GRAIL

Graph Edit Distance and Node Alignment using LLM-Generated Code

Samidha Verma*[1], Arushi Goyal*[1], Ananya Mathur*[1], Ankit Anand[2], Sayan Ranu[1]

[1]Indian Institute of Technology Delhi, India
[2]Google Deepmind, Montreal, Canada

ICML
International Conference On Machine Learning

Graph Edit Distance

Definition: Graph Edit Distance (GED) quantifies the dissimilarity between two graphs as the minimum number of edits required to transform one graph into another. An edit may comprise adding or deleting nodes and edges or replacing node and edge labels.

Computing GED is NP-hard [Bai et al., WSDM 2019] !

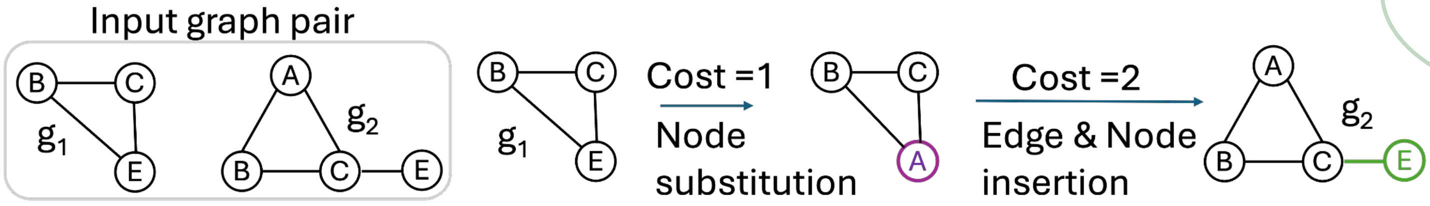


Illustration of edit path from g1 to g2 with GED 3.

Applications:

Graph Retrieval [Bai et al., WSDM 2019, Ding et al., AAAI 2020],
Pattern Recognition [Sanfeliu et al., Pattern Recognition 2002],
Image and Video Indexing [Tirthapura et al., Multimedia storage and archiving systems III, SPIE 1998],
Cheminformatics [Garcia-Hernandez et al., Journal of chemical information and modeling 2019]

Related Work: Strengths and Limitations

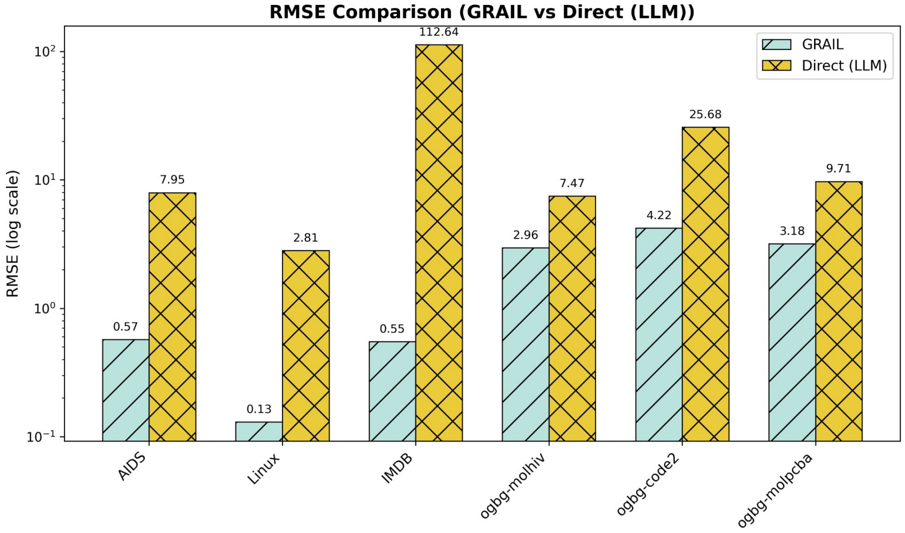
Name	End-to-end interpretable	Cross-domain generalization	Non-reliant on NP-hard supervision	Accurate
GREED [Ranjan et al., NeurIPS 2022]	❌	❌	❌	✅
GEDGNN [Piao et al., VLDB 2023]	⚪	❌	❌	✅
H^2MN [Zhang et al., KDD 2021]	❌	❌	❌	✅
ERIC [Zhuo et al., NeurIPS 2022]	❌	❌	❌	✅
GRAPHEDX [Jain et al., NeurIPS 2024]	❌	❌	❌	✅
Non-neural approaches [Blumenthal et al., VLDB Journal 2020]	⚪	✅	✅	❌
GRAIL (ours)	✅	✅	✅	✅

Non-satisfaction (❌) ; Partial-satisfaction (⚪) ; Satisfaction (✅)

GRAIL: A Paradigm Shift

Novel Problem Formulation for GED Approximation

- Shift the objective to learning a program that approximates GED.
- Enable end-to-end interpretability and superior generalization across datasets, domains, graph sizes, and label distributions.
- Programs generated by directly prompting the LLM are not effective (much higher RMSE).



GRAIL: A Paradigm Shift (Effective Alternate Formulation)

LLM-guided program discovery

The algorithmic framework of GRAIL is grounded on three novel design choices.

- Map the problem of approximating GED to maximum weight bipartite matching. Weights of the bipartite graph are computed using an LLM-generated program.
- Prompt provided to the LLM is tuned through an evolutionary algorithm based on FunSearch [Romera-Paredas et al., Nature 2023].
- Submodularity-based prompt-tuning methodology eliminates the need for ground-truth GED data, overcoming a critical bottleneck of existing neural approaches.

Experiments

Key Results: Approximation Errors

RMSE

Type	Methods	AIDS	Linux	IMDB	ogbg-molhiv	ogbg-code2	ogbg-molpcba	Avg. Rank
LLM	GRAIL	0.57	0.13	0.55	2.96*	4.22	3.18	2
	GRAIL-MIX	0.64	0.11	0.53	2.96	4.10	3.40	2.17
Neural	GREED	0.61	0.41	4.8	3.02	5.52	2.48	3.5
	GEDGNN	0.92	0.29	4.43	1.75	16.68	4.58	5
	ERIC	1.08	0.30	42.44	3.56	17.55	2.79	6.5
	H^{2}MN	1.14	0.60	57.8	12.01	11.96	5.50	8.33
	GRAPHEDX	0.78	0.27	32.36	14.14	21.46	10.01	8.33
Non Neural	ADJ-IP	0.85	0.50	42.18	10.21	14.94	8.06	7.33
	NODE	2.71	1.24	61.03	4.97	8.34	4.94	8.17
	LP-GED-F2	1.96	0.23	55.26	12.86	16.03	10.30	8.83
	BRANCH	3.31	2.45	7.36	9.86	12.64	11.31	9.33
	COMPACT-MIP	2.69	0.44	65.88	10.88	19.46	8.81	10
	IPFP	4.18	2.29	69.45	13.69	15.19	10.02	11.5

Exact Match Ratio (EMR)

Methods	AIDS	Linux	IMDB	ogbg-molhiv	ogbg-code2	ogbg-molpcba	Avg. Rank
GRAIL-MIX	0.80	≈ 1	≈ 1	0.20	0.12	0.12	1.83
GREED	0.58	0.79	0.17	0.23	0.09	0.21	2.17
ERIC	0.37	0.92	0.08	0.21	0.01	0.18	2.83
GEDGNN	0.35	0.85	0.07	0.57	0.01*	0.09	3.17

GRAIL MIX:

- Positive knowledge transfer across datasets.
- No dataset-specific training required – just train once!!
- 9.12 % relative gain on RMSE compared to the best baseline.

GRAIL-MIX: GRAIL, trained on a mixture of datasets

Takeaways

Autonomous LLM-Driven GED Computation

No NP-Hard Supervision

Effective

Mixture training (GRAIL-MIX)

Foundation Functions

Thank You !

Poster Session 1 - July 15, 2025

PAPER

CODE