ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN HỌC MÁY VỚI DỮ LIỆU ĐỒ THỊ

Cao học khóa K33

Đề tài:
Giải thích mạng nơ-ron đồ thị bằng đại diện có cùng phân phối
(Generating In-Distribution Proxy Graphs for Explaining Graph Neural Networks)

Giảng viên lý thuyết:	GS.TS LÊ HOÀI BẮC
	T.S LÊ NGỌC THÀNH
	Th.S NGUYỄN NGỌC ĐỨC
Học viên thực hiện:	23C11007 - Vũ Công Minh

TP HỒ CHÍ MINH - Năm 2025

Mục lục

1 Giới thiệu **4**
- 1.1 Mạng nơ-ron đồ thị GNN . . . 4
- 1.2 Khả năng giải thích của GNN . . . 5
- 1.3 Đóng góp của bài báo . . . 8

2 Cơ sở lý thuyết **8**
- 2.1 Ký hiệu và công thức hoá bài toán . . . 8
- 2.2 Vấn đề dữ liệu ngoài phạm vi phân phối (OOD: Out-Of-Distribution) . . . 9
- 2.3 Nút thắt thông tin trong đồ thị . . . 10

3 Giải thích mô hình với Đồ thị đại diện (Proxy Graph) **11**
- 3.1 Thành phần giải thích (Explainer) . . . 11
- 3.2 Thành phần tạo sinh Đồ thị đại diện (Proxy Graph Generator) . . . 12

4 Thực nghiệm **13**
- 4.1 Chuẩn bị dữ liệu và các mô hình . . . 14
- 4.2 Đánh giá định lượng (RQ1) . . . 15
- 4.3 Đánh giá độ chuyển dịch phân phối (RQ2) . . . 17
- 4.4 Đánh giá tầm ảnh hưởng của các thành phần (RQ3) . . . 18
- 4.5 Kết quả bổ sung . . . 18

5 Kết luận **20**

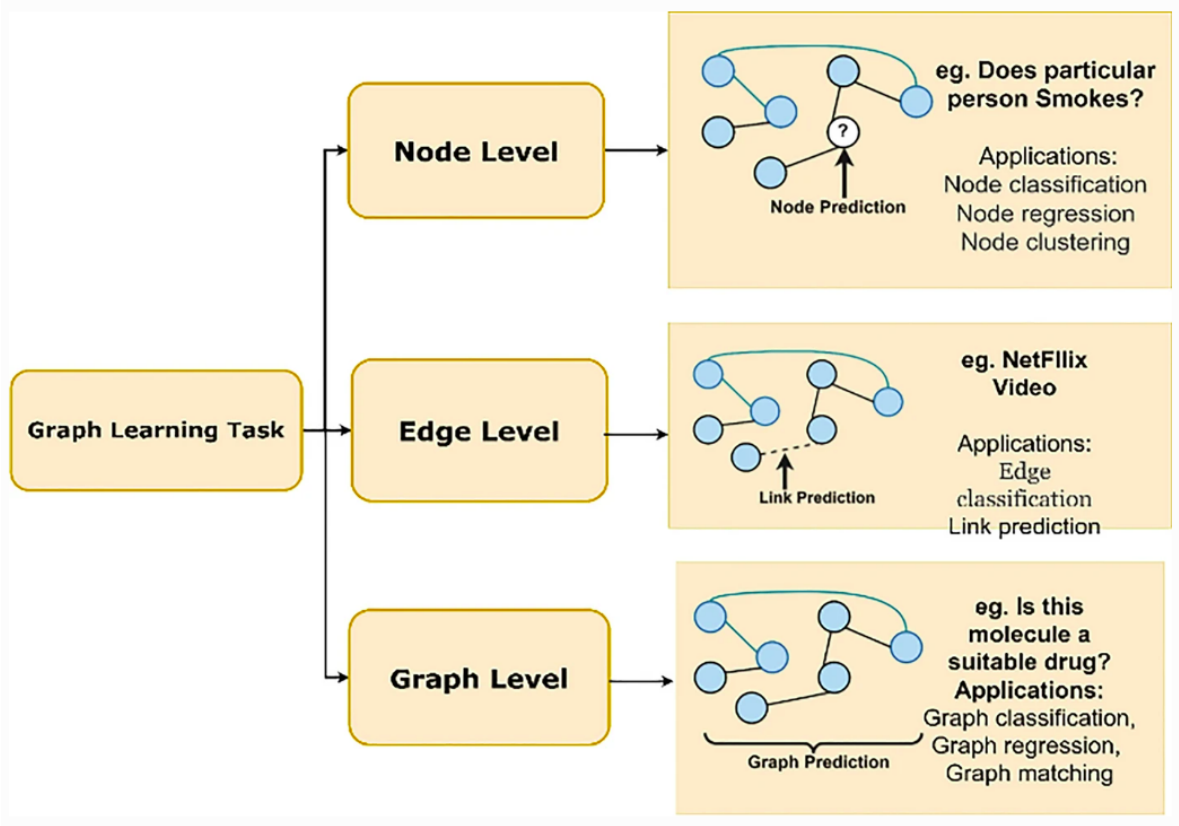
Tóm tắt nội dung

Với sự phát triển của Trí tuệ nhân tạo như hiện nay, mạng nơ-ron đồ thị (GNN) được xem là một trong những hướng nghiên cứu đầy tiềm năng bởi khả năng tích hợp thông tin, mở rộng và áp dụng trong nhiều lĩnh vực có cấu trúc phức tạp như y sinh, mạng xã hội, ... Chính vì nhu cầu triển khai GNN vào các ứng dụng quan trọng, các mô hình phức tạp như GNN cần đòi hỏi thêm về khả năng giải thích nhằm hỗ trợ người dùng hiểu và ra những quyết định chính xác hơn. Một phương pháp phổ biến để giải thích mô hình GNN là xác định các đồ thị con có thể giải thích bằng cách so sách nhãn của nó với đồ thị gốc. Tuy nhiên, công việc này bị ảnh hưởng lớn bởi sự chuyển dịch phân phối trong quá trình huấn luyện dẫn tới việc dự đoán nhãn không đạt được kết quả cao. Chính vì vậy, Zhuomin và các cộng sự đã đề xuất một phương pháp mới bằng cách tạo các đồ thị đại diện (proxy graph) từ những đồ thị con có thể giải thích, vừa đảm bảo khả năng giải thích cũng như phạm vi phân phối không quá lệch nhau trong quá trình huấn luyện. Trong phạm vi môn học, nhóm sẽ tiến hành tìm hiểu, trình bày lại những đóng góp của nhóm tác giả theo cách hiểu của mình, đồng thời diễn giải chi tiết hơn về các mô hình và phương pháp được liệt kê trong bài nghiên cứu. Ngoài ra, nhóm sẽ thử nghiệm thêm với một tập dữ liệu khác về giao thông cũng như bổ sung kết quả thực nghiệm mà tác giả không đề cập trong bài.

1 Giới thiệu

1.1 Mạng nơ-ron đồ thị GNN

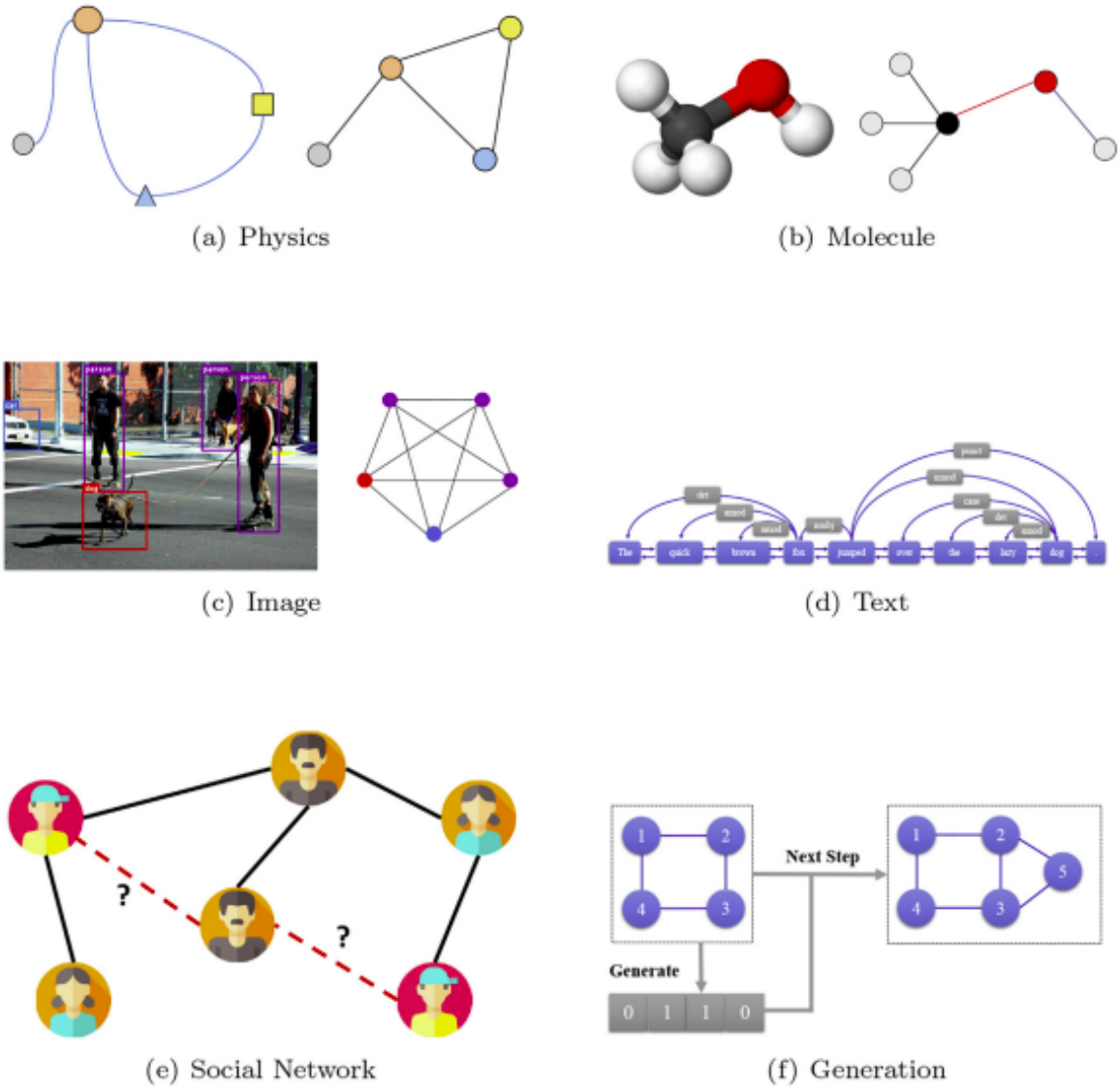
Hiện nay, mạng nơ-ron đồ thị (Graph Neural Network - GNN) ngày càng nhận được sự quan tâm bởi khả năng xử lý của nó trên các dữ liệu lớn có nhiều liên kết phức tạp như mạng xã hội, liên kết phân tử, bản đồ đường đi, ... [Zhou et al.(2018)]. Ý tưởng cốt lõi của GNNs là tận dụng cấu trúc đồ thị của dữ liệu để học các biểu diễn (representations) hiệu quả cho các đối tượng (nút) và liên kết của chúng (cạnh). Tahy vì xử lý dữ liệu một cách độc lập, GNNs xem xét mối quan hệ giữa các phần tử dữ liệu thông qua các kết nối trong đồ thị. Nguyên lý hoạt động của GNNs dựa trên việc truyền thông tin giữa các nút lân cận trong đồ thị. Mỗi nút sẽ tổng hợp thông tin từ các nút hàng xóm của nó và sử dụng thông tin này để cập nhật biểu diễn của chính nó. Quá trình này được lặp lại nhiều lần, cho phép thông tin lan truyền khắp đồ thị và các nút có thể học được các biểu diễn phức tạp, phản ánh cấu trúc và mối quan hệ của đồ thị. Những năm gần đây chứng kiến sự bùng nổ của GNNs với nhiều kiến trúc mới được đề xuất, chẳng hạn như Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), và nhiều biến thể khác. Sự phát triển này không chỉ giúp cải thiện hiệu suất của GNNs trên các bài toán khác nhau mà còn mở rộng phạm vi ứng dụng của chúng.



Hình 1: Các nhiệm vụ học trong GNN

Mạng nơ-ron đồ thị đặc biệt thích hợp với các dữ liệu có không gian phi Euclid. Khả n ăng tổng quát hóa tốt cũng là một điểm mạnh của GNNs. Mô hình được huấn luyện trên một tập đồ thị có thể được áp dụng cho các đồ thị mới với cấu trúc tương tự. Điều này cho thấy GNNs có khả năng thích ứng linh hoạt với dữ liệu mới, giúp tiết kiệm thời gian và công sức huấn luyện lại mô hình từ đầu. Bên cạnh đó, GNNs cũng tồn tại một số hạn chế về tài nguyên tính toán và thời gian vì trong thực tế các mạng lưới thông tin thật sự rất thưa và phức tạp. Mặc dù còn một số thách thức, GNNs đã chứng minh được tính hiệu quả của mình trong một loạt các ứng dụng thực tế. [Zhou et al.(2021)] Trong lĩnh vực mạng xã hội, GNNs được sử dụng để phân tích cấu trúc mạng, dự đoán hành vi người dùng, gợi ý bạn bè và phát hiện tin giả. Trong hóa học và sinh học, GNNs giúp dự đoán tính chất phân tử, tìm kiếm thuốc mới, phân tích tương tác protein và hiểu các quá trình sinh học phức tạp. Trong xử

lý ngôn ngữ tự nhiên, GNNs được áp dụng để phân tích cú pháp, trích xuất thông tin, dịch máy và hiểu ngữ nghĩa của câu. Trong lĩnh vực giao thông, GNNs giúp dự đoán lưu lượng giao thông, tối ưu hóa lộ trình và quản lý mạng lưới giao thông thông minh. Ngoài ra, GNNs còn được ứng dụng trong thương mại điện tử để gợi ý sản phẩm cho người dùng, phân tích hành vi mua hàng và phát hiện gian lận. Sự đa dạng trong ứng dụng thực tế cho thấy tiềm năng to lớn của GNNs trong việc giải quyết các bài toán phức tạp trong nhiều lĩnh vực khác nhau.

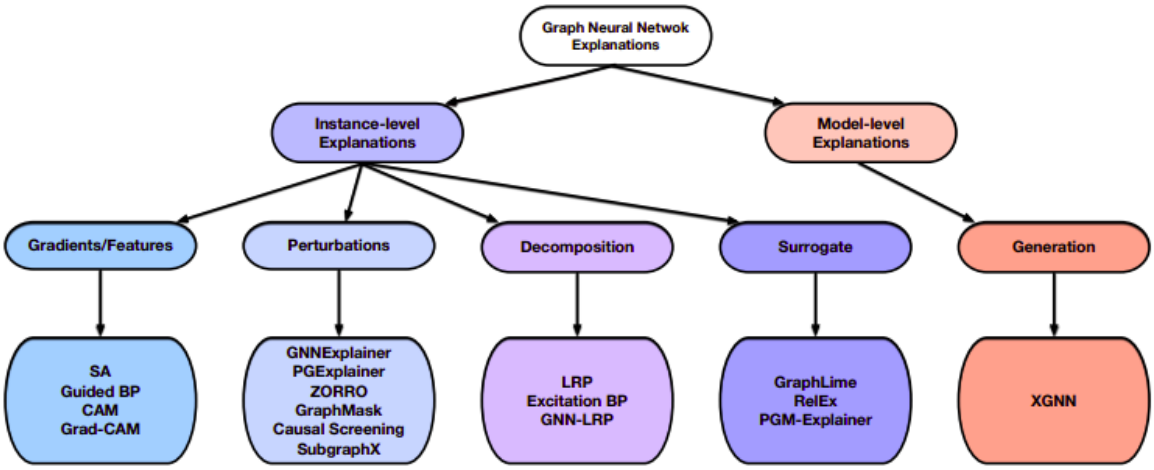


Hình 2: Những ứng dụng của GNN

1.2 Khả năng giải thích của GNN

Việc giải thích các mô hình học sâu nói chung và GNNs nói riêng có vai trò vô cùng quan trọng, đặc biệt trong bối cảnh chúng ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực như hiện nay. Những mô hình ấy được coi là "hộp đen"(blackbox) bởi tính phức tạp trong cách chúng tính toán và đưa ra quyết định. Việc diễn giải giúp làm sáng tỏ quá trình này, cho phép con người hiểu được "lý do"đằng sau các dự đoán. Điều này đặc biệt quan trọng trong các ứng dụng nhạy cảm như y tế, tài chính, pháp lý, nơi mà độ tin cậy và tính minh bạch là yếu tố then chốt. Không những thế, việc phân tích các lời giải thích, chúng ta có thể phát hiện ra các lỗi trong mô hình hoặc trong dữ liệu huấn luyện,

từ đó giúp cải thiện hiệu suất mô hình hơn. Theo [Yuan et al.(2020)], khả năng diễn giải của mô hình nơ-ron đồ thị được chia thành các thành phần sau:



Hình 3: Các khía cạnh của việc cách diễn giải của GNN

- Mức độ giải thích:
  - Instance-level Explanations: Giải thích lý do đưa ra dự đoán cho một trường hợp cụ thể (ví dụ: tại sao một nút được phân loại như vậy).
  - Model-level Explanations: Hiểu tổng quan về cách thức hoạt động của toàn bộ mô hình (ví dụ: đặc trưng nào quan trọng nhất cho việc dự đoán).
- Phương pháp tiếp cận:
  - Gradients/Features: Dựa trên gradient của mô hình để xác định các thành phần đặc trưng.
  - Perturbations: Tạo ra các phiên bản nhiễu loạn của đồ thị đầu vào.
  - Decomposition: Phân tích mức độ đóng góp của từng phần tử vào dự đoán.
  - Surrogate: Sử dụng mô hình đơn giản hơn để xấp xỉ và giải thích.
  - Generation: Tạo ra một đồ thị mới để giải thích.

SA (Saliency Maps) là một phương pháp giúp xác định các yếu tố quan trọng trong dự đoán của mô hình bằng cách tính toán đạo hàm của kết quả đối với các đặc trưng đầu vào. Phương pháp này rất hiệu quả khi áp dụng cho các mô hình có thể tính toán đạo hàm dễ dàng, như mạng nơ-ron thông thường hoặc GNNs. Để sử dụng SA, người ta thường làm nổi bật các nút hoặc cạnh quan trọng trong đồ thị, giúp người dùng hiểu rõ hơn về cách mà mô hình quyết định dựa trên các đặc trưng của đầu vào. Một điểm mạnh của phương pháp này là sự đơn giản và dễ dàng áp dụng cho nhiều loại mô hình, tuy nhiên, một số nhược điểm là nó có thể gặp khó khăn khi áp dụng cho các mô hình phức tạp hoặc khi gradient quá nhỏ.

Guided BP (Guided Backpropagation) là một cải tiến của phương pháp lan truyền ngược (backpropagation) để làm nổi bật các đặc trưng quan trọng hơn. Phương pháp này giúp giảm thiểu sự mơ hồ trong việc xác định các yếu tố ảnh hưởng đến kết quả dự đoán. Khi sử dụng phương pháp này trong GNNs, nó sẽ chỉ truyền lại các tín hiệu gradient từ những khu vực có ảnh hưởng lớn, từ đó giúp tăng tính rõ ràng trong giải thích. Tuy nhiên, phương pháp này có thể đòi hỏi tài nguyên tính toán lớn và không phải lúc nào cũng hiệu quả với các mô hình phức tạp.

CAM (Class Activation Mapping) là phương pháp dùng để làm nổi bật các vùng quan trọng trong mô hình dự đoán, chủ yếu được áp dụng cho các mô hình CNN. Tuy nhiên, CAM cũng có thể được áp dụng cho GNNs trong trường hợp các dữ liệu có cấu trúc giống với ảnh hoặc các đồ thị có cấu trúc lưới rõ ràng. CAM giúp tạo ra các heatmap để người dùng dễ dàng nhận diện các khu vực quan trọng mà mô hình tập trung vào. Dù vậy, CAM có hạn chế khi áp dụng cho các đồ thị không có cấu trúc rõ ràng hoặc phức tạp hơn.

Grad-CAM là phiên bản mở rộng của CAM, giúp giải thích kết quả của mô hình qua các lớp sâu hơn. Grad-CAM đã được chứng minh là rất hữu ích trong việc giải thích các mô hình phức tạp và sâu hơn, vì nó cho phép xác định các vùng quan trọng trong các lớp sâu của mạng. Phương pháp này có thể áp dụng cho GNNs, đặc biệt là trong các mô hình có nhiều lớp và cấu trúc phức tạp. Tuy nhiên, giống như CAM, Grad-CAM có thể không hiệu quả với các đồ thị không có cấu trúc rõ ràng hoặc các tác vụ yêu cầu phân tích chi tiết.

GNNExplainer là một phương pháp giải thích mô hình GNN bằng cách tìm ra các tập con của đồ thị có ảnh hưởng lớn đến dự đoán. Phương pháp này được sử dụng để xác định những phần quan trọng nhất trong đồ thị, từ đó giúp người dùng hiểu rõ hơn về cách mà mô hình đưa ra dự đoán. GNNExplainer có thể được áp dụng cho các tập dữ liệu đồ thị phức tạp như Cora hoặc Citeseer, nơi mà việc xác định các yếu tố quan trọng là rất cần thiết. Tuy nhiên, phương pháp này có thể tốn thời gian tính toán và không phải lúc nào cũng dễ hiểu đối với các đồ thị có cấu trúc quá phức tạp.

PGExplainer tương tự như GNNExplainer, nhưng thay vì tìm kiếm các subgraph, phương pháp này tập trung vào việc phân tích tác động của các đặc trưng hoặc phần tử cụ thể đối với kết quả dự đoán của mô hình. PGExplainer sử dụng các kỹ thuật perturbation để kiểm tra độ nhạy của mô hình với sự thay đổi trong các yếu tố đầu vào. Điều này giúp hiểu rõ hơn về các đặc trưng nào là quan trọng nhất trong việc tạo ra kết quả. Tuy nhiên, giống như GNNExplainer, PGExplainer cũng có thể đòi hỏi tài nguyên tính toán lớn khi áp dụng cho các đồ thị lớn.

ZORRO là một phương pháp sử dụng các perturbation để kiểm tra sự thay đổi trong dự đoán của mô hình khi thay đổi các yếu tố trong đồ thị. ZORRO rất hữu ích trong việc xác định sự ảnh hưởng của các yếu tố khác nhau trong mạng, giúp làm rõ các mối quan hệ nhân quả giữa các đặc trưng và kết quả. Mặc dù phương pháp này rất hiệu quả trong việc hiểu các mối quan hệ phức tạp, nhưng nó cũng có thể gặp khó khăn khi làm việc với các đồ thị chứa nhiều yếu tố không liên quan hoặc quá nhiều nhiễu.

GraphMask là phương pháp giải thích mô hình GNN thông qua việc che giấu các phần tử trong đồ thị và đánh giá tác động của chúng đối với kết quả dự đoán. Nó giúp hiểu rõ hơn về các nút hoặc cạnh quan trọng trong mô hình. Phương pháp này rất hữu ích trong việc phân tích các yếu tố quyết định của mô hình, nhưng nó cũng có thể gặp phải vấn đề khi áp dụng cho các đồ thị phức tạp hoặc khi các yếu tố có ảnh hưởng yếu nhưng vẫn cần thiết.

Causal Screening là một phương pháp giúp xác định các yếu tố có ảnh hưởng nhân quả đối với kết quả mô hình. Nó khác biệt so với các phương pháp perturbation thông thường ở chỗ nó không chỉ tìm kiếm các mối quan hệ tương quan mà còn kiểm tra xem sự thay đổi trong các yếu tố có thực sự gây ra sự thay đổi trong kết quả hay không. Phương pháp này rất mạnh mẽ khi cần làm rõ các mối quan hệ nhân quả trong các ứng dụng như hệ thống khuyến nghị hoặc chẩn đoán y tế. Tuy nhiên, để có thể sử dụng phương pháp này hiệu quả, người sử dụng phải có một hiểu biết vững về các giả thuyết nhân quả trong mô hình.

SubgraphX là một phương pháp giúp tìm kiếm các subgraph quan trọng trong đồ thị và phân tích ảnh hưởng của chúng đối với kết quả dự đoán. Nó rất phù hợp với các tác vụ phân loại đồ thị lớn, ví dụ như phân loại protein hoặc dự đoán mối quan hệ giữa các đối tượng trong mạng xã hội. Mặc dù phương pháp này giúp phát hiện các subgraph quan trọng, nhưng nó có thể tốn kém về mặt tính toán khi làm việc với các đồ thị có kích thước rất lớn.

LRP (Layer-wise Relevance Propagation) là một phương pháp phân tích sự đóng góp của các lớp trong mạng đến kết quả dự đoán của mô hình. LRP đặc biệt hữu ích trong các mô hình sâu như GNNs, giúp hiểu rõ hơn về cách mỗi lớp tác động vào quyết định cuối cùng. Tuy nhiên, phương pháp này có thể gặp phải một số vấn đề về hiệu suất khi áp dụng cho các mô hình GNN với số lượng lớp lớn, vì việc theo dõi sự lan truyền qua nhiều lớp có thể rất phức tạp và tốn thời gian tính toán.

Excitation BP là một biến thể của phương pháp lan truyền ngược, tập trung vào các phần quan trọng nhất của mô hình để xác định những yếu tố nào có ảnh hưởng lớn nhất đến kết quả. Phương pháp này giúp cải thiện độ chính xác của việc giải thích các quyết định của mô hình GNN, đặc biệt là khi mô hình rất sâu. Tuy nhiên, một nhược điểm của Excitation BP là nó có thể không hiệu quả khi áp dụng cho các mạng nơ-ron có quá nhiều lớp hoặc cấu trúc phức tạp.

GNN-LRP là phiên bản đặc biệt của LRP dành cho GNNs, giúp phân tích sự đóng góp của các lớp trong mạng đồ thị. Phương pháp này cho phép người dùng hiểu rõ hơn về cách các lớp trong mô hình GNN tương tác với nhau để đưa ra quyết định cuối cùng. Tuy nhiên, giống như LRP, GNN-LRP có thể tốn thời gian tính toán đối với các mạng nơ-ron đồ thị có kích thước lớn hoặc các mô hình quá phức tạp.

GraphLime là phương pháp giúp giải thích các mô hình GNN thông qua việc xây dựng các mô hình thay thế đơn giản hơn, ví dụ như cây quyết định, để mô phỏng quyết định của mô hình GNN phức tạp. Phương pháp này rất hữu ích khi người dùng cần giải thích các quyết định của mô hình GNN một cách dễ hiểu. Tuy nhiên, nhược điểm của GraphLime là nó chỉ cung cấp các mô phỏng, không phải là giải thích trực tiếp từ chính mô hình.

RelEx là một phương pháp tương tự như GraphLime, nhưng nó giúp giải thích các mô hình GNN thông qua các mô hình thay thế đơn giản, giúp xác định những yếu tố quyết định trong mô hình. Mặc dù RelEx có thể cung cấp các giải thích rõ ràng và dễ hiểu, nhưng nó có thể không phản ánh hoàn toàn sự phức tạp của mô hình GNN ban đầu.

PGM-Explainer áp dụng các mô hình đồ thị xác suất để giải thích các quyết định của GNNs. Phương pháp này rất phù hợp với các tác vụ cần giải thích các mối quan hệ xác suất giữa các nút và cạnh trong đồ thị. Tuy nhiên, PGM-Explainer có thể đòi hỏi kiến thức chuyên môn và tài nguyên tính toán lớn để triển khai hiệu quả.

XGNN là một phương pháp giúp sinh ra các đồ thị giải thích được, giúp người dùng dễ dàng hiểu cách mà mô hình GNN đưa ra quyết định. Phương pháp này giúp xây dựng các đồ thị con có thể giải thích, làm rõ các yếu tố quan trọng trong quá trình dự đoán. Tuy nhiên, việc sinh ra đồ thị giải thích có thể đòi hỏi tài nguyên tính toán lớn và phức tạp khi áp dụng cho các đồ thị có kích thước lớn.

1.3 Đóng góp của bài báo

Bài báo này tập trung vào phân tích và giải quyết một thách thức quan trọng: vấn đề "ngoài phân bố"(OOD - Out-of-Distribution) trong các mô hình GNN có khả năng giải thích. Vấn đề này có ý nghĩa then chốt để nâng cao độ tin cậy và khả năng diễn giải của GNN trong các ứng dụng thực tế. Những đóng góp chính của chúng tôi bao gồm:

- Phân tích và giải quyết vấn đề OOD: Chúng tôi đã tiến hành phân tích một cách có hệ thống và đề xuất giải pháp cho vấn đề dữ liệu nằm ngoài phân bố, một thách thức lớn đối với việc tạo ra lời giải thích đáng tin cậy cho GNN. Việc giải quyết vấn đề này giúp tăng cường tính ứng dụng thực tế của GNN.
- Phương pháp tham số mới: Chúng tôi giới thiệu một phương pháp tham số cải tiến, kết hợp các bộ tự mã hóa đồ thị để tạo ra các "đồ thị đại diện phân bố trong". Các đồ thị này không chỉ nằm trong phân bố dữ liệu gốc mà còn bảo toàn thông tin giải thích thiết yếu. Điều này giúp tạo ra các lời giải thích chính xác và dễ hiểu hơn cho các ứng dụng GNN.
- Thực nghiệm tàn diện: Thông qua các thực nghiệm toàn diện trên nhiều tập dữ liệu thực tế, chúng tôi chứng minh tính hiệu quả của phương pháp được đề xuất. Kết quả cho thấy phương pháp của chúng tôi có tính ứng dụng cao và vượt trội trong việc tạo ra các lời giải thích chất lượng.

2 Cơ sở lý thuyết

2.1 Ký hiệu và công thức hoá bài toán

Xét một đồ thị $G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ thuộc một tập đồ thị $\mathcal{G}$,

- $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ là Tập các nút (node set).
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ là Tập các cạnh (edge set).
- $\mathcal{X} \in \mathbb{R}^{n \times d}$ là Đặc trưng nút (node feature). với d là không gian đặc trưng (feature dimension) và dòng thứ i đại diện cho vec tơ đặc trưng (feature vector) của nút v_i. Ma trận kề của G được ký hiệu $A \in \{0,1\}^{n \times n}$ được xác định bởi tập cạnh $\mathcal{E}$ sao cho $A_ij = 1$ nếu $(v_i, v_j) \in \mathcal{E}, A_{ij} = 0$.

Nghiên cứu này sẽ tập trung giải quyết bài toán **Phân loại đồ thị**. Với mỗi đồ thị $\mathcal{G}$ có nhãn $Y \in \mathcal{Y}$, một mô hình GNN dùng để giải thích $f(\cdot)$ cần được huấn luyện sao cho có thể phân loại đồ thị $\mathcal{G}$ vào các lớp của nó $f : \mathcal{G} \mapsto \{1, 2, \ldots, |\mathcal{Y}|\}$. Đây được xem là cách tiếp cận ở cấp độ trường hợp (post-hoc instance level). Xét một đồ thị $G \in \mathcal{G}$, hàm tham số (parametric function) sẽ cần tìm được một đồ thị con $\mathcal{G}^* \subseteq \mathcal{G}$ có thể giải thích cho kết quả dự đoán $f(\mathcal{G})$, và ánh xạ tham số $\Psi_\psi : \mathcal{G} \mapsto \mathcal{G}^*$ được xem là hàm giải thích.

Table 5. Symbols and their descriptions.

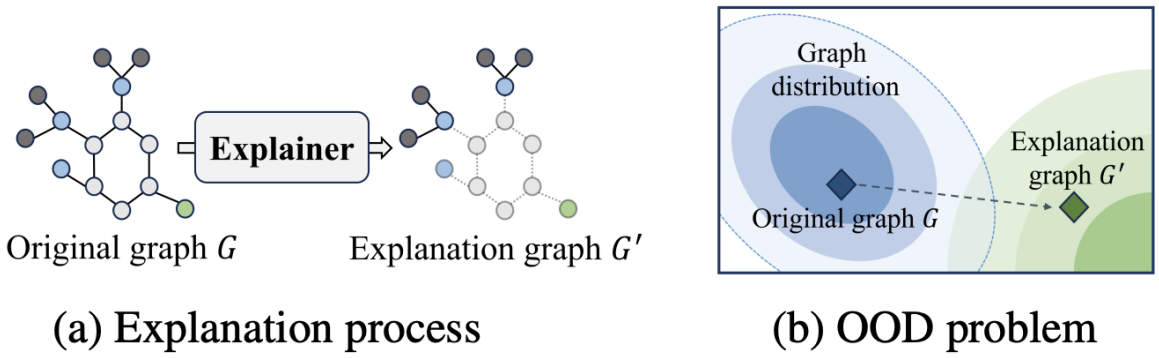
Symbols	Descriptions
$\mathcal{G}$	A set of graphs
$G, \mathcal{V}, \mathcal{E}$	Graph instance, node set, edge set
v_i	The i-th node
$\boldsymbol{X}$	Node feature matrix
$\boldsymbol{A}$	Adjacency matrix
$\boldsymbol{Z}$	Node representation matrix
Y	Label of graph G
$\mathcal{Y}$	A set of labels
G^*	Optimal explanatory subgraph
$\mathcal{G}^*$	A set of G^*
G'	Candidate explanatory subgraph
G^Δ	Non-explanatory graph
$\tilde{G}$	Proxy graph of G' with a fixed distribution
$\boldsymbol{h}, \boldsymbol{h}', \tilde{\boldsymbol{h}}$	Graph embeddings
d	Dimension of node feature
$f(\cdot)$	To-be-explained GNN model
$\Psi_\psi(\cdot)$	Explanation function
ψ	Parameter of the explanation function
$P_\mathcal{G}$	Distribution of original training graphs
$P_{\mathcal{G}'}$	Distribution of explanation subgraphs
$Q_{\boldsymbol{\phi}}$	Parameterized function of $P(\tilde{G}\|G')$
$\boldsymbol{\phi}$	Model parameters of $Q_{\boldsymbol{\phi}}$
$\boldsymbol{\phi}^*$	Optimal $\boldsymbol{\phi}$
α	Balance parameter between $I(G, G')$ and $I(Y, G')$
$\bar{\mathcal{E}}$	The set of node pairs that are unconnected in G
$\tilde{p}_{uv}$	Probability of node pair (u, v) in $\tilde{G}$
β	A hyper-parameter to get a trade-off between connected and unconnected node pairs
$f_{\text{enc}}(\cdot)$	The front part of GNNs that learns node representations
$f_{\text{cls}}(\cdot)$	The back part of GNNs that predicts graph labels based on node embeddings
$\sigma(\cdot)$	Sigmoid function
τ	Temperature hyper-parameter for approximation
λ	A hyper-parameter in Proxy loss function
$\mathcal{L}_{\text{dist}}$	Distribution loss between $\tilde{G}$ and G
$\mathcal{L}_{\text{KL}}$	KL divergence between distribution of $\boldsymbol{Z}^\Delta$ and its prior
$\mathcal{L}_{\text{proxy}}$	Proxy loss
$\mathcal{L}_{\text{exp}}$	Explainer loss

Hình 4: Bảng chú thích các ký hiệu

2.2 Vấn đề dữ liệu ngoài phạm vi phân phối (OOD: Out-Of-Distribution)

GNN là các mô hình mạnh mẽ cho việc xử lý dữ liệu có cấu trúc, như mạng xã hội, phân tử hóa học, hoặc các mạng phức tạp khác. Tuy nhiên, khi dữ liệu huấn luyện và dữ liệu kiểm tra đến từ các phân phối khác nhau, hiệu suất của GNN thường giảm đáng kể. Điều này xảy ra vì GNN thường được thiết kế để hoạt động tốt trong trường hợp dữ liệu huấn luyện và kiểm tra có cùng phân phối. Khi huấn luyện các mô hình Graph Neural Network (GNN), một trong những thách thức lớn là khả năng tổng quát hóa của mô hình đối với dữ liệu nằm ngoài phân phối huấn luyện, hay còn gọi là dữ liệu ngoài phạm vi phân phối (OOD). Việc các dữ liệu không thuộc phạm vi mà mô hình đã được huấn luyện làm ảnh hưởng đến khả năng dự đoán hoặc phân loại không chính xác. Đây cũng là một vấn đề khá ít được chú trọng trong các mô hình GNN có thể giải thích như: GCE [Finkelshtein et al.(2024)] quan tâm tối ưu hoá khả năng truyền đạt từ đồ thị "giáo viên"sang đồ thị "học sinh", MixupExplainer [Zhang et al.(2023)] đã giả định mô hình hỗn hợp có cùng phân phối với mô hình giải thích. Tuy nhiên trong thực tế, dữ liệu thường có sự thay đổi về phân phối do nhiều yếu tố như sự thay đổi trong môi trường, điều kiện thực tế, hoặc sự xuất hiện của các mẫu dữ liệu mới. Điều này đòi hỏi các mô hình

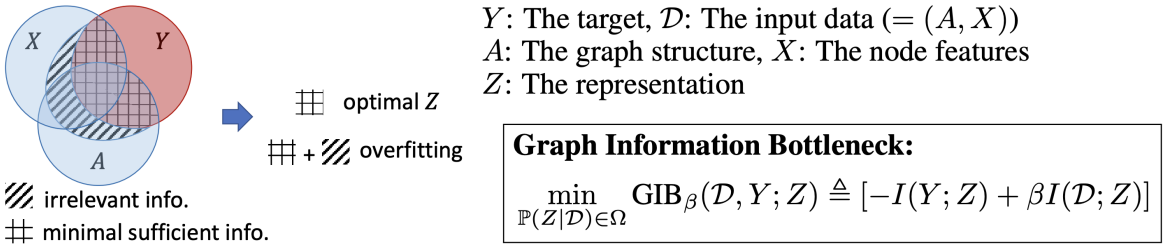
GNN phải có khả năng tổng quát hóa tốt đối với dữ liệu OOD để duy trì hiệu suất dự đoán.



Hình 5: (a) mô tả quá trình giải thích của mô hình GNN bằng cách phát hiện những nút/cạnh/đồ thị con đặc trưng ảnh hưởng đến kết quả dự đoán. (b) mô tả hiện tượng đồ thị giải thích G' nằm ngoài phân phối mà GNN được huấn luyện

2.3 Nút thắt thông tin trong đồ thị

Một trong những cách tiếp cận phổ biến để giải thích mạng nơ-ron đồ thị GNN là sử dụng nguyên tắc Nút thắt thông tin (GIB - Graph Information Bottleneck) được đề xuất các nhà nghiên cứu của đại học Stanford vào năm 2020 [Wu et al.(2020)]. Xét dữ liệu đầu vào $D = (A, X)$ bao gồm thông tin của cấu trúc đồ thị A và đặc trưng của điểm nút X. Trong trường hợp biểu diễn Z chứa thông tin không liên quan từ một trong hai thành phần này, nó sẽ gây hiện tượng quá khớp (Overfitting) và dễ bị tấn công đối kháng (Adversarial Attacks) và dễ bị thay đổi siêu tham số mô hình trong quá trình huấn luyện. Nguyên tắc này nhằm mục đích trích xuất một đồ thị con nhỏ gọn nhưng đầy đủ thông tin từ đồ thị đầu vào, đảm bảo rằng nó giữ lại đủ thông tin để mô hình có thể duy trì dự đoán ban đầu. Một điểm quan trọng của phương pháp GIB là kiểm tra khả năng dự đoán của đồ thị con này. Điều này thường được thực hiện bằng cách sử dụng đồ thị con làm đầu vào cho mô hình GNN và so sánh kết quả dự đoán của nó với kết quả từ đồ thị đầu vào đầy đủ.



Hình 6: GIB được sử dụng nhằm tối ưu hoá biểu diễn Z trong việc trích xuất những thông tin cần thiết tối thiểu từ dữ liệu đầu vào $D = (A, X)$ và dự đoán đầu ra Y.

Và tất nhiên, GIB chỉ hiệu quả khi và chỉ khi mô hình giải thích nằm trong phạm vi phân phối được huấn luyện. Để giải quyết vấn đề này, Zhoumin và các cộng sự đã đề xuất một ý tưởng mới sử dụng những **Đồ thị đại diện (Proxy Graphs)** nhằm mục đích vừa giữ nguyên thông tin nhãn có trong đồ thị G' cũng như tuân thủ theo phân phối của tập dữ liệu đồ thị gốc. Xét phân phối xác suất có điều kiện:

$$P(Y|G') = \mathbb{E}_{\tilde{G}\sim P_{\mathcal{G}}}[P(Y|\tilde{G}) \cdot P(\tilde{G}|G')] \tag{1}$$

Trong công thức (1) ở trên, chúng ta sẽ xử lý vấn đề OOD bằng cách sử dụng Đồ thị đại diện $\tilde{G}$ để dự đoán Y thay vì sử dụng trực tiếp Đồ thị giải thích G'. Và để thuận lợi cho việc tìm hàm tối ưu, chúng ta sẽ xấp xỉ $P(\tilde{G}|G')$ bằng một hàm tham số ký hiệu là $Q_\phi(\tilde{G}|G')$ với ϕ là tập tham số của mô

hình. Hàm tham số này còn được dùng để xấp xỉ phân phối không biết trước $P_{\mathcal{G}}$. Không những thế, chúng ta sẽ áp dụng thêm độ đo khoảng cách của các phân phối Kullback-Leibler (KL) divergence cho $Q_\phi(\tilde{G}|G')$ và $P_{\mathcal{G}}$. Kết quả, chúng ta sẽ được hàm mục tiêu cho đồ thị giải thích G^*:

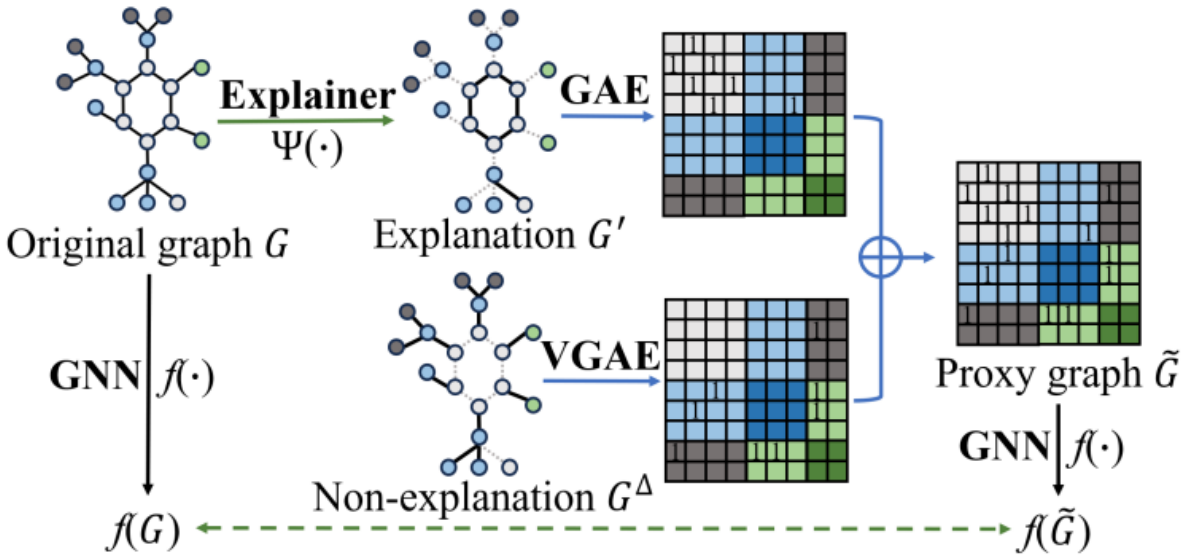
$$\arg\min_{G'} I(G, G') - \alpha \mathbb{E}_{G',Y}\left[\log \mathbb{E}_{\tilde{G}\sim Q^*_\phi(\tilde{G}|G')}[P(Y|\tilde{G})]\right] \tag{2}$$

với $\phi^* = \arg\min_{\phi} KL(Q_\phi(\tilde{G}|G'), P_{\mathcal{G}})$

Với khung hàm quan sát được xây dựng ở trên, chúng ta sẽ có cố gắng đạt được điểm tối ưu ở cả hai cấp độ lồng nhau (bi-level optimization): Tối ưu ngoài (Outer) cho thành phần Giải thích (Explainer) và Tối ưu trong (Inner) cho thành phần tạo sinh đồ thị đại diện (Proxy Graph Generator). Hai thành phần này sẽ được trình bày kỹ hơn ở phần 3 tiếp theo.

3 Giải thích mô hình với Đồ thị đại diện (Proxy Graph)

Dựa vào hướng giải quyết vấn đề của GIB ở phần trước, các tác giả đã đề xuất một kiến trúc mạng mới vừa đơn giản vừa có tính lý thuyết cao, được gọi là ProxyExplainer. Kiến trúc của mô hình sẽ bao gồm hai thành phần chính: Thành phần giải thích (Explainer) và thành phần tạo đồ thị đại diện (Generator). Bộ giải thích nhận đồ thị gốc G làm đầu vào và xuất ra một đồ thị con G' làm giải thích, được tối ưu hóa thông qua mục tiêu bên ngoài trong phương trình 2. Bộ tạo đồ thị đại diện tạo ra một đồ thị proxy trong phân phối, được tối ưu hóa với mục tiêu bên trong.



Hình 7: Đồ thị đại diện (Proxy Graph) bao gồm 2 thành phần chính: Thành phần giải thích (Explainer) và Thành phần tạo sinh(Proxy Graph Generator). Hai thành phần này sẽ được tổng hợp bằng hàm MAX.

3.1 Thành phần giải thích (Explainer)

Đối với thành phần giải thích (Explainer), tác giả sẽ sử dụng các thành phần tạo sinh (generative explainer) vì sự hiệu quả của chúng, đặc biệt là trong việc phan tích các biểu diễn nút (node representations) và nhãn của đồ thị (graph label). Chúng ta sẽ chia mô hình giải thích (to-be-explained model) $f(\cdot)$ làm 2 phần:

- $f_{\text{enc}}(\cdot)$: Đây là thành phần mã hóa (encoder), chịu trách nhiệm trích xuất các biểu diễn nút (node embeddings) dựa trên ma trận đặc trưng X và ma trận kề A. Ma trận X thường chứa thông tin về các đặc trưng của các nút, trong khi A biểu thị cấu trúc kết nối của đồ thị. Thành phần này thường là các mạng nơ-ron sâu để tạo các biểu diễn nút có chất lượng cao.

- $f_{\text{cls}}(\cdot)$: Đây là thành phần phân loại (classifier), thực hiện dự đoán nhãn của đồ thị Y dựa trên các biểu diễn nút Z, trong đó Z được tạo ra từ $f_{\text{enc}}(X, A)$. Thành phần này thường bao gồm một lớp tổng hợp (pooling layer) để tích hợp thông tin từ các nút và kết hợp với các lớp phân loại khác.

Với Z là ma trận biểu diễn các nút, và Y là nhãn dự đoán cho toàn bộ đồ thị, ta có:

$$Z = f_{\text{enc}}(X, A) \quad \text{và} \quad Y = f_{\text{cls}}(Z),$$

.

Và thành phần giải thích này sẽ đóng góp vào hàm mục tiêu tối ưu hóa ngoài (outer optimization objective). Chúng ta sẽ tập trung vào việc giải thích thuật ngữ đầu tiên $I(G, G')$, theo cách tiếp cận của Akin đến đồ thị gốc G. Ở đây, đồ thị giải thích con (explanation subgraph) G' được biểu thị bởi tập các nút V', tập các cạnh E', và ma trận đặc trưng X, với ma trận kề là A'. Chúng ta tham khảo công trình của Miao et al. (2022) để đưa vào một phân phối xấp xỉ biến thiên (variational approximation distribution) $R(G')$ cho phân phối $P(G')$. Sau đó, ta thu được một giới hạn trên (upper bound) như sau:

$$I(G, G') \leq \mathbb{E}_G[\text{KL}(P(G'|G) \| R(G'))].$$

Chúng ta áp dụng mô hình Erdős–Rényi (Erdős et al., 1960) và giả định rằng mỗi cạnh trong G' có xác suất tồn tại hoặc không tồn tại một cách độc lập với các cạnh khác. Cụ thể, sự tồn tại của một cạnh (u, v) trong G' được xác định bởi một biến Bernoulli $A'_{uv} \sim \text{Bern}(\pi'_{uv})$. Do đó, phân phối $P(G'|G)$ có thể được phân tích thành:

$$P(G'|G) = \prod_{(u,v)\in E'} P(A'_{uv}|\pi'_{uv}).$$

Điều này đúng cho bất kỳ phân phối tiên nghiệm (prior distribution) $R(G')$ nào. Chúng ta dựa vào công trình hiện có và giả định rằng sự tồn tại của một cạnh (u, v) trong G' được xác định bởi một biến Bernoulli khác $A'_{uv} \sim \text{Bern}(r)$, trong đó $r \in [0, 1]$ là một siêu tham số (Miao et al., 2022), độc lập với đồ thị G. Vì vậy, ta có:

$$R(G') = P(\mathcal{E}) \prod_{(u,v)\in E'} P(A'_{uv}).$$

Do đó, độ đo KL (KL divergence) giữa $P(G'|G)$ và phân phối trên được sử dụng để tính toán giới hạn trên cho mục tiêu tối ưu hóa.

$$\text{KL}(P(G'|G)||R(G')) = \sum_{(u,v)\in\mathcal{E}} \pi'_{uw} \log \frac{\pi'_{uw}}{r} + (1 - \pi'_{uw}) \log \frac{1 - \pi'_{uw}}{1 - r} + \text{Const.} \tag{3}$$

Biểu thức này đề cập đến việc tính toán KL divergence giữa $P(G'|G)$ và $R(G')$, với các yếu tố liên quan đến xác suất π'_{uw} và r, cùng với sự xuất hiện của hàm mất mát $\mathcal{L}_{exp}$ để huấn luyện bộ giải thích $\Psi(\cdot)$.

3.2 Thành phần tạo sinh Đồ thị đại diện (Proxy Graph Generator)

Chúng ta sẽ minh họa việc tổng hợp một đồ thị proxy thông qua sự kết hợp giữa đồ thị giải thích G' và sự xáo trộn từ đồ thị con không được giải thích $G^{\Delta} = (V, E^{\Delta}, X)$. Xét tập cạnh $E^{\Delta} \subseteq E'$ như một phần tử khác biệt giữa G và G', ma trận kề A^{Δ} được tạo ra từ $A \to A'$. Chúng ta sẽ có một bộ mã hóa để học ma trận Z' dựa trên A' và X', và một bộ giải mã tái tạo lại A' dựa trên Z'.

$$Z' = \text{ENC1}(A', X), \quad A' = \text{DEC}(Z')$$

Với sự xáo trộn vào phần không giải thích, phương pháp tác giả đề xuất cần sử dụng một thành phần Auto-Encoder(VGAE), một mô hình tạo sinh có khả năng tạo ra dữ liệu đồ thị với cấu trúc hợp lý. Quá trình này có vai trò quan trọng trong việc tạo ra các biến thể phức tạp của đồ thị con không giải thích.

$$\mu^{\Delta} = \text{ENC1}(A^{\Delta}, X), \quad \sigma^{\Delta} = \text{ENC2}(A^{\Delta}, X)$$

Các đại diện tiềm ẩn Z^{Δ} được lấy mẫu từ các phân phối này, đảm bảo rằng mỗi lần sinh ra là một biến thể duy nhất của đồ thị ban đầu. Bộ giải mã sau đó tái tạo lại đồ thị con không giải thích đã xáo trộn từ các đại diện tiềm ẩn được lấy mẫu.

$$Z^{\Delta} \sim \mathcal{N}(\mu^{\Delta}, \text{diag}(\sigma^{\Delta})^2), \quad \tilde{A}^{\Delta} = \text{DEC}(Z^{\Delta})$$

Cuối cùng, ma trận kề của đồ thị proxy là $\tilde{A} = A' + \tilde{A}^{\Delta}$, với A^{Δ} là ma trận kề của đồ thị con không giải thích. Việc này nhằm mục tiêu hướng đến tối ưu hóa bên trong (Inner Optimization), chúng ta sẽ triển khai ràng buộc đầu tiên bằng cách tối thiểu hóa khoảng cách phân phối giữa $Q_{\Phi}(G'|G')$ và P_G. Theo giả định Erdős–Rényi, khoảng cách phân phối tương đương với tổn thất entropy chéo giữa $\tilde{G}$ cho G và G' qua ma trận kề đầy đủ A.

$$L_{\text{dist}} = \beta \sum_{(u,v)\in\mathcal{E}} \log(\tilde{p}_{uv}) + \frac{1}{|\mathcal{E}|} \sum_{(u,v)\in\mathcal{E}} \log(1 - \tilde{p}_{uv})$$

Trong đó, $\mathcal{E}$ là tập các cặp nút không liên kết trong G, $\tilde{p}_{uv}$ là xác suất của cặp node (u, v) trong $\tilde{G}$, và β là tham số siêu tham số điều chỉnh sự đánh đổi giữa các cặp nút liên kết và không liên kết.

Ràng buộc thứ hai yêu cầu thông tin hỗn loạn của Y và $\tilde{G}$ giống như của Y và G'. Để giải quyết vấn đề ngoài phân phối (OOD), chúng ta cần triển khai ràng buộc này với một bộ tạo đồ thị, trong đó $\tilde{G}$ được tạo ra từ sự kết hợp giữa G' và đồ thị con không giải thích.

$$\mu^{\Delta} = \text{ENC1}(A^{\Delta}, X), \quad \sigma^{\Delta} = \text{ENC2}(A^{\Delta}, X)$$

Khung phương pháp để giúp tạo ra các sự xáo trộn đa dạng nhưng có tính đại diện, rất quan trọng trong việc cải thiện khả năng giải thích của các bộ giải thích trong khung đồ thị đại diện. Cuối cùng, xét không gian đại diện ẩn (latent space representations) $Z \sim \mathcal{N}(0, I)$, (trong đó I là ma trận đơn vị), hàm mất mát được định nghĩa như sau:

$$L_{\text{proxy}} = L_{\text{dist}} + \lambda L_{\text{KL}},$$

Trong đó, λ là một tham số siêu tham số điều chỉnh, L_{dist} tương đương với entropy chéo giữa $\tilde{G}$ và G. L_{KL} đại diện cho độ lệch Kullback-Leibler giữa phân phối của các đại diện tiềm ẩn Z^{Δ} và phân phối chuẩn Gauss đã giả định. Điều này đảm bảo rằng các sự xáo trộn được tạo ra là có ý nghĩa và có thể kiểm soát được. Hàm mất mát này thường được sử dụng trong các bài toán như Knowledge Distillation, trong đó một mô hình nhỏ (student model) học từ một mô hình lớn hơn (teacher model). Mô hình nhỏ cố gắng tái tạo phân phối xác suất của mô hình lớn thông qua KL-divergence, đồng thời giảm thiểu sai số dự đoán. Phương pháp này giúp mô hình nhỏ học được từ các phân phối xác suất của mô hình lớn mà vẫn giữ được khả năng dự đoán chính xác. Ngoài ra, hàm mất mát này cũng được áp dụng trong các mô hình sinh như Variational Autoencoders (VAEs), nơi KL-divergence được sử dụng để đảm bảo rằng phân phối của các ẩn biến trong mô hình sinh phù hợp với phân phối mục tiêu. Điều này giúp mô hình sinh ra dữ liệu có phân phối tương tự với dữ liệu thật. Việc điều chỉnh hệ số λ cho phép mô hình tìm được sự cân bằng giữa việc duy trì phân phối xác suất hợp lý và tối thiểu hóa sai số dự đoán.

4 Thực nghiệm

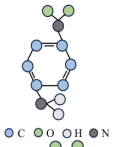
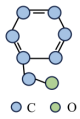
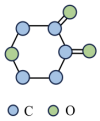
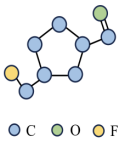
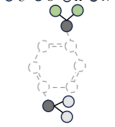
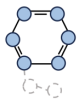
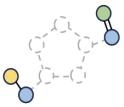
Trong phần thực nghiệm này, tác giả sẽ trình chi tiết các thực nghiệm cũng như kết quả tương ứng nhằm chứng minh độ hiệu quả của phương pháp ProxyExplainer. Các thí nghiệm này chủ yếu được thiết kế để khám phá 3 câu hỏi nghiên cứu sau:

- RQ1: Liệu khuôn khổ đề xuất có thể vượt trội hơn các đường cơ sở khác trong việc xác định các giải thích cho GNN không?
- RQ2: Phân phối có thay đổi nghiêm trọng trong các biểu đồ con giải thích không? Liệu phương pháp đề xuất có thể làm giảm bớt điều đó không?
- RQ3: Mỗi thành phần của ProxyExplainer tác động như thế nào đến hiệu suất chung trong việc tạo ra các giải thích?

4.1 Chuẩn bị dữ liệu và các mô hình

Để đánh giá hiệu suất của ProxyExplainer, các tác giả đã sử dụng 6 bộ dữ liệu chuẩn với các lời giải thích gốc: 4 bộ dữ liệu thực tế và 2 bộ dữ liệu BA bên dưới là nhân tạo.

Table 6. Statistics of molecule datasets for graph classification with ground-truth explanations.

	MUTAG	Benzene	Alkane-Carbonyl	Fluoride-Carbonyl	BA-2motifs	BA-3motifs
Graphs	4,337	12,000	4,326	8,671	1,000	3,000
Average nodes	29.15	20.58	21.13	21.36	25.00	21.92
Average edges	60.83	43.65	44.95	45.37	25.48	29.51
Node features	14	14	14	14	10	4
Original graph						 
Ground truth explanation						 
	NH_2, NO_2	Benzene Ring	Alkane,C=O	F^-,C=O	House, cycle	House,cycle, grid

Hình 8: Các bộ dữ liệu được sử dụng trong thực nghiệm.

- **MUTAG (Kazius et al., 2005)**: Bộ dữ liệu MUTAG bao gồm 4,337 đồ thị phân tử, mỗi đồ thị được phân loại thành hai nhóm dựa trên tác động đột biến của nó đối với vi khuẩn Gram âm S. Typhimurium. Việc phân loại này đến từ các nhóm tuyến virulence đặc biệt với tính đột biến trong việc ánh xạ phân tử theo nghiên cứu của Kazius et al. (Kazius et al., 2005).

- **Benzene (Sanchez-Lengeling et al., 2020)**: Bộ dữ liệu Benzene bao gồm 12,000 đồ thị phân tử từ cơ sở dữ liệu ZINC15 (Sterling, Irwin, 2015), có thể phân loại thành hai lớp. Mục tiêu chính là xác định xem vòng Benzene có tồn tại trong mỗi phân tử hay không. Trong trường hợp có nhiều vòng Benzene, mỗi vòng Benzene được coi là một lời giải thích riêng biệt.

- **Alkane-Carbonyl (Sanchez-Lengeling et al., 2020)**: Bộ dữ liệu Alkane-Carbonyl bao gồm tổng cộng 4,326 đồ thị phân tử, được phân loại thành hai lớp khác nhau. Các mẫu tích cực là các phân tử có cả nhóm chức alkane và carbonyl. Lời giải thích gốc bao gồm các nhóm chức alkane và carbonyl trong một phân tử cho trước.

- **Fluoride-Carbonyl (Sanchez-Lengeling et al., 2020)**: Bộ dữ liệu Fluoride-Carbonyl có 8,671 đồ thị phân tử. Lời giải thích gốc dựa trên sự kết hợp cụ thể của các nguyên tử fluoride và các nhóm chức carbonyl có mặt trong mỗi phân tử.

- **BA-2motifs (Luo et al., 2020)**: Bộ dữ liệu BA-2motifs bao gồm 1,000 đồ thị tổng hợp, mỗi đồ thị được tạo ra từ một mô hình cơ bản Barabasi-Albert (BA). Bộ dữ liệu được chia thành hai loại: một phần của các đồ thị thêm các mẫu mô phỏng cấu trúc của một ngôi nhà, và phần còn lại tích hợp các mẫu chu kỳ năm nút. Việc phân loại các đồ thị này phụ thuộc vào các mẫu cụ thể.

- **BA-3motifs (Chen et al., 2023b)**: BA-3motifs là một bộ dữ liệu mở rộng được lấy cảm hứng từ BA-2motifs và bao gồm 3,000 đồ thị tổng hợp. Mỗi đồ thị cơ bản đi kèm với một trong ba mẫu khác nhau: ngôi nhà, chu kỳ hoặc lưới.

Hai mô hình GCN và GIN được sử dụng trong thực nghiệm này với 3 lớp, hàm tối ưu Adam,.. giống như các cấu hình ở những nghiên cứu trước đó (Ying et al., 2019; Luo et al., 2020; Sanchez-Lengeling et al., 2020). Để đánh giá chất lượng giải thích, thực nghiệm sẽ thực hiện bài toán phân loại nhị phân

các cạnh. Các cạnh là một phần của đồ thị con thực tế được gắn nhãn là dương, trong khi tất cả các cạnh khác được coi là âm. Các trọng số quan trọng do các phương pháp giải thích đưa ra được diễn giải là điểm dự đoán. Một kỹ thuật giải thích hiệu quả là kỹ thuật gán trọng số cao hơn cho các cạnh bên trong đồ thị con thực tế so với các cạnh bên ngoài chúng. Chúng ta sử dụng AUC-ROC để đánh giá định lượng (Ying at el., 2019; Luo at el., 2020).

Algorithm 1 Algorithm of ProxyExplainer

Input: A set of graphs $\mathcal{G} = \{G_i\}_{i=0}^{N}$, with each $G_i = (\mathcal{V}_i, \mathcal{E}_i, \boldsymbol{X}_i)$,a pretrain to-be-explained model $f(\cdot)$, hyper parameters α, λ, M, epochs E.
Initialize an explainer function $\Psi_\psi(\cdot)$
epoch $\leftarrow 0$
while epoch $< E$ **do**
  for $G_i \in \mathcal{G}$ **do**
    $G_i' \leftarrow \Psi_\psi(G_i)$
    $G_i^{\Delta} \leftarrow G_i - G_i'$
    Compute $\tilde{\boldsymbol{A}}_i'$ with equation 14
    Compute $\boldsymbol{Z}^{\Delta}$ and $\tilde{\boldsymbol{A}}_i^{\Delta}$ with equation 16
    Compute proxy loss $\mathcal{L}_{\text{proxy}}$ with equation 18
    Update parameters in proxy graph generator with backpropagation
    if epoch$\%M == 0$ **then**
      Compute explainer loss $\mathcal{L}_{\text{exp}}$
      Update parameters in the explainer with backpropagation
    end if
  end for
  epoch $\leftarrow$ epoch $+ 1$
end while

Hình 9: Mã giả mô tả thuật toán của ProxyExplainer

4.2 Đánh giá định lượng (RQ1)

AUC-ROC là một độ đo phổ biến dùng để đánh giá hiệu suất của các mô hình phân loại nhị phân. Để giải thích chi tiết về AUC-ROC, chúng ta sẽ đi qua các khái niệm cơ bản và công thức liên quan.

- **ROC (Receiver Operating Characteristic)**: Đường cong ROC là một đồ thị thể hiện sự trao đổi giữa *Tỷ lệ Dương Thật (TPR)* và *Tỷ lệ Âm Giả (FPR)* tại các ngưỡng phân loại khác nhau.
- **AUC (Area Under the Curve)**: AUC là diện tích dưới đường cong ROC. AUC có giá trị từ 0 đến 1, với giá trị 1 biểu thị mô hình phân loại hoàn hảo và giá trị 0.5 biểu thị mô hình phân loại ngẫu nhiên.

Tỷ lệ Dương Thật (TPR): Còn được gọi là độ nhạy (recall), được tính theo công thức:

$$\text{TPR} = \frac{\text{Số lượng Dương Thật (True Positives, TP)}}{\text{Số lượng Dương Thật (TP)} + \text{Số lượng Âm Giả (False Negatives, FN)}}$$

Tỷ lệ Âm Giả (FPR): Được tính theo công thức:

$$\text{FPR} = \frac{\text{Số lượng Âm Giả (False Positives, FP)}}{\text{Số lượng Âm Giả (FP)} + \text{Số lượng Dương Giả (True Negatives, TN)}}$$

Đường cong ROC vẽ tỷ lệ Dương Thật (TPR) ở trục tung và tỷ lệ Âm Giả (FPR) ở trục hoành. Mô hình lý tưởng sẽ có đường cong ROC gần với điểm (0, 1), tức là có TPR cao và FPR thấp. Đường chéo từ góc dưới trái đến góc trên phải biểu thị một mô hình phân loại ngẫu nhiên (AUC = 0.5).

AUC là diện tích dưới đường cong ROC. AUC có thể được tính thông qua các kỹ thuật tích phân hoặc thông qua các phương pháp xấp xỉ. Một công thức xấp xỉ đơn giản để tính AUC có thể được viết như sau:

$$\text{AUC} = \sum_{i=1}^{N-1} \frac{(FPR_i - FPR_{i-1}) \cdot (TPR_i + TPR_{i-1})}{2}$$

Trong đó: - N là số lượng các điểm trên đường ROC. - FPR_i và TPR_i là các giá trị Tỷ lệ Âm Giả và Tỷ lệ Dương Thật tại điểm i trên đồ thị ROC.

- **AUC = 1**: Mô hình phân loại hoàn hảo, không có lỗi.
- **AUC = 0.5**: Mô hình phân loại ngẫu nhiên, tương đương với việc đoán ngẫu nhiên.
- **AUC < 0.5**: Mô hình phân loại kém hơn việc đoán ngẫu nhiên, có thể cần điều chỉnh hoặc thay đổi mô hình.

Nhìn chung, AUC-ROC là một độ đo được sử dụng để đánh giá hiệu suất của mô hình phân loại nhị phân. Đặc biệt, AUC rất hữu ích khi dữ liệu không cân bằng, vì nó xem xét cả tỷ lệ dương thật và tỷ lệ âm giả trên tất cả các ngưỡng phân loại, giúp cung cấp một cái nhìn tổng thể về khả năng phân loại của mô hình.

Table 1. Explanation accuracy in terms of AUC-ROC on edges.

	MUTAG	Benzene	Alkane-Carbonyl	Fluoride-Carbonyl	BA-2motifs	BA-3motifs
GradCAM	$0.727_{\pm 0.000}$	$0.740_{\pm 0.000}$	$0.448_{\pm 0.000}$	$0.694_{\pm 0.000}$	$0.714_{\pm 0.000}$	$0.709_{\pm 0.000}$
GNNExplainer	$0.682_{\pm 0.009}$	$0.485_{\pm 0.001}$	$0.551_{\pm 0.003}$	$0.574_{\pm 0.002}$	$0.644_{\pm 0.007}$	$0.511_{\pm 0.002}$
PGExplainer	$0.832_{\pm 0.032}$	$0.793_{\pm 0.054}$	$0.660_{\pm 0.036}$	$0.702_{\pm 0.018}$	$0.734_{\pm 0.117}$	$0.796_{\pm 0.010}$
ReFine	$0.612_{\pm 0.004}$	$0.606_{\pm 0.002}$	$0.768_{\pm 0.001}$	$0.571_{\pm 0.000}$	$0.698_{\pm 0.001}$	$0.629_{\pm 0.005}$
MixupExplainer	$0.863_{\pm 0.103}$	$0.611_{\pm 0.032}$	$0.811_{\pm 0.006}$	$0.706_{\pm 0.013}$	$0.906_{\pm 0.059}$	$0.859_{\pm 0.019}$
ProxyExplainer	$\mathbf{0.977}_{\pm 0.009}$	$\mathbf{0.845}_{\pm 0.036}$	$\mathbf{0.934}_{\pm 0.005}$	$\mathbf{0.758}_{\pm 0.068}$	$\mathbf{0.935}_{\pm 0.008}$	$\mathbf{0.960}_{\pm 0.008}$

real-world synthetic

Hình 10: Kết quả độ giải thích của các phương pháp dưới độ đo AUC-ROC

Xét kết quả AUC-ROC ở trên, ProxyExplainer mang lại những lời giải thích chính xác nhất trên tất cả các bộ dữ liệu. Cụ thể, nó cải thiện điểm AUC trung bình thêm 10,6% trên các bộ dữ liệu thực tế và 7,5% trên các bộ dữ liệu tổng hợp so với các phương pháp cơ bản hàng đầu. Việc so sánh với các phương pháp cơ bản làm nổi bật những lợi thế của khung giải thích mà tác giả đề xuất. Bên cạnh đó, ProxyExplainer ghi nhận các yếu tố giải thích tiềm ẩn một cách nhất quán trên nhiều bộ dữ liệu khác nhau. Ví dụ, MixupExplainer thể hiện khả năng tốt trên bộ dữ liệu tổng hợp BA-2motifs nhưng lại hoạt động kém trên bộ dữ liệu thực tế Benzene. Lý do là MixupExplainer phụ thuộc vào giả định độc lập giữa các subgraph giải thích và không giải thích, điều này có thể không đúng với các bộ dữ liệu thực tế. Ngược lại, ProxyExplainer luôn thể hiện hiệu suất cao trên các bộ dữ liệu khác nhau, chứng tỏ tính ổn định và khả năng thích ứng của nó.

Table 2. Explanation accuracy in terms of AP on MUTAG and BA-2motifs.

	MUTAG	BA-2motifs
GradCAM	$0.247_{\pm 0.000}$	$0.664_{\pm 0.000}$
GNNExplainer	$0.232_{\pm 0.001}$	$0.608_{\pm 0.004}$
PGExplainer	$0.611_{\pm 0.024}$	$0.682_{\pm 0.117}$
ReFine	$0.227_{\pm 0.001}$	$0.619_{\pm 0.002}$
MixupExplainer	$0.647_{\pm 0.083}$	$0.787_{\pm 0.073}$
ProxyExplainer	$\mathbf{0.756}_{\pm 0.107}$	$\mathbf{0.839}_{\pm 0.036}$

real-world synthetic

Table 3. Fidelity evaluation on MUTAG and BA-2motifs.

	MUTAG		BA-2motifs	
	$Fid_{\alpha_1,+} \uparrow$	$Fid_{\alpha_2,-} \downarrow$	$Fid_{\alpha_1,+} \uparrow$	$Fid_{\alpha_2,-} \downarrow$
GradCAM	$0.004_{\pm 0.000}$	$0.162_{\pm 0.000}$	$0.072_{\pm 0.000}$	$0.107_{\pm 0.000}$
GNNExp.	$0.031_{\pm 0.001}$	$0.148_{\pm 0.001}$	$0.057_{\pm 0.002}$	$0.132_{\pm 0.001}$
PGExp.	$0.034_{\pm 0.011}$	$0.148_{\pm 0.005}$	$0.065_{\pm 0.017}$	$0.126_{\pm 0.009}$
ReFine	$0.003_{\pm 0.000}$	$0.160_{\pm 0.001}$	$0.060_{\pm 0.005}$	$0.125_{\pm 0.001}$
MixupExp.	$0.037_{\pm 0.006}$	$0.146_{\pm 0.003}$	$0.074_{\pm 0.005}$	$0.112_{\pm 0.003}$
ProxyExp.	$\mathbf{0.040}_{\pm 0.002}$	$\mathbf{0.145}_{\pm 0.001}$	$\mathbf{0.086}_{\pm 0.003}$	$\mathbf{0.106}_{\pm 0.002}$

real-world synthetic

Hình 11: Kết quả so sánh trên AP và Fidelity

Để đánh giá hiệu suất của ProxyExplainer, chúng ta đã sử dụng điểm AP (Average Precision) vì tính quan trọng của độ chính xác đối với lớp dương trong bối cảnh của bài toán này. Như được trình

bày trong Bảng 2, với các điểm AP, ProxyExplainer liên tục vượt trội so với các phương pháp cơ bản khác trên hai bộ dữ liệu chuẩn MUTAG và BA-2motifs. Điều này cho thấy ProxyExplainer mang lại những lời giải thích chính xác hơn, đặc biệt là trong các trường hợp mà độ chính xác của lớp dương là quan trọng. Bên cạnh đó, một số nghiên cứu trước đây, chẳng hạn như SubgraphX (Yuan et al., 2021), đã sử dụng các chỉ số dựa trên độ trung thực để đánh giá. Tuy nhiên, những chỉ số này gặp phải vấn đề với vấn đề OOD (Out-of-Distribution) (Zheng et al., 2024; Amara et al., 2023), làm giảm độ tin cậy của kết quả. Do đó, chúng ta sử dụng các chỉ số Fidelity với độ trung thực mạnh mẽ hơn.

4.3 Đánh giá độ chuyển dịch phân phối (RQ2)

MMD (Maximum Mean Discrepancy) là một độ đo được sử dụng để so sánh khoảng cách giữa hai phân phối xác suất. Trong ngữ cảnh của thống kê đồ thị, MMD có thể được sử dụng để so sánh phân phối bậc, phân phối hệ số cụm (cluster coefficient), và phân phối phổ (spectrum distribution). Giả sử $k(x, y)$ là hàm kernel, MMD giữa hai tập mẫu từ phân phối p và q có thể được định nghĩa như sau:

$$\text{MMD}^2(p \parallel q) = \mathbb{E}_{x,y\sim p}[k(x,y)] - \mathbb{E}_{x,y\sim q}[k(x,y)] - 2\mathbb{E}_{x\sim p,y\sim q}[k(x,y)]$$

Trong đó:

- $k(x, y)$ là hàm kernel, có thể là một hàm đo độ tương đồng giữa hai điểm x và y.
- $\mathbb{E}_{x,y\sim p}[k(x, y)]$ là kỳ vọng của hàm kernel đối với các mẫu từ phân phối p.
- $\mathbb{E}_{x,y\sim q}[k(x, y)]$ là kỳ vọng của hàm kernel đối với các mẫu từ phân phối q.
- $\mathbb{E}_{x\sim p,y\sim q}[k(x, y)]$ là kỳ vọng của hàm kernel đối với các mẫu từ phân phối p và q.

MMD đo lường sự khác biệt giữa hai phân phối bằng cách so sánh các kỳ vọng của hàm kernel giữa các mẫu từ hai phân phối khác nhau. Độ đo này giúp đánh giá mức độ tương đồng hoặc khác biệt giữa các phân phối mà không yêu cầu giả định cụ thể về hình thức của các phân phối đó.

	MUTAG			Benzene			Alkane-Carbonyl			Fluoride-Carbonyl			BA-2motifs			BA-3motifs		
Metric	GT	PGE	Proxy	GT	PGE	Proxy	GT	PGE	Proxy	GT	PGE	Proxy	GT	PGE	Proxy	GT	PGE	Proxy
Deg.	0.614	0.468	**0.123**	0.843	0.393	**0.236**	0.872	0.665	**0.177**	0.638	0.488	**0.196**	0.759	0.496	**0.060**	0.541	0.149	**0.092**
Clus.	**0.003**	**0.003**	0.009	0.009	**0.002**	0.004	**0.011**	**0.011**	**0.011**	**0.012**	**0.012**	**0.012**	**0.447**	0.463	0.584	0.262	0.382	**0.245**
Spec.	0.414	0.341	**0.186**	0.295	0.163	**0.101**	0.596	0.447	**0.049**	0.351	0.315	**0.100**	0.245	0.256	**0.091**	0.217	0.063	**0.062**
Sum.	1.032	0.813	**0.317**	1.147	0.558	**0.341**	1.479	1.123	**0.237**	1.000	0.815	**0.308**	1.451	1.215	**0.735**	1.020	0.594	**0.399**

real-world synthetic

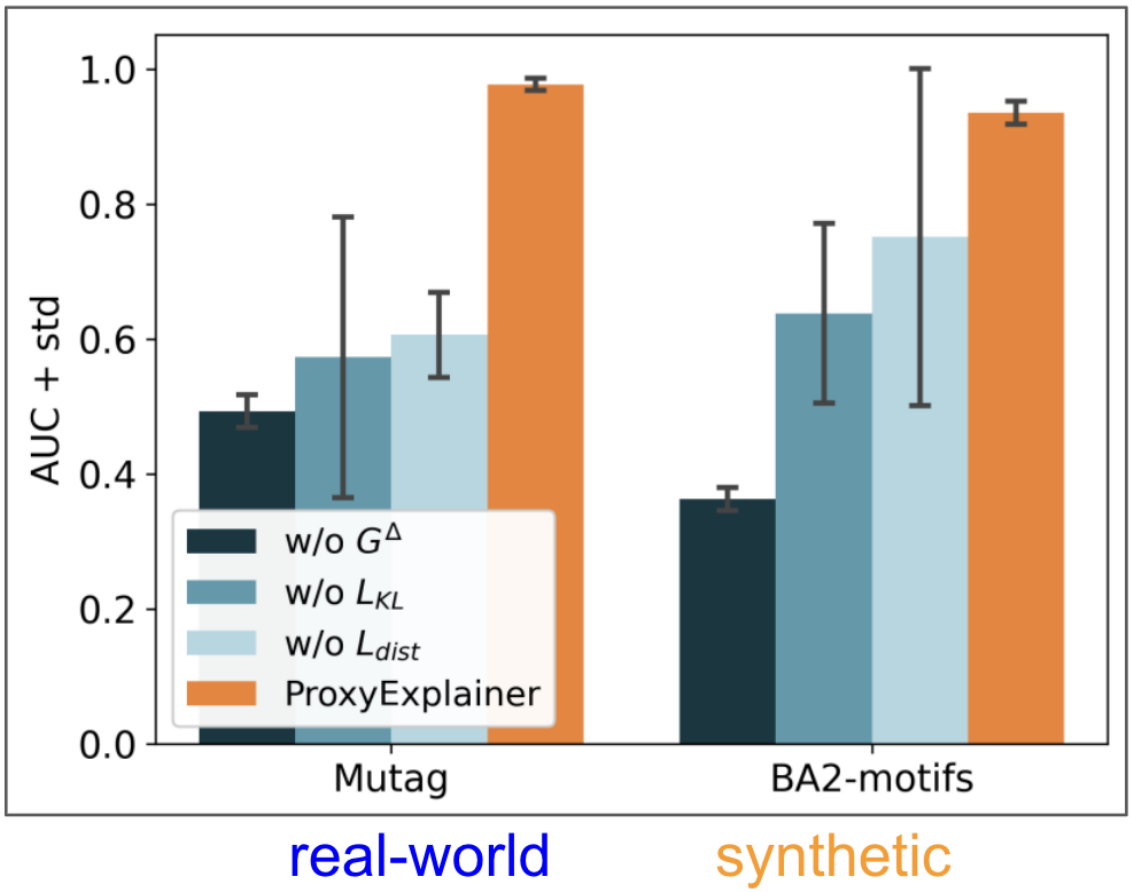
Hình 12: Kết quả MMD

Khả năng của ProxyExplainer trong việc sinh ra các đồ thị proxy nằm trong phân phối. Do tính không thể tính toán trực tiếp, tác giả theo các nghiên cứu trước đây (Chen et al., 2023b) để sử dụng độ chênh lệch trung bình tối đa (Maximum Mean Discrepancy - MMD) giữa các phân phối của nhiều thống kê đồ thị, bao gồm phân phối bậc, hệ số cụm và phân phối phổ, giữa các đồ thị proxy sinh ra và các đồ thị gốc. Cụ thể, tác giả sử dụng hàm kernel Gaussian Earth Mover's Distance khi tính toán MMD. Các giá trị MMD nhỏ hơn chỉ ra rằng phân phối đồ thị tương tự nhau. Để so sánh, tác giả cũng bao gồm các lời giải thích gốc và các lời giải thích được sinh ra bởi PGExplainer.

Kết quả được trình bày ở hình 12 trên. "GT"đại diện cho MMD giữa các lời giải thích gốc và các đồ thị gốc. "PGE"đại diện cho MMD giữa các lời giải thích được sinh ra bởi PGExplainer và các đồ thị gốc. "Proxy"đại diện cho MMD giữa các đồ thị proxy trong phương pháp của tác giả và các đồ thị gốc. tác giả có các quan sát sau đây. Thứ nhất, MMD giữa các lời giải thích gốc và các đồ thị gốc thường lớn, xác nhận động lực của tác giả rằng một mô hình được huấn luyện trên các đồ thị gốc có thể không đưa ra dự đoán chính xác trên các subgraph giải thích ngoài phân phối (OOD). Thứ hai, các lời giải thích được sinh ra bởi một công trình đại diện, PGExplainer, thường là OOD so với các đồ thị gốc, cho thấy rằng hàm mục tiêu dựa trên GIB gốc có thể là chưa tối ưu. Thứ ba, trong hầu hết các trường hợp, các đồ thị proxy được sinh ra bởi phương pháp của tác giả có MMD nhỏ hơn, chứng tỏ tính chất nằm trong phân phối của chúng.

4.4 Đánh giá tầm ảnh hưởng của các thành phần (RQ3)

Trong phần này, tác giả tiến hành các nghiên cứu loại bỏ các thành phần (ablation studies) để nghiên cứu vai trò của các thành phần khác nhau trong `ProxyExplainer`. Cụ thể, tác giả xem xét các biến thể sau của `ProxyExplainer`: (1) *w/o* $G\Delta$: trong biến thể này, tác giả loại bỏ bộ sinh subgraph không giải thích (VGAE), đây là phần dưới của mô hình như đã trình bày trong Hình 2; (2) *w/o LKL*: trong biến thể này, tác giả loại bỏ KL divergence khỏi hàm mất mát trong quá trình huấn luyện của `ProxyExplainer`; (3) *w/o Ldist*: trong biến thể này, tác giả loại bỏ mất mát phân phối khỏi `ProxyExplainer`. Kết quả của nghiên cứu loại bỏ các thành phần trên bộ dữ liệu MUTAG và BA-2motifs được trình bày trong bên dưới.



Hình 13: Kết quả đánh giá tầm ảnh hưởng của các thành phần

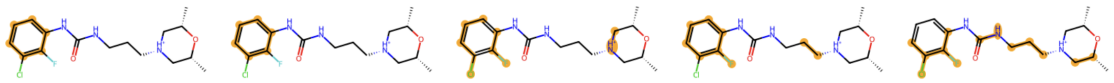
Có thể thấy sự suy giảm đáng kể về hiệu suất đối với tất cả các biến thể, cho thấy rằng mỗi thành phần đóng góp tích cực vào hiệu quả của `ProxyExplainer`. Đặc biệt, trên bộ dữ liệu thực tế MUTAG, khi không có ràng buộc về phân phối, biến thể *w/o Ldist* có hiệu suất kém hơn nhiều so với `ProxyExplainer`, điều này chỉ ra vai trò quan trọng của các đồ thị proxy trong phân phối đối với mô hình của tác giả. Việc loại bỏ mất mát phân phối đã ảnh hưởng đáng kể đến khả năng sinh ra các đồ thị proxy có tính chất nằm trong phân phối, một yếu tố quan trọng giúp mô hình đạt được hiệu quả cao trong các tác vụ thực tế.

4.5 Kết quả bổ sung

Thông tin không cần thiết hoặc thông tin nhiễu có thể gây ra sự mơ hồ trong quá trình giải thích mô hình, làm cho kết quả giải thích không chính xác hoặc khó hiểu. Đặc biệt, trong các bài toán phức tạp như phân tích đồ thị hoặc mạng nơ-ron, việc phải xử lý quá nhiều yếu tố không liên quan có thể dẫn đến các giải thích bị loãng và không tập trung vào những phần quan trọng. Điều này gây khó khăn lớn trong việc hiểu rõ mô hình đang hoạt động như thế nào và tại sao nó lại đưa ra những dự đoán

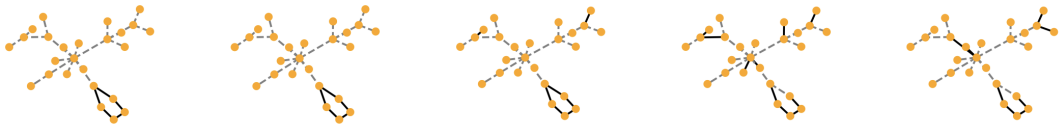
nhất định. Hơn nữa, nếu mô hình không loại bỏ được những yếu tố nhiễu này, người dùng sẽ gặp khó khăn trong việc đưa ra quyết định dựa trên các giải thích, vì các giải thích có thể không phản ánh đúng bản chất của mô hình hoặc gây hiểu nhầm về cách mà mô hình đưa ra kết quả.

Với `ProxyExplainer`, bằng cách học và điều chỉnh một bước về Non-Explanation, phương pháp này có thể giúp mô hình giảm thiểu sự ảnh hưởng của các yếu tố không quan trọng và giữ lại những yếu tố có giá trị giải thích thực sự. Điều này không chỉ giúp tăng cường tính chính xác của các giải thích mà còn giúp tăng cường sự minh bạch của mô hình, giúp người dùng hiểu rõ hơn về cách mà mô hình đưa ra quyết định. Sự cải thiện này đặc biệt quan trọng trong các ứng dụng thực tế, nơi mà việc hiểu và giải thích mô hình là điều kiện tiên quyết để có thể áp dụng mô hình vào các tình huống cụ thể và đưa ra các quyết định có cơ sở.



(a) Ground Truth (b) ProxyExplainer (c) MixupExplainer (d) PGExplainer (e) GNNExplainer

Figure 4. Visualization of explanation results from different explanation models on Benzene. The generated explanations are highlighted with bold orange edges.



(a) Ground Truth (b) ProxyExplainer (c) MixupExplainer (d) PGExplainer (e) GNNExplainer

Figure 5. Visualization of explanation results from different explanation models on BA-2motifs. The generated explanations are highlighted with bold black edges.

Hình 14: Mô tả một số kết quả giải thích của ProxyExplainer với những phương pháp khác

Một cách khác để xấp xỉ sự khác biệt phân phối giữa các đồ thị là so sánh các vector nhúng (embedding) của chúng. Ở đây, tác giả áp dụng cùng một cài đặt và sử dụng độ tương đồng Cosine (Cosine similarity) và khoảng cách Euclidean (Euclidean distance) để xấp xỉ trực quan sự tương đồng giữa các phân phối đồ thị. Bảng 8 trình bày các giá trị độ tương đồng Cosine và khoảng cách Euclidean được tính toán giữa các nhúng phân phối của đồ thị gốc h, subgraph giải thích gốc h', và đồ thị proxy sinh ra $\tilde{h}$. Đặc biệt, nhúng đồ thị proxy $\tilde{h}$ của tác giả thể hiện điểm độ tương đồng Cosine cao hơn và khoảng cách Euclidean thấp hơn với nhúng đồ thị gốc h so với nhúng giải thích gốc h'. tác giả quan sát được sự cải thiện trung bình là 19.5% về độ tương đồng Cosine và 35.6% về khoảng cách Euclidean. Đặc biệt, trên bộ dữ liệu BA-2motifs, có sự cải thiện đáng kể với 60.4% về độ tương đồng Cosine và 51.8% về khoảng cách Euclidean. Những phát hiện này làm nổi bật hiệu quả của phương pháp `ProxyExplainer` trong việc giảm thiểu sự chuyển dịch phân phối gây ra bởi độ thiên lệch suy luận trong mô hình GNN cần giải thích $f(\cdot)$, qua đó nâng cao hiệu suất giải thích.

Table 8. The Cosine similarity score and Euclidean distance between the distribution embeddings of the original graph $\boldsymbol{h}$, explanation subgraph $\boldsymbol{h}'$, and our proxy graph $\tilde{\boldsymbol{h}}$ on different datasets.

	MUTAG	Benzene	Alkane-Carbonyl	Fluoride-Carbonyl	BA-2motifs	BA-3motifs
Avg. Cosine($\boldsymbol{h}, \boldsymbol{h}'$) ↑	0.883	0.835	0.889	0.904	0.571	0.686
Avg. Cosine($\boldsymbol{h}, \tilde{\boldsymbol{h}}$) ↑	**0.985**	**0.905**	**0.938**	**0.908**	**0.916**	**0.918**
Avg. Euclidean($\boldsymbol{h}, \boldsymbol{h}'$) ↓	0.975	1.010	0.940	0.806	1.210	1.199
Avg. Euclidean($\boldsymbol{h}, \tilde{\boldsymbol{h}}$) ↓	**0.368**	**0.767**	**0.719**	**0.779**	**0.583**	**0.613**

Hình 15: Một số kết quả đo lường khoảng cách của các phân phối trong các bộ dữ liệu

5 Kết luận

Nhìn chung, đây là một nghiên cứu khá thú vị về vấn đề ít được chú trọng, đặc biệt là trong việc huấn luyện các mô hình mạng nơ-ron đồ thị có khả năng giải thích: hiện tượng OOD(Out-Of-Distribution). Hiện tượng này xảy ra khi mô hình được huấn luêyjn trên một tập dữ liệu nhất định, nhưng sau đó được áp dụng trên một dữ liệu mới có phân bố quá khác biệt. Việc này dẫn đến kết quả không chính xác và lời giải thích không đáng tin cậy. Và để giải quyết vấn đề này, nhóm tác giả đã cải tiến GIB (Graph Information Bottleneck - một nguyên lý thông tin trong diễn giải đồ thị) bằng cách tạo ra các Đồ thị đại diện nằm trong cùng phân phối với đồ thị huấn luyện (In-Distributed Proxy Graph). Cách tiếp cận này vừa giúp mô hình học được cách biểu diễn đồ thị chính xác hơn, giúp việc phân loại cạnh và khả năng giải thích của nó tốt hơn so với các mô hình khác được đề cập trên cả về tập dữ liệu thực tế lẫn tập dữ liệu nhân tạo. Điều này giúp mở rộng hơn các hướng phát triển mới nhằm giải quyết vấn đề OOD không chỉ ở cấp độ các nút mà còn các cấp độ cao hơn như đồ thị con,... hoặc các cấu trúc dữ liệu khác như hình ảnh, âm thanh, chuỗi thời gian,... Một số nhận xét khác khi thử nghiệm lại nghiên cứu của tác giả (https://github.com/MinhVu2018/GNN_ProxyExplainer), chính thành phần cần được giải thích từ những bộ dữ liệu này khá được kết nối dày đặc. Những bộ dữ liệu được thực hiện trong bài báo này đa phần là về sinh học nên cũng khá nhỏ và không thưa như các mạng lưới thực tế về giao thông, mạng xã hội,... Thêm vào đó, quá trình huấn luyện của ProxyExplainer đòi hỏi nhiều tài nguyên hơn khi phải học cả 2 thành phần Explanation và Non-Explanation. Điều này tuy giảm được độ nhiễu của những dữ liệu nằm ngoài vùng phân phối, tuy nhiên kết quả cũng tốt hơn không quá nhiều nhưng phải huấn luyện tốn kém hơn 2 lần.

Tài liệu

[Finkelshtein et al.(2024)] Ben Finkelshtein, Xingyue Huang, Michael Bronstein, and İsmail İlkan Ceylan. 2024. Cooperative Graph Neural Networks. arXiv:2310.01267 [cs.LG] https://arxiv.org/abs/2310.01267

[Wu et al.(2020)] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. 2020. Graph Information Bottleneck. *CoRR* abs/2010.12811 (2020). arXiv:2010.12811 https://arxiv.org/abs/2010.12811

[Yuan et al.(2020)] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. Explainability in Graph Neural Networks: A Taxonomic Survey. *CoRR* abs/2012.15445 (2020). arXiv:2012.15445 https://arxiv.org/abs/2012.15445

[Zhang et al.(2023)] Jiaxing Zhang, Dongsheng Luo, and Hua Wei. 2023. MixupExplainer: Generalizing Explanations for Graph Neural Networks with Data Augmentation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. ACM, 3286–3296. https://doi.org/10.1145/3580305.3599435

[Zhou et al.(2021)] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2021. Graph Neural Networks: A Review of Methods and Applications. arXiv:1812.08434 [cs.LG] https://arxiv.org/abs/1812.08434

[Zhou et al.(2018)] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. *CoRR* abs/1812.08434 (2018). arXiv:1812.08434 http://arxiv.org/abs/1812.08434