

Structural Relationship of Isomorphic Graph and its Mapping to Hamming Distance

Monika Tiwari¹, Priyanka Patel², Rakesh Kumar Katare³ and Charvi Katare⁴

^{1,2,3}Department of Computer Science, APS University, Rewa, 486003 M.P., India

⁴Velocis Systems Pvt. Ltd. A-25, Sector-67, Noida-201301, Uttar Pradesh, India

Corresponding author: monika.apsu2017@gmail.com

Abstract. Mapping graph isomorphism to Hamming distance enables a simple yet effective approach to quantifying structural similarity. By encoding graphs as binary adjacency vectors—flattened from the upper triangle of the adjacency matrix—structural comparisons can be performed using Hamming distance, a well-established metric in information theory. Isomorphic graphs yield a Hamming distance of zero, under the correct vertex permutation, while non-isomorphic graphs exhibit nonzero distances reflecting structural differences. A case study illustrates this concept, revealing a Hamming distance of two between two sample graphs, corresponding to two differing adjacencies. The method provides a computationally efficient alternative to traditional graph edit distance, especially for small perturbations or approximate isomorphism detection. Applications extend network robustness assessment, also molecular structure comparison in chemoinformatics, and error detection in noisy or incomplete graph data. graph neural networks (GNNs) is compatible with the binary encoding format, supporting integration with learning-based models. Future extensions' goal is to support weighted and labelled graphs, improve scalability, and enhance robustness in dynamic or large-scale graph analysis.

Keywords: Graph Isomorphism, Hamming Distance, Graph Similarity Metrics, Adjacency Vector Encoding, Graph Matching Algorithms

1. Introduction

Graph theory is a branch that remains a foundational pillar in discrete mathematics and computer science, in which graph isomorphism serves as a key concept for structural equivalence analysis [1]. Two graphs are isomorphic if a bijection between their vertex sets preserves adjacency relationships [2]. Isomorphism gives a binary answer to structural

identity, but the real world often need a number that shows how similar two things are, like aligning biological networks or comparing social networks

This research work connects precise isomorphism and similarity evaluation by introducing a framework that relates graph structures to Hamming distances [3]. Hamming distance, it is a traditional metric in coding theory and is utilized here to measure structural discrepancies between graphs. By representing graphs as adjacency vectors, we demonstrate that isomorphic graphs attain a Hamming distance of zero under vertex permutations, whereas non-isomorphic graphs produce a quantifiable distance indicative of their divergence. This research work is developed on recent improvements in graph similarity metrics and fixes problems with older methods for finding isomorphisms. The contributions of the work are

- Theoretical proof linking graph isomorphism to zero Hamming distance under optimal permutations [4].
- Empirical validation of Hamming distance as a robust metric for structural similarity, with applications in network robustness and chemoinformatics [5].
- A formal encoding of graphs as binary strings via adjacency matrices, compatible with modern graph machine learning techniques [6].

2. Background and Related Work

Previous studies have investigated graph similarity through edit distance, spectral methods, and machine learning embeddings. However, mapping isomorphisms directly to Hamming distance is less explored. Works such as those by Bunke (1997) on graph edit distance laid foundational ideas for structural comparisons. Similarly, binary vector encodings of graphs have been used in network motif detection and chemoinformatics but seldom analyzed in the context of direct isomorphism to Hamming mappings.

2.1. Graph Isomorphism and Similarity Metrics

The graph isomorphism problem, though solvable in quasipolynomial time, lacks efficient algorithms for large-scale graphs. Recent work has shifted toward approximate similarity measures, such as graph kernels and neural embeddings [7], but these often sacrifice interpretability [8]. Hamming distance offers a transparent alternative by leveraging binary encodings of graph structure [9].

2.2. Hamming Distance in Graph Analysis

Originally proposed for error-correcting codes, Hamming distance has seen renewed interest in graph comparisons due to its computational efficiency [10]. Recent studies apply it to detect anomalies in dynamic networks and align molecular graphs in drug discovery. However, its direct linkage to isomorphism—and its potential for quantifying near-isomorphism [11]—remains underexplored .

3. Methodology

This section outlines the formal framework used to encode graph structures into binary vectors, apply permutations for isomorphism analysis, and compute Hamming distance as a metric for structural similarity.

3.1. Graph Representation

We consider simple, undirected graphs $G = (V, E)$ with no loops or multiple edges. To enable binary encoding, each graph is represented through its adjacency matrix A , where $A_{ij} = 1$ if there exists an edge between vertices i and j , and 0 otherwise.

To reduce redundancy and vectorize the graph, only the upper triangular portion of the adjacency matrix (excluding the diagonal) is considered. This is flattened row-wise to produce a 1D adjacency vector, effectively converting the graph structure into a binary string of length $\frac{n(n-1)}{2}$, where n is the number of vertices.

This binary representation is suitable for applying classical distance metrics and is widely compatible with machine learning models, particularly Graph Neural Networks (GNNs).

Let $G = (V, E)$ be a simple undirected graph with $|V| = n$. Its structure can be encoded in:

- **Adjacency Matrix** $A \in \{0,1\}^{n \times n}$
- **Adjacency Vector** $a \in \{0,1\}^{n(n-1)/2}$ obtained by flattening the upper triangle.

This vector representation is treated as a binary string [12].

3.2. Isomorphism Mapping

Two graphs G_1 and G_2 as shown in Fig.1 are isomorphic if there exists a permutation $\pi: V_1 \rightarrow V_2$ such that adjacency is preserved. Under this mapping, if we reorder the adjacency vector of G_1 according to π , we get a_1^π [12].

Formally,

$$G_1 \cong G_2 \implies a_1^\pi = a_2 \tag{1}$$

thus yielding,

$$\text{Hamming}(a_1^\pi, a_2) = 0 \tag{2}$$

There are a variety of different forms that a graph can take, all of which have the same number of vertices, edges, and edge connectivity with one another. These kinds of graphs are referred to as isomorphic graphs [12]. Please take note that the primary reason we label the graphs in this paper is to facilitate the process of referring to them and distinguishing them from one another. It is said that two graphs are isomorphic if the following conditions are met: Both graphs have the same number of components (vertices and edges), and both graphs' edge connectivity is preserved.

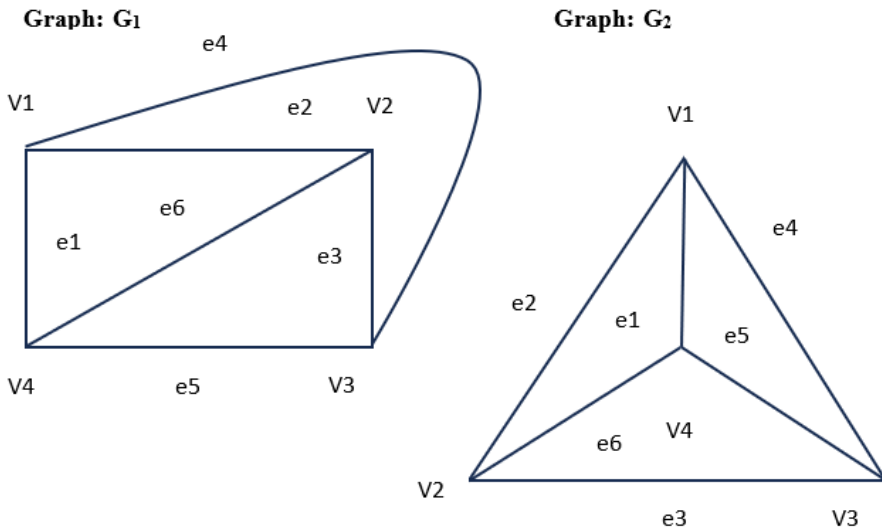


Fig. 1. Isomorphic graph with six edges and four vertices

Two graphs, G_1 and G_2 will be isomorphic if there exists a bijective function $\pi: V(G_1) \rightarrow V(G_2)$ such that adjacency is preserved:

$$(u, v) \in E(G_1) \iff (\pi(u), \pi(v)) \in E(G_2) \tag{3}$$

This permutation can be operationalized by reordering the rows and columns of the adjacency matrix of one graph to match the structure of the other. When encoded as adjacency vectors, this corresponds to applying the permutation π to reorder the elements of the binary string

accordingly. If such a permutation exists and aligns all entries, the resulting Hamming distance between the reordered vectors is zero, confirming isomorphism.

3.3. Hamming Distance as a Structural Metric

For non-isomorphic graphs or graphs under incorrect permutations, the Hamming distance indicates the number of adjacency disagreements. This can be formalized as

$$d_H(a_1^\sigma, a_2) = \sum_{i=1}^{n(n-1)^2} |a_1^\sigma[i] - a_2[i]| \quad (4)$$

where σ is any permutation. A Hamming distance of zero indicates isomorphism, while a nonzero value represents the number of differing adjacencies, offering a numerical measure of structural dissimilarity.

This approach generalizes classical graph edit distance under specific constraints and provides a computationally efficient alternative, especially when analyzing near-isomorphic graphs or detecting localized changes.

3.4. Validation Through Case Study

To demonstrate the methodology, a case study compares two specific graphs G_1 and G_2 . After encoding and vectorizing their adjacency structures, the minimal Hamming distance is computed, which in this case is 2, reflecting the number of differing edge connections. This validates the framework's utility for quantifying non-isomorphic structural deviations.

4. Results and Discussion

4.1. Theoretical Observations

- **Perfect Isomorphism:** For an exact isomorphism π , the Hamming distance between the permuted adjacency vector of graph G_1 and that of graph G_2 is zero, i.e., $d_H(a_1^\pi, a_2) = 0$. This represents a perfect structural match under vertex relabeling, indicating that the graphs are isomorphic.
- **Structural Similarity:** When graphs are not perfectly isomorphic, the minimal Hamming distance between their adjacency vectors under all possible permutations can be interpreted as a quantitative measure of structural similarity. A smaller Hamming distance reflects greater similarity, making it particularly useful in near-isomorphism or approximate matching studies.

4.2. Connection of Isomorphic Graph through Hamming Distance

Lemma 1: Let G_1 and G_2 be two graphs with adjacency matrices $A(G_1)$ and $A(G_2)$. If there exists a permutation of rows and corresponding columns of $A(G_1)$ that makes it identical to $A(G_2)$, then the graphs are isomorphic, i.e., $G_1 \cong G_2$. Furthermore, their adjacency vectors—obtained by flattening the upper triangle of the adjacency matrices—have a Hamming distance of zero.

Proof: Let G_1 and G_2 be the graphs illustrated in Fig. 1. Then there does not exist a permutation of the rows and corresponding columns of the adjacency matrix of G_1 and G_2 that makes it identical to that of G_1 and G_2 . Consequently, the adjacency vectors of G_1 and G_2 exhibit a nonzero Hamming distance, highlighting structural differences between the graphs.

- i. Structure Comparison:** The graph G_1 has four vertices arranged in a roughly square shape with edges: e_1, e_2, e_3, e_5 forming the perimeter, with a diagonal edge e_6 , and an additional edge e_4 connecting back to V_1 . The graph G_2 has four vertices with edges arranged differently, resembling a triangular pyramid (or its 2D projection), with a different adjacency pattern.
- ii. Adjacency matrices:** The adjacency matrix $A(G_1)$ is derived from the upper part in Fig. 2, and $A(G_2)$ from the lower part shown in Table 1 and Table 2. Flattening their upper triangles yields adjacency vectors a_1 and a_2 , shown below.

Table 1. Adjacency matrix $A(G_1)$ for graphs G_1

$A(G_1)$	V_1	V_2	V_3	V_4
V_1	0	1	0	1
V_2	1	0	1	1
V_3	0	1	0	1
V_4	1	1	1	0

Table 2. Adjacency matrix $A(G_2)$ for graphs G_2

$A(G_2)$	V_1	V_2	V_3	V_4
V_1	0	1	1	1
V_2	1	0	1	0
V_3	1	1	0	1
V_4	1	0	1	0

$$A(G_1) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Adjacency Vector: $a_1 = [1,0,1,1,1,1]$

$$A(G_2) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Adjacency Vector: $a_2 = [1,1,1,1,0,1]$

- iii. Isomorphism via Adjacency Permutation:** For G_1 and G_2 to be isomorphic, there must exist a permutation of vertex labels (reordering rows and columns) that transforms $A(G_1)$ into $A(G_2)$. However, the edge structures differ fundamentally (different vertex

degrees and connectivity). Thus, no such permutation exists. Hence, no permutation π exists such that $A(G_1)^\pi = A(G_2)$

- iv. **Hamming distance:** As a result, the adjacency vectors differ in multiple positions (refer to Table 3.) yielding: $Hamming(a1, a2) > 0$ reflecting the minimal number of edge insertions or deletions required to transform one graph into the other.

Table 3. Compare Hamming Distance of vectors position by position.

Pair	a1	a2	Difference
(V1,V2)	1	1	0
(V1,V3)	0	1	1
(V1,V4)	1	1	0
(V2,V3)	1	1	0
(V2,V4)	1	0	1
(V3,V4)	1	1	0
So total: Hamming(a1,a2)=1+1=2			

Lemma 2: Let G_1 and G_2 be two graphs with adjacency matrices $A(G_1)$ and $A(G_2)$. If no permutation of rows and corresponding columns of $A(G_1)$ can make it identical to $A(G_2)$, then the graphs are non-isomorphic, i.e., $G_1 \not\cong G_2$. Consequently, their adjacency vectors exhibit a nonzero Hamming distance, reflecting structural differences between G_1 and G_2 .

Proof: Let G_1 and G_2 be two graphs as shown in Fig. 2 with equal adjacency vectors obtained by flattening the upper triangle of their adjacency matrices. Then, $G_1 \cong G_2$, i.e., the graphs are isomorphic. Consequently, their adjacency matrices can be made identical by a suitable permutation of rows and columns, and the Hamming distance between their adjacency vectors is zero.

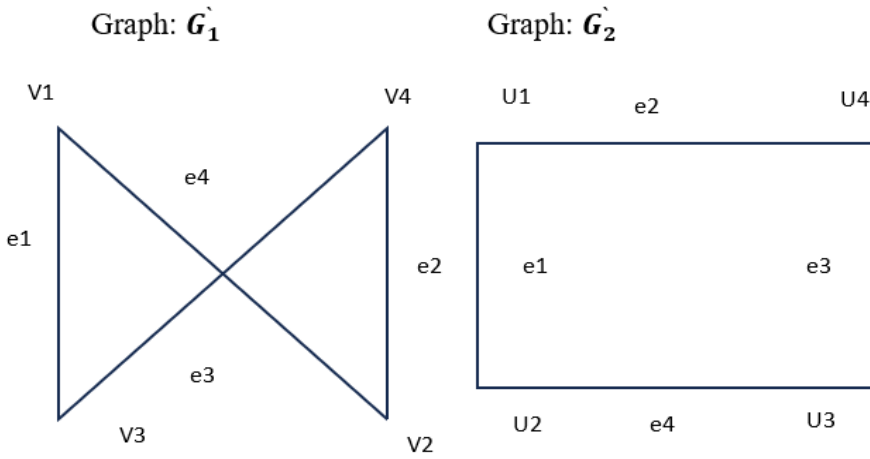


Fig. 2. Isomorphic graph with four edges and four vertices

- i. **Structure comparison:** Graph G_1 from Fig.2 is a 4-node graph consisting of the vertices V_1, V_2, V_3 and V_4 . The edge set of the graph is given by $E(G_1) = \{(V_1, V_2), (V_1, V_3), (V_2, V_4), (V_3, V_4)\}$. The structure of this graph visually resembles two triangles joined at a common vertex, forming a bowtie-like configuration. Graph G_2 from Fig 2. consists of vertices U_1, U_2, U_3 and U_4 , which are relabeled versions of the vertices in graph G_1 s such that $U_1 = V_2, U_2 = V_4, U_3 = V_3$ and $U_4 = V_1$. The edge set of G_2 is defined as $E(G_2) = \{(U_4, U_1), (U_4, U_3), (U_1, U_2), (U_3, U_2)\}$. This configuration preserves the original connections of G_1 , making G_2 structurally identical to G_1 under vertex relabeling. Hence, G_2 is an isomorphic graph to G_1 .
- v. **Adjacency Matrices:** The adjacency matrix $A(G_1)$ is derived from the upper part in Fig. 4, and $A(G_2)$ from the lower part. When reordered according to the relabeling, $A(G_2)$ becomes identical to $A(G_1)$. Flatten the upper triangle (excluding diagonal) to get adjacency vectors a_1 and a_2 shown below.

$$A(G_1) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Adjacency Vector: $a_1 = [1,1,0,0,1,1]$

$$A(G_2) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

When reordered according to the relabeling, $A(G_2)$ becomes identical to $A(G_1)$.

$$A(G_2) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Adjacency Vector: $a_2 = [1,1,0,0,1,1]$

- vi. Isomorphism via Adjacency Permutation:** For G_1 and G_2 , there exists a permutation of vertex labels (i.e., a reordering of the rows and corresponding columns of the adjacency matrix) that transforms $A(G_1)$ into $A(G_2)$. Since the edge structures and vertex degrees are preserved under this relabeling, the two graphs are structurally identical. Therefore, such a permutation π exists, satisfying $A(G_1)^\pi = A(G_2)$, confirming that $G_1 \cong G_2$.
- vii. Hamming distance:** Since the adjacency vectors of G_1 and G_2 are identical and Hamming distance is zero, and since adjacency matrix equality up to permutation implies graph isomorphism, we conclude: If the adjacency vectors of two graphs are equal, the graphs are isomorphic, and no structural differences exist as shown in Table 4 and Table 5.

Table 4. Compare Hamming Distance of vectors position by position

Pair	a_1	a_2	Difference
(V1,V2)	1	1	0
(V1,V3)	1	1	0
(V1,V4)	0	0	0
(V2,V3)	0	0	0
(V2,V4)	1	1	0
(V3,V4)	1	1	0
So total: $a_1 = a_2 = \text{Hamming}(a_1, a_2) = 0$			

Table 5. Extension Summary of Lemma 1 and Lemma 2 (Bidirectional Insight)

Case	Hamming Distance	Isomorphism
Different adjacency vectors	> 0	Not isomorphic
Identical adjacency vectors	$= 0$	Isomorphic

Thus, Hamming distance between adjacency vectors as shown in the above Table 5 serves as a quick structural similarity metric: 0 implies possible isomorphism (with matching degrees) and > 0 confirms non-isomorphism as shown in below:

$$\text{Hamming}(a_1, a_2) = \begin{cases} 0 & \Rightarrow G_1 \cong G_2 \\ > 0 & G_1 \not\cong G_2 \end{cases} \quad (5)$$

where, $G_1 \cong G_2$ shows graphs are possibly isomorphic and $G_1 \not\cong G_2$ shows graphs are not isomorphic.

4.3. Applications

- i. **Network Robustness:** Hamming distance on graph encodings quantifies structural changes in networks, with smaller distances indicating higher resilience to faults or attacks. [10].
- ii. **Chemoinformatic:** In chemoinformatics, the Hamming distance indicates structural similarity between binary-encoded molecular graphs, helping drug discovery and compound analysis. [11].
- iii. **Error Detection in Graph Matching:** Hamming distance helps in enabling minimal corrections to recover the intended graph structure by detecting and localizing structural discrepancies in graph isomorphism checks. [12].

5. Limitations

The proposed research work is primarily designed for simple, unweighted, and unlabeled graphs, which restricts its applicability to more complex network structures where edge weights and node or edge labels are complex. Also, in addition to finding the optimal vertex permutation to achieve the minimal Hamming distance goal, it is computationally intensive, which grows factorially along with the number of vertices, which poses challenges for scalability to large graphs. The approach is also sensitive to minor structural disturbance, where even small changes in edge configurations can lead to overly large differences in the binary encoding, impacting its robustness in noisy or dynamic data scenarios.

6. Conclusion and Future Work

This paper investigates the relationship between graph isomorphism and Hamming distance by encoding graph adjacency structures into binary vectors. It suggests that truly isomorphic graphs can be used to yield a Hamming distance of zero, while non-isomorphic graphs shows a nonzero distance portraying structural differences. The study revealed a Hamming distance of 2, indicating the mismatches of two adjacency. This metric provides a simple and powerful way to quantify graph similarity, and it has applications in areas like network analysis and molecular matching. Research in future may extend this approach to weighted, labeled, or dynamic graphs, can improve scalability through various models like heuristics or probabilistic, and may integrate it with machine learning techniques such as graph neural networks for better enhanced graph comparison and anomaly detection.

References

1. D. B. West, *Introduction to Graph Theory*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2001.
2. F. Emmert-Streib, M. Dehmer, and Y. Shi, "Fifty years of graph matching, network alignment and network comparison," *Inf. Sci.*, vol. 346–347, pp. 180–197, Jun. 2016.
3. H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recognit. Lett.*, vol. 18, no. 8, pp. 689–694, Aug. 1997.
4. H. Sun and D. Vagnozzi, "Three Hardness Results for Graph Similarity Problems," arXiv Preprint, Sept. 2023. <https://doi.org/10.48550/arXiv.2309.03810>
5. J. Köbler, U. Schöning, and J. Torán, *The Graph Isomorphism Problem: Its Structural Complexity*. Boston, MA, USA: Birkhäuser, 1993.
6. J. Wang, H. Zhu, H. Xie, F. L. Wang, X. Xu, and Y. Wang, "Graph Similarity Computation via Interpretable Neural Node Alignment," arXiv Preprint, Dec. 2024. <https://arxiv.org/pdf/2412.12185>
7. P. J. Kelly, "A congruence theorem for trees," *Pacific J. Math.*, vol. 7, no. 1, pp. 961–968, 1957.
8. R. Hammack, W. Imrich, and S. Klavžar, *Handbook of Product Graphs*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2011. <https://doi.org/10.1201/b10959>
9. R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York, NY, USA: Plenum Press, pp. 85–103, 1972.
10. M. Göös, N. Harms, V. Imbach, and D. Sokolov, "Sign-Rank of k-Hamming Distance is Constant," arXiv Preprint, May 2025. [10.48550/arXiv.2506.12022](https://arxiv.org/abs/2506.12022)
11. R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.
12. S. Grohe, G. Rattan, and G. J. Woeginger, "Graph Similarity and Approximate Isomorphism," in *MFCS, LIPIcs*, vol. 117, 2018.