Proceedings of the VLDB Endowment

Efficient Unsupervised Community Search with Pre-trained Graph Transformer

Jianwei Wang, Kai Wang, Xuemin Lin, Wenjie Zhang, Ying Zhang

jianwei.wang1@unsw.edu.au, w.kai@sjtu.edu.cn, xuemin.lin@sjtu.edu.cn,
wenjie.zhang@unsw.edu.au, ying.zhang@zjgsu.edu.cn

UNSW SYDNEY

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

浙江工商大学 ZHEJIANG GONGSHANG UNIVERSITY

Outline

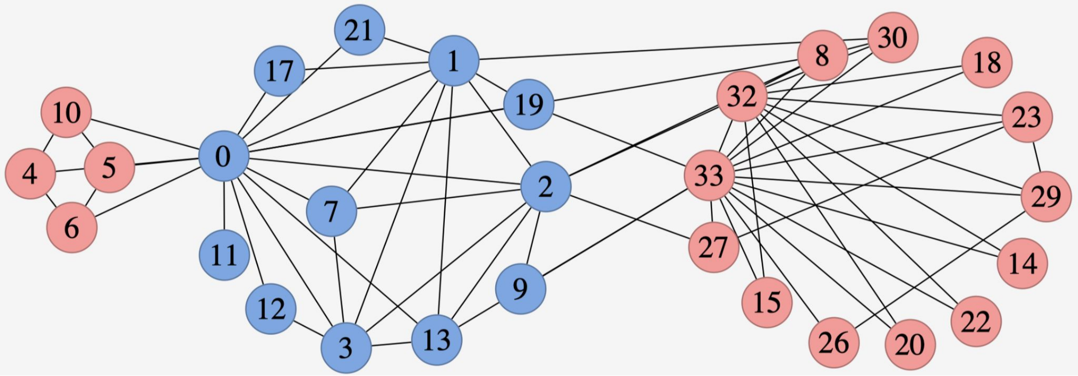
- Problem definition
- Motivations and Challenges
- Methods
- Experiments
- Summary

Problem Definition

- **Community:** Normally, a set of nodes that are densely connected.

- **Community Search:** Given a graph $G(V, E)$, and a query q where q is a set of query nodes, the task of community search (CS) aims to find a query-dependent community where nodes in the found community are densely intra-connected.

- **Applications**
  - Fraud detection.
  - Friend recommendation.
  - Protein complex identification.



Existing works and Motivations

- **Existing non-learning methods:**
  - ➢*k*-core based CS model
  - ➢*k*-truss based CS model
  - ➢*k*-ECC based CS model

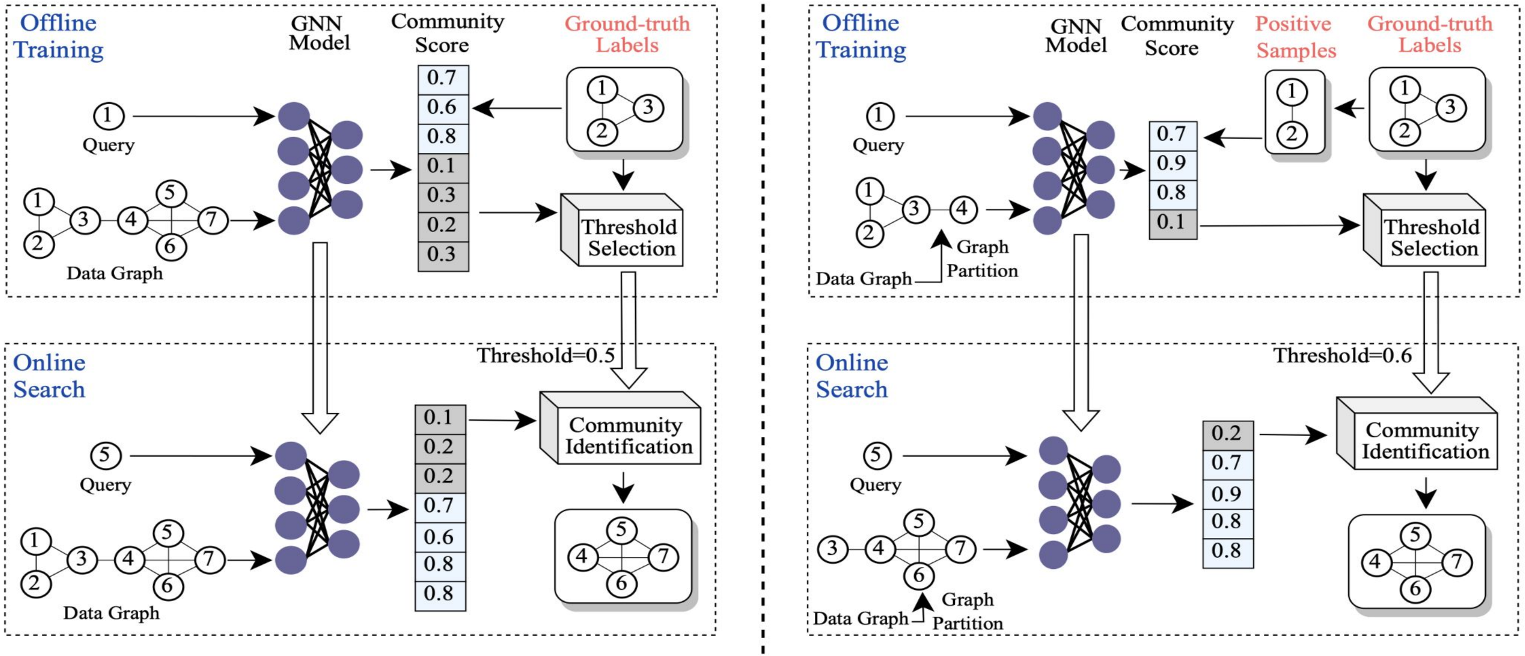
✔ Label Free
✖ Structure Flexibility

- **Existing learning-based methods:**
  - ➢QD-GNN
  - ➢COCELP

✖ Label Free
✔ Structure Flexibility

✔ Label Free
✔ Structure Flexibility

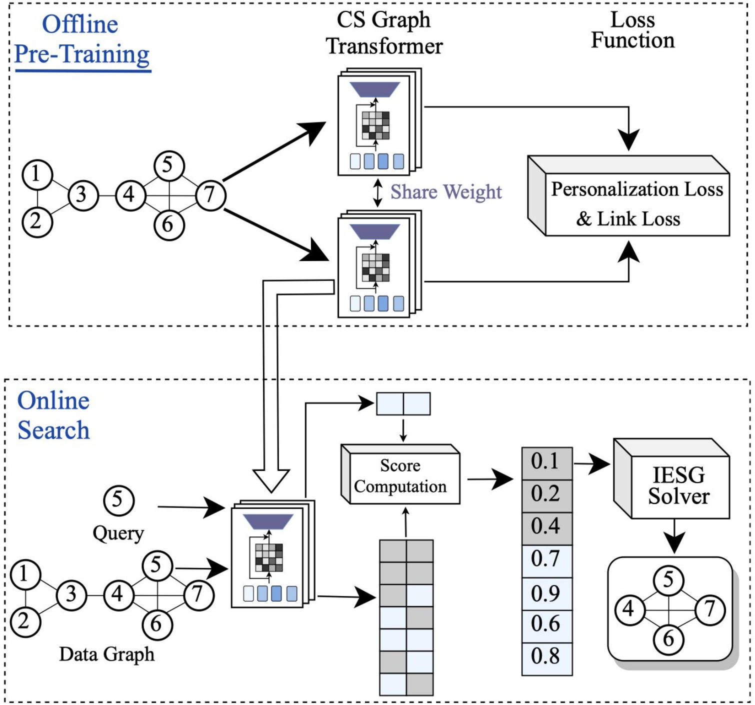
Existing Learning Frameworks



(a) QD-GNN (Supervised) (b) COCLEP (Semi-Supervised)

- Two-stage framework: Offline training phase and Online search phase
- Using labels for Community score learning and Community Identification

Our Method



- Two-stage framework:
  - Offline pre-training and Online search
- Unsupervised community score learning:
  - Offline pre-training with CSGphormer
  - && Online score computation via similarity
- Unsupervised community identification:
  - Identification with Expected Score Gain
  - && Local Search && Global Search

Offline Pre-training

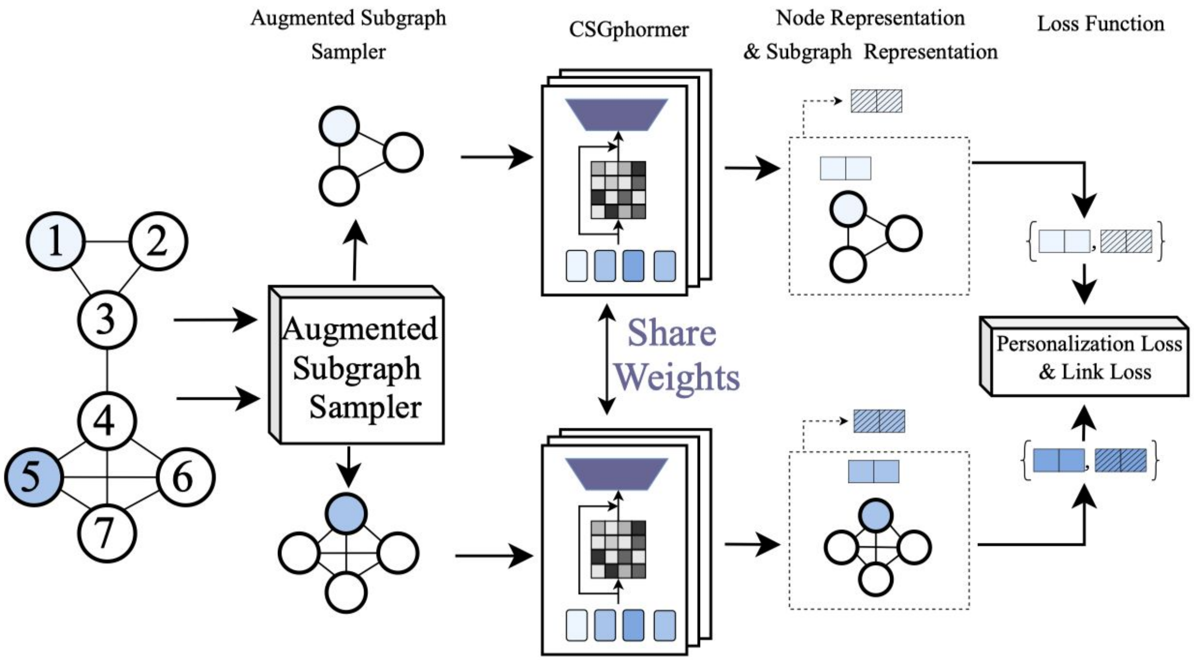


Figure 2: Illustration of the offline pre-training phase

Augmented subgraph Sampler && CSGphormer && Loss functions

Offline Pre-training: Augmented Subgraph Sampler

DEFINITION 2. *(Conductance [6, 46]). Given a graph $G(V, E)$ and a community C, the conductance of C is defined as:*

$$\Phi(G, C) = \frac{|e(C, \overline{C})|}{min(d_C, d_{\overline{C}})} \tag{1}$$

where $\overline{C} = V \backslash C$ is complement of C. $e(C, \overline{C})$ is the edges between nodes in C and nodes in $\overline{C}$. d_C is the sum of degrees of the nodes in C.

- Conductance-based augmented subgraph sampler
- K-hop subgraph with lowest conductance value

Offline Pre-training: CSGphormer

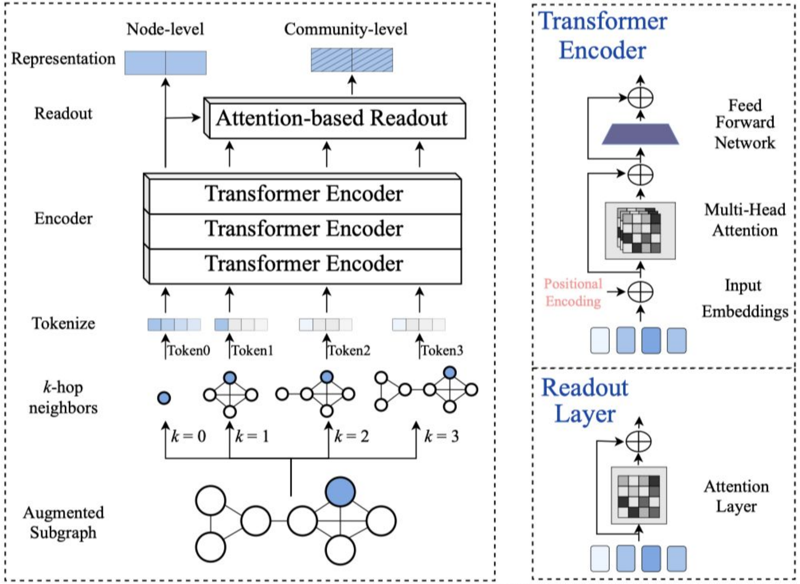


Figure 3: Architecture of *CSGphormer*

Algorithm 1: Forward Propagation of *CSGphormer*.

Input: center node v, feature matrix X, adjacent matrix A, transformer layers L.

Output: The node representation Z_v^{node} and community-level representation Z_v^{com}.

$\mathcal{X}_v \leftarrow \{{}^0x_v, {}^1x_v, \cdots, {}^Kx_v\}$

$H_v^{(0)} \leftarrow \mathcal{X}_v W$

`// L-layers transformer encoder.`

for $l = 0, \cdots, L-1$ **do**

- $P \leftarrow$ Position Encoding Construction
- $H_v^{(l)} \leftarrow H_v^{(l)} + P$
- $H_v^{(l+1)} = \mathrm{MHA}(\mathrm{LN}(H_v^{(l)})) + H_v^{(l)}$
- $H_v^{(l+1)} = \mathrm{FFN}(\mathrm{LN}(H_v^{(l+1)})) + H_v^{(l+1)}$

`// Readout layer.`

$Z_v^{node} \leftarrow {}^0H_v^{(L)}$; $Z_v^{com} \leftarrow$ Zero Tensor

for $k = 1, \cdots, K$ **do**

- $\alpha_k = \dfrac{\exp(({}^0H_v^{(L)} || {}^kH_v^{(L)})W_a^T)}{\sum_{i=1}^{K} \exp(({}^0H_v^{(L)} || {}^iH_v^{(L)})W_a^T)}$
- $Z_v^{com} \leftarrow Z_v^{com} + \alpha_k \, {}^kH_v^{(L)}$

return Z_v^{node}, Z_v^{com}

Offline Pre-training: Loss functions

- Personalization loss: central node is similar to its community while different from other's community

$$\mathcal{L}_p = \frac{1}{|V|^2}\sum_{v\in V}\sum_{u\in V}\left(-\max\left(\sigma(Z_v^{node} Z_v^{com}) - \sigma(Z_v^{node} Z_u^{com}) + \epsilon, 0\right)\right)$$

Contrastive loss

- Link loss: nodes that have a link should be close in the latent space

$$\mathcal{L}_k = \frac{1}{|V|^2}\sum_{v\in V}\sum_{u\in V} -A(u,v)(Z_u^{node} Z_v^{node}) + (1 - A(u,v))(Z_u^{node} Z_v^{node})$$

Generative loss

- Overall loss: $\mathcal{L} = \mathcal{L}_p + \alpha\mathcal{L}_k$

Online Search: Score Computation

Algorithm 3: Community Score Computation

Input: The query V_q, graph G, pre-trained network $f^{\theta}(\cdot)$.

Output: The community score S.

Initialize $S \leftarrow \{s_v = 0 \text{ for } v \in V\}$

for $\{v\} \in V$ **do**

  for $\{u\} \in V_q$ **do**

$$s_v \leftarrow s_v + \frac{\sum_{i=0}^{d_m^{(L)}} f_i^{\theta}(v) f_i^{\theta}(u)}{\sqrt{\sum_{i=0}^{d_m^{(L)}} f_i^{\theta}(v) f_i^{\theta}(v)} \times \sqrt{\sum_{i=0}^{d_m^{(L)}} f_i^{\theta}(u) f_i^{\theta}(u)}}$$

  $s_v \leftarrow \frac{s_v}{|V_q|}$;

return S

Pairwise

Cosine Similarity

Online Search: IESG

- Expected Score Gain: $ESG(S, C, G) = \frac{1}{|V_C|^{\tau}}\left(\sum_{v \in V_C} s_v - \frac{\sum_{u \in V} s_u}{|V|}|V_C|\right)$

# of internal nodes

τ is a hyperparameter to control granularity

sum of internal scores

expected score for nodes in the community

- Identification with expected score gain

Definition 4. *(Identification with Expected Score Gain). Given a graph $G(V, E)$, the query V_q, the community score S and a profit function $ESG(\cdot)$, IESG aims to select a community C of G, such that:*

(1) V_C contains nodes in V_q, and C is connected;

(2) $ESG(S, C, G)$ is maximized among all feasible choices for C.

query-driven && cohesive constraint

nodes with high community score

- The problem of IESG is NP-hard

Online Search: IESG Solver

Algorithm 4: *Local Search* Algirithm

Input: The community score S, graph G and query V_q.

Output: The identified community $\tilde{C}_q$.

$\tilde{C}_q, Q \leftarrow V_q$; $max\_esg \leftarrow -inf$

while $|Q| < |V|$ **do**

  $u \leftarrow \text{argmax}_{v \in \zeta \overline{Q}} s_v$

  $Q = Q \cup u$;

  if $ESG(S, \tilde{C}_q \cup \{u\}, G) > max\_esg$ **then**

    $max\_esg \leftarrow \text{ESG}(S, \tilde{C}_q \cup \{u\}, G)$

    $\tilde{C}_q = \tilde{C}_q \cup \{u\}$

  else

    Terminate

return $\tilde{C}_q$

highest score in the neighborhood

Algorithm 5: *Global Search* Algorithm

Input: The community score S, graph G and query V_q.

Output: The identified community $\tilde{C}_q$.

$\tilde{C}_q \leftarrow V_q; t_s = 0; t_e = |S|$

$\hat{S} \leftarrow$ sort S from large to small

while $t_s < t_e$ **do**

  $C_{mid} = \{v_i | \hat{s}_i \geq \hat{s}_{\frac{ts+te}{2}}\}$

  $C_{left} = \{v_i | \hat{s}_i \geq \hat{s}_{\frac{ts+te}{2}-1}\}$

  if $ESG(\hat{S}, C_{mid}, G) > ESG(\hat{S}, C_{left}, G)$ **then**

    $t_s \leftarrow \frac{t_s + t_e}{2}$

  else

    $t_e \leftarrow \frac{t_s + t_e}{2}$

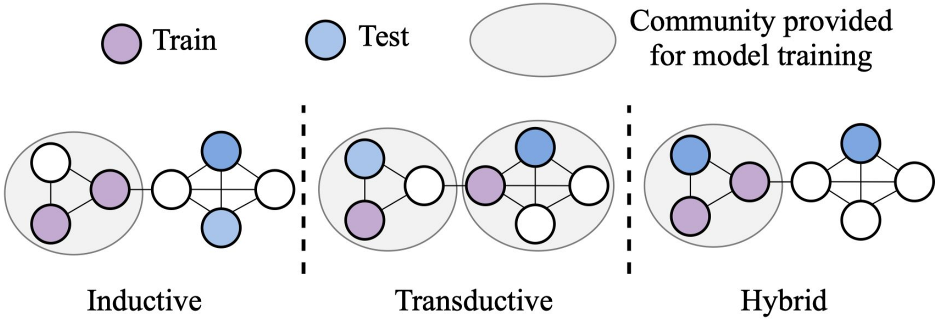
return $\tilde{C}_q = \tilde{C}_q \cup \{v_i | \hat{s}_i \geq \hat{s}_{t_e}\}$

global highest score

Experiments: Dataset and query generation

Table 3: Statistics of the datasets

Datasets	$\lvert V \rvert$	$\lvert E \rvert$	$\lvert C \rvert$	d
Texas	183	325	5	1,703
Cornell	183	298	5	1,703
Wisconsin	251	515	5	1,703
Cora	2,708	10,556	7	1,433
Citeseer	3,327	9,104	6	3,703
Photo	7,650	238,162	8	745
DBLP	17,716	105,734	4	1,639
CoCS	18,333	163,788	15	6,805
Physics	34,493	495,924	5	8,415
Reddit	232,965	114,615,892	41	602



- **Metrics**
  - F1-score
  - Normalized Mutual Information (NMI)
  - Jaccard similarity (JAC)

- **Query settings**
  - Inductive (the ability for unseen community)
  - Transductive
  - Hybrid

Experiments: F1-score results

Table 4: F1-score results under different settings

Settings	Models	Texas	Cornell	Wisconsin	Cora	Citeseer	Photo	DBLP	CoCS	Physics	Reddit	Average +/-
Inductive	CST	0.1986	0.1975	0.2251	0.2111	0.1423	0.2019	0.2854	0.1252	0.2276	0.1463	-27.12%
	EquiTruss	0.3120	0.3168	0.3079	0.2384	0.2240	0.2166	0.3252	0.1225	0.2471	0.2163	-21.46%
	M*k*ECS	0.3581	0.3177	<u>0.3404</u>	0.2364	0.2015	0.1975	0.2768	0.1152	0.2193	0.2068	-22.03%
	CTC	0.3211	<u>0.3482</u>	0.3327	0.2558	0.2418	0.2626	0.3417	0.1059	0.2511	0.2431	-19.69%
	QD-GNN	0.0821	0.0669	0.0683	0.0322	0.0536	0.0018	0.0372	0.0145	OOM	OOM	-41.50%
	COCLEP	<u>0.4044</u>	0.2960	0.1804	0.3094	0.3058	0.4413	0.3066	0.4253	0.3389	0.2696	-13.95%
	TransZero-LS	0.1801	0.1583	0.2074	<u>0.5467</u>	<u>0.3906</u>	<u>0.5725</u>	**0.4407**	<u>0.4292</u>	<u>0.5075</u>	**0.4879**	-7.52%
	TransZero-GS	**0.4283**	**0.3716**	**0.3755**	**0.5764**	**0.4535**	**0.6018**	<u>0.4326</u>	**0.4374**	**0.5113**	<u>0.4848</u>	-
Transductive	QD-GNN	0.6703	0.8408	0.6247	0.5062	0.4726	0.2205	0.4918	0.6356	OOM	OOM	+9.81%
	COCLEP	0.4020	0.3167	0.3206	0.3685	0.3331	0.5060	0.3763	0.3549	0.4388	0.3270	-9.29%
Hybrid	QD-GNN	0.3852	0.3644	0.5956	0.4789	0.4097	0.0833	0.3902	0.4969	OOM	OOM	-5.91%
	COCLEP	0.3883	0.3313	0.2938	0.3615	0.3067	0.4388	0.3733	0.4027	0.4693	0.3071	-10.01%

∗ CST, EquiTruss, M*k*ECS, CTC and *TransZero* have consistent results under three settings as they are label-free. *TransZero* with *Local Search* is denoted as *TransZero*-LS, and *TransZero* with *Global Search* is denoted as *TransZero*-GS. OOM indicates out-of-memory. The last column presents the average margin compared to *TransZero*-GS.

- ✔ TransZero has an outstanding performance, especially under the inductive setting.

Experiments: NMI and JAC results

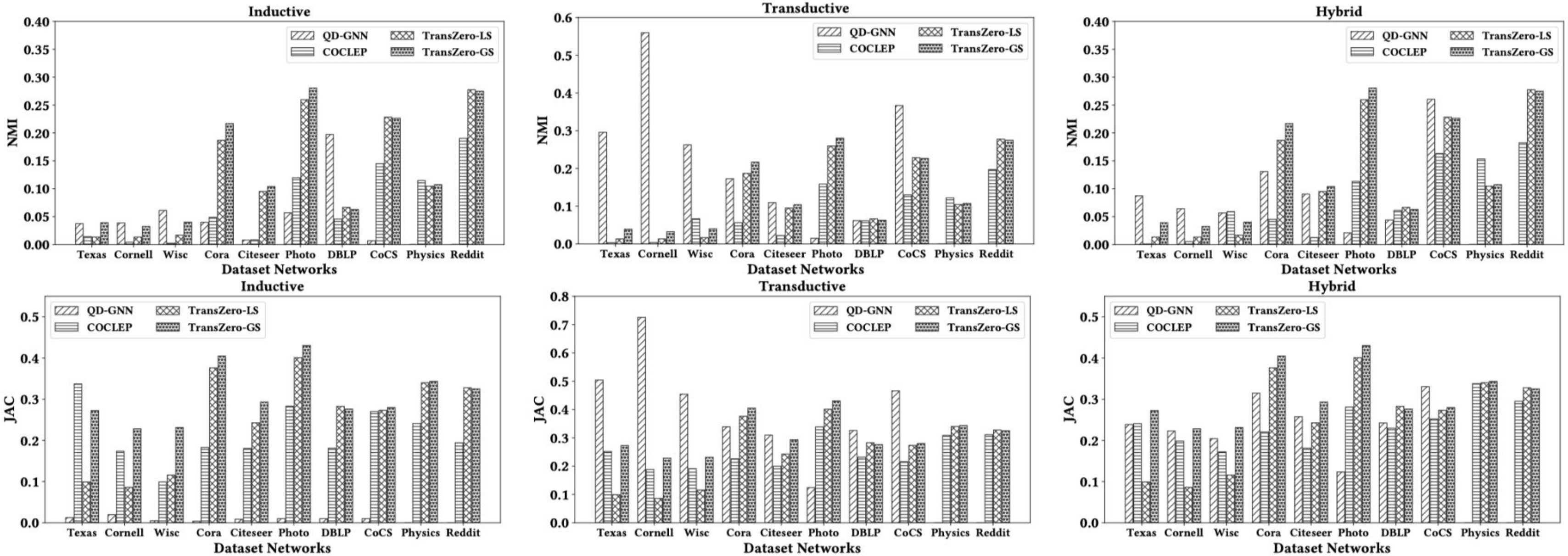
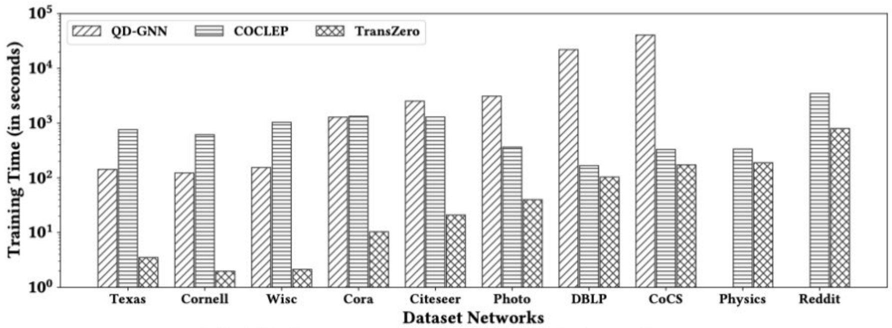


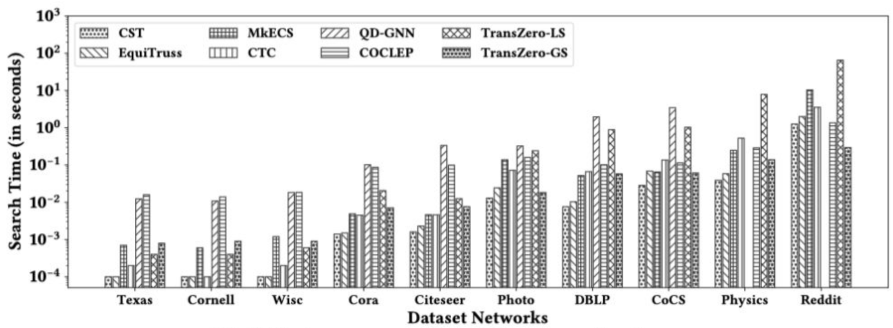
Figure 6: NMI and JAC results under different settings

TransZero has a competitive performance using NMI and JAC as metrics

Experiments: Efficiency



(a) Efficiency results of the training phase

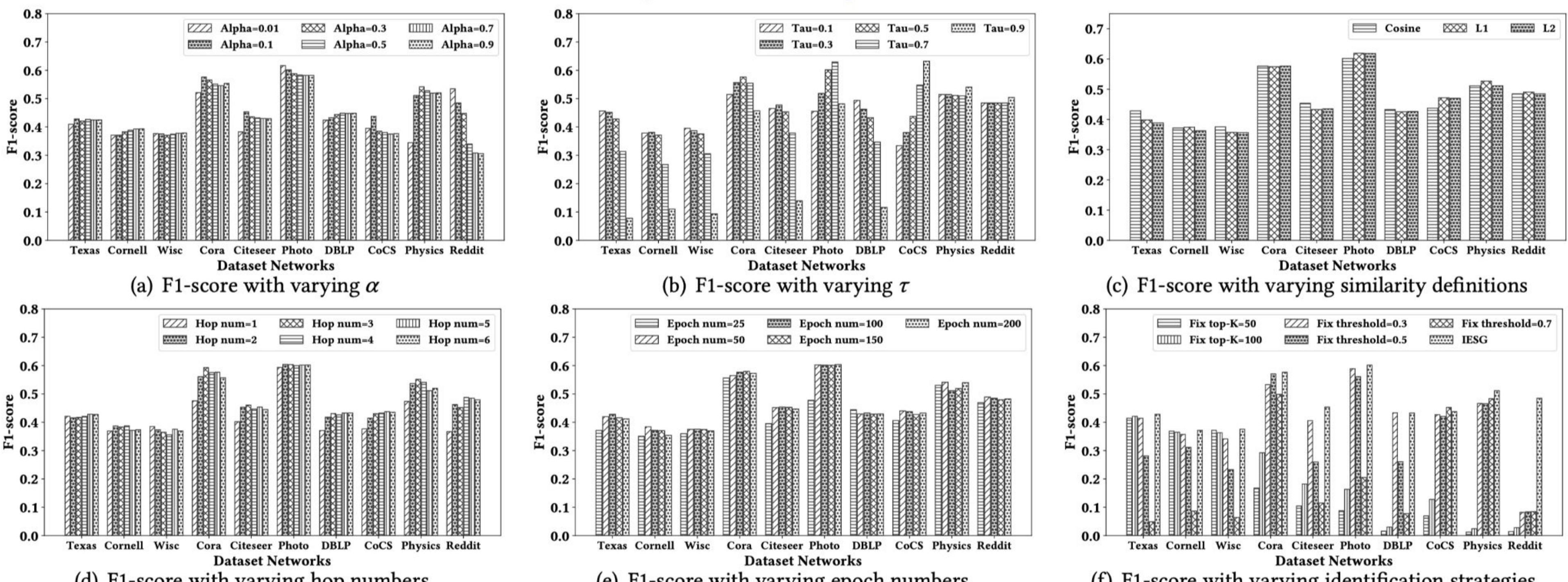


(b) Efficiency results of the search phase

Figure 7: Efficiency results

- TransZero has a better efficiency in the offline training phase and can deal large graph
- TransZero-GS has a better efficiency in the online search phase compared to learning methods.

Experiments-Hyperparameter



(a) F1-score with varying α (b) F1-score with varying τ (c) F1-score with varying similarity definitions

(d) F1-score with varying hop numbers (e) F1-score with varying epoch numbers (f) F1-score with varying identification strategies

Figure 8: Hyper-parameter analysis results

Experiments: Ablation study

Table 5: Ablation study

Models	Texas	Cornell	Wisconsin	Cora	Citeseer	Photo	DBLP	CoCS	Physics	Reddit	Average +/−
Full model	0.4283	0.3716	0.3755	0.5764	0.4535	0.6018	0.4326	0.4374	0.5113	0.4848	-
w/o $\mathcal{L}_p$	0.4215	0.3749	0.3773	0.5462	0.4259	0.5716	0.4501	0.3502	0.5183	0.2981	-3.19%
w/o $\mathcal{L}_k$	0.3894	0.3576	0.3579	0.4203	0.3044	0.6116	0.4087	0.4532	0.3506	0.5076	-5.12%
w/o Conductance Aug	0.4212	0.3692	0.3848	0.4755	0.4019	0.5935	0.3708	0.3766	0.4738	0.4167	-3.89%
w/o *CSGphormer*	0.3317	0.2421	0.2169	0.4048	0.2780	0.4473	0.2708	0.3074	0.3435	0.3649	-14.65%

- All the designed components can enhance the performance
- CSGphormer can bring the largest enhancement

Summary

- **We propose a learning-based unsupervised community search framework, named TransZero.**

- **In the offline phase, an efficient graph transformer CSGphormer.**

- **In the online phase, we calculate the community score by similarity of learned similarity. We model the community identification as Identification with Expected Score Gain (IESG). We propose Local Search and Global Search for IESG.**

- **Extensive experiments over 10 popular public datasets demonstrate the effectiveness of TransZero.**

Q & A

Code and Data available in: