectively. In contrast, aryl–azo molecules containing multiple sulfonate groups exhibited positive rates of 62.40%, 54.55%, and 68.42% across the three official splits. The corresponding rates among non-azo molecules with multiple sulfonates were 20.71%, 22.73%, and 15.38%. A similar enrichment was observed for singly sulfonated aryl–azo molecules in the training and test splits, although the validation stratum contained only two such molecules and was therefore uninformative.

To determine whether this pattern could be explained by coarse differences in molecular composition or by the official data split, we fitted a descriptive logistic model containing aryl–azo status, sulfonate-count indicators, their interactions, standardized log heavy-atom count, standardized aromatic-atom fraction, and validation/test split indicators. The fitted model was

$$\begin{aligned}\operatorname{logit}\Pr(Y_i = 1) = \beta_0 + \beta_{\text{azo}} Z_i + \beta_1 Q_{i,1} + \beta_{2+} Q_{i,2+} \\ + \beta_{\text{azo}\times 1} Z_i Q_{i,1} + \beta_{\text{azo}\times 2+} Z_i Q_{i,2+} + \boldsymbol{\gamma}^\top \mathbf{c}_i,\end{aligned} \tag{164}$$

where Z_i indicates an aryl–azo structure, $Q_{i,1}$ and $Q_{i,2+}$ indicate one and multiple sulfonates, respectively, and $\mathbf{c}_i$ contains the molecular-size, aromaticity, and split covariates. The adjusted estimates are reported in Table 28.

The aryl–azo main effect was not significant after adjustment, whereas both aryl–azo–sulfonate interaction terms remained strong and significant. These results refine the initial local observation into a composite aryl–azo–sulfonate hypothesis: the activity enrichment is associated with the joint chemical context rather than with the aryl–azo linkage in isolation. More broadly, this case

Table 29: Split-dependent association of `aromatic=True` × `atom_01` with MolBACE labels. Cohen's d is computed as active minus inactive, so a negative value indicates stronger activation among inactive molecules.

Split	Inactive mean	Active mean	Cohen's d
Train	0.2157	0.1556	-0.459
Valid	0.2485	0.2022	-0.436
Test	0.1910	0.2403	$+0.363$

demonstrates how an explicit learned component can be traced to its activating occurrences, resolved beyond the finite patch boundary, and converted into a chemically defined hypothesis that can be examined directly against the original molecular data. The result concerns statistical enrichment in the historical whole-cell MolHIV screen and does not assign a specific molecular target or biochemical mechanism.

F.3 MolBACE: Auditing a Split-Specific Structural Shortcut

Identifying a traceable negative-transfer carrier. We applied the occurrence-level tracing procedure described above to a previously locked MolBACE model containing a 69-dimensional composition/global-statistics block and a 161-dimensional structural block. Because the purpose of this experiment was to explain an observed validation–test discrepancy, candidate discovery was conducted as a retrospective, test-aware audit rather than as prospective feature selection. We searched for root-conditioned dictionary components whose label association was consistent between train and validation but reversed on test, and then traced their strongest activations to the original molecular patches. The clearest carrier was `aromatic=True` × `atom_01`. Its highly activating occurrences correspond to an aromatic carbon root connected to two aromatic carbons, each continuing through the ring. We therefore describe it as an *aryl-like local carbon environment*, rather than as a complete phenyl group, because the radius-2 observation does not necessarily contain the entire ring. Consistent interpretations were recovered through overlapping root conditions, including aromaticity, degree three, SP2-like hybridization, and one attached hydrogen, indicating that the result was not specific to a single semantic indicator.

Association reversal across the official splits. The activation of this component exhibited a clear reversal in its marginal association with the class label (Table 29). Active molecules had lower activation than inactive molecules in both train and validation, with Cohen's $d = -0.459$ and -0.436, respectively. The fitted model could therefore learn that this aryl-like environment was evidence against BACE activity, and the validation split supported the same direction. On test, however, the association reversed to $d = +0.363$, making the train-supported use of this coordinate unreliable under the test distribution.

Exact occurrence-level intervention. The explicit representation allows this observation to be tested directly within the already fitted predictor. For every real rooted patch satisfying the aromatic-root condition, we set only its atom-01 sparse coefficient to zero. We then recomputed the five graph-level pools used by the original representation—activation frequency, positive mass, negative mass, RMS activation, and maximum absolute activation—while leaving every other coefficient and input feature unchanged. Finally, the intervened representations were passed through the same fitted XGBoost models without retraining. Reconstructing the unmodified representation from the stored patch coefficients reproduced the original graph features with zero numerical error, confirming that the intervention operated on the actual model input rather than on an explanatory surrogate.

Let

$$\Delta\mathrm{AUC} = \mathrm{AUC}_{\mathrm{intervened}} - \mathrm{AUC}_{\mathrm{original}}.$$

Deleting the conditioned atom-01 activations reduced validation ROC–AUC by 0.00875 on average, with a decrease in all ten model seeds. In contrast, it increased test ROC–AUC by 0.00402, again consistently across all ten seeds (Table 30). Thus, the same learned component supports validation performance but causes negative transfer on test.

Table 30: Frozen-model intervention on the aryl-like atom-01 component. No model is retrained after deleting the specified patch coefficients. Standard deviations are measured over ten fitted XGBoost seeds.

Evaluation split	ΔROC–AUC	Seed consistency
Valid	-0.00875 ± 0.00339	$10/10$ decrease
Test	$+0.00402 \pm 0.00248$	$10/10$ increase

Connection to split composition. An independent audit of the official partitions explains why the reliability of this component changes. The validation split contains 130 active but only 21 inactive molecules, whereas the test split contains 81 active and 71 inactive molecules. Using typed radius-2 WL-token fingerprints independently of the learned dictionary, validation negatives have a mean maximum training similarity of 0.825, compared with 0.701 for test negatives. Nevertheless, the nearest training molecule agrees with the label for only 47.6% of validation negatives, versus 87.3% of test negatives; the mean active fraction among their 15 nearest training neighbors is correspondingly 0.41 and 0.20. Validation therefore contains a small set of difficult negative decoys resembling active training molecules, whereas test negatives are more readily separated by coarse molecular composition.

The partition asymmetry is also coupled with the ordering of the original source table: all 21 validation negatives originate from a later inactive block, whereas all 71 test negatives originate from an earlier inactive block. Although source-row indices are never provided to the model, this pattern further indicates that the official splits select different chemical series and class-conditional difficulty regimes.

Taken together, these split differences explain why a structural relation supported by train and validation can transfer poorly to test. The fused representation improves validation ROC–AUC over the composition/global-statistics view by 0.0801, but decreases test ROC–AUC by 0.0352. The aryl-like atom-01 family provides a traceable occurrence-level instance of this mismatch: validation supports its use as negative evidence, whereas its label association reverses in the differently composed test partition. The exact deletion experiment establishes that this component is a reproducible negative-transfer carrier within the fixed model. Other reversed components were also detected, and neutralizing atom-01 alone does not explain the complete performance gap, indicating that the shortcut is distributed rather than attributable to a single motif. More broadly, this case shows how explicit structural coordinates can identify both a learned relation that fails to transfer and the molecular occurrences and distribution shift underlying that failure.