Research Track

Graph Algorithms and Modeling for the Web

Diffusion-based Negative Sampling on Graphs for Link Prediction

Trung-Kien Nguyen and Yuan Fang

In Proceedings of the 2024 ACM Web Conference, May 13-17, 2024

Motivation

Problem Negative sampling in contrastive learning for Link Prediction

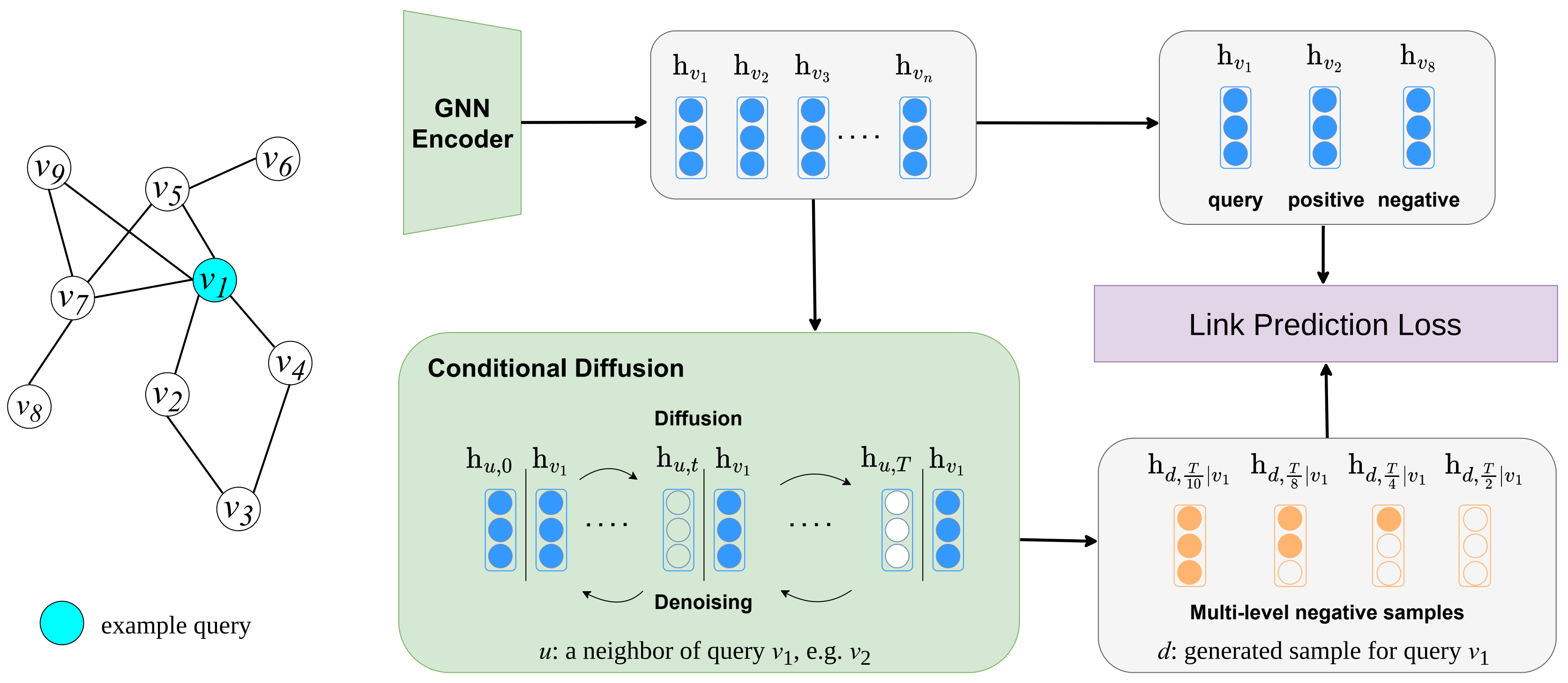
- Requires positive and negative samples for a given query node
- Negative sampling: huge search space and many false negatives

Challenges

1. *How to flexibly model and control the quality of negative nodes?*
   → **Multi-level** negative sampling strategy
2. *How do we find sufficient negative examples of variable hardness?*
   → **Diffusion models:** generating multi-level samples at different steps

Proposed model: DMNS

Overall Framework



GNN Encoder
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
example query
h_{v_1} h_{v_2} h_{v_3} .... h_{v_n}
h_{v_1} h_{v_2} h_{v_8}
query
positive
negative
Link Prediction Loss
Conditional Diffusion
Diffusion
$h_{u,0}$ h_{v_1} .... $h_{u,t}$ h_{v_1} .... $h_{u,T}$ h_{v_1}
Denoising
u: a neighbor of query v_1, e.g. v_2
$h_{d,\frac{T}{10}|v_1}$ $h_{d,\frac{T}{8}|v_1}$ $h_{d,\frac{T}{4}|v_1}$ $h_{d,\frac{T}{2}|v_1}$
Multi-level negative samples
d: generated sample for query v_1

(a) Input Graph (b) DMNS (c) Prediction

GNN Encoder

$$\mathbf{h}_v^l = \sigma\left(\textsc{Aggr}(\mathbf{h}_v^{l-1}, \{\mathbf{h}_i^{l-1} : i \in \mathcal{N}_v\}; \omega^l)\right)$$

Conditional Diffusion

Forward Process

$$\mathbf{h}_{u,t} = \sqrt{\bar{\alpha}_t}\mathbf{h}_u + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \quad \forall u \in \mathcal{N}_v,\ \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Reverse Process

$$\epsilon_{t,\theta|v} = (\gamma + \mathbf{1}) \odot \mathbf{h}_{u,t} + \eta,$$

$$\gamma = \mathrm{FCL}(\mathbf{t} + \mathbf{h}_v; \theta_\gamma), \quad \eta = \mathrm{FCL}(\mathbf{t} + \mathbf{h}_v; \theta_\eta),$$

$$[\mathbf{t}]_{2i} = \sin(t/10000^{\frac{2i}{d_h}}) \qquad [\mathbf{t}]_{2i+1} = \cos(t/10000^{\frac{2i}{d_h}})$$

Overall Loss

Multi-level Negative Sampling

$$\mathbf{h}_{d,T|v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

$$\mathbf{h}_{d,t-1|v} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{h}_{d,t|v} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_{t,\theta|v}\right) + \sigma_t\mathbf{z},$$

Diffusion Loss (MSE)

$$\mathcal{L}_D = \|\epsilon_t - \epsilon_{t,\theta|v}\|^2$$

Link Prediction Loss

$$\mathcal{L} = -\log\sigma(\mathbf{h}_v^\top\mathbf{h}_u) - \log\sigma(-\mathbf{h}_v^\top\mathbf{h}_{u'}) - \textstyle\sum_{d_i \in D_v} w_i \log\sigma(-\mathbf{h}_v^\top\mathbf{h}_{d_i}))$$

$$(v, u, u', D_v),$$

query node

positive node

real negative node

DMNS negative sets:

$$D_v = \{\mathbf{h}_{d,t|v} : t = \tfrac{T}{10}, \tfrac{T}{8}, \tfrac{T}{4}, \tfrac{T}{2}\}$$

Theoretical Analysis

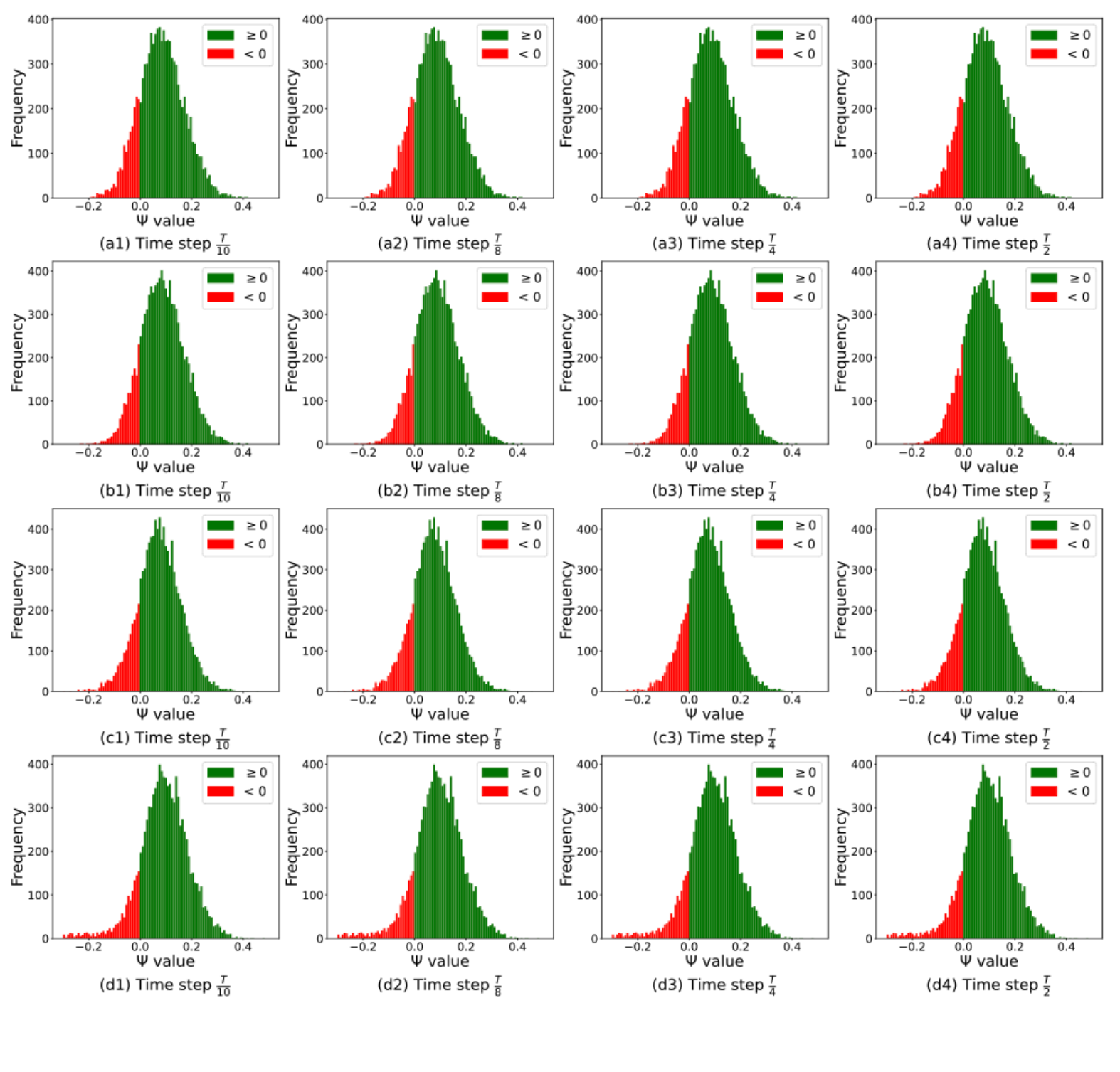
The majority of negative examples from DMNS follow the Sub-linear Positivity Principle [6]: which balances the trade-off between the embedding objective and expected risk for robust negative sampling.

Theorem 1 (Sub-linear Positivity Diffusion). *Consider a query node v. Let $\mathbf{x}_n \sim \mathcal{N}(\mu_{t,\theta}, \Sigma_{t,\theta})$ and $\mathbf{x}_p \sim \mathcal{N}(\mu_{0,\theta}, \Sigma_{0,\theta})$ represent samples drawn from the negative and positive distributions of node v, respectively. Suppose the parameters of the two distributions are specified by a diffusion model θ conditioned on the query node v at time $t > 0$ and 0, respectively. Then, the density function of the negative samples f_n is sub-linearly correlated to that of the positive samples f_p:*

$$f_n(\mathbf{x}_n|v) \propto f_p(\mathbf{x}_p|v)^\lambda, \quad \text{for some } 0 < \lambda < 1,$$

as long as $\Psi \geq 0$, which is a random variable given by $\Psi = 2\Delta^\top\sqrt{\bar{\alpha}_t}(\mathbf{x}_0 - \mu_0) + \Delta^\top\Delta \geq 0$, where $\Delta = \sqrt{\bar{\alpha}_t}\mu_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0 - \mu_t$, $\mathbf{x}_0$ is generated by the model θ at time 0, and $\epsilon_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. □

See the paper for the proof.



(a1) Time step $\frac{T}{10}$ (a2) Time step $\frac{T}{8}$ (a3) Time step $\frac{T}{4}$ (a4) Time step $\frac{T}{2}$
(b1) Time step $\frac{T}{10}$ (b2) Time step $\frac{T}{8}$ (b3) Time step $\frac{T}{4}$ (b4) Time step $\frac{T}{2}$
(c1) Time step $\frac{T}{10}$ (c2) Time step $\frac{T}{8}$ (c3) Time step $\frac{T}{4}$ (c4) Time step $\frac{T}{2}$
(d1) Time step $\frac{T}{10}$ (d2) Time step $\frac{T}{8}$ (d3) Time step $\frac{T}{4}$ (d4) Time step $\frac{T}{2}$

Figure 2: Empirical distributions (histograms) of Ψ on (a1–a4) Cora, (b1–b4) Citeseer, (c1–c4) Coauthor-CS, (d1–d4) Actor, across different time steps.

Experiments

Datasets

Datasets	Nodes	Edges	Features	Property
Cora	2708	5429	1433	homophilous
Citeseer	3327	4732	3703	homophilous
Coauthor-CS	18333	163788	6805	homophilous
Actor	7600	30019	932	heterophilous

Baselines

Classic GNNs
- GCN [1]
- GAT [2]
- SAGE [3]

Heuristic NS
- PNS [4]
- DNS [5]
- MCNS [6]

Generative NS
- GraphGAN [7]
- ARGVA [8]
- KBGAN [9]

Subgraph-based GNNs
SEAL [10]
ScaLed [11]

Link Prediction

Table 2: Evaluation of link prediction against baselines using GCN as the base encoder.

Methods	Cora		Citeseer		Coauthor-CS		Actor	
	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
GCN	.742 ± .003	.805 ± .003	.735 ± .011	.799 ± .008	.823 ± .004	.867 ± .003	.521 ± .004	.634 ± .003
GVAE	<u>.783</u> ± .003	<u>.835</u> ± .002	.743 ± .004	.805 ± .003	.843 ± .011	.882 ± .008	<u>.587</u> ± .004	<u>.684</u> ± .003
PNS	.730 ± .008	.795 ± .006	.748 ± .006	.809 ± .005	.817 ± .004	.863 ± .003	.517 ± .006	.631 ± .006
DNS	.735 ± .007	.799 ± .005	<u>.777</u> ± .005	<u>.831</u> ± .004	.845 ± .003	.883 ± .002	.558 ± .006	.663 ± .005
MCNS	.756 ± .004	.815 ± .003	.750 ± .006	.810 ± .004	.824 ± .004	.868 ± .004	.555 ± .005	.659 ± .004
GraphGAN	.739 ± .003	.802 ± .002	.740 ± .011	.803 ± .008	.818 ± .007	.863 ± .005	.534 ± .007	.644 ± .005
ARVGA	.732 ± .011	.797 ± .009	.689 ± .005	.763 ± .004	.811 ± .003	.858 ± .002	.526 ± .012	.638 ± .009
KBGAN	.615 ± .004	.705 ± .003	.568 ± .006	.668 ± .005	<u>.852</u> ± .002	<u>.888</u> ± .002	.472 ± .003	.596 ± .002
SEAL	.751 ± .007	.812 ± .005	.718 ± .002	.784 ± .002	.850 ± .001	.886 ± .001	.536 ± .001	.641 ± .001
ScaLed	.676 ± .004	.752 ± .003	.630 ± .004	.712 ± .003	.828 ± .001	.869 ± .001	.459 ± .001	.558 ± .001
DMNS	**.793** ± .003	**.844** ± .002	**.790** ± .004	**.841** ± .003	**.871** ± .002	**.903** ± .001	**.600** ± .002	**.696** ± .002

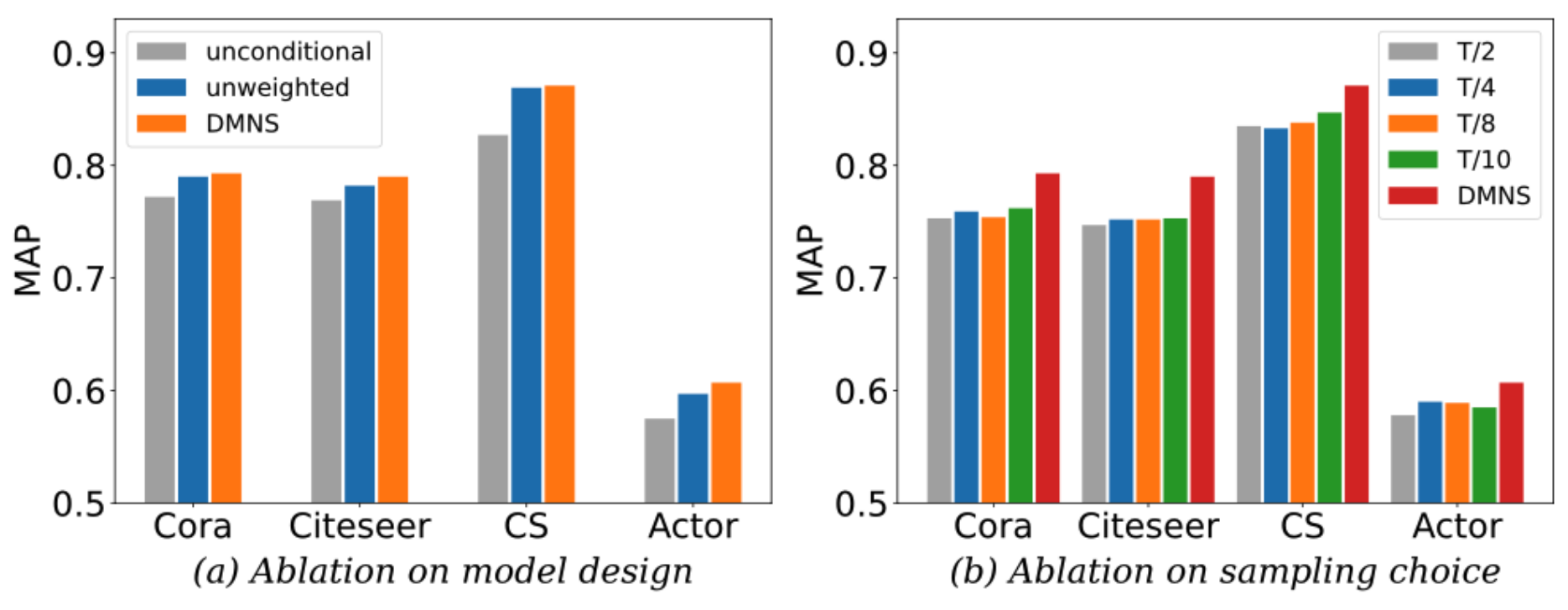
*Best is **bolded** and runner-up <u>underlined</u>.

Table 3: Evaluation of link prediction on DMNS with various base encoders.

Methods	Cora		Citeseer		Coauthor-CS		Actor	
	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
GAT	.766 ± .006	.824 ± .004	.767 ± .007	.763 ± .062	.833 ± .003	.874 ± .002	.479 ± .004	.603 ± .003
DMNS-GAT	**.813** ± .004	**.859** ± .003	**.788** ± .007	**.840** ± .006	**.851** ± .002	**.889** ± .002	**.573** ± .007	**.675** ± .005
SAGE	.598 ± .014	.668 ± .013	.622 ± .012	.713 ± .009	.768 ± .005	.826 ± .004	.486 ± .004	.604 ± .003
DMNS-SAGE	**.700** ± .007	**.773** ± .005	**.669** ± .013	**.749** ± .010	**.843** ± .004	**.883** ± .003	**.582** ± .017	**.682** ± .013

- DMNS outperforms competing baselines on all datasets and metrics, showing effectiveness of multi-level negative sampling strategy.
- DMNS improves performance of various base GNN encoders, demonstrating its flexibility.

Ablation Study



(a) Ablation on model design *(b) Ablation on sampling choice*

Figure 3: Ablation studies.

(a) On model design
- Unconditional diffusion performs worse than conditional counterparts
- Unweighted negative examples exhibits drop in performance

(b) On sampling choice
- Performance of each single time step varies, but all are worse than combining them together
- Smaller time steps often outperform larger time steps

Visualization

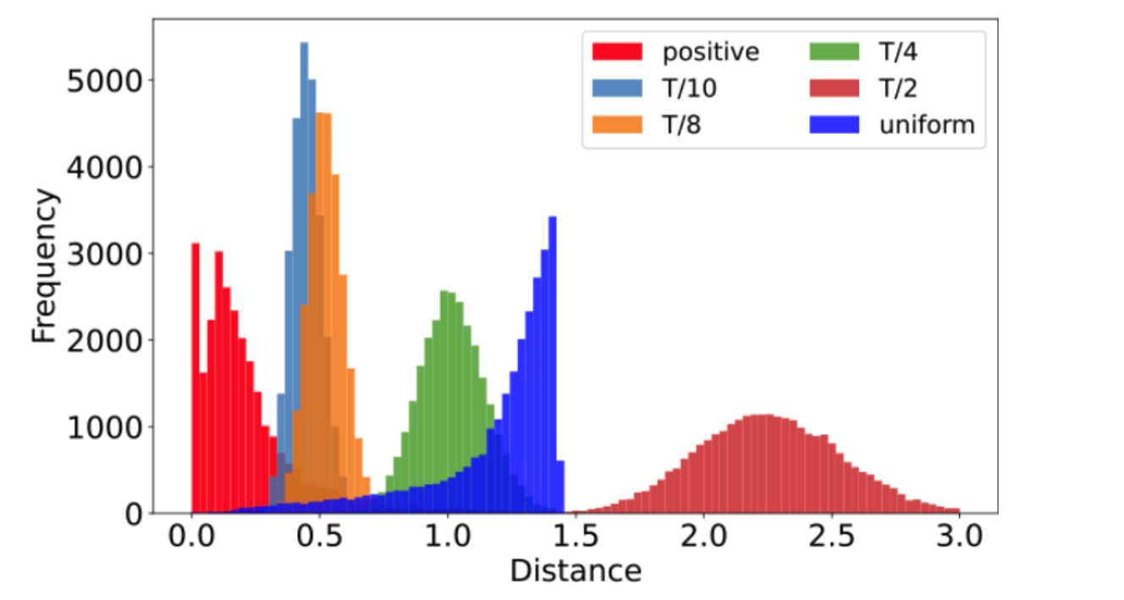


Figure 5: Histogram of embedding distances from query.

The embedding distance as proxy to hardness
- Smaller distances from the query node imply harder examples
- Examples of DMNS are generally harder uniform sampling, but not too hard (not closer than the positives) to impair the performance
- Utilizing multi-level samples allows to capture a wide range of hardness levels for negative sampling.

Conclusions

Problem
- Multi-level negative sampling for graph link prediction

Proposed model: DMNS
- Empowers the sampling of multi-level negative examples, by sampling at different denoised steps of diffusion models
- Adheres the sub-linear positivity principle for robust negative sampling

Experiments
- Extensive experiments demonstrate the effectiveness of DMNS

Key References

[1] Kipf et al. 2017. Semi-supervised classification with graph convolutional networks. ICLR.
[2] Veličković et al. 2018. Graph attention networks. ICLR.
[3] Hamilton et al. 2017. Inductive representation learning on large graphs. NeurIPS.
[4] Mikotov et al. 2013. Distributed representations of words and phrases and their compositionality. NeurIPS.
[5] Zhang et el. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. SIGIR.
[6] Yang et al. 2020. Understanding negative sampling in graph representation learning. KDD.
[7] Wang et al. 2018. Graphgan: Graph representation learning with generative adversarial nets. AAAI.
[8] Pan et al. 2018. Adversarially regularized graph autoencoder for graph embedding. IJCAI.
[9] Cai et al. 2017. Kbgan: Adversarial learning for knowledge graph embeddings. ACL.
[10] Zhang et al. 2018. Link prediction based on graph neural networks. NeurIPS.
[11] Louis et al. 2022. Sampling Enclosing Subgraphs for Link Prediction. CIKM.

Acknowledgements
This research / project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Proposal ID: T2EP20122-0041). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.