

组合超图中的跨尺度结构模式识别

Cross-Scale Structural Pattern Recognition in Composite Hypergraphs

参赛学生姓名：赵小博

中学：清华大学附属中学

省份：北京市

国家 / 地区：中国

指导老师姓名：甘霖、郭建超

指导老师单位：河套数学与交叉学科研究院、清华大学附属中学

Cross-Scale Structural Pattern Recognition in Composite Hypergraphs

Xiaobo Zhao¹, Lin Gan², and Jianchao Guo¹

¹Tsinghua University High School, Beijing, China

²Hetao Institute of Mathematics and Interdisciplinary Sciences, Shenzhen, China

Abstract—Hypergraphs generated by the same structural family can differ in size and local incidence structure. The challenge is to identify the generating families of an anonymous composite hypergraph when its component sizes exceed those used for fitting. Exact hypergraph Weisfeiler-Lehman (WL) kernels match local descriptions through discrete labels, without measuring structural similarity between unequal descriptions. We propose a kernel framework that combines exact hypergraph WL counts with a typed adaptation of generalized WL tree comparison. The relaxed branch compares vertex- and hyperedge-rooted computation trees, maps them to fitted representatives, and records normalized frequencies. Exact counts preserve structural distinctions and multiplicity, while relaxed features allow nonidentical local structures to share coordinates. Because their relative usefulness may differ across families, validation selects an exact, relaxed, or mixed kernel for each label, using linear support vector machines. Under fixed fitted representations, disjoint replication multiplies exact counts but preserves relaxed frequencies, revealing the absolute multiplicity discarded by normalization. We also prove that the restricted tree distance is a metric and that the resulting kernels are isomorphism-invariant and positive semidefinite. Experiments on ten generating families assess recognition beyond the final fitting ranges. A variant selecting one kernel for all labels improves recognition of large isolated instances after fitting only small ones. When all three composite components exceed the fitted size ranges, label-wise selection achieves a mean macro-F1 of 86.78% and reduces label errors by 16.8% relative to the exact-count baseline, although complete-set accuracy remains lower. These results support combining precise structural distinctions with shared local descriptions for cross-scale pattern recognition.

Index Terms—Hypergraphs, structural pattern recognition, size generalization, Weisfeiler-Lehman kernels, kernel selection.

I. INTRODUCTION

Cross-scale structural pattern recognition asks which structural families contributed to a composite hypergraph when its components are larger than those encountered during learning. A graph edge connects two vertices, whereas a hyperedge can connect more than two vertices and represent a group relation (Fig. 1). The difficulty is that changes in component size can alter local structures while leaving the generating families unchanged. For example, composites formed from a flower, a cycle, and a lattice may have different numbers of vertices and hyperedges but the same three family labels (Fig. 2). The classifier receives only the anonymous hypergraph; component memberships and the generating recipe are unavailable. Each structural pattern denotes a generating family, and its output label records whether that family contributed to the composite.

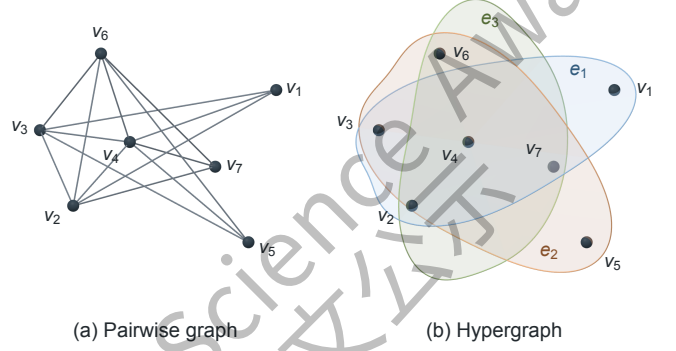


Fig. 1. Pairwise and hypergraph representations of the same seven vertices. In (a), an unweighted edge joins vertices belonging to a common hyperedge. In (b), the colored sheets represent $e_1 = \{v_1, v_2, v_3, v_4\}$, $e_2 = \{v_3, v_4, v_5, v_6\}$, and $e_3 = \{v_2, v_4, v_6, v_7\}$; v_4 belongs to all three. Both panels use the same coordinates and viewpoint. The spatial layout is illustrative; projected overlap alone does not indicate membership.

Two hypergraphs are isomorphic if a relabeling maps one to the other while preserving vertex-hyperedge incidences. Instances of the same structural pattern can nevertheless be non-isomorphic, as with cycles containing different numbers of vertices. The Weisfeiler-Lehman (WL) subtree kernel represents a graph by counts of labels obtained through iterative neighborhood aggregation [1]. The hypergraph WL test of Feng et al. extends this refinement to vertices and hyperedges [2]. Its labels describe local structures and provide the basis for hypergraph kernels. Exact matching treats unequal labels as distinct features without quantifying the similarity of their underlying structures. Changes in hyperedge cardinality or local connectivity can produce new labels even when the generating family is unchanged. Related work on graph neural networks has identified changes in local structures across graph sizes as a challenge to size generalization [3]. For the present task, the question is how different local descriptions can share features that remain useful for recognizing the same family at a larger size.

Generalized WL kernels compare the unfolding trees associated with labels and group similar structures [4]. This provides a way to share features between local structures with different exact labels. A separate choice is whether to represent each local structure by its number of occurrences or its relative frequency. Raw counts retain absolute multiplicity, which may carry information about the generating family; frequencies

discard it. Different output labels may therefore benefit from different readouts.

We form an exact hypergraph WL representation by counting vertex and hyperedge labels, and complement it with a relaxed representation that compares local computation trees. The comparison preserves vertex and hyperedge types, and the representation records the frequencies of groups of similar trees. For each generating family, we use validation data to select an exact, relaxed, or mixed kernel for a linear support vector machine (SVM). We evaluate this label-wise method on composite hypergraphs built from ten generating families. In the all-scale evaluation, where all three components take larger structural sizes at test time, the method achieves a mean macro-F1 of 86.78%, exceeding the exact-count baseline and the evaluated neural baselines. Its complete-set accuracy exceeds that of HGNN+ [5] by 26.20 percentage points.

The contributions of this work are as follows:

- We develop a label-wise kernel selection method for cross-scale structural pattern recognition in composite hypergraphs, combining exact hypergraph WL counts with a typed adaptation of generalized WL similarity.
- We characterize local label propagation and the response of counts and frequencies to disjoint replication. We prove that the typed tree distance is a metric and establish isomorphism invariance and positive semidefiniteness of the kernels.
- We conduct a ten-family evaluation across structural sizes, with ablations and one-family-at-a-time comparisons to examine the contribution of each representation and the scope of label-wise selection.

II. RELATED WORK

Research on graph and hypergraph isomorphism provides a foundation for comparing structures independently of vertex numbering. Isomorphism-invariant representations assign identical descriptions to isomorphic inputs. WL color refinement constructs such descriptions from local structure; with consistent initialization and encoding, differing label counts certify non-isomorphism, while equal counts do not establish isomorphism. WL kernels use the resulting labels to measure structural similarity for classification [1], [2]. These methods connect isomorphism testing with statistical learning. In cross-scale pattern recognition, the classifier must also associate non-isomorphic instances of different sizes with the same generating family.

A. Structural Comparison of Graphs and Hypergraphs

Many graph kernels compare structures through substructure representations. Graphlet kernels count small induced subgraphs by isomorphism type [6]. Walk and path kernels compare connectivity through matching walks or path-based features [1]. WL subtree kernels count labels obtained by recursively aggregating neighboring vertex labels [1]. We build on the WL route because its hypergraph extension

provides explicit structural features [2], while the unfolding-tree interpretation of its labels supports the relaxed comparison reviewed below [4].

For hypergraphs, local structure also includes membership in hyperedges that may contain more than two vertices. Under an unweighted pairwise projection, a single three-vertex hyperedge and the three distinct pairwise edges on those vertices both yield a triangle, despite their different hyperedge memberships.

Wachman and Khardon introduced a rooted kernel for ordered hypergraphs based on walks [7]. Bai et al. constructed a hypergraph kernel by applying an isomorphism test to directed line graphs [8]. The hypergraph WL test of Feng et al. updates hyperedge labels from their incident vertices and then vertex labels from the updated incident hyperedges [2]. The resulting subtree and hyperedge kernels provide the basis for our exact representation.

Simple graphs are 2-uniform hypergraphs, with two vertices in every edge. Feng et al. show that their hypergraph WL subtree kernel agrees with the graph WL subtree kernel on simple graphs under corresponding initialization and counting conventions [2]. Colored incidence graphs provide a further connection by representing vertices and hyperedges as different node types. Böker relates hypergraph color refinement to refinement on these incidence graphs and characterizes it using homomorphisms from Berge-acyclic hypergraphs [9]. Zhang et al. analyze the discrimination power of hypergraph WL and rooted kernels from a homomorphism perspective [10].

Feng et al. evaluate their kernels separately on graph and hypergraph datasets [2]. Our synthetic task retains the ten hypergraph families used in their RHG-10 dataset. RHG-10 classifies individual instances by family; we combine instances from several families and predict which families participated in generating the composite. The shared real-world graph benchmarks are IMDB-BINARY and IMDB-MULTI from actor collaboration networks [11], MUTAG and NCI1 from molecular data [12], [13], and PROTEINS from protein structures [14]. For complementary graph-classification evaluation, we treat these graphs as 2-uniform hypergraphs and predict their original application labels.

B. Relaxed WL and Structural Similarity

Exact label matching treats unequal local descriptions as distinct features without measuring their similarity. The generalized WL graph kernel of Schulz et al. compares the unfolding trees associated with labels using a structure- and depth-preserving tree edit distance [4]. Clustering similar trees gives relaxed features in which distinct exact labels can contribute to the same coordinate. Structural smoothing provides a related approach to graph comparison, using relationships between substructures to smooth sparse or unobserved substructure statistics [15].

Other graph methods introduce similarity at different stages of structural comparison. Gradual WL slows the refinement process through neighborhood clustering while preserving the eventual WL partition under its refinement constraints [16].

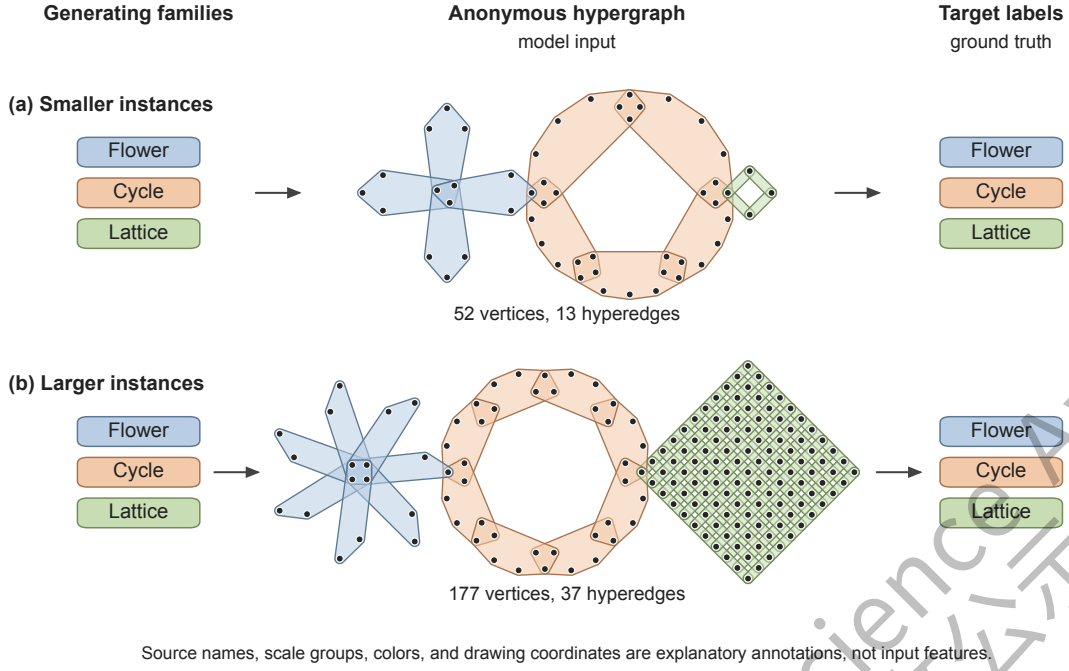


Fig. 2. Cross-scale structural pattern recognition in composite hypergraphs. The two examples are existing small-scale training and large-scale test instances formed from Flower, Cycle, and Lattice. The test example uses larger values of the three family-specific scale parameters, while the generating-family labels remain unchanged. Black dots denote vertices and colored closed curves denote hyperedges. Colors indicate the generating families for illustration only. The classifier receives the anonymous hypergraph; colors, family names, scale descriptions, drawing coordinates, and ground-truth labels are not input features. These are representative instances from different splits, not a paired intervention or a display of model predictions.

Wasserstein WL compares distributions of WL node representations at the graph level [17]. Tree Mover’s Distance and Wasserstein WL subtree distances instead use comparisons between computation or unfolding trees to define graph distances [18], [19]. Sliced Wasserstein WL kernels have also been used for Gaussian process regression on large scientific meshes [20]. These graph-level comparisons differ from grouping local structures to form explicit features. Following generalized WL, we adapt local-tree comparison to computation trees that distinguish vertices from hyperedges, obtaining explicit features for hypergraph classification.

C. Size Generalization and Structural Counts

Yehudai et al. show that shifts in local patterns can obstruct generalization by graph neural networks even when their expressive power is sufficient for the task [3]. Existing approaches address size dependence through both representations and training. Bevilacqua et al. use subgraph densities to construct approximately size-invariant representations under a specified graph-generating model [21]. SizeShiftReg regularizes representations across graph coarsenings [22], while DISGEN separates size-related and task-related representations [23]. This motivates examining which local hypergraph structures remain useful when component sizes change.

Structural classification under size shift is also studied in GOOD-Motif [24]. Each graph combines one motif with a background graph, and the label identifies the motif category; its size splits vary the background-graph size. Wasserstein

Barycenter Matching (WBM) evaluates this task using small training graphs, intermediate-size validation graphs, and larger test graphs [25]. Size generalization also arises in RNA structure evaluation, where EquiRNA predicts the root-mean-square deviation of candidate structures from native structures and evaluates transfer to longer RNAs [26]. Our task predicts the presence of multiple generating families in an anonymous hypergraph and varies the internal sizes of all components or one family at a time. Section IV specifies the sizes used for fitting, selection, and testing.

At the readout level, the issue is how a multiset of local features is aggregated. The multiset analysis underlying graph isomorphism networks shows that suitable sum representations can preserve multiplicity, whereas mean aggregation is insensitive to uniform replication [27]. Levin et al. relate size generalization to the correspondence between inputs of different sizes, showing that sum and mean are compatible with different size transformations under their respective assumptions [28]. Disjoint replication provides a theoretical comparison in which local incidence neighborhoods are preserved. The family-specific parameter changes used in our experiments can alter these neighborhoods as well as the multiplicities of local structures. Our labels indicate family presence, but counts of local structures may still help identify a generating family. We retain exact counts and relaxed frequencies as candidate representations and use label-wise validation to assess their suitability for this task.

TABLE I
MAIN NOTATION USED IN THE METHOD.

Notation	Description
$H = (V, E)$	Hypergraph with vertex set V and hyperedge set E .
q, ℓ, y_ℓ	Number of families, family index, and binary participation label.
$d(v), r(e)$	Vertex degree and hyperedge arity.
$\mathcal{F}$	Indexed collection used to fit the representation.
$h, h_{\max}$	Exact refinement round and maximum round.
$c_v^h(v), c_e^h(e)$	Vertex and hyperedge colors at round h .
$\Phi_{\text{ex}}, \phi_{\text{ex}, \mathcal{F}}$	Full exact counts and fitted exact feature map.
$P_{\mathcal{F}}$	Projection onto the fitted color vocabulary.
$\tau, X_\tau(H)$	Node type and its vertex or hyperedge set.
t, T	Tree truncation depth and maximum depth.
$S_\tau^t(x)$	Typed computation tree rooted at x at depth t .
$D_{\tau, t}$	Tree distance within one type and depth.
$w(S)$	Number of node occurrences in tree S .
$k, R_{\tau, t, \mathcal{F}}$	Representative cap and fitted representatives.
$\rho_{\tau, t, \mathcal{F}}$	Fixed nearest-representative map.
$\phi_{\text{rel}, \mathcal{F}}$	Concatenated block-frequency feature map.
bH	Disjoint union of b copies of H .
$K_{\text{ex}}, K_{\text{rel}}$	Fitted exact and relaxed inner-product kernels.
$s_{\mathcal{F}}, \lambda$	Fitted kernel scale and mixture coefficient.
C	SVM regularization parameter.

III. METHOD

We study structural pattern recognition from an anonymous composite hypergraph. Let $H = (V, E)$ be a finite simple hypergraph with nonempty vertex and hyperedge sets and nonempty hyperedges; isolated vertices are allowed. The target $y \in \{0, 1\}^q$ records participation of the q generating families: $y_\ell = 1$ when family ℓ participated. The predictor receives only the terminal hypergraph, without component memberships, interfaces, source parameters, or a generating recipe. Cross-scale evaluation changes family sizes beyond those used to fit the final model; Section IV specifies the scales at each fitting stage. These labels describe the generation process, without assuming a unique source decomposition.

Increasing a family's scale can change both its local incidence structures and their numbers of occurrences. Exact counts retain structural distinctions and multiplicity, while relaxed frequencies allow different local descriptions to share coordinates. We combine these representations and select a kernel for each label, using linear support vector machines (SVMs). Figure 3 shows the method. Throughout, $\mathcal{F}$ is the nonempty finite indexed collection used to fit the representation at the current stage. Its vocabulary, representatives, and kernel scale remain fixed when transforming other graphs. Table I gives the main notation; Appendix A-A supplies additional definitions and deterministic fitting rules.

A. Exact Hypergraph WL Representation

We retain exact local descriptions as a reference for cross-scale recognition. We use the edge-first hypergraph WL refinement of Feng et al. [2], with vertex degree $d(v) = |\{e \in E : v \in e\}|$ and hyperedge arity $r(e) = |e|$ as structural initial

labels. Write enc_0 and enc_1 for injective canonical encodings with disjoint initialization and refinement namespaces. Vertex and hyperedge type tags are v and e . Initialization and round $h \geq 1$ are

$$\begin{aligned}
 c_v^0(v) &= \text{enc}_0(v, d(v)), \\
 c_e^0(e) &= \text{enc}_0(e, r(e)), \\
 c_e^h(e) &= \text{enc}_1(e, c_e^{h-1}(e), \\
 &\quad \{\{c_v^{h-1}(v) : v \in e\}\}), \\
 c_v^h(v) &= \text{enc}_1(v, c_v^{h-1}(v), \\
 &\quad \{\{c_e^h(e) : e \ni v\}\}).
 \end{aligned} \tag{3.1}$$

Here $\{\{\cdot\}\}$ denotes a multiset; vertices use the newly computed edge colors. For $h_{\max} \in \mathbb{N}_0$, let $\Phi_{\text{ex}}(H)$ count every vertex and hyperedge color occurrence over rounds $0, \dots, h_{\max}$. The fitted feature map is $\phi_{\text{ex}, \mathcal{F}} = P_{\mathcal{F}}\Phi_{\text{ex}}$, where $P_{\mathcal{F}}$ retains fitting colors in canonical order; unseen colors have no coordinate. Appendix A-C proves the projection identities needed for SVM prediction.

Adding a new degree-one vertex to an existing edge changes its arity but preserves the old endpoints' degrees. Their round-one colors nevertheless change (Proposition A.1, Appendix A-C). The relaxed branch compares such different descriptions; a color change alone does not imply a classification error.

B. Typed Relaxed Representation

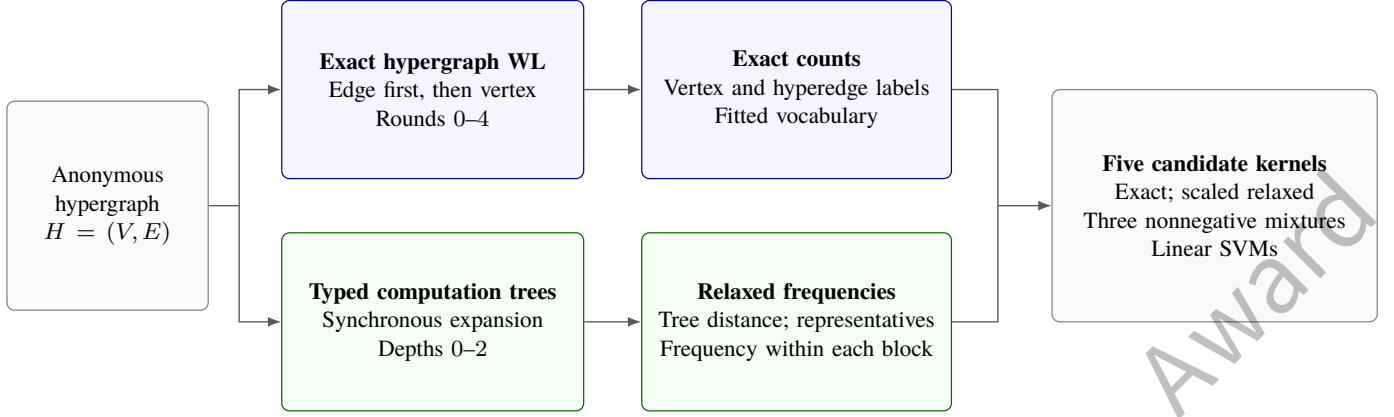
The relaxed branch compares local structures even when their exact descriptions differ. We adapt the structure- and depth-preserving tree comparison and structural grouping of generalized WL kernels [4] to typed incidence computation trees. Let $\tau \in \{v, e\}$, $X_v(H) = V$, $X_e(H) = E$, and $\bar{\tau}$ be the opposite type. Write $x \sim z$ for incidence adjacency and set $a_v(v) = d(v)$, $a_e(e) = r(e)$. For $T \in \mathbb{N}_0$, define

$$\begin{aligned}
 S_\tau^0(x) &= (\tau, 0, a_\tau(x), \emptyset), \\
 S_\tau^t(x) &= (\tau, t, a_\tau(x), \{\{S_{\bar{\tau}}^{t-1}(z) : z \sim x\}\}), \\
 &\quad t = 1, \dots, T.
 \end{aligned} \tag{3.2}$$

Both types use the preceding depth synchronously, with fixed root labels, unlike the edge-first recursion (3.1). Signatures represent unordered trees of alternating vertex and hyperedge occurrences; a node may recur along different walks. Truncation depth t need not equal tree height: isolated vertices have no children even for $t > 0$.

We compare two signatures by matching their child subtrees, deleting or inserting each unmatched subtree in full. Comparisons are restricted to the same type and truncation depth. Let a_S be the root label of S and let $w(S)$ be the total number of nodes in the represented tree, counting occurrences. At depth zero, set $D_{\tau, 0}(S, T') = \mathbf{1}\{a_S \neq a_{T'}\}$. At positive depth, enumerate the child occurrences as $(S_i)_{i=1}^p$ and $(T'_j)_{j=1}^r$. With $\mathcal{M}_{p, r}$ the set of partial one-to-one matchings between their index sets, define

(a) Representations and candidate kernels



Vocabulary, representatives, and kernel scale are fixed within each fitting stage.

(b) Selection and prediction: composite all-scale protocol

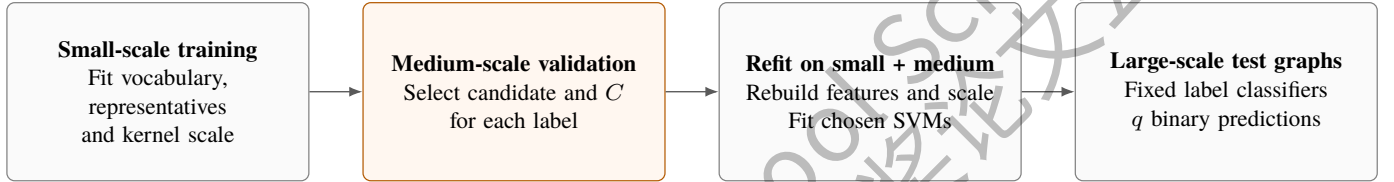


Fig. 3. Method construction and use. (a) An anonymous hypergraph supplies an edge-first exact count representation and a synchronous typed-tree representation. Restricted tree distances, fitted representatives, and block frequencies construct the relaxed features. Their inner products form five candidate kernels. (b) In the composite all-scale protocol, small-scale training graphs determine the representations and kernel scale used for selection on medium-scale validation graphs. Each label’s candidate identity and C are then fixed, and the representations, kernel scale, and classifiers are refitted on small plus medium graphs. Large-scale test graphs use the resulting fixed classifiers.

$$\begin{aligned}
 D_{\tau,t}(S, T') &= \mathbf{1}\{a_S \neq a_{T'}\} \\
 &+ \min_{M \in \mathcal{M}_{p,r}} \left[\sum_{(i,j) \in M} D_{\tau,t-1}(S_i, T'_j) \right. \\
 &\quad \left. + \sum_{i \notin \text{dom } M} w(S_i) + \sum_{j \notin \text{ran } M} w(T'_j) \right]. \quad (3.3)
 \end{aligned}$$

Thus unmatched subtrees are charged in full. Appendix A-B gives the equivalent augmented assignment.

Proposition 3.1 (Typed tree metric). *For each fixed (τ, t) , $D_{\tau,t}$ is a metric on the space of finite recursive typed signatures defined in Appendix A-A, hence on its hypergraph-realized subset. Zero distance means identical signatures. Appendix A-B gives the proof.*

For each type/depth block, deterministic farthest-first traversal [29] selects at most $k \in \mathbb{N}_{>0}$ representatives $R_{\tau,t,\mathcal{F}} = (R_1, \dots, R_{k_{\tau,t}})$ from observed fitting signatures, starting with the most frequent. Let $\rho_{\tau,t,\mathcal{F}}(S)$ index a nearest representative

under $D_{\tau,t}$. Both steps use canonical tie rules, specified in Appendix A-A. Every block is nonempty. Define

$$\begin{aligned}
 \phi_{\text{rel},\mathcal{F};\tau,t,j}(H) &= \frac{1}{|X_\tau(H)|} \\
 &\cdot \sum_{x \in X_\tau(H)} \mathbf{1}\{\rho_{\tau,t,\mathcal{F}}(S_\tau^t(x)) = j\}. \quad (3.4)
 \end{aligned}$$

Concatenating all blocks gives $\phi_{\text{rel},\mathcal{F}}(H)$, with unit sum per block. Representative mapping supplies shared structural coordinates; nearby trees need not receive the same coordinate. Normalization removes uniform multiplicity: under fixed fitted objects, b disjoint copies preserve the relaxed vector and multiply exact counts by b (Proposition A.2, Appendix A-C). This result does not cover internal family expansion or interface changes.

C. Kernel Construction and Selection

Exact counts and relaxed frequencies preserve different information about structure and multiplicity. We therefore include both representations and their mixtures as candidates, allowing validation data to determine their use. At each fitting

Algorithm 1 Label-wise selection for composite hypergraphs

Input: Nonempty labeled training and validation collections; every training label has both classes. Anonymous evaluation graphs.

Configuration: $h_{\max} = 4$, $T = 2$, $k = 8$ per block; candidates in (3.5); $C \in \{0.01, 0.1, 1, 10, 100\}$.

- 1: Fit the exact vocabulary and typed-tree representatives on training graphs only.
- 2: Transform training and validation graphs with these fixed objects. Compute training–training and validation–training Grams for both representations, and obtain $s_{\mathcal{F}}$ from training diagonals.
- 3: For each label, candidate, and C , fit a weighted binary SVM on training labels and score unweighted validation F1. Choose the largest F1, breaking exact ties by candidate order, then smaller C . Undefined F1 is zero.
- 4: Fix the candidate identity and C for each label.
- 5: Refit vocabulary, representatives, and scale on training plus validation, retaining observation multiplicities. Recompute sample weights on the combined labels and fit the selected classifiers.
- 6: Transform evaluation graphs with the refitted objects. Predict a label when its decision score is at least zero.

No threshold search is performed. Evaluation labels are excluded from fitting and selection.

Output: Fitted representations, per-label candidate identities and C values, classifiers, and predictions.

stage, let K_{ex} and K_{rel} be the inner products of the two fitted representations, suppressing their $\mathcal{F}$ subscript here. Set $s = s_{\mathcal{F}}$ to the median exact self-kernel on $\mathcal{F}$ divided by the median relaxed self-kernel on $\mathcal{F}$; both medians are positive. Define the mixed kernels and ordered candidate family by

$$\begin{aligned} K_{\lambda} &= K_{\text{ex}} + \lambda s K_{\text{rel}}, & \lambda \in \{0.1, 1, 10\}, \\ K_{\mathcal{F}} &= (K_{\text{ex}}, K_{0.1}, K_1, K_{10}, s K_{\text{rel}}). \end{aligned} \quad (3.5)$$

The mixed kernel multiplies relaxed features by $\sqrt{\lambda s}$. All five candidates are explicit feature-space inner products used with linear soft-margin SVMs through precomputed Gram matrices [30].

Our Global variant selects one kernel and one C for all labels, using validation macro-F1 and the four candidates consisting of Exact and the three mixtures. Experts selects separately for each label from all five candidates in (3.5). The restricted Label-wise mixture variant also selects separately, but excludes the pure relaxed endpoint. These variants compare shared and label-wise selection within the same representation construction.

Algorithm 1 selects a candidate and C for each composite-task label by unweighted validation F1, then refits on training plus validation. Every fitting label has both classes. Binary SVM weights balance their total weights, with mean sample weight one, and are recomputed from the current fitting labels. Section IV gives the separate cross-validation protocol for isolated families and multiclass protocol for real graphs.

Proposition 3.2 (Invariance, kernel validity, and endpoints). *For fixed fitted objects and canonical tie rules, both feature maps are invariant to vertex and hyperedge reindexing. Their kernels and nonnegative mixtures are positive semidefinite (PSD). Within the same fitting stage, the endpoints recover*

the exact baseline Gram and $s_{\mathcal{F}}$ times the relaxed baseline Gram. Appendix A-C proves these claims directly from the feature maps, independently of tree metricity.

Appendix A-C gives brief computational bounds; dense fitting-Gram storage remains quadratic in the number of fitting graphs.

IV. EXPERIMENTS AND DISCUSSION

We evaluate whether exact and relaxed representations support recognition of generating families at unseen structural sizes. We first examine isolated instances, for which one predictor fitted on small-scale data is evaluated at two larger scales. We then turn to the main task of recognizing several families in anonymous composite hypergraphs, comparing shared and label-wise selection. Real-world graph classification provides a separate evaluation with semantic labels.

A. Evaluation Setup

The method in Section III preserves exact structural counts while allowing local descriptions to share relaxed coordinates. To test these choices under scale changes, we define scale through each family’s natural construction parameters. The cross-scale tests use parameter ranges beyond those available when fitting the final predictor, while retaining the same family labels.

Data and task. We use the ten synthetic hypergraph families introduced by Feng et al. [2] (Fig. 4), with small, medium, and large ranges of their natural construction parameters. A composite contains instances from three distinct families, joined by two vertex identifications of the form illustrated in Fig. 5; no connecting hyperedge is added. The final vertices are randomly reindexed. The classifier receives only the resulting hypergraph and predicts ten binary labels indicating which families contributed. Component memberships, interfaces, construction parameters, and the generating recipe are unavailable to the classifier. Each isolated instance has one positive label and each composite has three, but predictions are made independently without an argmax or top-three correction. Appendix B-A lists the families, parameter ranges, sample counts, and checks for duplicate or ambiguous structures.

Scale conditions. The isolated-family comparison uses a source-only protocol. We select configurations from concatenated out-of-fold predictions within 970 small-scale instances, then refit on all 970 instances. The 340 medium-scale and 656 large-scale instances are used only for evaluation. Its SVM weighting assigns half the total weight to the positive family and divides the other half equally among the remaining nine families, as specified in Appendix B-B; it differs from the binary class balancing used for composites.

For composites, Table II distinguishes small-scale IID evaluation, simultaneous growth of all three components (all-scale), and one-family-at-a-time growth (OFAT). For each generation seed, the conditions share 720 small-scale training records and each has 360 validation and 360 test records. The data cover all 120 combinations of three families. We use three generation seeds. For Experts, Algorithm 1 is applied separately to each

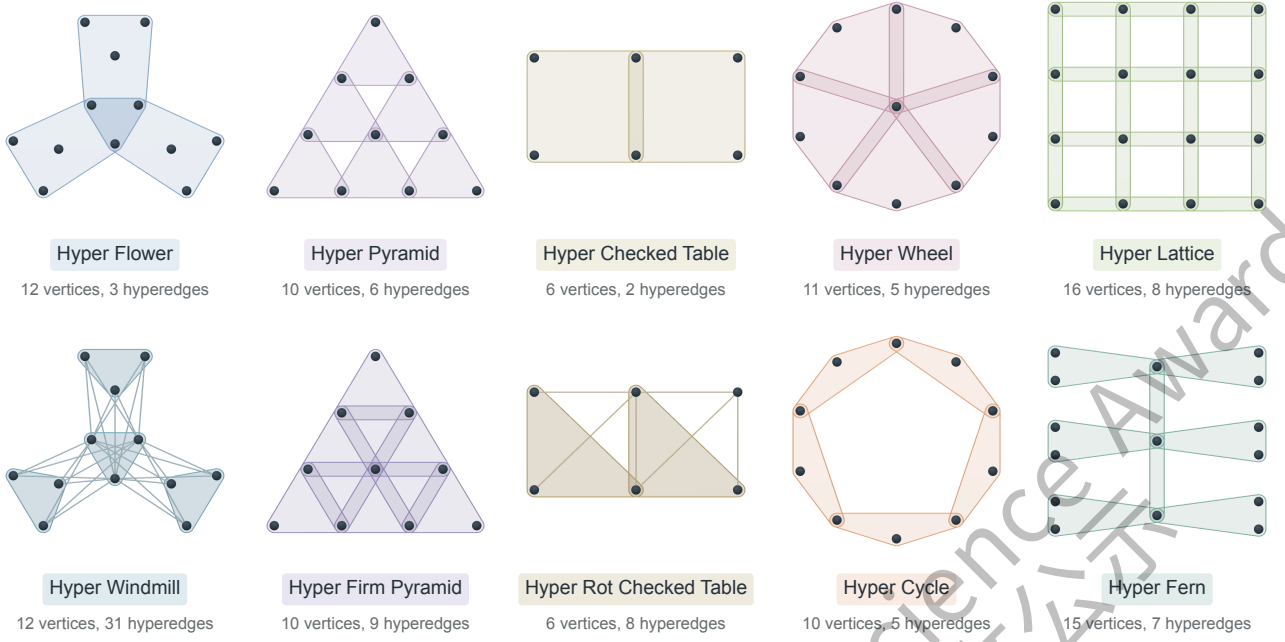


Fig. 4. Representative instances of the ten generating families of Feng et al. [2], with names following their Table 6. Spheres denote vertices; tubes denote two-vertex hyperedges, and shaded surfaces denote larger hyperedges. The Hyper Windmill example has four separate group hyperedges and pairwise connections between central and peripheral vertices. Each cell of Hyper Rot Checked Table contains one three-vertex hyperedge and three two-vertex hyperedges. In Hyper Lattice, each complete row and column forms a hyperedge. Counts describe the displayed instances; coordinates, colors, and panel scales are illustrative.

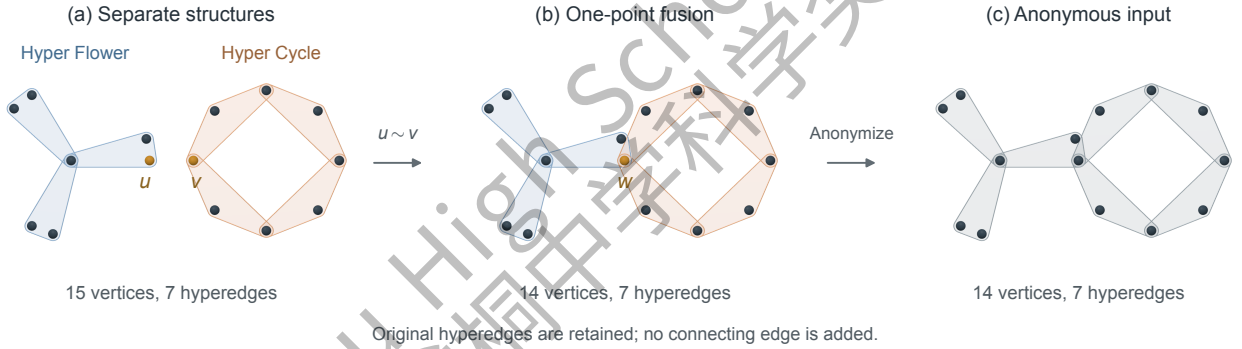


Fig. 5. One local step of composite-hypergraph construction. (a) Hyper Flower and Hyper Cycle are initially disjoint. (b) The designated vertices u and v are identified as w , reducing the vertex count from 15 to 14. All seven hyperedges are retained; no connecting edge is added. (c) Vertex identifiers are permuted, and source colors and interface marks are withheld from the predictor. The drawing layout remains fixed to show the unchanged incidences. The benchmark applies two such steps to three distinct families.

condition, including validation selection and final refitting. Thus, all-scale and OFAT final models include medium-scale supervision before evaluation at larger sizes. Scores across conditions compare separately selected models. The paired OFAT diagnostics and their reference protocol are summarized in Appendix B-C.

Compared methods. Exact is the inherited hypergraph WL count baseline [2]. Relaxed uses our typed adaptation of generalized WL tree comparison [4] alone. Relaxed, Global, Label-wise mixture, and Experts are our adaptations and variants, marked [†] in the tables. Within the construction in Section III-C, Global uses shared kernel and C selection, but

each label still has its own SVM. Label-wise mixture excludes the pure relaxed candidate from the full Experts method. These comparisons examine the relaxed branch, shared selection, and the additional relaxed endpoint. The representation settings follow Algorithm 1.

A 73-dimensional structural-statistics representation (Stats-73), HGNN [31], and HGNN+ [5] provide additional comparisons on the same splits. Kernel and statistics methods use linear SVMs. Neural inputs, loss weights, and training budgets are reported separately in Appendix B-B.

Evaluation measures. Macro-F1 averages F1 over the ten labels; exact-match measures the fraction of completely correct

TABLE II

SCALE CONDITIONS FOR THE COMPOSITE TASK. EVERY CONDITION STARTS FROM THE SAME SMALL-SCALE TRAINING SET FOR A GIVEN GENERATION SEED. FINAL FITTING USES TRAINING PLUS THE INDICATED VALIDATION SET AND REFITS THE REPRESENTATIONS, KERNEL SCALE, AND CLASSIFIERS.

Condition	Initial training	Validation	Test
Small-IID	All components small	All components small	All components small
All-scale	All components small	All components medium	All components large
OFAT	All components small	Target component medium	Target component large

label sets; Hamming loss measures the fraction of incorrect binary labels. Composite results report means and sample standard deviations over the three generation seeds. Family-level summaries appear in Appendix B-C. The synthetic experiments are development evaluations on the retained parameter grids. Generation seeds may share structures, so pooled prediction counts are used descriptively rather than as independent samples for significance tests.

Together, these protocols test the two abilities needed for cross-scale pattern recognition: reusing family-level structural information beyond the fitting range, and recognizing several contributing families in an anonymous composite. The isolated-family test examines the first ability under small-scale supervision; the composite tests examine recognition under simultaneous or individual component growth.

B. Cross-Scale Recognition of Isolated Families

Under the source-only protocol above, we first examine scale transfer without composition. For each kernel variant, one predictor selected and fitted entirely on small instances is evaluated on both the medium and large sets without further fitting. This setting tests whether the learned structural representation remains useful beyond its fitting range before several families are combined.

Table III and Fig. 6 show two observations. First, combining representations substantially improves recognition on the large unseen-size set. On the large set, Global increases macro-F1 from 79.77% for Exact to 96.61%, with exact-match increasing from 90.70% to 94.36%. It corrects 35 instances misclassified by Exact while introducing 11 new complete-set errors, a net gain of 24 correct instances. The 16.84-percentage-point macro-F1 gain is especially relevant because this measure gives each family equal weight despite unequal test supports. Relaxed alone has lower macro-F1 than Exact on both sets. The combined variant therefore outperforms either representation alone in this setting. It retains exact distinctions and multiplicity while also allowing different local structures to contribute to shared relaxed coordinates.

Second, label-wise selection provides an additional gain on the medium set. Experts gives the highest medium-set macro-F1, 95.12%, compared with 88.71% for Global and 87.65% for Exact. Its medium-set exact-match is 94.71%, however, below the 96.47% of Label-wise mixture. On the large set, Global leads the kernel variants on all three reported measures, while Experts has a macro-F1 of 80.50%. The two sets contain different instances and family supports, so this change in ranking does not describe a continuous size-growth trajectory.

The principal result is that combining exact and relaxed descriptions recovers family recognition beyond the small-scale fitting range. Global provides the strongest large-set performance, and Experts improves family-balanced recognition on the medium set. These results support keeping both representations available when transferring structural information to unseen sizes.

C. Cross-Scale Recognition of Composite Hypergraphs

We now test these combinations when several families occur within one anonymous hypergraph at unseen component sizes. This main experiment examines the value of combining representations and the additional choices made available to each label.

Under all-scale growth, Experts achieves the highest macro-F1 and the lowest Hamming loss among the evaluated methods (Table IV). Compared with Global, macro-F1 increases from 84.68% to 86.78%, while Hamming loss decreases from 9.99% to 8.43%. Both measures also improve over Exact. The gains accompany lower exact-match: 42.69%, compared with 44.63% for Global and 43.98% for Exact. Against the evaluated HGNN+ configuration, Experts improves macro-F1 from 72.95% to 86.78% and exact-match by 26.20 percentage points.

The restricted variants connect this result to the available representations. Relaxed alone reaches 77.39% macro-F1 under all-scale growth, below Exact, whereas Label-wise mixture reaches 84.98%. Adding the pure relaxed endpoint raises macro-F1 to 86.78% and reduces Hamming loss by 1.31 percentage points. Allowing some labels to use the relaxed representation alone therefore has empirical value in this setting. Global and Experts differ in both shared versus label-wise parameter selection and the availability of that endpoint; the four-candidate label-wise variant isolates the latter choice.

The other scale conditions distinguish the value of representation mixing from the additional flexibility of full label-wise selection. On small-scale IID data, Label-wise mixture gives the best means on all three measures, and Global slightly exceeds Experts. Under OFAT, Global gives the highest macro-F1, 86.17%, and the lowest Hamming loss, 8.26%; Exact retains the highest exact-match. Shared mixing therefore remains useful when only one component grows, while the clearest gain from full label-wise selection occurs when all three components grow together.

The aggregate errors explain why better label measures need not improve complete-set recognition. Across 1,080 all-scale test prediction records, Experts reduces the number of

TABLE III

ISOLATED-FAMILY RECOGNITION AT UNSEEN SIZES (%). FOR EACH KERNEL VARIANT, THE SAME PREDICTOR IS SELECTED AND FITTED ON 970 SMALL-SCALE INSTANCES AND EVALUATED ON BOTH SETS WITHOUT FURTHER FITTING. RESULTS ARE DETERMINISTIC. [†] DENOTES OUR ADAPTATIONS AND VARIANTS. EXPERTS IS THE FULL METHOD.

Method	Medium (340 graphs)			Large (656 graphs)		
	Macro-F1 $\uparrow$	Exact-match $\uparrow$	Hamming $\downarrow$	Macro-F1 $\uparrow$	Exact-match $\uparrow$	Hamming $\downarrow$
Exact	87.65	95.00	0.56	79.77	90.70	0.98
Relaxed [†]	73.40	64.41	4.09	59.46	60.98	4.54
Global [†]	88.71	96.18	0.47	96.61	94.36	0.59
Label-wise mixture [†]	89.26	96.47	0.41	84.85	93.90	0.66
Experts [†]	95.12	94.71	0.56	80.50	89.94	1.14

Medium and large are different instance sets. Pyramid, Firm Pyramid, and Lattice each have only two instances in each set. Counts for all ten families appear in Table VI.

TABLE IV

COMPOSITE RECOGNITION (%). MEANS $\pm$ SAMPLE STANDARD DEVIATIONS OVER THREE GENERATION SEEDS, WITH 360 TEST RECORDS PER SEED AND CONDITION. BOLD DENOTES THE BEST MEAN IN A COLUMN, NOT STATISTICAL SIGNIFICANCE. NEURAL VARIABILITY INCLUDES INITIALIZATION AS WELL AS DATA GENERATION. EACH CONDITION HAS ITS OWN SELECTED AND REFITTED MODELS. [†] DENOTES OUR ADAPTATIONS AND VARIANTS. EXPERTS IS THE FULL METHOD.

Method	All-scale			Small-IID			OFAT		
	Macro-F1 $\uparrow$	Exact-match $\uparrow$	Hamming $\downarrow$	Macro-F1 $\uparrow$	Exact-match $\uparrow$	Hamming $\downarrow$	Macro-F1 $\uparrow$	Exact-match $\uparrow$	Hamming $\downarrow$
Exact	84.50 $\pm$ 1.10	43.98 $\pm$ 0.42	10.13 $\pm$ 0.30	91.95 $\pm$ 0.66	63.06 $\pm$ 2.42	4.56 $\pm$ 0.34	86.05 $\pm$ 0.65	45.65 $\pm$ 2.78	8.36 $\pm$ 0.47
Relaxed [†]	77.39 $\pm$ 0.20	18.52 $\pm$ 2.49	13.93 $\pm$ 0.32	73.28 $\pm$ 0.43	13.98 $\pm$ 1.37	17.23 $\pm$ 0.50	68.96 $\pm$ 1.01	10.09 $\pm$ 1.79	20.05 $\pm$ 0.99
Global [†]	84.68 $\pm$ 1.38	44.63 $\pm$ 1.53	9.99 $\pm$ 0.53	92.03 $\pm$ 0.54	62.96 $\pm$ 2.84	4.53 $\pm$ 0.30	86.17 $\pm$ 0.37	45.46 $\pm$ 3.83	8.26 $\pm$ 0.32
Label-wise mixture [†]	84.98 $\pm$ 1.20	46.11 $\pm$ 0.73	9.74 $\pm$ 0.30	92.09 $\pm$ 0.90	63.61 $\pm$ 3.63	4.49 $\pm$ 0.49	85.86 $\pm$ 0.35	45.09 $\pm$ 3.58	8.43 $\pm$ 0.30
Experts [†]	86.78 $\pm$ 0.73	42.69 $\pm$ 0.32	8.43 $\pm$ 0.40	91.84 $\pm$ 0.66	62.87 $\pm$ 3.05	4.63 $\pm$ 0.35	85.84 $\pm$ 0.32	45.09 $\pm$ 3.58	8.40 $\pm$ 0.35
Stats-73	76.32 $\pm$ 1.70	20.19 $\pm$ 1.85	16.77 $\pm$ 1.45	77.87 $\pm$ 0.25	22.59 $\pm$ 1.31	14.48 $\pm$ 0.14	72.29 $\pm$ 1.74	16.02 $\pm$ 2.31	18.22 $\pm$ 1.29
HGNN	72.58 $\pm$ 0.91	13.98 $\pm$ 2.16	16.97 $\pm$ 0.90	67.82 $\pm$ 0.82	7.78 $\pm$ 1.21	22.21 $\pm$ 1.12	65.85 $\pm$ 1.43	7.13 $\pm$ 1.43	22.56 $\pm$ 1.35
HGNN+	72.95 $\pm$ 3.18	16.48 $\pm$ 2.51	17.09 $\pm$ 2.13	68.06 $\pm$ 0.81	7.13 $\pm$ 1.25	22.12 $\pm$ 1.34	66.02 $\pm$ 1.09	7.04 $\pm$ 2.26	21.79 $\pm$ 0.39

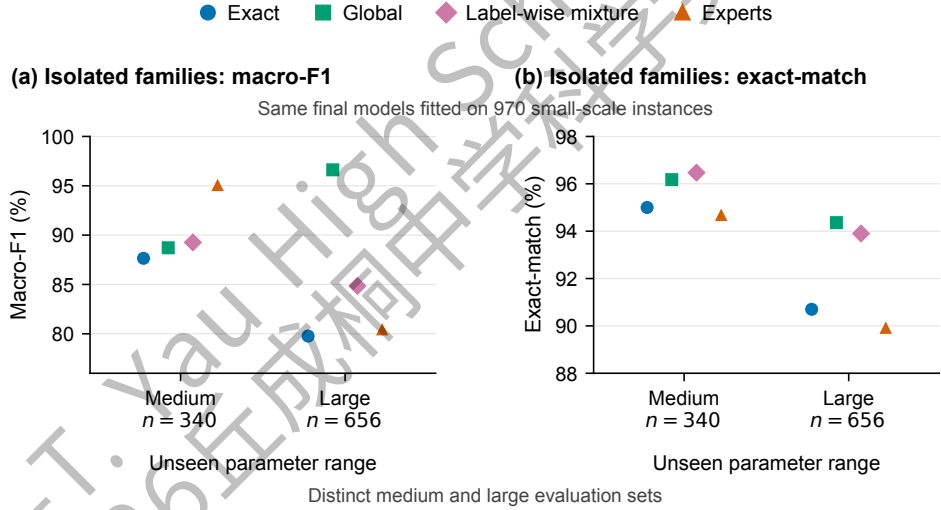


Fig. 6. Cross-scale recognition of isolated families: (a) macro-F1 and (b) exact-match. For each kernel variant, the same predictor fitted on 970 small-scale instances is evaluated on the medium (340) and large (656) sets. The two sets contain different instances; points are not a paired growth trajectory, and deterministic results have no repeat-based error bars. Full results appear in Table III.

incorrect labels from 1,094 for Exact to 910, a reduction of 16.8%. Records with at least three incorrect labels decrease from 131 to 51, while completely correct records decrease from 475 to 461. Thus, fewer graphs incur three or more label errors. It makes fewer errors per incorrect graph on average, although more graphs retain at least one error. This distribution accounts for the Hamming and exact-match trade-off; macro-

F1 also depends on which family labels incur those errors.

The main contribution of Experts is thus improved source-label recognition under simultaneous component growth. Relative to shared mixing, access to label-specific representations improves macro-F1 and reduces label error; relative to Exact, it also reduces the frequency of records with several incorrect labels. This is the setting in which the additional selection

TABLE V
REAL-WORLD GRAPH ACCURACY (%). MEANS $\pm$ SAMPLE STANDARD DEVIATIONS
ACROSS FIVE WL-GROUPED FOLDS. THESE FOLDS ARE NOT INDEPENDENT
REPETITIONS. GLOBAL USES MULTICLASS SELECTION WITHOUT A
TRAIN-PLUS-VALIDATION REFIT. $\dagger$ DENOTES OUR ADAPTATIONS AND VARIANTS.

Dataset	Exact	Relaxed $\dagger$	Global $\dagger$
IMDB-BINARY	66.28 $\pm$ 5.87	66.51 $\pm$ 1.91	66.38 $\pm$ 5.95
IMDB-MULTI	43.03 $\pm$ 9.52	39.17 $\pm$ 7.12	45.90 $\pm$ 5.85
MUTAG	85.09 $\pm$ 4.52	81.17 $\pm$ 7.27	85.09 $\pm$ 4.52
NCI1	79.17 $\pm$ 1.07	61.24 $\pm$ 2.42	79.15 $\pm$ 1.24
PROTEINS	70.26 $\pm$ 2.38	63.79 $\pm$ 4.65	68.56 $\pm$ 3.72

flexibility most directly benefits the paper’s cross-scale recognition task.

D. Evaluation on Real-World Graphs

The preceding experiments establish the value of combining representations for generating-family recognition at unseen sizes. We now examine how this construction relates to the structural properties of real-world graph datasets. We use IMDB-BINARY, IMDB-MULTI, MUTAG, NCI1, and PROTEINS [11]–[14], comprising 7,911 graphs with their original semantic labels.

The two IMDB datasets contain actor-collaboration graphs with recurring dense structures. In particular, 789 of 1,500 IMDB-MULTI records (52.6%) are complete graphs, compared with 139 of 1,000 IMDB-BINARY records (13.9%). Both contain complete graphs of different sizes. These graphs share a connectivity rule while vertex degrees and structural counts vary with size, providing a concrete reason to examine similarity-based descriptions alongside exact counts. This comparison concerns whole-graph topologies; the corresponding data counts are detailed in Appendix B-A.

Only topology is used: edges are treated as two-vertex hyperedges, and original node attributes are excluded. The targets are mutually exclusive classes, so this experiment uses our Global multiclass adaptation. All methods share five outer folds grouped by graph WL fingerprints, with an inner validation split; there is no imposed size split. The SVMs use no sample reweighting. Validation accuracy determines selection, and selected models are retained without refitting on training plus validation. The ten-label Experts method is not evaluated here. Appendix B-B gives the selection details.

Global achieves the highest mean accuracy among the seven evaluated methods on IMDB-MULTI, increasing Exact’s 43.03% to 45.90% (Table V; full comparison in Appendix B-C). The structural-group analysis identifies where this improvement occurs. A complete-graph topology, K_7 , accounts for 307 records. WL grouping places all of them in one test fold, so their predictor is fitted on other structural groups. Global makes 60 more correct predictions than Exact in this held-out group. The observed gain therefore has a concrete structural location: combining the representations improves the class assignment for a frequent topology absent from that fold’s fitting data. Gains are concentrated in this group; the aggregate prediction decomposition appears in Appendix B-C.

The remaining datasets clarify the role of the exact branch. Global stays close to Exact on IMDB-BINARY, MUTAG, and NCI1, while accuracy decreases on PROTEINS. Relaxed alone is weaker on MUTAG and NCI1, indicating the predictive value of the exact descriptions retained by the combination. The IMDB-MULTI improvement shows the additional value that joint exact and relaxed features can provide when classifying a held-out repeated structure. Together with the synthetic results, it supports combining structural distinctions with similarity-based descriptions across different classification settings.

V. CONCLUSION

In this paper, we developed a kernel framework for recognizing the generating families of anonymous composite hypergraphs at unseen structural sizes. To compare local structures whose exact WL labels change with size, we adapted generalized WL tree comparison to vertex- and hyperedge-typed computation trees. We paired the resulting relaxed frequencies with exact hypergraph WL counts, and introduced selection among exact, relaxed, and mixed kernels for each generating family. This construction lets different families use different combinations of structural distinctions, multiplicity, and local similarity within a linear SVM framework. Our analysis makes the information trade-off explicit: under fixed fitted representations, disjoint replication preserves relaxed frequencies but multiplies exact counts. Replication invariance thus discards absolute multiplicity, motivating the retention of both readouts. We also establish the metric property of the typed tree distance and construct isomorphism-invariant positive semidefinite kernels.

Experiments on ten generating families demonstrate the value of this framework beyond the fitting range. Our shared-selection variant, Global, gives the strongest large-scale isolated-family results using only small-scale fitting data. When all three components of a composite grow, Experts achieves the highest macro-F1 and lowest Hamming loss among the evaluated methods, although complete-set accuracy remains below that of Exact. These results show that preserving exact structural information and sharing features across nonidentical local structures can jointly support cross-scale recognition.

REFERENCES

- [1] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman Graph Kernels," *Journal of Machine Learning Research*, vol. 12, no. 77, pp. 2539–2561, 2011. [Online]. Available: <https://www.jmlr.org/papers/v12/shervashidze11a.html>
- [2] Y. Feng, J. Han, S. Ying, and Y. Gao, "Hypergraph Isomorphism Computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 3880–3896, 2024. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/38215323/>
- [3] G. Yehudai, E. Fetaya, E. Meir, G. Chechik, and H. Maron, "From Local Structures to Size Generalization in Graph Neural Networks," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 11975–11986. [Online]. Available: <https://proceedings.mlr.press/v139/yehudai21a.html>
- [4] T. H. Schulz, T. Horváth, P. Welke, and S. Wrobel, "A generalized Weisfeiler-Lehman graph kernel," *Machine Learning*, vol. 111, pp. 2601–2629, 2022. [Online]. Available: <https://link.springer.com/article/10.1007/s10994-022-06131-w>
- [5] Y. Gao, Y. Feng, S. Ji, and R. Ji, "HGNN+: General Hypergraph Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3181–3199, 2023. [Online]. Available: <https://www.computer.org/csdl/journal/tp/2023/03/09795251/1Eb15xX3fHy>
- [6] N. Shervashidze, S. V. N. Vishwanathan, T. H. Petri, K. Mehlhorn, and K. M. Borgwardt, "Efficient Graphlet Kernels for Large Graph Comparison," in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 5, 2009, pp. 488–495. [Online]. Available: <https://proceedings.mlr.press/v5/shervashidze09a.html>
- [7] G. Wachman and R. Khardon, "Learning from Interpretations: A Rooted Kernel for Ordered Hypergraphs," in *Proceedings of the 24th International Conference on Machine Learning*, 2007. [Online]. Available: <https://icml.cc/Conferences/2007/paperlist.html>
- [8] L. Bai, P. Ren, and E. R. Hancock, "A Hypergraph Kernel from Isomorphism Tests," in *2014 22nd International Conference on Pattern Recognition*, 2014. [Online]. Available: <https://doi.org/10.1109/ICPR.2014.665>
- [9] J. Böker, "Color Refinement, Homomorphisms, and Hypergraphs," 2019, arXiv preprint arXiv:1903.12432. [Online]. Available: <https://arxiv.org/abs/1903.12432>
- [10] Y. Zhang, S. Du, Y. Feng, S. Ying, and Y. Gao, "Reinterpreting Hypergraph Kernels: Insights Through Homomorphism Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 48, no. 1, pp. 466–481, 2026. [Online]. Available: <https://github.com/FuYou0723/RethinkHGK>
- [11] P. Yanardag and S. V. N. Vishwanathan, "Deep Graph Kernels," in *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1365–1374.
- [12] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity," *Journal of Medicinal Chemistry*, vol. 34, no. 2, pp. 786–797, 1991.
- [13] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 347–375, 2008.
- [14] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. Suppl. 1, pp. i47–i56, 2005.
- [15] P. Yanardag and S. V. N. Vishwanathan, "A Structural Smoothing Framework For Robust Graph Comparison," in *Advances in Neural Information Processing Systems* 28, 2015. [Online]. Available: <https://papers.nips.cc/paper/2015/hash/7810ccd41bf26faaa2c4e1f20db70a71-Abstract.html>
- [16] F. Baüse and N. M. Kriege, "Gradual Weisfeiler-Leman: Slow and Steady Wins the Race," in *Proceedings of the First Learning on Graphs Conference*, vol. 198, 2022, pp. 20:1–20:18. [Online]. Available: <https://proceedings.mlr.press/v198/bause22a.html>
- [17] M. Togninalli, E. Ghisu, F. Llinares-López, B. Rieck, and K. Borgwardt, "Wasserstein Weisfeiler-Lehman Graph Kernels," in *Advances in Neural Information Processing Systems* 32, 2019. [Online]. Available: https://papers.neurips.cc/paper_files/paper/2019/hash/73fed7fd472e502d8908794430511fd-Abstract.html
- [18] C.-Y. Chuang and S. Jegelka, "Tree Mover's Distance: Bridging Graph Metrics and Stability of Graph Neural Networks," in *Advances in Neural Information Processing Systems* 35, 2022. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/139ae969f49abd9a113981c1f7fce5ce-Abstract-Conference.html
- [19] Z. Fang, J. Huang, X. Su, and H. Kasai, "Wasserstein Graph Distance Based on L1-Approximated Tree Edit Distance between Weisfeiler-Lehman Subtrees," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7539–7549. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/25916>
- [20] R. Carpintero Perez, S. Da Veiga, J. Garnier, and B. Staber, "Gaussian process regression with Sliced Wasserstein Weisfeiler-Lehman graph kernels," in *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, vol. 238, 2024, pp. 1297–1305. [Online]. Available: <https://proceedings.mlr.press/v238/carpintero-perez24a.html>
- [21] B. Bevilacqua, Y. Zhou, and B. Ribeiro, "Size-Invariant Graph Representations for Graph Classification Extrapolations," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 837–851. [Online]. Available: <https://proceedings.mlr.press/v139/bevilacqua21a.html>
- [22] D. Buffelli, P. Liò, and F. Vandin, "SizeShiftReg: a Regularization Method for Improving Size-Generalization in Graph Neural Networks," in *Advances in Neural Information Processing Systems* 35, 2022. [Online]. Available: https://papers.nips.cc/paper_files/paper/2022/hash/cee3fa5be458f08fbb12a5bb783aac8-Abstract-Conference.html
- [23] Z. Huang, Q. Yang, D. Zhou, and Y. Yan, "Enhancing Size Generalization in Graph Neural Networks through Disentangled Representation Learning," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235, 2024, pp. 20365–20381. [Online]. Available: <https://proceedings.mlr.press/v235/huang24ac.html>
- [24] S. Gui, X. Li, L. Wang, and S. Ji, "GOOD: A Graph Out-of-Distribution Benchmark," in *Advances in Neural Information Processing Systems* 35, *Datasets and Benchmarks Track*, 2022. [Online]. Available: https://papers.nips.cc/paper_files/paper/2022/hash/0dc91de822b71c66a7f54fa121d8ebb9-Abstract-Datasets_and_Benchmarks.html
- [25] X. Chu, Y. Jin, X. Wang, S. Zhang, Y. Wang, W. Zhu, and H. Mei, "Wasserstein Barycenter Matching for Graph Size Generalization of Message Passing Neural Networks," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, 2023, pp. 6158–6184. [Online]. Available: <https://proceedings.mlr.press/v202/chu23a.html>
- [26] Z. Li, J. Cen, W. Huang, T. Wang, and L. Song, "Size-Generalizable RNA Structure Evaluation by Exploring Hierarchical Geometries," in *International Conference on Learning Representations* 2025, 2025. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2025/hash/691fe5a436d53e23c08fbb2da529617-Abstract-Conference.html
- [27] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://arxiv.org/abs/1810.0826>
- [28] E. Levin, Y. Ma, M. Díaz, and S. Villar, "On Transferring Transferability: Towards a Theory for Size Generalization," in *Advances in Neural Information Processing Systems* 38, 2025. [Online]. Available: https://proceedings.nips.cc/paper_files/paper/2025/hash/60b393761eea35bf491687c46e4f1a5a-Abstract-Conference.html
- [29] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0304397585902245>
- [30] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011. [Online]. Available: <https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>
- [31] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph Neural Networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/4235>
- [32] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016. [Online]. Available: <https://doi.org/10.1109/TAES.2016.140952>