Transitivity Recovering Decompositions: Interpretable and Robust Fine-Grained Relationships

Abhra Chaudhuri[1,5,6] **Massimiliano Mancini**[2] **Zeynep Akata**[3,4] **Anjan Dutta**[5,6] *

[1] University of Exeter [2] University of Trento [3] University of Tübingen

[4] MPI for Informatics [5] The Alan Turing Institute [6] University of Surrey

Abstract

Recent advances in fine-grained representation learning leverage local-to-global (emergent) relationships for achieving state-of-the-art results. The relational representations relied upon by such methods, however, are abstract. We aim to deconstruct this abstraction by expressing them as interpretable graphs over image views. We begin by theoretically showing that abstract relational representations are nothing but a way of recovering transitive relationships among local views. Based on this, we design Transitivity Recovering Decompositions (TRD), a graph-space search algorithm that identifies interpretable equivalents of abstract emergent relationships at both instance and class levels, and with no post-hoc computations. We additionally show that TRD is *provably robust* to noisy views, with empirical evidence also supporting this finding. The latter allows TRD to perform at par or even better than the state-of-the-art, while being fully interpretable. Implementation is available at `https://github.com/abhrac/trd`.

1 Introduction

Identifying discriminative object parts (local views) has traditionally served as a powerful approach for learning fine-grained representations [95, 85, 43]. Isolated local views, however, miss out on the larger, general structure of the object, and hence, need to be considered in conjunction with the global view for tasks like fine-grained visual categorization (FGVC) [94]. Additionally, the way in which local views combine to form the global view (local-to-global / emergent relationships [59]) has been identified as being crucial for distinguishing between classes that share the same set of parts, but differ only in the way the parts relate to each other [32, 97, 10, 9, 61, 12]. However, representations produced by state-of-the-art approaches that leverage the emergent relationships are encoded in an abstract fashion – for instance, through the summary embeddings of transformers [9, 10], or via aggregations on outputs from a GNN [97, 27]. This makes room for the following question that we aim to answer through this work: *how can we make such abstract relational representations more human-understandable?* Although there are several existing works that generate explanations for localized, fine-grained visual features [11, 78, 36, 63], providing interpretations for abstract representations obtained for discriminative, emergent relationships still remains unaddressed.

Illustrated in Figure 1, we propose to make existing relational representations interpretable through bypassing their abstractions, and expressing all computations in terms of a graph over image views, both at the level of the image as well as that of the class (termed *class proxy*, a generalized representation of a class). At the class level, this takes the form of what we call a Concept Graph: a generalized relational representation of the salient concepts across all instances of that class. We use graphs, as they are a naturally interpretable model of relational interactions for a variety of tasks [86, 18, 1, 46], allowing us to visualize entities (e.g., object parts, salient features), and their relationships.

*Abhra Chaudhuri (ac1151@exeter.ac.uk) is the corresponding author.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Figure 1: Instead of learning representations of emergent relationships that are abstract aggregations of views, our method deconstructs the input, latent, and the class representation (proxy) spaces into graphs, thereby ensuring that all stages along the inference path are interpretable.

We theoretically answer the posed question by introducing the notion of transitivity (Definition 3) for identifying subsets of local views that strongly influence the global view. We then show that the relational abstractions are nothing but a way of encoding transitivity, and design Transitivity Recovering Decompositions (TRD), an algorithm that decomposes both input images and output classes into graphs over views by recovering transitive cross-view relationships. Through the process of transitivity recovery (Section 3.2), TRD identifies maximally informative subgraphs that co-occur in the instance and the concept (class) graphs, providing end-to-end relational transparency. In practice, we implement TRD by optimizing a GNN to match a graph-based representation of the input image to the concept graph of its corresponding class (through minimizing the Hausdorff Edit Distance between the two). The input image graph is obtained by decomposing the image into its constituent views [10, 84, 94], and connecting complementary sets of views. The concept graph is obtained by performing an online clustering on the node and edge embeddings of the training instances of the corresponding class. The process is detailed in Section 3.3.

We also show that TRD is provably robust to noisy views in the input image (local views that do not / negatively contribute to the downstream task [18]). We perform careful empirical validations under various noise injection models to confirm that this is indeed the case in practice. The robustness allows TRD to always retain, and occasionally, even boost performance across small, medium, and large scale FGVC benchmarks, circumventing a known trade-off in the explainability literature [31, 73, 23, 37]. Finally, the TRD interpretations are ante-hoc – graphs encoding the learned relationships are produced as part of the inference process, requiring no post-hoc GNN explainers [88, 92, 34, 68, 90].

In summary, with the purpose of bringing interpretability to abstract relational representations, we make the following contributions – (1) show the existence of graphs that are equivalent to abstract local-to-global relational representations, and derive their information theoretic and topological properties; (2) design TRD, a provably robust algorithm that generates such graphs in an ante-hoc manner, incorporating transparency into the relationship computation pipeline at both instance and class levels; (3) extensive experiments demonstrating not only the achieved interpretability and robustness, but also state-of-the-art performance on benchmark FGVC datasets.

2 Related Work

Fine-grained visual categorization: Learning localized image features [2, 96, 47], with extensions exploiting the relationship between multiple images and between network layers [52] were shown to be foundational for FGVC. Strong inductive biases like normalized object poses [6, 87] and data-driven methods like deep metric learning [14] were used to tackle the high intra-class and low inter-class variations. Unsupervised part-based models leveraged CNN feature map activations [35, 94] or identified discriminative sequences of parts [4]. Novel ways of training CNNs for FGVC included boosting [55], kernel pooling [15], and channel masking [19]. Vision transformers, with their ability to learn localized features, had also shown great promise in FGVC [81, 29, 50]. FGVC interpretability has so far focused on the contribution of individual parts [11, 20]. However, the usefulness of emergent relationships for learning fine-grained visual features was demonstrated by

[83, 97, 9, 10]. Our method gets beyond the abstractions inherent in such approaches, and presents a fully transparent pipeline by expressing all computations in terms of graphs representing relationships, at both the instance and the class-level.

Relation modeling in deep learning: Relationships between entities serve as an important source of semantic information as has been demonstrated in graph representation learning [72, 18], deep reinforcement learning [93], question answering [67], object detection [33], knowledge distillation [60], and few-shot learning [70]. While the importance of learning relationships between different views of an image was demonstrated in the self-supervised context [61, 9], [10] showed how relational representations themselves can be leveraged for tasks like FGVC. However, existing works that leverage relational information for representation learning typically (1) model all possible ways the local views can combine to form the global view through a transformer, and distill out the information about the most optimal combination in the summary embedding [9, 10], or (2) model the underlying relations through a GNN, but performing an aggregation on its outputs [27, 97]. Both (1) and (2) produce vector-valued outputs, and such, cannot be decoded in a straightforward way to get an understanding of what the underlying emergent relationships between views are. This lack of transparency is what we refer to as "abstract". The above set of methods exhibit this abstraction not only at the instance, but at the class-level as well, which also appear as vector-valued embeddings. On the contrary, an interpretable relationship encoder should be able to produce graphs encoding relationships in the input, intermediate, and output spaces, while also representing a class as relationships among concepts, instead of single vectors that summarize information about emergence. This interpretability is precisely what we aim to achieve through this work.

Explainability for Graph Neural Networks: GNNExplainer [88] was the first framework that proposed explaining GNN predictions by identifying maximally informative subgraphs and subset of features that influence its predictions. This idea was extended in [92] through exploring subgraphs via Monte-Carlo tree search and identifying the most informative ones based on their Shapley values [69]. However, all such methods inherently provided either local [88, 34, 68] or global explanations [90], but not both. To address this issue, PGExplainer [51] presented a parameterized approach to provide generalized, class-level explanations for GNNs, while [82] obtained multi-grained explanations based on the pre-training (global) and fine-tuning (local) paradigm. [48] proposed a GNN explanation approach from a causal perspective, which led to better generalization and faster inference. On the other hand, [3] improved explanation robustness by modeling decision regions induced by similar graphs, while ensuring counterfactuality. Although ViG [27] presents an algorithm for representing images as a graph of its views, it exhibits an abstract computation pipeline lacking explainability. A further discussion on this is presented in Appendix A.10. Also, generating explanations for graph metric-spaces [99, 45] remains unexplored, which we aim to address through this work.

3 Transitivity Recovering Decompositions

Our methodology is designed around three central theorems - (i) Theorem 1 shows the existence of semantically equivalent graphs for every abstract representation of emergent relationships, (ii) Theorem 2 establishes an information theoretic criterion for identifying such a graph, and (iii) Theorem 3 shows how transitivity recovering functions can guide the search for candidate solutions that satisfy the above criterion. Additionally, Theorem 4 formalizes the robustness of transitivity recovering functions to noisy views. Due to space constraints, we defer all proofs to the Appendix A.9.

Preliminaries: Consider an image $x \in \mathbb{X}$ with a categorical label $y \in \mathbb{Y}$ from an FGVC task. Let $g = c_g(x)$ and $\mathbb{L} = \{l_1, l_2, \ldots, l_k\} = c_l(x)$ be the global and set of local views of an image x respectively, jointly denoted as $\mathbb{V} = \{g\} \cup \mathbb{L}$, where c_g and c_l are cropping functions applied on x to obtain such views. Let f be a semantically consistent, relation-agnostic encoder (Appendices A.4 and A.7) that takes as input $v \in \mathbb{V}$ and maps it to a latent space representation $z \in \mathbb{R}^n$, where n is the representation dimensionality. Specifically, the representations of the global view g and local views $\mathbb{L}$ obtained from f are then denoted by $z_g = f(g)$ and $\mathbb{Z}_{\mathbb{L}} = \{f(l) : l \in \mathbb{L}\} = \{z_{l_1}, z_{l_2}, \ldots, z_{l_k}\}$ respectively, together denoted as $\mathbb{Z}_{\mathbb{V}} = \{z_g, z_{l_1}, z_{l_2}, \ldots, z_{l_k}\}$. Let ξ be a function that encodes the relationships $r \in \mathbb{R}^n$ between the global (g) and the set of local ($\mathbb{L}$) views. DiNo [9] and Relational Proxies [10] can be considered as candidate implementations for ξ. Let $\mathcal{G}$ be a set of graphs $\{(\mathbb{V}, \mathbb{E}_1, \mathbb{F}_{\mathbb{V}_1}, \mathbb{F}_{\mathbb{E}_1}), (\mathbb{V}, \mathbb{E}_2, \mathbb{F}_{\mathbb{V}_2}, \mathbb{F}_{\mathbb{E}_2}), \ldots\}$, where the nodes in each graph constitute of the view set $\mathbb{V}$, and the edge set $\mathbb{E}_i \in \mathcal{P}(\mathbb{V} \times \mathbb{V})$, where $\mathcal{P}$ denotes the power set. $|\mathcal{G}| = |\mathcal{P}(\mathbb{V} \times \mathbb{V})|$, meaning that G is the set of all possible graph topologies with V as the set of nodes. The node features F_{V_i}, and the edge features F_{E_i} 2 R^n.

3.1 Decomposing Relational Representations

De nition 1 (Semantic Relevance Graph). A graph G_s 2 G of image views where each view-pair is connected by an edge with strength proportional to their joint relevance in forming the semantic label y of the image. Formally, the weight of an edge $E_{S_{ij}}$ connecting v_i and v_j is proportional to $I(v_i v_j; y)$.

Intuitively, $E_{S_{ij}}$ is a measure of how well the two views pair together as compatible puzzle pieces in depicting the central concept in the image.

Theorem 1. For a relational representation r that minimizes the information gap $I(x; y|Z_V)$, there exists a metric space (M ; d) that de nes a Semantic Relevance Graph such that:

$$d(z_i; y) > \quad \wedge \ d(z_j; y) > \quad)$$
$$9 \quad : R^n !^{sem} R^n \ j \ r = \ (' \ (z_i); ' \ (z_j)); d(r; y)$$

where M 2 R^n, $!^{sem}$ and d denote semantically consistent transformations and distance metrics respectively, ' : Z_V ! M^{sem}, and is some nite, empirical bound on semantic distance.

Intuition: The theorem says that, even if z_i and z_j individually cannot encode anymore semantic information, the metric space (M ; d) encodes their relational semantics via its distance function d. Assuming the theorem to be false would mean that the outputs of the aggregation do not allow for a distance computation that is semantically meaningful, and hence, there is no source from which r (the relational embedding) can be learned. This would imply that either (i) the information gap (Appendix A.2) does not exist, which is false, because f is a relation-agnostic encoder, or (ii) all available metric spaces are relation-agnostic and hence, the information gap would always persist. The latter is again in contrary to what is shown in Appendix A.6, which derives the necessary and suf cient conditions for a model to bridge the information gap. Thus, (M ; d) de nes G_s with weights $1{=}d(' \ (v_i); ' \ (v_j)) \ / \ I(v_i v_j; y)$, where ' : V ! M .

Corollary 1.1 (Proxy / Concept Graph). Proxies can also be represented by graphs.

Intuition: Since (M ; d) is equipped with a semantically relevant distance function, the hypersphere enclosing a locality/cluster of local/global view node embeddings can be identi ed as a semantic relevance neighborhood. The centers of such hyperspheres can be considered as a proxy representation for each such view, generalizing some salient concept of a class. The centers of each such view cluster can then act as the nodes of a proxy/concept graph, connected by edges, that are also obtained in a similar manner by clustering the instance edge embeddings.

Theorem 2. The graph G underlying r is a member of G with maximal label information. Formally,

$$G = \arg\max_{G 2 G} I(G; y)$$

Intuition: The label information $I(x; y)$ can be factorized into relation-agnostic and relation-aware components (Appendix A.2). The node embeddings in G already capture the relation-agnostic component (being derived from f). From the implication in Theorem 1, we know that given the node embeddings, is able to reduce further uncertainty about the label by joint observation of view pairs (or in other words, edges in G). Hence must be relation-aware, as the relation-agnostic component is already captured by f. Thus, G , which is obtained as a composition of f and , must be a suf cient representation of x (Appendices A.2 and A.3). Since suf ciency is the upper-bound on the amount of label information that a representation can encode (Appendix A.3), G is the graph that satis es the condition in the theorem. In the section below, we show that the way to obtain G from G is by recovering transitive relationships among local views.

3.2 Transitivity Recovery

De nition 2 (Emergence). The degree to which a set of local views $(v_1; v_2; ::: v_k)$ contributes towards forming the global view. It can be quanti ed as $I(v_1 v_2 ::: v_k; g)$.

Figure 2: The TRD pipeline: We begin by computing a Complementarity Graph (G_c) on the view embeddings obtained from f. The Graph Encoder derives the Semantic Relevance Graph (G_s) through transitivity recovery on G_c. We perform a class-level clustering of the instance node and edge embeddings, producing a proxy graph, representing the salient concepts of a class and their interrelationships. The sufficiency of G_s is guaranteed by minimizing its Hausdorff distance with its proxy graph.

Definition 3 (Transitivity). Local views v_1, v_2, and v_3 are transitively related iff, when any two view pairs have a high contribution towards emergence, the third pair also has a high contribution. Formally,

$$I(v_i v_j; g) > \alpha \wedge I(v_j v_k; g) > \alpha \implies I(v_i v_k; g) > \alpha,$$

where $(i, j, k) \in \{1, 2, 3\}$ and α is some threshold for emergence. A function $\phi(v)$ is said to be *transitivity recovering* if the transitivity between v_1, v_2, and v_3 can, in some way, be inferred from the output space of ϕ. This helps us quantify the *transitivity* of *emergent* relationships across partially overlapping sets of views.

Theorem 3. *A function $\varphi(z)$ that reduces the uncertainty $I(v_i v_j; g | \varphi(z_i) \varphi(z_j))$ about the emergent relationships among the set of views can define G_s by being transitivity recovering.*

Intuition: Reducing the uncertainty about transitivity among the set of views is equivalent to reducing the uncertainty about the emergent relationships, and hence, bridging the information gap. If v_1, v_2, and v_3 are not transitively related, then it would imply that, at most, only one of the three views (v_j) is semantically relevant, and the other two (v_i, v_k) are not. This leads to a *degenerate* form of emergence. Thus, transitivity is the key property in the graph space that filters out such degenerate subsets in L, helping identify view triplets where all the elements contribute towards emergence. A further discussion can be found in Appendix A.11. Now, we know that relational information must be leveraged to learn a sufficient representation. So, for sufficiency, a classifier operating in the graph space must leverage transitivity to match instances and proxies. In the proof, we hence show that transitivity in G is equivalent to the relational information $r \in \mathbb{R}^n$. Thus, the explanation graph G_s is a maximal MI member of G obtained by applying a transitivity recovering transformation on the Complementarity Graph.

3.3 Generating Interpretable Relational Representations

Depicted in Figure 2, the core idea behind TRD is to ensure relational transparency by expressing all computations leading to classification on a graph of image views (Theorem 1), all the way up to the class-proxy, by decomposing the latter into a concept graph (Corollary 1.1). We ensure the sufficiency (Theorem 2) of the instance and proxy graphs through transitivity recovery (Theorem 3) by Hausdorff Edit Distance minimization.

Complementarity Graph: We start by obtaining relational-agnostic representations Z_V from the input image x by encoding each of its views $v \in V$ as $f(v)$. We then obtain a Complementarity Graph $G_c \in G$ with node features $F_V = Z_V$, and edge strengths inversely proportional to the value of the mutual information between the local view embeddings $I(z_i; z_j)$ that the edge connects. Specifically, we instantiate the edge features F_E as learnable n-dimensional vectors, all with the same value of

$1/|z_i - z_j|$. This particular choice of calculating the edge weights is valid under the assumption that f is a semantically consistent function. Intuitively, local views with low mutual information (MI) are complementary to each other, while ones with high MI have a lot of redundancy. The inductive bias behind the construction of such a graph is that, strengthening the connections among the complementary pairs suppresses the flow of redundant information during message passing, thereby reducing the search space in $\mathcal{G}$ for finding G. The global view g, however, is connected to all the local-views with a uniform edge weight of 1. This is because the local-to-global emergent information, i.e., $I(l_i l_j \ldots; g)$, is what we want the model to discover through the learning process, and hence, should not incorporate it as an inductive bias.

Semantic Relevance Graph: We compute the Semantic Relevance Graph G_s by propagating G_c through a Graph Attention Network (GAT) [77]. The node embeddings obtained from the GAT correspond to the (z) in our theorems. We ensure that is transitivity recovering by minimizing the Learnable Hausdorff Edit Distance between the instance and the proxy graphs (Corollary 1.1), which is known to take into account the degree of local-to-global transitivity for dissimilarity computation [64]. Below, we discuss this process in further detail.

Proxy / Concept Graph: We obtain the proxy / concept graph for each class via an online clustering of the Semantic Relevance node and edge embeddings ($_n(z)$, $'_e(z)$ respectively) of the train-set instances of that class, using the Sinkhorn-Knopp algorithm [16, 8]. We set the number of node clusters to be equal to $|V|$. The number of edge clusters (connecting the node clusters) is equal to $|V|(|V| - 1)/2$. Note that this is the case because all the graphs throughout our pipeline are complete. Based on our theory of transitivity recovery, sets of semantically relevant nodes should form cliques in the semantic relevance and proxy graphs, and remain disconnected from the irrelevant ones by learning an edge weight of 0. We denote the set of class proxy graphs by $P = \{G_{P_1}; G_{P_2}; \ldots; G_{P_k}\}$, where k is the number of classes.

Inference and Learning objective: To recover end-to-end explainability, we need to avoid converting the instance and the proxy graphs to abstract vector-valued embeddings in $\mathbb{R}^n$ at all times. For this purpose, we perform matching between G_s and $G_p \in P$ via a graph kernel [2]. This kernel trick helps us bypass the computation of the abstract relation $: \mathcal{G} \to \mathbb{R}^{n^2}$, and perform the distance computation directly in $\mathcal{G}$, using only the graph-based decompositions, and thereby keeping the full pipeline end-to-end explainable.

We choose to use the Graph Edit Distance (GED) as our kernel [58]. Although computing GED has been proven to be NP-Complete [79], quadratic time Hausdorff Edit Distance (HED) [25] and learning-based approximations [65] have been shown to be of practical use. However, such approximations are rather coarse-grained, as they are either based on linearity assumptions [65], or consider only the substitution operations [25]. Learnable (L)-HED [64] introduced two additional learnable costs for node insertions and deletions using a GNN, thereby making it a more accurate approximation. We observe that our already performs the job of the encoder prefix in L-HED (Appendix A.12). Thus, we simply apply a fully connected MLP head $M : \to \mathbb{R}$ with shared weights on the (z) embeddings and the vertices of $G_p \in P$ to compute the node insertion and deletion costs. Thus, in our case, L-HED takes the following form:

$$h(G_s; G_p) = \sum_{u \in G_s} \min_{v \in G_p} c(u; v) + \sum_{v \in G_p} \min_{u \in G_s} c(u; v) \quad (1)$$

$$c(u; v) = \begin{cases} ||(u)||; & \text{for deletions } (u \to \varepsilon), \\ ||(v)||; & \text{for insertions } (\varepsilon \to v), \\ ||u - v||/2; & \text{for substitutions}, \end{cases}$$

where $= 1/(2|V|)$, and ε is the empty node [64]. We use $h(\cdot;\cdot)$ as a distance metric for the Proxy Anchor Loss [39], which forms our final learning objective, satisfying Theorem 2. Below we discuss how transitivity recovery additionally guarantees robustness to noisy views.

3.4 Robustness

Let D be the data distribution over $X \times Y$. Consider a sample $x \in X$ composed of views $V = \{v_1; v_2; \ldots; v_k\}$ with label $y \in Y$. A view $v \in V$ is said to be *noisy* if $\text{label}(v) \neq y$ [13]. The

[2] Graph kernels, in general, help bypass mappings $\mathcal{G} \to \mathcal{H}$, where $\mathcal{H}$ is the Hilbert space [22].