Appendix

Datasets

We select four real-world datasets as follows. The detailed statistics of these datasets are shown in Table 1.

Facebook/Twitter Facebook is one of the largest social media platforms that allows users to connect with friends and family. Twitter is a microblogging social media platform that allows users to post short messages and interact with other users. About.Me is a third-party platform that allows users to create a personal profile that includes links to social media accounts, e.g., Facebook and Twitter. The dataset was collected by (Cao and Yu 2016) and constructed by (Du et al. 2022), containing 1043 nodes in each layer.

Twitter/YouTube YouTube is a video-sharing platform that allows users to upload, share, and view video content. This dataset has been obtained starting from Friendfeed, a social media aggregator. In Friendfeed, users can directly post messages and comments on other messages much like in Twitter and other similar OSNs, where they can also register their accounts on other systems. From these, a multilayer network dataset was collected by (Dickison, Magnani, and Rossi 2016), including the Twitter layer and YouTube layer. We select subnetworks of these two layers, with 2177 interlayer links.

Douban-online/Douban-offline Douban is a social media platform in China, which allows users to generate content related to movies, books, music, and other topics. This dataset was collected in 2010 by (Zhong et al. 2012). Each user has information including offline event participation. The offline network contains 1118 users according to users' co-occurrence in social gatherings, and the online network with 3,906 nodes contains all these offline users (Zhang and Tong 2019).

Douban/Weibo Weibo is a microblogging social media platform in China. Douban users can present their Weibo accounts on the homepage, providing interlayer information. The Douban/Weibo dataset was collected by (Hong et al. 2022). We select subnetworks of these two layers, with 4293 interlayer links.

Datasets	Nodes#	Edges#	Anchors#
Facebook	1043	4734	1043
Twitter	1043	4860	
Twitter	2562	6967	2177
YouTube	2409	7862	
Douban-onine	3906	8164	1118
Douban-offline	1118	1511	
Douban	4607	62018	4293
Weibo	4393	29893	

Table 1: Structural statistics of the datasets.

Hyperparameter Settings and Training Details

The depth of the CGNN model $K = 2$, where the first layer is a GCN-form layer, and the following layer is a cross-GNN layer. Parameters in the cross-GNN layer for both two networks are shared. The input node features are the adjacency matrices of the two networks, the hidden dimension is 256, and the embedding dimension is 128. Then to construct the random-walk-based intralayer loss, the window size is 10, walks per node is 10, walk length is 20, $Q = 1$, and $b_r = 512$. To construct the interlayer loss, $\eta = 1, b_s = |S^+|$, and ϵ is selected among $\{0, 0.1, \ldots, 0.8, 0.85, 0.9, 0.95, 1\}$. To construct the total loss, α is selected among $\{0, 0.01, 0.05, 0.1, 0.15, 0.2, 0.3, \ldots, 0.9, 1\}$. The learning rate is selected among $\{0.01, 0.001, 0.0001\}$, and we use Adam optimizer. Note that we directly use cosine similarity to predict the linkage probability on Facebook/Twitter dataset, as overfitting occurs when using a Logistic Regression classifier. All experiments were carried out 10 times and averaged. Some important hyperparameter settings of different datasets are demonstrated in Table 2.

Datasets	α	ϵ	epochs
Facebook/Twitter	0.05	0.7	2000
Twitter/YouTube	0.9	0.9	50
Douban-online/Douban-offline	0.1	0.9	500
Douban/Weibo	0.9	0.9	40

Table 2: Some hyperparameter settings of different datasets.

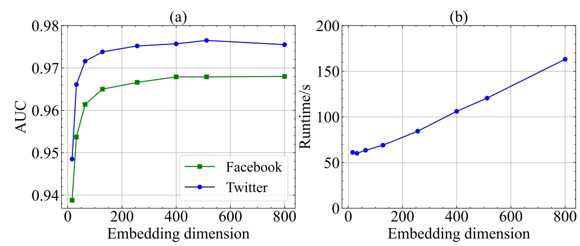

Figure 1: Performance with different embedding dimensions. (a) AUC *vs.* embedding dimension; (b) Runtime *vs.* embedding dimension.

Additional Experimental Results

Effectiveness Analysis We also compare our proposed method with baselines on more real-world datasets such as Douban-online/Douban-offline and Douban/Weibo. For each dataset, we randomly select 90% edges as the training set, and the rest 10% edges as the test set. Then we randomly sample negative test edges of the same size as the test set for evaluation. The performance in Douban-online/Douban-offline and Douban/Weibo is shown in Table 3. CGNN is almost the best, showing the effectiveness of our proposed method.

Parameter Sensitivity Study Figure 1 shows the impact of embedding dimension d, whose value varies as {16, 32, 64, 128, 256, 400, 512, 800}. One can observe that performance first grows rapidly, and then grows slowly when $d > 64$. Meanwhile, the runtime exhibits linear growth, confirming that the complexity of the proposed method is in proportion to the embedding dimension. The performance at $d = 128$ comes to $99.7\%+$ of that at $d = 800$, so we se-

Dataset	Method	Ratio of Interlayer Links			
		0	30%	60%	90%
Douban-online ↕ Douban-offline	CN	0.5969 / 0.6892	0.5976 / 0.6959	0.6000 / 0.7094	0.6031 / 0.7508
	RA	0.5973 / 0.6910	0.5979 / 0.6978	0.6004 / 0.7112	0.6035 / 0.7548
	SVD	0.6544 / 0.7383	0.6496 / 0.7396	0.6540 / 0.7655	0.6564 / 0.7819
	n2v	0.6194 / 0.7489	0.6189 / 0.7772	0.6451 / 0.8154	0.6530 / 0.9226
	GAE	0.6239 / 0.7097	0.6225 / 0.7695	0.6328 / 0.7899	0.6642 / 0.9066
	GAT	0.6337 / 0.7326	0.6427 / 0.7849	0.6440 / 0.8240	0.6735 / 0.9093
	MAA	0.5973 / 0.6910	0.5979 / 0.6969	0.5985 / 0.7139	0.5986 / 0.7537
	LPIS	0.8362 / 0.8419	0.8362 / 0.8304	0.8158 / 0.8288	0.7716 / 0.9183
	CLF	0.8122 / 0.8660	0.8188 / 0.8898	0.8205 / **0.9409**	0.8231 / **0.9649**
	n2v-e	0.6194 / 0.7489	0.6441 / 0.8213	0.6419 / 0.8933	0.6563 / 0.9165
	EG-mini	0.6828 / 0.5000	0.6922 / 0.6691	0.6987 / 0.7700	0.7234 / 0.8084
	CELP	0.7580 / 0.8280	0.7313 / 0.8411	0.7441 / 0.8861	0.7553 / 0.9559
	CGNN	**0.8323 / 0.8775**	**0.8345 / 0.8918**	**0.8348** / 0.9048	**0.8410** / 0.9464
Douban ↕ Weibo	CN	0.8344 / 0.7942	0.8357 / 0.8011	0.8414 / 0.8191	0.8455 / 0.8375
	RA	0.8397 / 0.7986	0.8418 / 0.8063	0.8462 / 0.8244	0.8531 / 0.8479
	SVD	0.7286 / 0.7105	0.7261 / 0.7119	0.7259 / 0.7175	0.7233 / 0.7131
	n2v	0.7408 / 0.7632	0.7443 / 0.7706	0.7504 / 0.7776	0.7707 / 0.8046
	GAE	0.8297 / 0.8084	0.8343 / 0.8067	0.8284 / 0.8138	0.8366 / 0.8343
	GAT	0.8132 / 0.8000	0.8074 / 0.7927	0.8118 / 0.8046	0.8169 / 0.8079
	MAA	0.8399 / 0.7990	0.8389 / 0.8003	0.8408 / 0.8083	0.8437 / 0.8207
	LPIS	0.8855 / 0.8934	0.8761 / 0.8650	0.8548 / 0.8116	0.8328 / 0.7665
	CLF	0.8376 / 0.8457	0.8333 / 0.8411	0.8229 / 0.8276	0.8147 / 0.8171
	n2v-e	0.7408 / 0.7632	0.8241 / 0.8284	0.8298 / 0.8501	0.8383 / 0.8662
	EG-mini	0.8891 / 0.9097	0.8903 / 0.9116	0.8946 / 0.9205	**0.9043** / 0.9232
	CELP	0.8929 / 0.8882	0.8671 / 0.8797	0.8612 / 0.8776	0.8627 / 0.8931
	CGNN	**0.8950 / 0.9267**	**0.8964 / 0.9225**	**0.8948 / 0.9228**	0.8957 / **0.9257**

Table 3: AUC with different ratios of interlayer links.

Dataset	Method	Ratio of Interlayer Links		
		30%	60%	90%
Facebook/Twitter	CGNN	0.9105 / 0.9164	0.9444 / 0.9550	0.9653 / 0.9740
	$\mathrm{CGNN_e}$	0.9107 / 0.9165	0.9444 / 0.9552	0.9650 / 0.9739
Twitter/YouTube	CGNN	0.9327 / 0.9094	0.9353 / 0.9125	0.9353 / 0.9108
	$\mathrm{CGNN_e}$	0.9322 / 0.9122	0.9350 / 0.9119	0.9370 / 0.9123
Douban-online/Douban-offline	CGNN	0.8345 / 0.8918	0.8348 / 0.9048	0.8410 / 0.9464
	$\mathrm{CGNN_e}$	0.8347 / 0.8888	0.8358 / 0.9064	0.8394 / 0.9443
Douban/Weibo	CGNN	0.8964 / 0.9225	0.8948 / 0.9228	0.8957 / 0.9257
	$\mathrm{CGNN_e}$	0.8917 / 0.9249	0.8951 / 0.9241	0.8980 / 0.9246

Table 4: Impact of network extension on CGNN.

Method	SVD	CELP	GAE	GAT	CGNN
Runtime/s	37.26	55.69	77.40	86.34	68.46

Table 5: Runtime of different methods on Facebook/Twitter dataset.

lect $d = 128$ to balance performance and time cost in other experiments.

Runtime We recorded the runtime of several methods for link prediction in Facebook/Twitter dataset with 90% training edges and 90% interlayer links. SVD, GAE, and GAT are modified by the network extension strategy. GAE and GAT are both GNN-based methods and have the same settings as CGNN. SVD, GAE, GAT, and CGNN are implemented by PyTorch and were run on an NVIDIA A100-40G GPU in a CentOS Linux 7.5 server. CELP is a random-walk-based method implemented by Gensim and was run on an 11th Gen Intel(R) Core(TM) i5-11400 @ 2.60GHz CPU in a desktop computer. The runtime results are shown in Table 5. One can find that CGNN has similar complexity to GAE and GAT.

Network Extension Network extension is helpful for applying single-layer link prediction methods to multilayer networks. *Does network extension help with our proposed method?* To answer this question, we conducted network extension before using our method (referred to as $\mathrm{CGNN_e}$) on different datasets with 90% training edges. Results are demonstrated in Table 4. We can observe that network extension helps little with our method, and even has a negative impact sometimes. It indicates that our proposed method has contained the additional information provided by network extension.

References

Cao, X.; and Yu, Y. 2016. BASS: A bootstrapping approach for aligning heterogenous social networks. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD-16)*, 459–475.

Dickison, M. E.; Magnani, M.; and Rossi, L. 2016. *Multilayer social networks*. Cambridge University Press.

Du, X.; Yan, J.; Zhang, R.; and Zha, H. 2022. Cross-Network Skip-Gram Embedding for Joint Network Alignment and Link Prediction. *IEEE Transactions on Knowledge & Data Engineering*, 34(03): 1080–1095.

Hong, H.; Li, X.; Pan, Y.; and Tsang, I. W. 2022. Domain-Adversarial Network Alignment. *IEEE Transactions on Knowledge & Data Engineering*, 34(07): 3211–3224.

Zhang, S.; and Tong, H. 2019. Attributed Network Alignment: Problem Definitions and Fast Solutions. *IEEE Transactions on Knowledge & Data Engineering*, 31(09): 1680–1692.

Zhong, E.; Fan, W.; Wang, J.; Xiao, L.; and Li, Y. 2012. Comsoc: adaptive transfer of user behaviors over composite social network. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD-12)*, 696–704.