JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 27, No. 2, pp. 493-501, Feb. 2026

Check for updates

Hybrid-GM: 실행시간 예측 정확도 향상을 위한 쿼리 비용 평가 모델

박 화 진[1*] · 최 옥 주[2]
[1]숙명여자대학교 공과대학 인공지능공학부 교수
[2]배재대학교 AI·소프트웨어공학부 조교수

Hybrid-GM: A Query Cost Estimation Model for Improving Execution Time Prediction Accuracy

Hwa-Jin Park[1*] · Ok-Joo Choi[2]

[1]Professor, Division of AI Engineering, Sookmyung Women's University, Seoul 04310, Korea
[2]Assistant Professor, Department of AI·Software Engineering, PAI CHAI University, Daejeon 35345, Korea

[요 약]

데이터베이스 시스템의 성능 향상에 필수적인 SQL 쿼리 최적화는 지속적인 연구 주제이다. 기존의 비용 기반 최적화 방식은 데이터 규모 증가 및 쿼리 구조 복잡성 증대에 따라 통계적 오류와 부정확한 카디널리티 추정으로 인해 실제 실행 비용을 정확히 예측하기 어려운 한계점을 노출하고 있다. 이 문제를 해결하기 위해 본 연구는 GNN의 그래프 구조와 전역 특성 통계값을 통합하는 Hybrid-GM모델을 제안한다. 이 모델은 MLP 인코더로 전역 쿼리 특성을, GNN 인코더로 실행 계획의 구조적 특성을 학습하여 쿼리 실행 비용을 예측한다. TPC-H 벤치마크 실험 결과, Hybrid-GM 모델은 기존 MLP 및 GNN 단독 모델 대비 RMSE와 MAE 지표에서 74.3% 및 68.6% 등의 향상된 성능을 보였다. 본 연구는 학습 기반 쿼리 최적화 분야의 발전에 크게 기여할 것으로 기대한다.

[Abstract]

SQL query optimization, essential for improving database system performance, remains an active area of research. Traditional cost-based optimization methods, however, are limited in accurately predicting actual execution costs due to statistical errors and imprecise cardinality estimations, especially with increasing data scales and query complexity. This study proposes a Hybrid-GM model that integrates the graph structure of graph neural network (GNNs) with global query feature statistics. This model employs an multi-layer perceptron (MLP) encoder to capture global query features and a GNN encoder to learn the structural properties of query execution plans, enabling accurate prediction of query execution costs. The Hybrid-GM model demonstrated improved performance compared to standalone MLP and GNN models, achieving relative reductions of 74.3% and 68.6% in root mean squared error (RMSE) and mean absolute error (MAE), respectively. This study is expected to make a significant contribution to the advancement of machine learning-based query optimization.

색인어 : 쿼리 옵티마이저, 쿼리실행계획, 그래프 신경망, 다층 퍼셉트론, 하이브리드 GNN-MLP

Keyword : Query Optimizer, Query Execution Plan, Graph Neural Network (GNN), Multi-Layer Perceptron (MLP), Hybrid GNN-MLP

http://dx.doi.org/10.9728/dcs.2026.27.2.493

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 28 December 2025; **Revised** 15 January 2026
Accepted 21 January 2026

***Corresponding Author; Hwa-Jin Park**

Tel: ※ 개인정보 표시제한
E-mail: phj2000@sookmyung.ac.kr

Ⅰ. 서 론

데이터베이스 시스템(Database System, DBS)에서 쿼리 옵티마이저는 사용자가 작성한 논리적 쿼리를 실제로 실행할 수 있는 물리적 계획으로 변환한다. 이러한 쿼리의 실행 계획(Execution Plan)은 실행비용을 낮추기 위한 최적의 경로를 찾는 핵심 역할을 한다[1]. 동일한 SQL(Structured Query Language)쿼리라도 인덱스 사용 유무, 조인 순서 및 알고리즘, 물리적 연산자에 따라 실행비용이 수백 배 차이가 발생할 수 있으므로 최소의 실행비용을 찾는 쿼리 옵티마이저가 DBS성능을 좌우한다[2].

최근에는 데이터 세트의 규모가 커지고 다양한 데이터 요구사항으로 인해 쿼리 구조가 점점 복잡해지고 있다. 이러한 상황에서는 기존의 규칙기반이나 비용 기반 쿼리 옵티마이저는 최적의 실행계획을 찾는 데 한계가 발생한다. 전통적인 쿼리 옵티마이저는 미리 정해진 경험적인 규칙에 의존하거나 실행 통계 자료에 기반하고 있어서 비용 모델이 정확하지 않거나 오래된 통계를 사용할 경우 비효율적인 실행 계획이 나오기 때문이다[3],[4].

이러한 한계를 극복하기 위해 최근에는 머신러닝 기술을 기반으로 학습형 쿼리 옵티마이저에 대해 활발하게 연구되고 있다[5]. 학습형 쿼리 옵티마이저는 기존의 쿼리 최적화 방식에 머신러닝 기법을 접목하여 보다 효율적인 쿼리 계획을 생성하는 기술이다. 특히 쿼리 실행 계획이나 실행 그래프를 분석하기 위한 그래프 신경망(Graph Neural Network, GNN) 활용에 대한 연구가 활발히 진행되고 있다[6]. GNN은 쿼리 연산자들의 복잡한 관계와 데이터 흐름을 분석하여 기존의 옵티마이저가 최적화하기 어려운 실행 계획 패턴을 학습한다.

본 논문에서는 SQL 쿼리 성능을 향상시키기 위해 쿼리 최적화 모델의 예측력을 개선하기 위해 GNN과 다층 퍼셉트론(Multi-Layer Perceptron, MLP)을 결합한 Hybrid GM(GNN-MLP) 모델을 제안한다. 본 모델은 쿼리 옵티마이저에서 MLP가 연산자 노드의 특성을 학습하는 능력과 GNN의 구조적 관계를 파악하는 능력을 결합한다. GNN은 쿼리 계획을 구성하는 연산자들 간의 부모-자식 관계, 데이터 흐름, 조인 순서 등 구조적 정보를 통합해 쿼리 계획 전체의 맥락과 연산자 간 상호작용을 효과적으로 모델링한다. 또한 MLP는 각각의 쿼리 연산자의 특성(선택도, 카디널리티, 사용 인덱스 등)을 세밀하게 분석하고 이를 고차원 특성 공간으로 바꾼다. 이로 인해 각각의 연산자가 전체 쿼리 계획 비용에 어떤 영향을 미치는지 개별적으로 파악할 수 있다. Hybrid-GM모델은 개별 연산자의 세세한 정보와 쿼리 실행 계획 내의 구조적 특징을 모두 고려하여 GNN 기반 쿼리 최적화 방식에서의 부분적인 패턴을 보완하고 쿼리 실행 비용 예측의 정확도를 높이는 게 목적이다.

본 연구의 주요 기여는 다음과 같이 요약된다.

① 쿼리 실행 계획의 구조적 임베딩과 전역 특성 임베딩을 결합하는 Hybrid GM 프레임워크를 제안한다.

② 동일 쿼리에서 파생된 실행 계획 간의 데이터 누수를 방지하기 위해 쿼리 단위 분할(query-level split) 기반의 학습 및 평가 프로토콜을 설계하고, 반복 실험을 통해 모델의 안정성과 일반화 성능을 검증한다.

③ GNN 단독 모델, MLP 단독 모델과의 비교 실험을 통해, 제안 모델의 성능 향상과 결합 효과를 정량적으로 분석한다.

실험 결과, 제안하는 Hybrid GM 모델은 기존 기준선 모델 대비 실행 시간 예측 정확도 측면에서 일관된 성능 향상을 보였으며, 특히 구조적 복잡성이 높은 실행 계획 환경에서 그 효과가 더욱 두드러짐을 확인하였다.

본 연구를 통해 복잡한 쿼리 실행 비용 예측 정확도를 제고함으로써 데이터베이스 관리자가 쿼리 옵티마이저의 의사결정에 대한 신뢰성을 높이는 데 기여할 것으로 기대된다.

본 논문의 구성은 다음과 같다.

제 2장에서는 연구 배경 및 관련 연구로서 쿼리 옵티마이저의 연구 동향을 파악하고 제 3장에서는 본 연구의 기본 모델인 GNN 기반 쿼리옵티마이저와 다층 퍼셉트론(MLP)의 이론적 배경을 검토한다. 또한 기존 GNN 모델의 한계를 극복하기 위해 제안하는 Hybrid 모델의 구조와 설계 방법을 상세히 기술한다. 제 4장에서는 실험 환경과 데이터셋을 소개하고, 기존 모델 대비 개선시킨 실험 결과를 분석한다. 마지막으로 제5장에서는 연구의 결론을 맺고 향후 연구 방향을 제시한다.

Ⅱ. 연구 배경 및 관련 연구

데이터베이스 시스템(Database System, DBS)은 대규모 데이터 처리와 높은 사용자 요구사항을 만족시키기 위해 효율적인 쿼리 실행이 필수적이다. 단일 SQL 쿼리는 다양한 실행 계획(Query Execution Plan, QEP)으로 변환될 수 있으며 선택된 실행 계획의 품질에 따라 처리 시간은 수 밀리초에서 수 분 이상까지 차이가 발생한다. 따라서 최적의 실행계획을 구하는 쿼리 옵티마이저는 DBS 성능에 매우 중요한 기능이다.

2-1 전통적인 쿼리 옵티마이저

대부분 현대의 관계형 데이터베이스 관리 시스템(Relational Database Management System, RDBMS)은 비용 기반 또는 규칙과 비용 기반 쿼리 옵티마이저 혼합 형태로 사용하고 있다.

1) 규칙 기반 쿼리 옵티마이저

규칙 기반 쿼리 옵티마이저(Rule-based Query Optimizer, RBO)는 쿼리의 논리 형태를 정해진 규칙 및 휴리스틱 형태로 변환한다[7]. RBO는 쿼리의 구조적 규칙에 따라 최적의 접근 경로 및 조인 순서를 미리 정해 놓는다. 쿼

리가 입력되면 쿼리를 실행하기 전에 다양한 변환 규칙을 적용하여 쿼리 표현을 개선한 후에 최적화 단계를 진행한다. 구현이 용이하고 이해하기 쉬우나 복잡한 쿼리나 조인 순서에서 한계가 발생한다.

2) 비용 기반 쿼리 옵티마이저

비용 기반 쿼리 옵티마이저(Cost-based Query Optimizer, CBO)는 쿼리의 다양한 실행계획의 비용을 비교 분석하여 가장 효율적인 실행계획을 선택한다[8]. CBO는 테이블의 크기 인덱스 유무, 조인 카디널리티 등의 통계정보를 사용하는 비용 모델이다. 데이터 정보 반영과 좋은 실행계획을 얻을 수 있는 반면 통계정보의 정확성이나 매우 복잡한 쿼리의 카디널리티 추정 오류에 따라 실행계획의 품질이 달라질 수 있다.

2-2 학습형 쿼리 옵티마이저

최근 인공지능의 발전과 함께 기존의 전통적인 옵티마이저 한계를 극복하기 위한 머신러닝(Machine Learning, ML) 모델을 활용한 학습형 쿼리 옵티마이저(Learned Query Optimizer, LQO)에 대한 연구가 활발하게 진행되고 있다. Neo는 전통적인 옵티마이저를 기반으로 딥러닝 모델을 사용하여 쿼리 실행계획의 비용을 예측하는 모델이다[9]. Balsa는 강화학습을 기반으로 쿼리 옵티마이저를 학습하는 프레임워크이다[10]. LEON은 기존의 옵티마이저를 대체하기보단 보조적으로 ML을 결합한 프레임워크로서 실제 DBMS 환경에 적용 가능성을 고려하여 설계되었다[11].

2-3 그래프 신경망 기반 쿼리 옵티마이저

그래프 신경망(Graph Neural Network, GNN)은 그래프 구조 데이터를 학습하기 위해 설계된 딥러닝 모델이다. GNN은 노드, 엣지, 특성(feature)으로 구성된 그래프를 입력으로 받아서 이웃 노드들의 정보를 집계하여 노드 표현을 갱신하는 과정을 반복하면서 그래프의 구조적 의미를 학습한다[12]. GNN은 노드 간 관계, 조인 연산 순서, 연산 종속성 등을 직접 모델링할 수 있어 기존 머신러닝 모델보다 학습 효율이 높다는 장점이 있다. 초기의 GNN 기반 쿼리 옵티마이저는 쿼리를 조인 그래프로 표현하여 GNN 메시지 패싱으로 조건/조인 관계를 학습하여 기존의 옵티마이저 대비 오류를 크게 개선하였다[13]. MSCN(Multi-Set Convolutional Network)이나 NeuroCard 연구는 GNN 기반 옵티마이저 베이스라인으로 사용됨으로써 테이블 조인. 필터를 임베딩하여 카디널리티를 추정하였다[14],[15]. 또한, GNN과 강화학습을 결합하여 조인의 순서를 최적화한 쿼리 계획을 탐색하는 연구가 진행되고 있다[16].

Ⅲ. Hybrid-GM 모델 설계

3-1 Hybrid-GM 모델 설계 배경

1) MLP 기반 쿼리 옵타마이저

초기 연구들은 데이타베이스 쿼리 성능 예측을 위해 주로 전역 통계 특성을 입력으로 하는 다층 퍼셉트론(MLP) 기반 회귀 모델을 사용하였다. 이러한 접근법에는 옵티마이저가 제공하는 추정 카디널리티, 연산비용, 테이블 크기, 조인 수 등과 같은 요약 통계가 모델의 주요 입력으로 활용되었다. MLP는 비교적 단순한 구조에도 불구하고 비선형 관계를 근사할 수 있으므로, 전통적인 회귀 방법에 비해 성능 향상이 있었다. 그러나 MLP 기반 접근법은 쿼리 실행 계획의 구조적 특성을 명시적으로 반영하지 못한다는 근본적인 한계를 가진다. 동일한 전역 통계값을 갖는 실행계획도 연산자의 배치 순서, 트리 깊이 및 서브플랜 간 상호작용에 따라 실제 실행 시간은 크게 달라질 수 있다. 이러한 구조적 요인은 단순 전역 특성 벡터로는 충분히 표현되지 않으며, 결과적으로 모델의 일반화 성능 저하로 이어질 수 있다. 그런데도 MLP 기반 모델은 계산 비용이 적고 구현이 간단하며 전역 특성 벡터를 반영할 수 있다는 점에서 본 연구의 주요 구성 모델로 채택하였다.

2) GNN 기반 쿼리 옵타마이저

최근 연구에서는 쿼리 실행계획을 트리 또는 일반 그래프로 간주하고 이를 처리하기 위해 그래프 신경망(Graph Neural Network, GNN)을 활용하는 접근이 활발히 수행되고 있다. 이 방법은 쿼리 계획을 노드와 엣지의 집합으로 정의하고, 각 노드에 연산자 타입 및 통계 특성을 할당하여 실행계획의 구조정보를 직접적으로 모델링한다.

메시지 패싱(Message Passing)기반의 GNN 노드가 이웃 노드로부터 반복적으로 정보를 수집하고 이를 통합함으로 이웃의 범위를 넓혀가며 상호작용을 하여 학습할 수 있다. 이와 같은 메커니즘은 특히 조인 순서 결정, 파이프라인 병렬성, 연산자 중복성 등 구조적 의존성이 중요한 데이터베이스 성능 예측 문제에서 유리하다. 실제로 단순한 MLP 기반 모델보다 실행비용 및 지연시간을 더욱 정확하게 예측하기도 한다.

그러나 GNN 기반 모델 역시 한계가 있다. 대부분의 연구에서 노드 중심의 지역구조 학습에 집중하며 데이터베이스 전체 수준의 통계나 워크로드 수준 맥락과 같은 전역 정보를 직접적으로 통합하지 못하는 경우가 많다. 또한 쿼리에 따라 모델 복잡도가 높고 학습비용이 많이 드는 경우도 있다.

3) Hybrid GM 기반 쿼리 옵타마이저

본 연구에서는 두 모델을 활용하여 두 개의 상반된 특성들을 결합하고자 하는 의도에서 출발하였다. 그리하여 GNN을 통한 구조적 표현학습과 MLP를 통한 전역 특성 표현학습을 수행하고 두 표현을 융합하여 비용을 예측하는 Hybrid GM(GNN-MLP)을 제안하고자 한다.

3-2 Hybrid-GM 모델 설계 개요

본 연구에서는 SQL 쿼리 실행 계획(Query Execution Plan, QEP)을 기반으로 하여 쿼리 실행계획내 연산자들의 구조적 상호작용과 쿼리 전체 수준에서 관측되는 전역 통계정보를 동시에 반영하는 모델을 제안한다.

모델은 그림 1에서 보는 바와 크게 두 부분, 즉 특성 변환(feature transformation)과 융합 및 비용 예측(fusion & cost prediction)으로 분류된다. 그리고 특성 변환에는 그래프 구조 특성 변환과 전역 특성 변환으로 구분한다.

1) 특성 변환

실행계획을 입력으로 받아 각 변환 유형별로 임베딩을 출력한다.

• GNN 기반 그래프 임베딩

Graph repesentaion, GNN 인코더, Graph-level Embedding으로 구성되어 있다. 실행 계획 트리를 그래프 구조로 변환한 후 GNN 인코더에서는 GCN(Graph Convolution Network)를 사용하여 노드 간 종속성 및 조인 순서 정보를 통합한다. 직접 연결된 이웃 노드를 1-hop이라 하며 k-hop 의 message passing을 통하여 그래프 전체를 대표하는 그래프 레벨 임베딩을 생성한다.

• MLP (Multi Layer Perceptron) 기반 전역특성표현

특성추출, MLP 인코더 및 MLP embedding으로 구성되어 있다. 실행 계획에서 추출한 통계적 특성을 입력으로 사용하여 특성 벡터를 고차원으로 변환하여 노드 임베딩을 한다.

2) 융합 및 비용 예측

융합층과 MLP 회귀 네트워크로 구성되어 있다. 그래프 레벨 임베딩과 MLP 특성 노드 임베딩을 고차원으로 연결하여 MLP 회귀 네트워크에 입력한다. 예측의 목표변수는 실제 실행 시간으로 세팅하여 예측정확도를 산출한다.

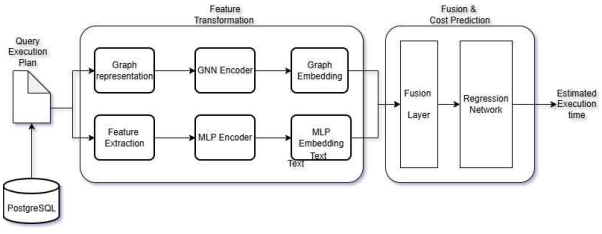

Fig. 1. Hybrid GM model framework
그림 1. Hybrid GM 모델 프레임워크

3-3 세부 구성요소

Hybrid GM 모델의 구성요소를 세부적으로 설명한다.

1) 실행계획

입력 SQL 쿼리는 PostgreSQL 기반의 실행 계획 추출 과정을 통해 정량적 특성과 그래프 구조 표현으로 분리된다. 쿼리 전역 속성(예: Estimated Cost, Join Count, Predicate Selectivity)은 다층 퍼셉트론(MLP)을 통해 임베딩 되며 실행 계획의 트리구조는 그래프 구조 즉 노드 및 엣지로 변환되어 GNN 인코더에 입력된다.

2) 그래프 표현

JSON 파일의 실행계획에는 트리 형태로 구성되어 있다. 이러한 구조를 그래프 구조에 맞춰 노드와 엣지의 형태로 표현한다. 여기서 노드는 실행 연산자로 seq scan, index scan, hash join, aggregate 등으로 구성되어 있고 노드 특성은 연산자 종류, 추정 카디널리티, 추정 비용, 출력 행수 등이 포함된다. 엣지는 연산자 간의 데이터 흐름 또는 상하관계를 연결한다.

3) GNN 인코더

쿼리 실행계획은 방향그래프 G = (V, E)로 표현되며 각 노드 $v \in V$ 는 조인, 스캔, 정렬과 같은 연산자를 나타낸다.

각 노드는 연산자 수준의 특성 벡터 $X_v \in R^{d_n}$ 을 가지며 GNN은 메시지 패싱 과정을 통해 다음과 같이 노드 임베딩을 업데이트한다.

$$h_v^{(k)} = \phi(h_v^{(k-1)}, \bigoplus_{u \in N(v)} \psi(h_u^{(k-1)})) \quad (1)$$

여기서 ϕ와 ψ는 학습가능한 함수이며 $N(v)$는 이웃 노드의 집합이다. 최종적으로 이 노드 임베딩들은 풀링연산을 통해 그래프 수준 임베딩으로 집계된다[18].

4) 그래프 수준 임베딩

그래프 수준 임베딩을 생성하는 것을 READOUT 연산이라고 하며 노드별 벡터들을 하나의 고정길이 벡터로 변환하는 역할을 한다. 이 READOUT을 실제로 구현하는 방법에는 합계, 최대, 평균 풀링등을 통하여 구해지는데 본 연구에서는 평균 풀링을 적용하였다. 즉, 메시지 패싱을 통해 얻은 노드 임베딩들을 평균함으로써 그래프 전체를 대표하는 고정길이 벡터를 구성하였다. 수식으로 표현하면 다음과 같다.

$$z_G = \frac{1}{N}\sum_{i=1}^{N} h_i^{(L)} \quad (2)$$

여기서 $h_i^{(L)}$ 는 최종 GNN 레이어 L에서의 노드 i 의 은닉표현이며 z_G는 그래프 수준 임베딩을 의미한다.

5) 특성 추출(feature Extraction)

실행계획으로부터 MLP 인코더에 사용될 전역적 특성을 추출하기 위해 정량적 속성을 구성하였다. 실행계획은 주어진 SQL 쿼리를 수행하기 위해 연산자들의 구조와 실행전략을 기술하는 트리 형태의 질의 수행계획으로 쿼리의 복잡도 및 예상 실행 자원 요구량을 반영한다. 본 연구에서는 시스템 전역적인 지표를 추출하여 MLP의 입력으로 사용한다. 전역 특성 추출 과정은 다음과 같은 단계로 구성된다. QEP의 각 연산자 노드 및 서브플랜 단위에서 제공되는 통계정보와 비용정보를 수집한다. 구체적으로는 추정 카디널리티, 연산자별 예상 I/O 비용, CPU 비용, 전체 계획 비용, 조인수, 연산자 깊이, 실행계획 트리의 높이, 및 리프노드 수 등을 포함한다. 이러한 특성들은 모두 스칼라값으로 정규화된 후 하나의 전역 특성 벡터로 결합하며 이는 다음과 같이 표현할 수 있다.

$$g = [f_1, f_2, ..., f_k]^T \in R^K \tag{3}$$

여기서 f_k는 QEP로부터 추출된 k번째 전역 특성이며, K는 전역 특성의 수를 의미한다.

6) MLP 인코더(MLP Encoder)

전역 특성 벡터

$$g \in R^{d_g} \tag{4}$$

는 실행계획 전체에 대한 통계정보 즉, 추정비용, 카디널리티, I/O 통계 등을 포함한다. MLP Encoder는 입력 벡터를 x_{MLP} 라고 할 때 MLP는 다다음과 같은 계층적 변환으로 정의된다.

$$\begin{aligned} h^{(1)} &= \alpha(W_1 x_{MLP} + b_1) \\ h^{(2)} &= \alpha(W_2 h^{(1)} + b_2) \\ z_{MLP} &= W_3 h^{(2)} + b_3 \end{aligned} \tag{5}$$

여기서 α는 ReLU 와 같은 활성화 함수이며 x_{MLP} 는 수치 특성기반 잠재 표현이다. MLP 인코더는 구조적 관계보다 전역 수준의 요약 통계와 스칼라적 경향성을 반영한다[18].

7) MLP 임베딩(MLP Embedding)

MLP 임베딩 모듈은 QEP로부터 추출된 전역특성을 입력으로 받아 이들 특성을 고차원 표현 공간으로 사상하여 예측에 유용한 잠재 표현을 생성하는 역할을 수행한다. 수식으로 표현하면

$$z_{MLP} \in R^d \tag{6}$$

여기서 d는 임베딩 차원이고 z_{MLP} 는 전역적 특성을 요약한 잠재적 벡터이다.

8) 융합 방식 및 회귀 네트워크

Hybrid GM에서 주요한 부분은 기존의 두 모델의 출력을 어떻게 융합하는가에 있다. 그래프 임베딩은 실행계획의 구조적 패턴을 제공하며 특성 임베딩은 실행계획의 수치적 특성을 제공하고 있어서 두 모델에서 각각 생성된 임베딩은 서로 보완적 정보를 제공한다고 할 수 있다.

따라서 본 연구에서는 가장 직관적이면서 정보에 대한 손실이 없는 연결 방식을 선택하여 결합표현을 형성하였다.

$$z_{fusion} = [z_{GNN} \| z_{MLP}] \tag{7}$$

이후 결합표현은 선형 층과 활성화 함수를 통해 회귀 헤드로 전달되어 최종 실행 시간 예측값 $\hat{y}$를 산출한다.

$$\hat{y} = f_{reg}(z_{fusion}) \tag{8}$$

이 방법은 단순 평균 또는 가중합 기반 앙상블과 달리, 결합된 표현 공간에서 공동 학습을 가능하게 한다는 점에서 의의가 있다.

9) 학습 목적 및 손실함수

본 모델은 실행시간(비용)을 예측하는 회귀문제를 대상으로 하며 평균제곱오차(Mean Square Error)를 기본 손실함수로 사용한다.

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \tag{9}$$

Ⅳ. 실 험

4-1 실험목표

본 논문에서 제안한 Hybrid GM 모델의 실행시간 예측 성능을 정량적으로 평가하기 위해, 표준 회귀 성능지표로 RMSE, MAE, R2을 사용하여 GNN 기반의 옵티마이저와 딥러닝에서의 MLP 옵티마이저와 성능을 비교분석하였다.

4-2 데이타 세트

실험에는 데이터베이스 벤치마크 쿼리로 널리 사용되는 TPC-H 데이터 세트를 활용하였다[17]. 스케일펙터 (SF) = 10 환경에서 쿼리를 생성하고, 각 쿼리에 대해 다수의 실행계

획을 생성하였다.

사용쿼리 수는 20개이며 쿼리당 생성된 플랜 수는 12개씩 생성하여 총 플랜 수가 240개가 되게 확장하였다. 또한 각 플랜에 대해 다음 정보를 수집하였다. 이 자료들은 그래프로 표현하기 위한 자료들이다. 즉, 연산자 종류(Hash join, Nested Loop, Seq Scan등), 각 노드의 추정 카디널리티 및 비용, 조인 그래프 구조 (edge, node feature)와 실제 실행 시간 및 또는 실행비용(ground truth) 등 이다,

4-3 실험 셋업

모든 실험은 동일한 하드웨어 환경에서 수행하였다.

CPU는 Intel i5 processor이고 GPU는 NVIDIA GeForce RTX 4080이며 RAM은 32GB이며 Windows 11 환경에서 실험하였고 DBMS는 PostgreSQL 17에서 수행하였다.

또한 본 연구는 Hybrid GM 모델의 일반화 성능을 공정하게 평가하기 위해 쿼리레벨에서 데이터 분할 전략을 채택하였다. 이 전략은 TCP-H벤치마크에서 제공되는 SQL 쿼리의 실행계획을 생성하는 과정에서 유사한 실행계획이 훈련집합과 테스트 집합에 동시에 포함될 경우 데이터누수가 발생할 수 있기 때문이다. 이를 방지하기 위해 본 실험에서는 쿼리레벨에서 Train/Validation/Test = 70%/15%/15% 비율로 분할하였다.

4-4 베이스라인 모델과 비교 범위

본 연구에서는 제안하는 Hybrid GM 모델의 성능 향상이 단순한 모델 결합에 따른 효과가 아니라, 그래프 구조 기반 표현 학습과 전역 특성 기반 학습 간의 상호보완적 결합에서 기인함을 체계적으로 검증하기 위해 구성 요소 기반 기준선 모델을 설정하였다. 이를 위해 GNN 단독 모델과 MLP 단독 모델을 주요 비교 대상으로 채택하고, 구조적 정보와 통계적 특성이 실행 시간 예측 성능에 미치는 영향을 분리하여 분석하였다.

한편, Neo, Bao, Lero와 같은 최신 학습형 쿼리 옵티마이저는 특정 데이터베이스 엔진 및 실행 파이프라인과 긴밀하게 결합된 통합형 시스템으로 설계되어 있어, 본 연구에서 사용하는 실행 계획 그래프 기반의 독립적 비용 예측 환경과 직접적인 공정 비교에는 기술적 제약이 존재한다. 이에 따라 본 논문에서는 연구 범위를 모델 구조 수준에서의 일반화 성능 및 결합 효과의 검증에 한정하였다.

4-5 측정 방법

실행 비용 예측의 성능을 평가하기 위해 다음의 표준 회귀 성능 지표를 사용하였다.

1) MAE (Mean Absolute Error)

예측값과 실제값 간의 평균 절대 오차를 측정하는 지표이다. 값이 작을수록 모델 성능이 우수하다고 평가되며 주로 이상치가 많은 데이터셋에 적합하다.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y_i}\right| \tag{10}$$

여기서 y_i 는 실제값이고 $\hat{y_i}$는 예측값을 의미한다.

2) RMSE (Root Mean Squared Error)

예측값과 실제값 간 오차의 제곱 평균에 루트를 씌운값으로 모델 성능을 평가하는 대표지표이다. MAE와 달리 더 큰 패널티를 부여해 모델의 정확성을 엄격히 측정한다. RMSE 값이 작을수록 모델이 우수하다.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y_i}\right)^2} \tag{11}$$

여기서 y_i 는 실제값이고 $\hat{y_i}$는 예측값을 의미한다.

3) R^2

독립변수가 종속변수(실제값)의 변동성을 얼마나 설명하는지 나타내는 지표로 독립변수는 학습에 반영되어 나온 예측값을 의미하고 종속변수는 실제값을 의미한다. 값이 1에 가까울수록 모델이 설명력이 높아 데이터 적합도가 우수함을 의미한다.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y_i})^2}{\sum_i (y_i - \bar{y})^2} \tag{12}$$

여기서 y_i 는 실제값이고 $\hat{y_i}$는 예측값을 의미한다.

4-6 측정 결과

본 연구에서 회귀 모델의 목표변수는 실행계획의 실측 실행시간인 Actual Total Time이며 이는 PostgreSQL에 의해 밀리초(ms) 단위로 제공된다. 따라서 모든 실험에서 실행시간의 단위는 일관되게 밀리초로 설정하였다. 또한 본 연구에서는 repeated random split을 30회 수행하여 매 회 RMSE, MAE, R^2를 측정하고 평균과 분산을 구해 95%의 신뢰구간을 계산하였다. 결과는 표 1과 같이 나타났다.

표 1. 모델별 RMSE, MAE, R^2 (k=30, 95% 신뢰구간)
Table 1. RMSE, MAE and R^2 for each model (k=30, 95% CI)

Model	RMSE	MAE	R^2
MLP	337.81 (271.18 -404.44)	170.83 (143.71 – 197.94)	0.653 (0.551~0.754)
GNN	536.56 (-56 – 1129.68)	358.33 (11.23- 705.42)	-13.01 (39.93-13.90)
Hybrid GM	**92.42 (33.62 – 151.23)**	**57.20 (26.80 – 87.6)**	**0.760 (0.645~1.09)**

4-7 결과분석

실험 결과, 표 1에서 보는 바와 같이 제안된 Hybrid GM 모델이 비교 대상 모델들 가운데 가장 우수한 예측 성능을 보였다. GNN은 대체로 좋으나 극단적으로 이상치가 3~4개가 있는 것이 문제점으로 보이고 이상치를 제거한다면 hybrid와 근소한 차이로 매우 정확한 수치를 보인다. 반면 MLP는 성능이 제일 낮으나 전반적으로 매우 안정적인 수치를 보여준다. RMSE 기준으로 Hybrid 모델은 92.42 (33.62-151.23)을 기록하였으며 이 수치는 GNN 모델의 536.56 및 MLP 모델의 337.81에 비해 현저히 낮은 값이다. 이러한 결과는 GNN 기반의 관계적 표현학습과 MLP 기반의 비정형 특성변환을 상호보완적으로 활용하여 예측오차를 효과적으로 감소시킨다는 점을 보여준다. 또한 GNN에서 R^2는 0보다 작아서 이상치로 보이며 이상치 제거 후 모델별 성능은 훨씬 개선된다. 다만 이상치를 기록한 이유를 분석하는 것은 향후 연구로 진행할 예정이다. MLP는 특히 표 2와 같이 각 모델 대비 성능 향상도를 산출하였는데, GNN 모델 대비 Hybrid 모델의 RMSE는 약 71.87% 감소하였으며, 이는 단순한 그래프 기반 학습만으로 모든 비용 요인을 추정하지 못한다는 점을 시사한다.

표 2. 모델 간 성능 비교 분석
Table 2. Model performance comparison analysis

Model	RMSE (decline rate)	MAE (decline rate)	R^2 (growth rate)
MLP-> Hybrid GM	74.3%	68.6%	32.3%
GNN-> Hybrid GM	**86.68%**	**53.68%**	**13.87 (-13.01 -> 0.86)**

또한 본 연구는 결정계수 R^2를 추가 지표로 사용하여 제안하는 모델의 설명력을 평가하였다. 측정값이 1에 가까울수록 실제값의 분산을 모델이 더 많이 설명함을 의미한다. 실험결과 Hybrid GM모델은 0.760(0.645~1.09)을 기록하여 가장 높은 설명력을 보였다. 이는 GNN 모델의 측정치와 MLP 모델의 측정치를 비교할 때, Hybrid GM 모델이 실행비용 변동의 대부분을 효과적으로 설명하고 있음을 나타낸다. MLP 모델의 R2 = 0.653은 입력 특성만으로는 쿼리 실행비용의 구조적 변동을 충분히 알아내기 어렵다는 점을 보여주며, 특히 Hybrid 모델이 GNN 모델보다 음수에서 양수 0.86으로 향상된 것은 그래프 기반 표현학습과 특성 기반 표현학습이 상호보완적으로 작용하여 두 접근방식 중 어느 하나만으로는 알아내기 어려운 잔차 변동성까지 효과적으로 설명했음을 의미한다.

이러한 결과는 다음의 중요한 시사점을 제공한다.

- 쿼리플랜은 본질적으로 그래프 구조를 가진 데이터이며
- 관계정보학습이 비용예측 성능 향상에 결정적인 역할을 하며
- 구조적정보(GNN)와 비구조적특성(MLP)를 통합하는 Hybrid방식이 단일모델보다 예측정확도를 향상시키며 높은 설명력을 달성한다.

V. 결 론

본 연구는 TPC-H 벤치마크 기반의 SQL 쿼리 실행 계획을 대상으로, 실행 시간을 더욱 정확하게 예측하기 위한 Hybrid GM 모델을 제안하였다. 이 모델은 쿼리플랜을 그래프 구조로 표현하여 관계적 종속성을 학습하는 GNN과 집계된 특성 기반 표현을 학습하는 MLP를 결합함으로써, 두 모델의 상호보완적 특성을 활용하도록 설계하였다.

실험 결과, 제안된 Hybrid GM 모델은 기존의 GNN 및 MLP 단일모델과 비교하여 RMSE와 MAE 측면에서 가장 낮은 오차를 보였으며, 설명력을 나타내는 R2 지수도 높은 수치를 보였다. 특히 GNN 모델에서의 치명적인 오차가 있을 때도 Hybrid GM은 안정적인 수치를 보였고 이상치 빈도수도 감소한 것을 알 수 있었다.

이 결과는 쿼리 실행 비용 예측에 있어 그래프 구조정보와 비구조적 특성정보를 동시에 활용하는 접근법이 통계적으로 유의한 성능 향상을 제공함을 의미한다.

하지만 본 연구에서는 몇 가지 한계가 존재한다. 첫째, 실험데이터는 TPC-H 벤치마크에 기반하고 있어 실제 상용시스템의 다양한 워크로드를 모두 반영하지 못한다. 둘째, 본 연구는 정적학습을 전제로 하였으며 데이터 분포변화 및 스키마 진화에 대한 적응적 학습은 고려하지 않았다. 셋째, 본 연구에서 주요 목표는 실제 실행 시간 및 실제 비용 지표를 예측하는 것이었으므로 메모리 사용량 등 다중 목적 최적화를 이루지 않았다.

향후 연구에서는, Hybrid GM을 시각화로 확장하여 MLP에 사용하는 SHAP와 GNN에 사용되는 GNN-Explainer를 접목할 예정이다. 두 모델을 단순히 동시에 보여주는 방식이 아닌 보다 더 상호작용하는 방법으로 융합화한 모델을 시각화하여 제공하는 것이 목표이다. 또한 이 제안 모델을 실제 데이터베이스 옵티마이저 파이프라인에 통합하여 종단 간(end-to-end) 성능 비교를 수행하여 성능비교를 할 계획이다. 그 외 제안된 모델의 주요 목표인 쿼리 최적화를 위해 실행 비용 예측을 넘어 조인 순서 생성 및 인덱스 추천 등 구체적인 최적화 의사결정으로 본 연구를 확장하는 것이 다음 연구 목표이다.

감사의 글

본 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-지역지능화혁신인재양성사업의 지원을 받아 수행된 연구임(IITP-2026-RS-2022-00156334).

참고문헌

[1] Y. E. Ioannidis, “Query Optimization,” *ACM Computing Surveys*, Vol. 28, No. 1, pp. 121-123, March, 1996. https://doi.org/10.1145/234313.234367

[2] S. Chaudhuri, “An Overview of Query Optimization in Relational Systems,” in *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, Seattle, Washington, pp. 34-43, May 1998. https://doi.org/10.1145/275487.275492

[3] V. Leis, A. Gubichev, A. Mirchev, P. Boncz, A. Kemper, and T. Neumann, “Still Asking: How Good Are Query Optimizers, Really?,” *Proceedings of the VLDB Endowment*, Vol. 18, Issue 12, pp. 5531-5536, August 2025. https://doi.org/10.14778/3750601.3760521

[4] W. Wu, Y. Chi, S. Zhu, J. Tatemura, H. Hacigümüs, and J. F. Naughton, “Predicting Query Execution Time: Are Optimizer Cost Models Really Unusable?,” in *Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE)*, Brisbane, Australia, pp. 1081-1092, April 2013. https://doi.org/10.1109/ICDE.2013.6544899

[5] R. Zhu, L. Weng, B. Ding, and J. Zhou, “Learned Query Optimizer: What is New and What is Next”, in *Proceedings of the Companion of the 2024 International Conference on Management of Data*, Santiago, Chile, pp. 561-569, 2024. https://doi.org/10.1145/3626246.3654692

[6] J. Chem, G. Ye, Y. Zhao, S. Liu, L. Deng, X. Chen, ... and K. Zheng, “Efficient Join Order Selection Learning with Graph-based Representation,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington DC, pp. 97-107, August 2022. https://doi.org/10.1145/3534678.3539303

[7] H. Pirahesh, J. M. Hellerstein, and W. Hasan, “Extensible/Rule Based Query Rewrite Optimization in Starburst,” *ACM SIGMOD Record*, Vol.21, No. 2, pp 39-48, June 1992. https://doi.org/10.1145/141484.130294

[8] P. Griffiths Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price, “Access Path Selection in a Relational Database Management System,” in *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data*, Boston MA, pp. 23-34, 1979. https://doi.org/10.1145/582095.582099

[9] R. Marcus, P. Negi, H. Mao, C. Zhang, M.Alizadeh, T. Kraska, ... and N. Tatbul, “Neo: A Learned Query Optimizer,” *Proceedings of the VLDB Endowment*, Vol. 12, No. 11, pp. 1705-1718, 2019. https://doi.org/10.14778/3342263.3342644

[10] Z. Yang, W.-L. Chiang, S. Luan, G. Mittal, M. Luo, and I. Stoica, “Balsa: Learning a Query Optimizer Without Expert Demonstrations,” in *Proceedings of the 2022 International Conference on Management of Data*, Philadelphia: PA, pp. 931-944, 2022. https://doi.org/10.1145/3514221.35178

[11] X. Chen, H. Chen, Z. Liang, S. Liu, J. Wang, K. Zeng, ... and K. Zheng, “LEON: A New Framework for ML-Aided Query Optimization,” *Proceedings of the VLDB Endowment*, Vol. 16, No. 9, pp. 2261-2273, 2023. https://doi.org/10.14778/3598581.3598597

[12] C. Gallicchio and A. Micheli, “Fast and Deep Graph Neural Networks,” in *Proceeding of The 34th AAAI Conference on Artificial Intelligence*, pp. 3898-3905, 2020. https://doi.org/10.48550/arXiv.1911.08941

[13] Z. Yang, E. Liang, A. Kamsetty, C. Wu, Y. Duan, X. Chen, ... and I. Stoica, “Deep Unsupervised Cardinality Estimation,” *Proceedings of the VLDB Endowment*, Vol. 13, No. 3, pp. 279-292, 2019. https://doi.org/10.14778/3368289.3368294

[14] A. Kipf, T. Kipf, B. Radke, V. Leis, P. Boncz, and A. Kemper, “Learned Cardinalities: Estimating Correlated Joins with Deep Learning,” in *Proceedings of the 9th Biennial Conference on Innovative Data Systems Research (CIDR ‘19)*, California, January 2019. https://doi.org/10.48550/arXiv.1809.00677

[15] Z. Yang, A. Kamsetty, S. Luan, E. Liang, Y. Duan, X. Chen, and I. Stoica, “NeuroCard: One Cardinality Estimator for All Tables,” *Proceedings of the VLDB Endowment*, Vol. 14, No. 1, pp. 61-73, 2020. https://doi.org/10.14778/3421424.342143

[16] Q. Q. Fan, K.-H. Han, and S.-S. Shin, “Query Execution Plan Optimization Model Based on Graph Query Optimization,” *Advanced Industrial SCIence*, Vol. 4, No. 4, pp. 23-35, 2025. https://doi.org/10.23153/AI-Science.2025.4.4.023

[17] TPC-H. Dataset [Internet]. Available: https://www.tpc.org/tpc_documents_current_versions/current_specifications5.asp.

[18] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?,” in *Proceedings of International Conference on Learning Representation*, 2019. https://doi.org/10.48550/arXiv.1810.00826

박화진(Hwa-Jin Park)

1989년 : 숙명여자대학교 대학원 (이학석사-지능형 CAI)
1997년 : 미국 아리조나 주립 대학교 대학원 (공학박사-컴퓨터 그래픽스)

1997년~2000년: 삼성SDS
2000년~2002년: 평택대학교
2002년~현 재: 숙명여자대학교 정교수
※관심분야 : 컴퓨터 그래픽스, 가상/증강현실, 게임, GNN 등

최옥주(Ok-Joo Choi)

1987년 : 숙명여자대학교 (이학사)
1990년 : 숙명여자대학교 대학원 (이학석사-전산학/SW공학)
2008년 : 숙명여자대학교 대학원 (이학박사-컴퓨터공학/데이터베이스)

1990년~1996년: LG전자 생산기술원 시스템실 주임원구원
1996년~2009년: 한국오라클 컨설팅서비스본부 Technical Consulting Manager
2009년~2022년: 한국과학기술원 전산학부 연구부교수
2023년: 강원국립대학교 산학협력중점교수
2024년~현 재: 배재대학교 AI.소프트웨어공학부 조교수
※관심분야 : AI-based DB, XAI, Data Analytics, Data/SW Quality, Project Management