Algorithm 1: Training process of LMCH

Input: Source HG datasets $\mathcal{S}$, the target HG dataset $\mathcal{G}$
Output: The learned parameters Θ_{LM}, Θ_{GNN}, LM-encoded embedding $\mathbf{Z}$ of the target HG

Initialize Θ_{LM}, Θ_{GNN}
for each node $v \in \mathcal{S}$ **do**
  Generate metapath-based corpus W_v by node/edge textualization and metapath textualization
  Obtain the LM-encoded embedding z_v using (1)
  Decode z_v to an N-way classification space $\hat{y}_{v\text{-LM}}$ using softmax$(\text{MLP}(\boldsymbol{z}_v))$
  Calculate $\mathcal{L}_{\text{LM}}$ using (2)
end for
for each node $v \in \mathcal{G}$ **do**
  Generate $\boldsymbol{h}_v^{(0)}$ for GNN in the first iteration using (1)
  while Θ_{LM} and Θ_{GNN} have not converged **do**
    Obtain GNN-encoded node embedding $\boldsymbol{h}_v^{(L)}$ using (3)
    Calculate $\mathcal{L}_{\text{LM}\to\text{GNN}}$ using (4)
    Back propagation and update Θ_{GNN}
    Update soft labels $\overline{y}_v$ using MLP$(\boldsymbol{h}_v^{(L)})$
    Obtain LM-encoded node embedding $\boldsymbol{z}_v$ using (1)
    Calculate $\mathcal{L}_{\text{GNN}\to\text{LM}}$ using (5)
    Back propagation and update Θ_{LM}
    Calculate $\mathcal{L}_{\text{Align}}$ using (6)
  end while
end for
return Θ_{LM}, Θ_{GNN}, LM-encoded node embeddings $\mathbf{Z}$

A Appendix

A.1 Pseudo Algorithm

Details of our LMCH training process are shown in Algorithm 1.

A.2 Hyper-parameter Study

We assess the model's sensitivity to the number of LM fine-tuning epochs and the LM embedding dimension b, with results presented in Figure 4. As the number of LM fine-tuning epochs increases, performance initially improves but eventually declines. This occurs because more epochs provide the LM with greater opportunities to learn general information from the source heterogeneous graph. However, since the amount of general knowledge in the source HGs is limited, beyond a certain number of epochs, the LM has captured sufficient information. An increase in epochs beyond this point leads to the incorporation of noise detrimental to knowledge transfer, resulting in decreased performance. The trend of LM embedding dimension variation in Figure 4 (b) follows a similar pattern. This is because an optimal dimension is required to effectively encode heterogeneous information, and excessively high dimensions may introduce redundancy.

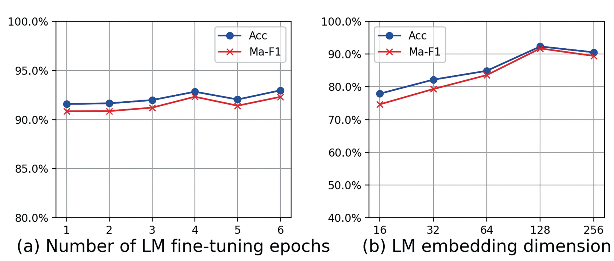

Figure 4: Node classification accuracy and Macro-F1 score of LMCH with different hyper-parameter settings on DBLP dataset in a 3-way 3-shot setting.

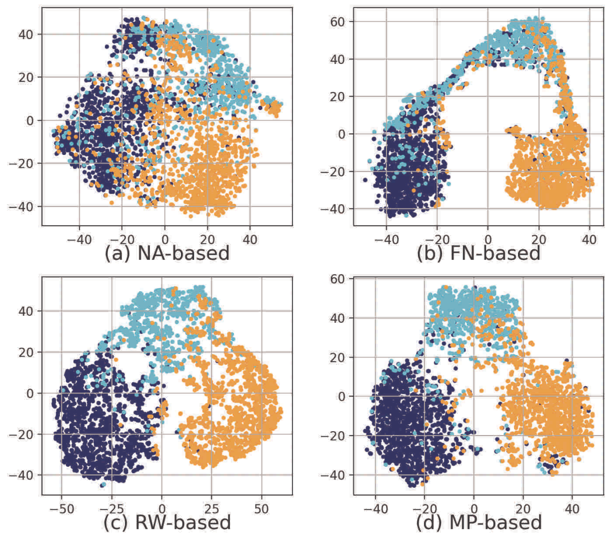

Figure 5: Visualization embedding on DBLP. Each point indicates one node and its color indicates the class.

A.3 Case Study

We projected the DBLP dataset's node embeddings into a 2-dimensional vector space using t-SNE, with different colors denoting distinct node classes. From Figure 5, it is evident that the MP-based method yields the best result because metapaths can simultaneously consider both local structures and long-range semantic relationships inherent in HGs. The RW-based method achieves near-optimal results for the reason that random walk introduces additional noise and finally leads to a performance degradation. The NA-based and FN-based methods that only consider local structures represent poor results on the DBLP dataset. The embeddings learned using the NA-based and FN-based methods show unclear boundaries between different classes and low intra-class similarity.

A.4 Dataset Statistics

Table 4 displays the metapaths used for corpus construction and other detailed statistics for IMDB, DBLP (Wang et al. 2019b), YELP (Lu et al. 2019) and PubMed (Tan et al. 2024) datasets.

Dataset	Node	Metapath	Labeled Node Type	Class
IMDB	Movie (M): 4,278 Actor (A): 5,257 Director (D): 2,081	MAM, MDM AMA, AMDMA DMD, DMAMD	Movie	4
DBLP	Author (A): 4,057 Paper (P): 14,328 Term (T): 7,723 Conference (C): 20	APA, APCPA, APTPA PAP, PTP, PCP TPT, TPCPT, TPAPT CPC, CPAPC, CPTPC	Author	4
YELP	Business (B): 2,614 User (U): 1,286 Service (S): 9 Star Level (L): 9 Reservation (R): 2	BUB, BSB, BLB, BRB UBU, UBSBU, UBLBU, UBRBU SBS, SBUBS, SBLBS, SBRBS LBL, LBUBL, LBSBL, LBRBL RBR, RBUBR, RBSBR, RBLBR	Business	3
PubMed	Disease (D): 20,163 Gene (G): 13,561 Chemical (C): 26,522 Species (S): 2,863	DGD, DCD, DSD GDG, GCG, GSG CDC, CGC, CSC SDS, SGS, SCS	Disease	8

Table 4: Statistics of datasets.

A.5 Implementation Details

Our experiments are conducted on a Linux operating system using a single NVIDIA A800 GPU with 80GB of memory. The implementation of our LMCH leverages libraries such as PyTorch (Paszke et al. 2019), PyTorch Geometric (Fey and Lenssen 2019), DGL (Wang et al. 2019a), scikit-learn (Pedregosa et al. 2011), and Transformers (Wolf et al. 2020). Hyper-parameter configurations are detailed in Table 5.

Training Phase	Hyper-Parameter	Value
Cross-Heterogeneity LM Fine-Tuning	training epochs	4
	dropout rate	0.4
	batch size	32
	optimizer	AdamW
	learning rate	1e-5
	weight decay	0.01
	L2 regularization λ_{LM}	1e-2
GNN-Supervised LM Training	max iteration	
	embedding dimension	10
	GNN/LM training epochs	128
	GNN/LM batch size	200/5
	LM batch size	500/32
	optimizer	AdamW
	GNN/LM learning rate	5e-4/1e-5
	GNN/LM weight decay	1e-5/0.01
	GNN/LM dropout rate	0.4/0.1
	LM dropout rate	1e-5
	GNN L2 regularization λ_{GNN}	
LM-GNN Contrastive Alignment	training epochs	25
	optimizer	Adam
	learning rate	2e-5

Table 5: Hyper-parameter configurations.

A.6 Baseline Descriptions

Detailed descriptions of the baselines are as follows:

- **GCN** (Kipf and Welling 2016) is a semi-supervised graph convolutional network specifically designed for homogeneous graphs. In our experiments, we convert heterogeneous graphs into homogeneous graphs to facilitate the training process.
- **GAT** (Veličković et al. 2017) is a homogeneous graph netural network considering the attention mechanism. Here we also convert heterogeneous graphs into homogeneous graphs for training.
- **RGCN** (Schlichtkrull et al. 2018) views a HG as multi-relational homogeneous graphs, transforming the problem into aggregating features from multiple homogeneous graphs.
- **HAN** (Wang et al. 2019b) develops a hierarchical attention structure that includes node-level attention and semantic-level attention.
- **MAML** (Finn, Abbeel, and Levine 2017) is a model-agnostic meta-learning method based on learning easily adaptable model parameters through gradient descent.
- **ProtoNet** (Snell, Swersky, and Zemel 2017) is a few-shot learning method which represents each class by the mean of its examples in a representation space learned by a neural network.
- **GPN** (Ding et al. 2020) is a ProtoNet-based method that considers the importance of nodes in the support set when constructing the class prototypes.
- **G-Meta** (Huang and Zitnik 2020) applies the ProtoNet algorithm and leverages local subgraphs of nodes to transfer subgraph-specific information.
- **GLEM** (Zhao et al. 2022) is a variational expectation maximization framework that alternatively updates a LM and a GNN.

	Pre-Training	Fine-Tuning	Inference
GLEM	141.25s	218.89s	26.15s
LMBot	143.71s	219.18s	27.52s
LMCH	328.70s	567.02s	30.45s

Table 6: Time comparison of three stages (pre-training, fine-tuning, and inference) with the IMDB 3-way 3-shot cross-heterogeneity node classification task.

- **LMBot** (Cai et al. 2024) is a Twitter bot detection method that integrates GNN-learned graph information into a language model and then realized graph-less deployment.
- **CGFL** (Ding, Wang, and Liu 2023) is a meta-learning method which encodes shared relations in HGs and evaluates source HGs by a three-level score module to achieve knowledge transfer.

A.7 Time Complexity

To provide a more intuitive comparison, we consider the time cost of three stages (pre-training, fine-tuning, and inference) with the IMDB *3*-way *3*-shot cross-heterogeneity node classification task using LMCH, along with two other competitive methods, GLEM and LMBot. As shown in the Table 6, due to the longer token length of metapath-based corpora, LMCH incurs more time than GLEM and LMBot. However, the absolute training time of LMCH is still under 20 minutes, and its inference time is comparable to the others. Given the performance improvements, the additional time complexity is acceptable.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Cai, Z.; Tan, Z.; Lei, Z.; Zhu, Z.; Wang, H.; Zheng, Q.; and Luo, M. 2024. LMbot: distilling graph knowledge into language model for graph-less deployment in twitter bot detection. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 57–66.

Chien, E.; Chang, W.-C.; Hsieh, C.-J.; Yu, H.-F.; Zhang, J.; Milenkovic, O.; and Dhillon, I. S. 2021. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, K.; Wang, J.; Li, J.; Shu, K.; Liu, C.; and Liu, H. 2020. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 295–304.

Ding, P.; Wang, Y.; and Liu, G. 2023. Cross-heterogeneity graph few-shot learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 420–429.

Dong, Y.; Tang, J.; Wu, S.; Tian, J.; Chawla, N. V.; Rao, J.; and Cao, H. 2012. Link prediction and recommendation across heterogeneous social networks. In *2012 IEEE 12th International conference on data mining*, 181–190. IEEE.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Fu, X.; Zhang, J.; Meng, Z.; and King, I. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, 2331–2341.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, K.; and Zitnik, M. 2020. Graph meta learning via local subgraphs. *Advances in neural information processing systems*, 33: 5862–5874.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Liu, J.; Shi, C.; Yang, C.; Lu, Z.; and Philip, S. Y. 2022. A survey on heterogeneous information network based recommender systems: Concepts, methods, applications and resources. *AI Open*, 3: 40–57.

Lu, Y.; Shi, C.; Hu, L.; and Liu, Z. 2019. Relation structure-aware heterogeneous information network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 4456–4463.

Ma, A.; Wang, X.; Li, J.; Wang, C.; Xiao, T.; Liu, Y.; Cheng, H.; Wang, J.; Li, Y.; Chang, Y.; et al. 2023. Single-cell biological network inference using a heterogeneous graph transformer. *Nature Communications*, 14(1): 964.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12: 2825–2830.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, 593–607. Springer.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Tan, Y.; Zhou, Z.; Lv, H.; Liu, W.; and Yang, C. 2024. Walklm: A uniform language model fine-tuning framework for attributed graph embedding. *Advances in Neural Information Processing Systems*, 36.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, M.; Zheng, D.; Ye, Z.; Gan, Q.; Li, M.; Song, X.; Zhou, J.; Ma, C.; Yu, L.; Gai, Y.; et al. 2019a. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Wang, X.; Bo, D.; Shi, C.; Fan, S.; Ye, Y.; and Philip, S. Y. 2022. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 9(2): 415–436.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019b. Heterogeneous graph attention network. In *The world wide web conference*, 2022–2032.

Wen, Z.; and Fang, Y. 2023. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 506–516.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Xie, H.; Zheng, D.; Ma, J.; Zhang, H.; Ioannidis, V. N.; Song, X.; Ping, Q.; Wang, S.; Yang, C.; Xu, Y.; et al. 2023. Graph-aware language model pre-training on a large graph corpus can help multiple graph applications. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5270–5281.

Yan, B.; Cao, Y.; Wang, H.; Yang, W.; Du, J.; and Shi, C. 2024. Federated heterogeneous graph neural network for privacy-preserving recommendation. In *Proceedings of the ACM on Web Conference 2024*, 3919–3929.

Yasunaga, M.; Leskovec, J.; and Liang, P. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 793–803.

Zhang, Q.; Wu, X.; Yang, Q.; Zhang, C.; and Zhang, X. 2022a. Few-shot heterogeneous graph learning via cross-domain knowledge transfer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2450–2460.

Zhang, Q.; Wu, X.; Yang, Q.; Zhang, C.; and Zhang, X. 2022b. HG-Meta: Graph meta-learning over heterogeneous graphs. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, 397–405. SIAM.

Zhang, Z.; Wang, X.; Zhou, H.; Yu, Y.; Zhang, M.; Yang, C.; and Shi, C. 2024a. Can Large Language Models Improve the Adversarial Robustness of Graph Neural Networks? *arXiv preprint arXiv:2408.08685*.

Zhang, Z.; Zhang, M.; Yu, Y.; Yang, C.; Liu, J.; and Shi, C. 2024b. Endowing Pre-trained Graph Models with Provable Fairness. In *Proceedings of the ACM on Web Conference 2024*, 1045–1056.

Zhao, J.; Qu, M.; Li, C.; Yan, H.; Liu, Q.; Li, R.; Xie, X.; and Tang, J. 2022. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019. Meta-gnn: On few-shot node classification in graph meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2357–2360.

Zhu, J.; Cui, Y.; Liu, Y.; Sun, H.; Li, X.; Pelger, M.; Yang, T.; Zhang, L.; Zhang, R.; and Zhao, H. 2021. Textgnn: Improving text encoder via graph neural network in sponsored search. In *Proceedings of the Web Conference 2021*, 2848–2857.

Zhuang, Z.; Xiang, X.; Huang, S.; and Wang, D. 2021. Hinfshot: A challenge dataset for few-shot node classification in heterogeneous information network. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 429–436.