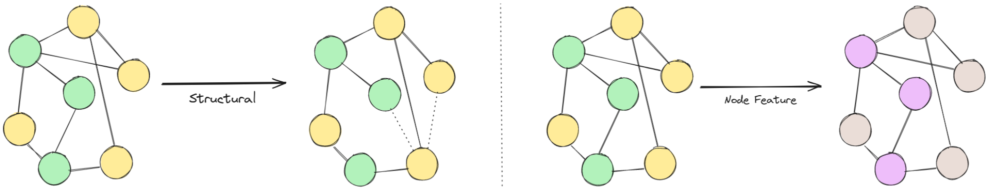
# A Learned Generalized Geodesic Distance Function-Based Approach for Node Feature Augmentation on Graphs

Amitoz Azad
Yuan Fang

# Contents

# Graph Augmentation[1]

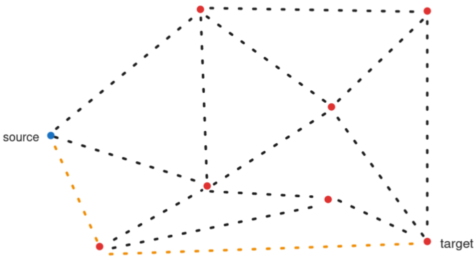


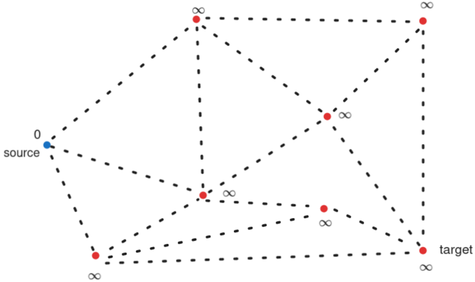

[1] Tong Zhao et al. “Graph data augmentation for graph machine learning: a survey”. In: *arXiv* (2022).

# Contents

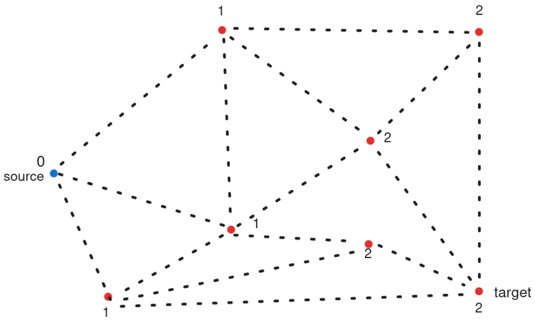
1. Problem
2. Motivation
3. Method
4. Results

# Geodesic on Graphs



# Geodesic Distance Function on Graphs[2]




## Geodesic Distance Function

$$f(x) = \min_{y \in N(x)} \{f(y) + 1\} \quad x \in V \setminus V_0$$

$$f(x) = 0 \quad x \in V_0$$

[2] Moshe Sniedovich. "Dijkstra's algorithm revisited...". In: *Control and cybernetics* (2006).

# Generalized Geodesic Distance Function[3]

Geodesic Distance Function

$$f(x) = \min_{y \in N(x)} \{f(y) + 1\} \quad x \in V \setminus V_0$$

$$f(x) = 0 \quad x \in V_0$$

Generalized Geodesic Distance Function

$$\rho(x) \|\nabla_w^- f(x)\|_p = 1 \quad x \in V \setminus V_0$$

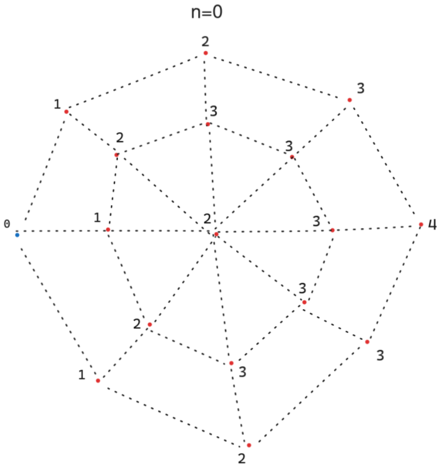
$$f(x) = 0 \quad x \in V_0$$

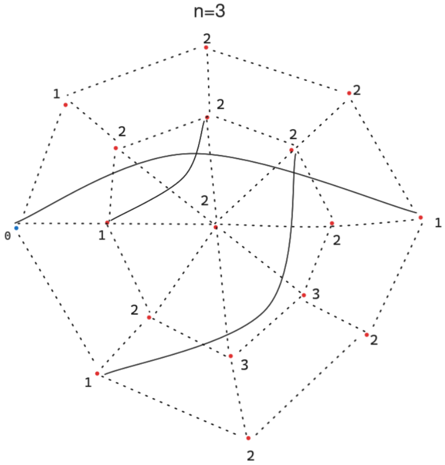
[3] Jeff Calder and Mahmood Ettehad. "Hamilton-Jacobi equations on graphs ...". In: *JMLR* (2022).

# Connection Between The Two

**Proposition 1**: *For an uweighted graph with a constant potential function* $\rho(x) = 1$, *any valid solution of the generalized geodesic distance function equation with supremum norm (i.e.* $p = \infty$*) yields geodesic distance function.*

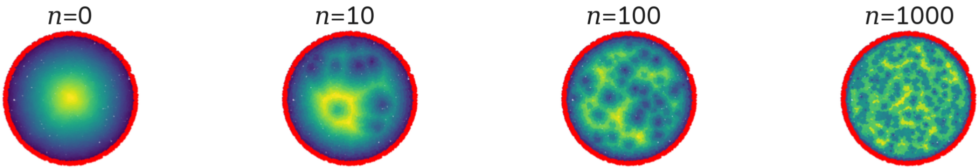
# Robustness[4]




[4] Leonard Kaufman and Peter J. Rousseeuw. “Graph k-medoids”. In: *Wiley Probability and Statistics* (1990).

# Robustness




Top-row: Robustness of Geodesic (Shortest-path) Distances to random edge corruptions.

Bottom-row: Robustness of Generalized Geodesic Distances for $p$ = 1 to random edge corruptions.

The $n$ represents the number of random corrupted edges added to a given graph. The graph construction: 20,000 points (nodes) were randomly sampled from a unit ball in $R^2$ . An $\epsilon$-neighborhood unweighted graph was constructed using these sampled points with $\epsilon$ = 0.05. All points within $\epsilon$ distance of the boundary of the unit ball are considered boundary nodes. Colors represent the distance from the boundary, with red indicating the boundary where the distance function is zero, and yellow indicating the maximum distance.

# Contents

# Model

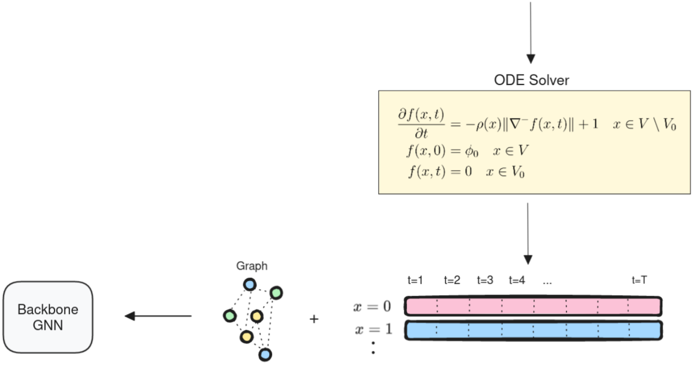
$$\begin{aligned} \rho(x)\|\nabla^- f(x)\| &= 1 \quad x \in V \setminus V_0 \\ f(x) &= 0 \quad x \in V_0 \end{aligned}$$

$$\begin{aligned} \frac{\partial f(x,t)}{\partial t} &= -\rho(x)\|\nabla^- f(x,t)\| + 1 \quad x \in V \setminus V_0 \\ f(x,0) &= \phi_0 \quad x \in V \\ f(x,t) &= 0 \quad x \in V_0 \end{aligned}$$

# Model[5]

$$
\begin{aligned}
\frac{\partial f(x,t)}{\partial t} &= -\rho(x)\|\nabla^{-} f(x,t)\| + 1 \quad x \in V \setminus V_0 \\
f(x,0) &= \phi_0 \quad x \in V \\
f(x,t) &= 0 \quad x \in V_0
\end{aligned}
$$

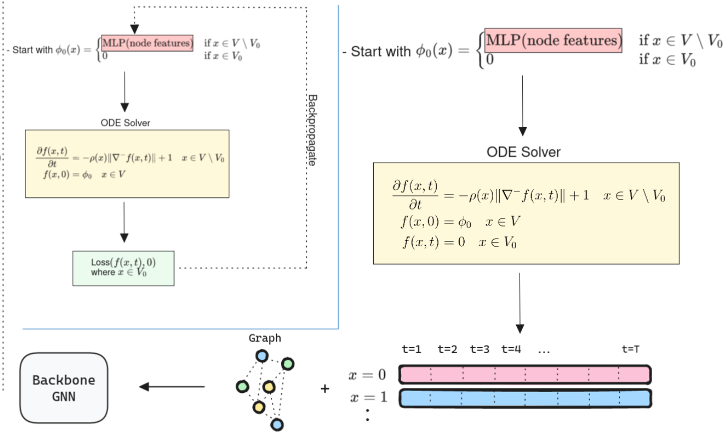
- Solve it for each class for every node $x$ for $T$ different time steps.
- Start with $\phi_0(x) = \begin{cases} \infty & \text{if } x \in V \setminus V_0 \\ 0 & \text{if } x \in V_0 \end{cases}$




[5] Ricky T. Q. Chen. *Torchdiffeq*. 2018. URL: https://github.com/rtqichen/torchdiffeq.

# Model[6]

$$
\begin{aligned}
\frac{\partial f(x,t)}{\partial t} &= -\rho(x)\|\nabla^{-} f(x,t)\| + 1 \quad x \in V \setminus V_0 \\
f(x,0) &= \phi_0 \quad x \in V \\
f(x,t) &= 0 \quad x \in V_0
\end{aligned}
$$



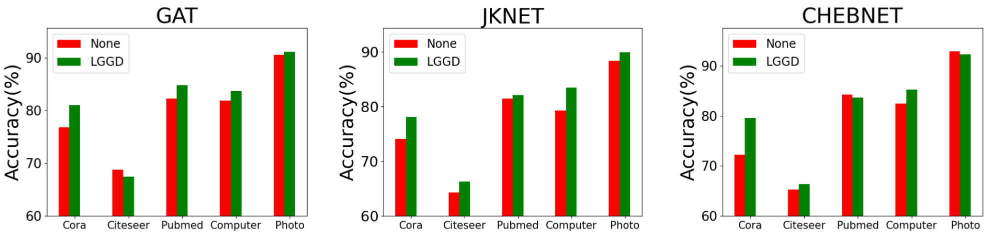

[6] Maziar Raissi et al. "PINNS: Physics informed neural networks ...". In: *Journal of Computational physics* (2019).

# Contents

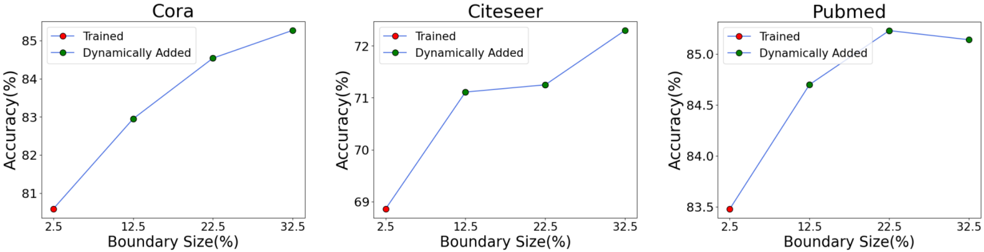
# Results

| Model | Cora | Citeseer | Pubmed | Computers | Photo |
|---|---|---|---|---|---|
| 01 GCN | 74.13 ± 2.08 | 66.08 ± 2.16 | 79.73 ± 0.71 | 81.72 ± 1.78 | 87.57 ± 1.18 |
| 02 MixUp | 72.72 ± 1.78 | 64.14 ± 1.75 | 80.02 ± 0.52 | 80.76 ± 1.40 | 88.67 ± 0.80 |
| 03 DropEdge | 72.28 ± 1.39 | 65.73 ± 1.83 | 81.89 ± 0.84 | 81.45 ± 1.02 | 88.29 ± 1.27 |
| 04 GAug-M | 72.14 ± 1.37 | 66.38 ± 1.29 | 82.18 ± 1.36 | 84.82 ± 0.78 | 91.05 ± 1.21 |
| 05 GAug-O | 71.30 ± 1.54 | 67.22 ± 1.06 | OOM* | 83.03 ± 0.50 | 90.62 ± 0.30 |
| 06 GDC (heat) | 77.52 ± 1.74 | 65.38 ± 1.36 | 82.16 ± 0.93 | 80.18 ± 1.31 | 88.12 ± 2.21 |
| 07 GDC (ppr) | 78.13 ± 2.13 | 66.33 ± 1.84 | 80.86 ± 0.78 | 82.88 ± 1.14 | 89.07 ± 2.19 |
| 08 GGD | 69.95 ± 2.51 | 43.21 ± 2.44 | 76.49 ± 0.87 | 78.89 ± 1.61 | 85.69 ± 0.92 |
| 09 LGGD | 80.18 ± 1.53 | 67.23 ± 1.79 | 83.24 ± 1.79 | 85.23 ± 2.18 | 92.02 ± 2.33 |
| 10 LGGD w. $\rho(x)$ | **81.56 ± 2.29** | **68.63 ± 1.70** | 83.36 ± 1.88 | **85.49 ± 1.09** | **92.39 ± 2.11** |
| 11 GPR-GNN | 79.45 ± 1.66 | 67.18 ± 1.84 | **84.11 ± 0.38** | 82.80 ± 2.01 | 91.48 ± 1.59 |
| 12 GOAL | 76.07 ± 1.56 | 66.57 ± 1.26 | 81.83 ± 1.28 | 83.43 ± 1.04 | 91.65 ± 0.69 |

# Results


GAT
JKNET
CHEBNET
None
LGGD
Accuracy(%)
60
70
80
90
Cora
Citeseer
Pubmed
Computer
Photo

# Results


Cora
Trained
Dynamically Added
85
84
83
82
81
Accuracy(%)
2.5
12.5
22.5
32.5
Boundary Size(%)
Citeseer
Trained
Dynamically Added
72
71
70
69
Accuracy(%)
2.5
12.5
22.5
32.5
Boundary Size(%)
Pubmed
Trained
Dynamically Added
85.0
84.5
84.0
83.5
Accuracy(%)
2.5
12.5
22.5
32.5
Boundary Size(%)

# Thank You :)