Do Neural Scaling Laws Exist on Graph Self-Supervised Learning

Qian Ma
Rensselaer Polytechnic Institute
maq@rpi.edu

Haitao Mao
Michigan State University
haitaoma@msu.edu

Jingzhe Liu
Michigan State University
liujin33@msu.edu

Zhehua Zhang
Rensselaer Polytechnic Institute
zhangz45@rpi.edu

Chunlin Feng
Rensselaer Polytechnic Institute
fengc5@rpi.edu

Yu Song
Michigan State University
songyu5@msu.edu

Yihan Shao
Rensselaer Polytechnic Institute
shaoy9@rpi.edu

Yao Ma
Rensselaer Polytechnic Institute
may13@rpi.edu

Abstract

Self-supervised learning (SSL) is essential to obtain foundation models in NLP and CV domains via effectively leveraging knowledge in large-scale unlabeled data. The reason for its success is that a suitable SSL design can help the model to follow the neural scaling law, i.e., the performance consistently improves with increasing model and dataset sizes. However, it remains a mystery whether existing SSL in the graph domain can follow the scaling behavior toward building Graph Foundation Models (GFMs) with large-scale pre-training. In this study, we examine whether existing graph SSL techniques can follow the neural scaling behavior with the potential to serve as the essential component for GFMs. Our benchmark includes comprehensive SSL technique implementations with analysis conducted on both the conventional SSL setting and many new settings adopted in other domains. Surprisingly, despite the SSL loss continuously decreasing, no existing graph SSL techniques follow the neural scaling behavior on the downstream performance. The model performance only merely fluctuates on different data scales and model scales. Instead of the scales, the key factors influencing the performance are the choices of model architecture and pretext task design. This paper examines existing SSL techniques for the feasibility of Graph SSL techniques in developing GFMs and opens a new direction for graph SSL design with the new evaluation prototype. Our code implementation is available online to ease reproducibility [1].

1 Introduction

Self-supervised learning (SSL) [1] is to leverage the informative patterns from abundant unlabeled data via pre-training. SSL techniques serve an indispensable role in building Foundation Models in CV and NLP domains with successful applications [2, 3, 4]. A successful SSL design can observe the neural scaling law behavior where the test performance can continuously improve and the test loss continuously decreases with increasing pre-training data size and the model parameter size [5]. Neural scaling laws serve as the key principle for the success of foundation models in CV and NLP domains.

[1] https://github.com/HaitaoMao/GraphSSLScaling

Q. Ma et al., Do Neural Scaling Laws Exist on Graph Self-Supervised Learning. *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, PMLR 269, Virtual Event, November 26–29, 2024.

SSL techniques [6, 7, 8] are also successfully adopted in the graph domain while there is no Graph Foundation Model (GFM) with SSL so far . It remains unclear whether graph SSL techniques follow the scaling law. To this end, we benchmark existing graph SSL techniques to examine whether they can follow the neural scaling behavior with the potential to build GFMs [9, 10]. We focus on the graph classification task instead of transductive node classification and link prediction as the unidentified relationship between train and test nodes. An inductive graph classification setting helps to construct clear-control data scaling settings. We also provide more discussion on the inductive settings in Appendix D. Initial observations demonstrate that the graph SSL loss continuously decreases on the test set with increasing data scale and model scale. However, despite the decreasing SSL loss, the downstream task performance does not observe the scaling behavior correspondingly. Instead of data scale and model scale, key factors that influence the downstream performance are the non-parametric aggregations derived from model architecture design and the SSL objective design. Such observations illustrate that the current graph SSL objective may not be a good choice for training a GFM [9, 10]. Therefore, we introduce a new evaluation perspective on scaling law. The setting helps state a new position on the graph SSL design in the GFM era towards better scaling.

Organizations. The main focus of this work is to explore the scaling law on the existing GraphSSL methods from both data scaling and model scaling. We tend to examine whether the scaling law can be applied to the existing GraphSSL methods so that they can have the potential to serve as a part of the Graph Foundation Model. The following sections will be arranged in the following way: In Section 2, we briefly introduce the neural scaling law and the SSL learning on Graph. In Section 3, we introduce the basis settings of our experiments and the existing GraphSSL methods to investigate. In Section 4, we present the results related to data scaling, and our analysis reveals that there is a gap between the SSL pre-training task and the downstream task resulting in the vanishing of data scaling law. In Section 5, we present the results related to model scaling. Our observations indicate that the model architecture is a key factor influencing the model's performance on SSL tasks instead of the simple number of parameters or model scale.

2 Related Works

Neural Scaling Law The general idea of the neural scaling law is that the model's performance will keep improving with the scaling of training data or model parameters [11]. The quantitive formulation of the Neural Scaling Law is typically described in a power-law form as follows, which is first proposed by Hestness et al [12].

$$\epsilon = \mathbf{a}\mathbf{X}^{-\mathbf{b}} + \epsilon_\infty \tag{1}$$

The variable $\mathbf{X}$ represents the size of the model or the training set. The ϵ is the prediction error of the model. $\mathbf{a}$, $\mathbf{b}$ and $\epsilon_\infty > 0$ are all positive parameters. Under the guidance of neural scaling law, researchers could predict the performance of large models based on small-scale experiments, which greatly saves the costs of redundant runs. Moreover, the scaling laws can be applied to benchmark different models for the backbone of foundation models. Hence, neural scaling law has helped the development of large models in computer vision [12, 13, 14, 15] and natural language processing [11, 13, 16, 17, 18, 19, 20].

Liu et al [21] take an initial step of developing the neural scaling laws in the general graph domain. Specifically, it verifies the general forms of neural scaling laws on graphs. It also discovers some unique phenomena of model scaling and proposes a proper metric for data scaling on graphs. Within specific graph domains *e.g.,* molecular graphs, there are existing works [22] that discovered the scaling of GNNs. These works provide a foundation for our study but are limited to supervised learning, while our focus is self-supervised learning.

Self-Supervised Learning and its applications on Graph. The rise of self-supervised learning in Natural Language Processing (NLP) and Computer Vision (CV) [23, 24, 25] has shifted attention to learning paradigms that do not depend on annotated data. The burgeoning interest in self-supervised learning methodologies presents an invaluable opportunity for graph learning research, particularly in overcoming the reliance on annotated data. A growing body of work has introduced a variety of self-supervised learning strategies for graph data [26, 27, 28, 29, 30, 31, 32, 6], marking a critical evolution in the field. These methods aim to reproduce the success of self-supervised learning in graph learning research. However, whether scaling law exists under these Graph SSL methods are still mysterious.

By leveraging abundant unlabeled data in the real world and expanding model scale, there are a lot of models [2, 3] in CV, and NLP areas that serve as foundation models as they benefit from the scaling law during the pre-training stage. If there exists a GraphSSL method following the neural scaling law, we believe that it has a solid basis to serve as a part of the graph foundation model.

3 Experiment Setups

To ensure the comprehensiveness of our exploration, we implement the existing representative Graph SSL methods on various datasets. We select graph classification as the downstream task for evaluation. Each experiment setting is repeated with five different random seeds are deployed. Here we provide some basic description of the methods and datasets.

Graph SSL Methods. We conducted experiments on the following Graph SSL methods. **(1) InfoGraph:** As a pioneer work of Graph SSL, InfoGraph [8] maximizes mutual information between global graph embeddings and local sub-structure embeddings, leveraging JSD as its contrastive loss. **(2) GraphCL:** GraphCL [7] is a general contrastive learning framework. By maximizing the representations similarity between two different randomly perturbed local sub-graphs of the same node, the encoder can be pre-trained in a SSL manner. **(3) JOAO:** JOAO [33] can automatically and dynamically select augmentations during GraphCL training. **(4) GraphMAE:** GraphMAE [34] is a generative SSL methods, which aims at reconstructing the feature and information of the data. The encoder of GraphMAE is trained by reconstructing the masked data feature with provided context.

Datasets. We used the following Datasets for conducting experiments whose detailed statistics are outlined in Appendix A. **reddit-threads** [35] contains graphs presenting the task to predict whether a thread is discussion-based. **ogbg-molhiv**,**ogbg-molpcba** are curated by ogb [36], all of them are molecular property prediction datasets and the task performance metrics are ROC-AUC and AP correspondingly.

Experiments Settings. We pre-train the encoder with the existing Graph SSL methods using unlabeled data and the details for the specific settings *e.g.,* training hyper-parameters and backbone selections are outlined in Appendix B. We evaluate the GraphSSL methods by applying the pre-trained encoder to downstream task via linear probing following the existing evaluation protocol [37] and more details about the evaluation protocol are outlined in Appendix C.

Data Split. All datasets are split with the ratio 8:1:1 for training, validation, and testing set and only the pre-split training set is used to pre-train the model. To ensure the reproducibility of our experiments, we fix the split for all datasets and all methods will use the same split for experiments.

4 Data scaling

In this section, we conduct experiments to explore the data scaling of Graph SSL methods and our findings. The main observation is that there is no obvious downstream performance gain along with the scaling-up data, indicating no data-scaling effect.

4.1 Data Scaling on Downstream Performance

To explore how the performance of downstream tasks improves with the scale-up pre-training data for Graph SSL methods, we introduce the settings and then present our observations.

Settings. To verify the existence of the data scaling phenomenon of GraphSSL methods, we construct the following pipeline for pre-training and evaluation to verify data scaling.

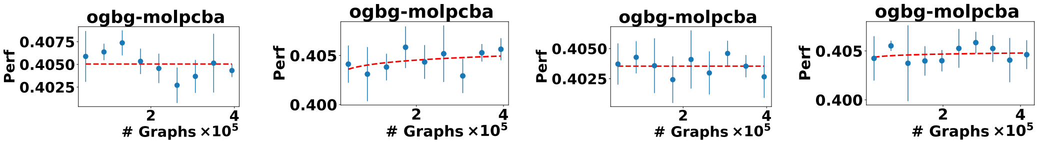
- For a reasonable data scaling setting, we gradually increase the ratio for pre-training data with a fixed interval by containing all data used in the previous ratios.
- For each dataset, we further slice the pre-split training data with the fixed interval with 0.1 as different pre-training data ratio settings. The order of indices is fixed after generation, so we can gradually increase the data ratio for pre-training from 0.1 to 1 by slicing the indices to make sure the data from previous lower ratio can be included.
- For the evaluation on the downstream tasks, we trained an SVM classifier with the pre-trained model fixed following the existing protocols and reported its performance on the held-out test set as the metrics for evaluation.

- To examine whether the Graph SSL methods can consistently exhibit the scaling effect, we fit the equation of scaling law to our empirical results on different data scales with the least square and calculate the coefficient of determination R^2 for examining the quality of the fitting to the scaling.

Observation 1. With gradually scaled-up pre-training data, no obvious scaling effect can be observed from the downstream performance.

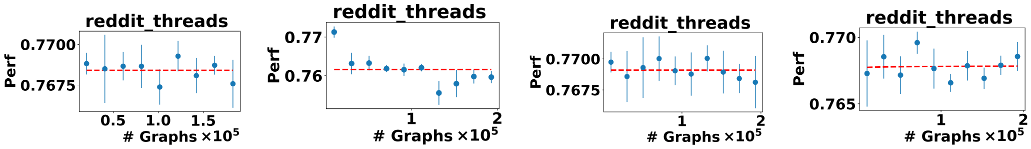
We conduct experiments to gradually increase the data ratio for pre-training to observe whether the performance gain, along with the increasing data, serves as evidence for the data-scaling behavior. Figure 1 and 2 illustrate the performance on ogbg-molpcba and reddit-threads datasets and more results on more datasets can be found in Appendix E. The x-axis indicates the number of graphs for pre-training, and the y-axis indicates the downstream performance, respectively. The data points are used to fit the parameters of the scaling law and R^2 is calculated to examine the overall quality of fitting. Typically, a R^2 value larger than 0.5 can be considered significant.

Our key observation is that the performance of all investigated Graph SSL methods does not exhibit a consistent and obvious scaling behavior despite the consistently increased data ratio for pre-training. There is no fitted curve nor large R^2 value to indicate that the downstream performance can scale up along with more pre-training data amount in Figure 1 and 2.



(a) GraphCL R^2=0.0 (b) GraphMAE R^2=0.0 (c) InfoGraph R^2=0.0 (d) JOAO R^2=0.03

Figure 1: Data Scaling of Performance on ogbg-molpcba with standard deviation. x-axis indicates the data amount used for pre-training and y-axis indicates the downstream performance. No obvious scaling behavior can be observed.



(a) GraphCL $R^2 = 0.0$ (b) GraphMAE $R^2 = 0.0$ (c) InfoGraph $R^2 = 0.0$ (d) JOAO $R^2 = 0.009$

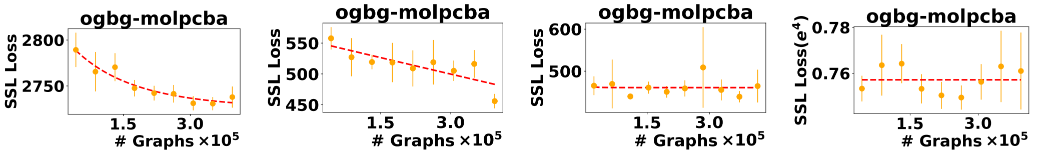
Figure 2: Data Scaling of Performance on reddit-threads with standard deviation. x-axis indicates the data amount used for pre-training and y-axis indicates the downstream performance. No obvious scaling behavior can be observed.

The key conclusion derived from the above results is that, unlike the SSL methods in the NLP [11, 13, 16, 17, 18, 19, 20] and CV [12, 13, 14, 15] domains , *Graph SSL methods do not observe data scaling behavior across graphs.* A further investigation to understand why such a phenomenon happens can be found as follows.

During the pre-training stage, the Graph SSL methods will try to optimize their own SSL task objectives or SSL Losses as we introduced in Section 2. These objectives or losses are improved during the pre-training stage. Therefore, we would like to investigate whether they benefit the downstream loss *i.e.,* whether the knowledge can be generalized to the unseen datasets. If such a reduction can be observed on the downstream loss, then it indicates that the GraphSSL methods fail to scale up on downstream performance due to the huge gap between pre-training and downstream tasks. Therefore, our further investigation aims to examine if the capability obtained along with increasing large-scale pre-training data doesn't *correspond* to the **downstream performance gain**.

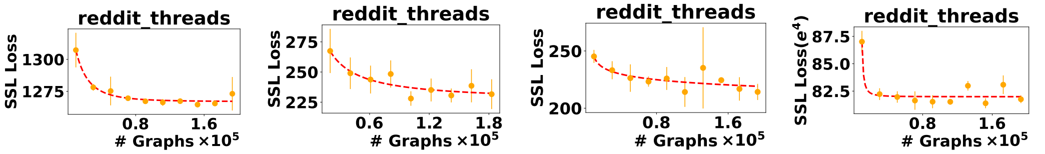
4.2 Data Scaling on SSL loss

Settings. We investigate the above question by changing the metrics we observed from the downstream performance to the same SSL task objectives. More specifically, we utilize the held-out test



(a) InfoGraph $R^2 = 0.93$ (b) GraphCL $R^2 = 0.58$ (c) JOAO $R^2 = 0.01$ (d) GraphMAE $R^2 = 0.0$

Figure 3: Data Scaling of SSL Loss on ogbg-molpcba with standard deviation. x-axis indicates the data amount used for pre-training and y-axis indicates the SSL Loss on the held-out test data. More obvious scaling behavior can be observed on InfoGraph and GraphCL compared to performance.



(a) InfoGraph $R^2 = 0.98$ (b) GraphCL $R^2 = 0.83$ (c) JOAO $R^2 = 0.63$ (d) GraphMAE $R^2 = 0.87$

Figure 4: Data Scaling of SSL Loss on reddit-threads with standard deviation. x-axis indicates the data amount used for pre-training and y-axis indicates the SSL Loss on the held-out test data. More obvious scaling behavior can be observed compared to performance.

set to compute the same SSL loss used in the pre-training stage with the pre-trained model fixed. In this way, we can examine if there is a gain on the same SSL task from improved capability obtained by scale-up pre-training data.

Observation 2. With gradually scaled-up pre-training data, consistent scaling behavior can be observed on the SSL loss.

We conduct data scaling experiments to examine if the gain can be observed on SSL tasks with the scale-up pre-training data. The results presented in Figure 3 and 4 illustrate how the SSL Loss improves on ogbg-molpcba and reddit-threads datasets with scale-up pre-training data. The x-axis is the pre-training data amount and the y-axis is the SSL Loss. The overall fitting quality of the fitted curve obtained with the data points is examined by the R^2 value.

Compared with the observation on the downstream task performance, the scaling behavior on the SSL loss is consistent and obvious. However, the scaling behavior could be method-specific *i.e.,* some methods behave more consistently and stably in a scaling manner while others do not. Taking the InfoGraph as an example, as shown in Figure 3(a) and 4(a) , the SSL loss evaluated on the testing data decreases as pre-training data scales. Meanwhile, for other methods *e.g.,* GraphCL the scaling effect is less obvious and consistent as shown in Figure 3(b) and 4(b).

According to the above results, we observe that the scale-up pre-training data can improve the capability of SSL tasks in a data-scaling manner. Notably, we do not observe the scaling behavior on the downstream performance in Section 4.1. Therefore, it could be the gap between the pre-training and downstream tasks that block the GraphSSL methods from following the scaling law on the downstream performance.

5 Model Scaling

In this section, we conduct experiments to explore the model scaling of Graph SSL methods. Specifically, we aim to observe how the performance improves as we increase the number of model parameters. Our key observation is that no consistent scaling behaviour can be observed with model scaling on performance.

Settings. To observe the model scaling effects on the Graph SSL methods, we scale up the model parameters by increasing **(1) size of hidden dimensions** and **(2) number of layers** for the encoders applied in the Graph SSL methods. Following the settings in the Section 4, we monitor both the graph classification downstream performance and the value of SSL objectives on the held-out test set.

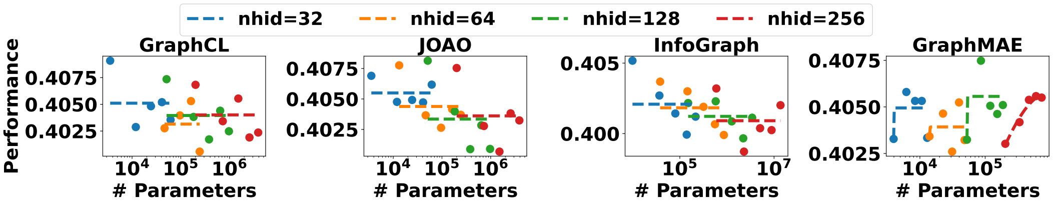


Figure 5: Performance on ogbg-molpcba. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed on all methods.The R^2 values for each method are listed as follows.GraphCL:0.0,JOAO:0.0,InfoGraph:0.0,GraphMAE:0.46. The standard deviation is shown in Figure 22.

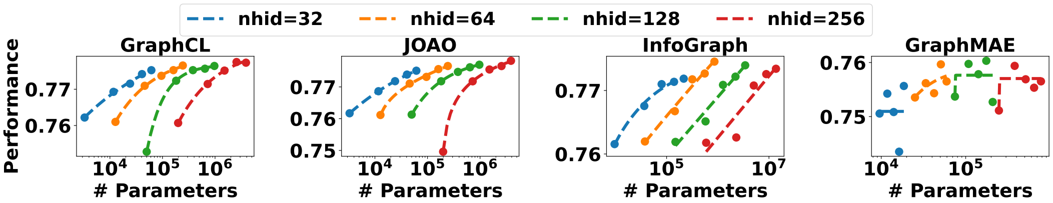


Figure 6: Performance on reddit-threads. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. Obvious scaling behavior can be observed on all methods except GraphMAE. However, the range is quite narrow .The R^2 values for each method are listed as follows, GraphCL:0.99,JOAO:0.99,InfoGraph:0.95,GraphMAE:0.36.The standard deviation is shown in Figure 23.

5.1 Model Scaling on Downstream Performance

Observation 3. Under the two different manners of model scaling settings, there is no obvious scaling effect can be observed from the downstream performance.

We present the results showing how the downstream performance of investigated methods improve with scale-up model size on two datasets, ogbg-molpcba in Figure 5, 6, and reddit-threads in Figure 7, 8, where the number of parameters is indicated by the x-axis and the downstream performance is indicated by the y-axis, and different colors indicate different settings of hidden size or number of layers. Due to the space limit, the results on more datasets can be found in Appendix F.

By increasing the number of layers with the hidden size fixed, no consistent scaling behavior can be observed as shown in Figure 5 and 6. Even for GraphCL, JOAO, and InfoGraph, there seems to be an obvious scaling effect exhibited only on Reddit Threads datasets as shown in Figure 6, however, this could be limited to the scale of plotting as the differences between the downstream performance metrics are very marginal and far away from being called 'scaling', especially compared with the number of parameters increasing in a exponential way. Similarly, by increasing the number of hidden size with the number of layers fixed, no scaling effect can be consistently presented for all methods across all datasets, as shown in Figure 7 and 8.

Therefore, our key observation is that there is no consistent scaling behavior in the downstream performance of GraphSSL methods with either scale-up hidden dim or number of layers, unlike the scaling effect that universally exists in CV and NLP domains by increasing the total number of parameters of the model. Consequently, we conducted further investigation on SSL loss to examine if the capability of scale-up model parameters benefits the SSL tasks without corresponding to the downstream performance gain.

5.2 Model Scaling on SSL loss

To examine if the scaled-up model parameters can improve the capability of SSL task to reveal scaling law, we target the SSL loss on the downstream data as a metric. To better examine the GraphCL and JOAO, we select proper settings for data augmentation for contrastive learning as it is indicated as the key component in their original papers [7, 33]. The details are deferred to the Appendix G.

Observation 4. Under the model scaling setting with increasing numbers of layers, the scaling effect on the SSL loss can be observed on particular datasets and SSL methods.

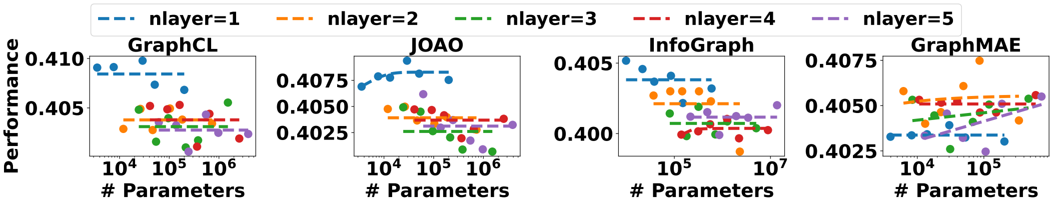


Figure 7: Performance on ogbg-molpcba grouped by layers. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed.The R^2 values for each method are listed as follows,GraphCL:0.10,JOAO:0.09,InfoGraph:0.02,GraphMAE:0.18.The standard deviation is shown in Figure 24.

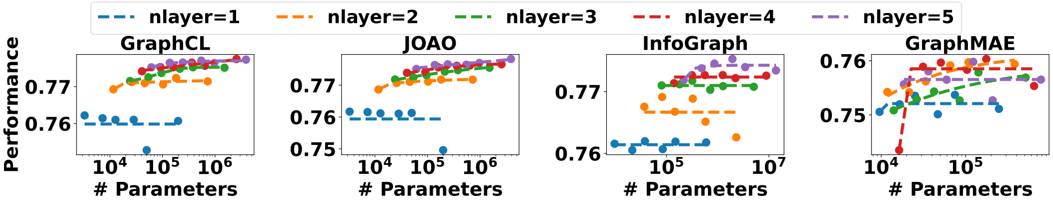


Figure 8: Performance on reddit-threads grouped by layers. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. The R^2 values for each method are listed as follows.GraphCL:0.56,JOAO:0.79,InfoGraph:0.49,GraphMAE:0.64. The standard deviation is shown in Figure 25.

We present the results of all methods on the same dataset in Figure 9 and 10, where the x-axis denotes the total number of parameters of the model and the y-axis denotes the metrics of SSL loss respectively. Different colors represent different hidden size settings.

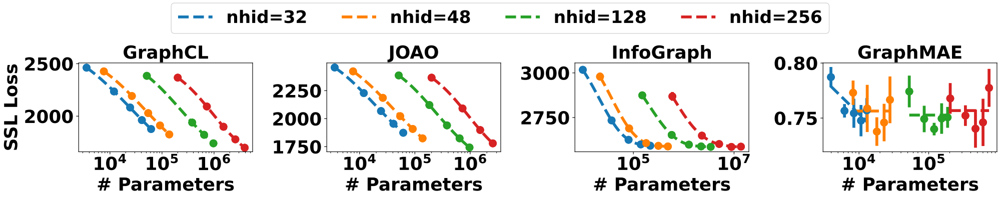


Figure 9: SSL Loss on ogbg-molpcba with standard deviation. Obvious scaling behaviour can be observed except GraphMAE. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.99, JOAO:0.99, InfoGraph:0.99, GraphMAE:0.17.

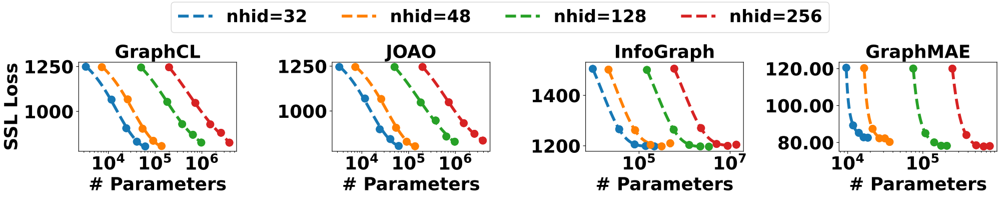


Figure 10: SSL Loss on reddit-threads with standard deviation. Obvious scaling behaviour can be observed on all methods. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.99, JOAO:0.99, InfoGraph:0.99, GraphMAE:0.99.

Our key conclusion from the above results is that there is method-specific model scaling behavior can be observed with the scale-up number of layers and fixed hidden size. For **InfoGraph**, as shown in Figure 9, it can exhibit more obvious scaling behavior compared to the trend observed with the performance metrics. Moreover, the scaling behavior is consistently obvious on all other datasets as shown in Figure 9 and 10. Compared with the representative contrastive SSL method InfoGraph, **GraphMAE** is a generative method, the scaling effect is not consistent nor obvious on its SSL loss *i.e.,* its feature reconstruction cosine loss. As shown in Figure 9 , the scaling effect indicated by the R^2 value is not as obvious and consistent as that presented in Figure 10. These differences indicate that the SSL task design of GraphMAE can not consistently benefit from the scaling up of model parameters. As contrastive SSL methods, **GraphCL** and **JOAO** can exhibit similar scaling behavior as InfoGraph on all datasets as shown in Figure 9 and 10.

Observation 5. Under the model scaling settings with increasing hidden size, there is no obvious scaling effect can be observed from the SSL loss across datasets.

We also grouped the results by the same number of layers. The results are presented in Figure 11 and 12, where different colors indicate different number of layers settings. The fitted curve and R^2 value to examine the fitting quality, all indicate that no consistent and obvious scaling behavior can be observed.

Our key conclusion from the above results is that there is no consistent or obvious scaling behavior can be observed with the scale-up hidden size while fixing the number of layers. Taking InfoGraph as an example, the SSL Loss results grouped by the same number of layers shown in Figure 12 indicates that there is no improvement on the objectives by increasing the hidden size with the number of layers fixed. Moreover, compared to the scaling behavior exhibited by InfoGraph with scale-up number of layers in the model shown in Figure 10, the different behaviors with different model scaling manners indicate that the improvement is more likely to be the result of more aggregations by stacking more layers instead of the capability of transformation with more learnable model parameters. Consequently, we conducted a further investigation on InfoGraph for this phenomenon to examine whether the aggregation benefits the capability of InfoGraph on SSL objectives rather than the transformation with more learnable parameters.

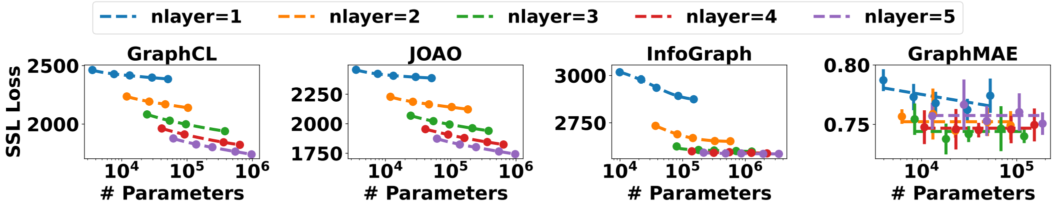


Figure 11: SSL Loss on ogbg-molpcba grouped by layer. No obvious scaling behavior can be consistently observed. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R2 values for each method are listed as follows. GraphCL:0.99, JOAO:0.99,InfoGraph:0.95, GraphMAE:0.38.

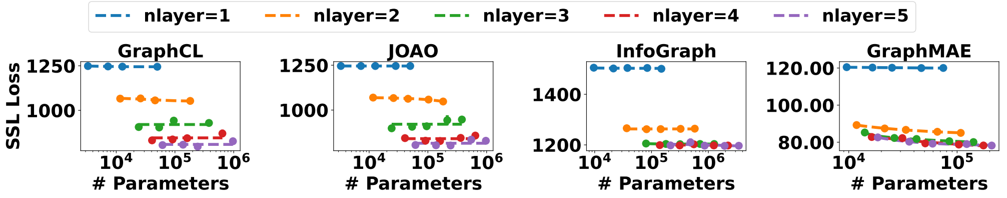


Figure 12: SSL Loss on reddit-threads grouped by layer with standard deviation. No obvious scaling behavior can be observed. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.36, JOAO:0.18, InfoGraph:0.00, GraphMAE:0.99. Kindly note that R^2 is the metric to evaluate how well the fitted curve is instead of a direct metric to indicate the existence of the scaling effect.

Observation 6. By fixing the hidden size for transformation and increasing the number of aggregations, an obvious and consistent scaling behavior can be observed on SSL Loss for the new implementation with transformation and aggregation decoupled.

To decouple the aggregation and transformation of the model, we modify the implementation of the original InfoGraph accordingly. More specifically, we replaced the layers except for the last layers from the GINConv layer to Message Passing layers. For the original implementation, all layers

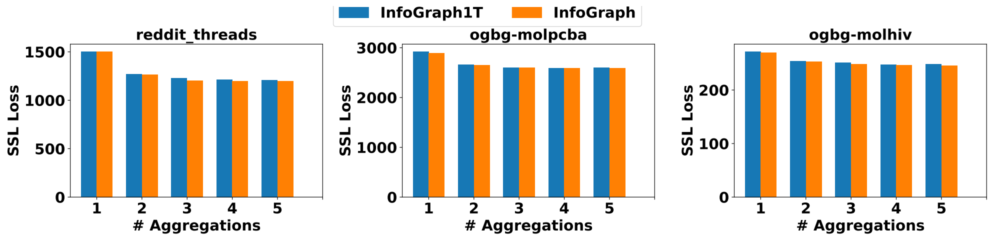


Figure 13: SSL loss Comparison on three datasets with nhid=96. x-axis denotes the number of aggregations and y-axis denotes SSL Loss on the held-out test set. Decoupled new implementation of InfoGraph with only one transformation(InfoGraph1T) can also exhibit consistent scaling behavior with marginal difference to original results

are the same, *i.e.,* for each layer, the GINConv layer will aggregate the features and then transform them. Our new implementation only remains the one transformation layer, which is the same as the original version for feature transformation while the other layers are only message passing layers for aggregation. In this way, the aggregation and transformation are decoupled. The embedding obtained with the trained encoder will be fed into the projection head to map the embeddings into another latent space for calculating the SSL Loss. Our new implementation with aggregation and transformation decoupled is denoted as **InfoGraph1T**. We fit the the empirical results to the curve of scaling law and compare the original(InfoGraph) and new implementation(InfoGraph1T) pairwise. Figure 13 illustrates how the SSL Loss improves with more aggregations compared to the original InfoGraph, where the x-axis and y-axis indicate the number of aggregation and the SSL Loss metrics respectively. Our key observation is that the newly implemented **InfoGraph1T**, with decoupled aggregation and transformation, exhibits scaling behavior with scale-up number of aggregation, which is similar to the scaling behavior with scale-up number of layers exhibited in Figure 9 and 10 . The fitted curves of the original implementation with more transformation layers and decoupled implementation are almost overlapped. Therefore, we can draw a conclusion that the improvement in the SSL Loss is primarily due to the model architecture or structure with more aggregations, rather than the capability with more learnable parameters for transformation.

6 Conclusion

In this work, we take the first step to explore the neural scaling laws on the existing Graph SSL methods. Specifically, we try to verify the existence of two basic forms of neural scaling laws: the model scaling law and the data scaling law. Our attempts obtain some key observations and provide some insights for future work. **Obs** 1 and 3 indicate that no scaling behavior can be observed in the downstream performance. Meanwhile, **Obs** 2 and 4 indicate that scaling behavior can only be observed in the SSL loss with the increasing number of layers of the encoder in GraphSSL methods. The above observations can draw a conclusion that the gain in the downstream performance does not correspond to SSL loss. These results indicate that there is a huge gap between the existing SSL and downstream tasks in Graph domain. Therefore, for further GFM design, we believe that a proper SSL task design is critical to mitigate this gap to exhibit scaling behavior on the downstream tasks. **Obs** 5 and 6 indicate that the scaling behavior we observed is mainly caused from the characteristics of the model architecture *i.e.,* more aggregations instead of the improved capability with more learnable parameters. These observations provide insights to future work like verifying the existence of neural scaling law on more powerful backbones *e.g.,* Graph Transformer for GraphSSL methods. Moreover, the scaling behavior exhibited in SSL loss for contrastive methods is more consistent than generative methods. These results suggest that SSL task design and the component design of GraphSSL methods should be considered as the key factors to reveal the potential of scaling law. Therefore, for further GFM design, we believe that a powerful and representative backbone is critical to be able to scale up to accommodate continuously increasing pre-training data. Our findings shed light on the absence of the scaling behaviors of existing GraphSSL methods and point to critical components that should be considered in future design.

7 Acknowledgement

This research is supported by the National Science Foundation (NSF) under grant numbers NSF-2406647 and NSF-2406648.

References

[1] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 3

[4] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[5] Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312, 2022. 1

[6] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022. 2

[7] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020. 2, 3, 6, 13

[8] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019. 2, 3, 13

[9] Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Michael Galkin, and Jiliang Tang. Graph foundation models. *arXiv preprint arXiv:2402.02216*, 2024. 2

[10] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations*, 2023. 2

[11] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2, 4

[12] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. 2, 4

[13] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. 2, 4

[14] Utkarsh Sharma and Jared Kaplan. Scaling laws from the data manifold dimension. *The Journal of Machine Learning Research*, 23(1):343–376, 2022. 2, 4

[15] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022. 2, 4

[16] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2, 4

[17] Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, 2021. 2, 4

[18] Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. *arXiv preprint arXiv:2109.07740*, 2021. 2, 4

[19] Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. Data scaling laws in NMT: The effect of noise and architecture. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1466–1482. PMLR, 17–23 Jul 2022. 2, 4

[20] Patrick Fernandes, Behrooz Ghorbani, Xavier Garcia, Markus Freitag, and Orhan Firat. Scaling laws for multilingual neural machine translation. *arXiv preprint arXiv:2302.09650*, 2023. 2, 4

[21] Jingzhe Liu, Haitao Mao, Zhikai Chen, Tong Zhao, Neil Shah, and Jiliang Tang. Neural scaling laws on graphs. *arXiv preprint arXiv:2402.02054*, 2024. 2, 14

[22] Maciej Sypetkowski, Frederik Wenkel, Farimah Poursafaei, Nia Dickson, Karush Suri, Philip Fradkin, and Dominique Beaini. On the scalability of gnns for molecular graphs. *arXiv preprint arXiv:2404.11568*, 2024. 2

[23] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021. 2

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[25] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[26] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018. 2

[27] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020. 2

[28] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080, 2021. 2

[29] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. *Advances in Neural Information Processing Systems*, 34:76–89, 2021. 2

[30] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021. 2

[31] Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Simple unsupervised graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7797–7805, 2022. 2

[32] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. Spectral feature augmentation for graph contrastive learning and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11289–11297, 2023. 2

[33] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021. 3, 6

[34] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022. 3

[35] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020. 3, 13

[36] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020. 3, 13

[37] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116*, 2021. 3, 13

[38] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018. 13

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 13

[40] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pages 4116–4126. PMLR, 2020. 13

[41] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017. 22, 23

A Dataset

All the datasets can be obtained from TU Dataset [35] and OGB Dataset [36]. Here we provide their statistics in Table 1.

Table 1: Datasets statistics.

Name	# Graphs	# Avg. nodes	# Avg. edges	Metric
ogbg-molhiv	41,127	25.5	27.5	ROC-AUC
ogbg-molpcba	437,929	26.0	28.1	AP
ogbg-ppa	158,100	243.4	2,266.1	Accuracy
Reddit-Threads	203,088	23.93	24.99	Accuracy
ZINC-Full	249,456	23.15	24.90	RMSE
PCQM4MV2	3,746,619	14.11	14.52	MAE

B Experiment Specific Settings

All hyper-parameter configuration files for the methods used in this study are provided in the code repository. Below, we outline the settings for some general hyper-parameters:

- **Backbone Selection**: For all methods, the backbone architecture is GIN [38].
- **Hidden Size and Number of Layers**: For all methods involved in data scaling experiments, the hidden size is set to 32, and the number of layers is set to 2.
- **Learning Rate and Optimizer**: Across all experiments, the initial learning rate is set to 0.001, and the Adam optimizer [39] is employed for training.
- **Graph Classification Task**: For downstream evaluation, an SVM classifier is trained, with the C parameter selected via grid search over the range [0.001, 0.01, 0.1, 1, 10, 100, 1000].
- **Graph Regression Task**: For downstream evaluation, a two-layer MLP is trained with a hidden size matching that of the pre-trained encoder.

C More Details about Evaluation Protocols

In this work, we only use the split training set for pre-training to ensure the testing set used for evaluation will not be leaked in the pre-training stage. For the evaluation of SSL methods, we follow the existing setting in previous works [7, 8, 37, 40] by fixing the pre-trained encoder and use the embeddings obtained by this specific fixed pre-trained encoder to conduct a downstream task, such as training a classifier for graph classification. The pre-training data and downstream data are from the identical dataset. Then the metrics on the downstream task will be reported to serve as the performance of the pre-trained encoder to reflect the feasibility of the SSL method correspondingly. Specifically, for the downstream task settings, we only used the pre-first 10% of the split training data with labels. For each experiment settings, five different random seeds are deployed. We stored the model pre-trained after 100 epochs to further evaluate them with the downstream tasks.

D More dicussions on data scaling settings

Here we would like to provide more illustration on why we focused on the inductive graph classification to construct clear-control data scaling settings.

For the node classification, under the transductive settings, the whole graph structure will be used during both training and testing stage. During the training stage, the test nodes will be masked, *i.e.,* without label information. Moreover, the edges and nodes are all contributing to the message passing process. Then if we want to investigate the data scaling on the node classification dataset, we need to gradually enlarge the graph, i.e., attach the edges and nodes correspondingly. In reverse, we will need to remove the edges and nodes from the original dataset to construct the subsets. However, there is no principle for conducting such a process, then a natural idea is to random remove the nodes and their connected edges. An extreme situation is that we removed all 1-hop neighbors of the testing nodes during the training stage to make the testing nodes isolated. In this case, the model's ability to classify these isolated test nodes would be diminished, making it difficult to evaluate meaningful scaling behavior in node classification. It could also skew our conclusions, as the extreme isolation might lead to more generalization issues than those related purely to data scaling.

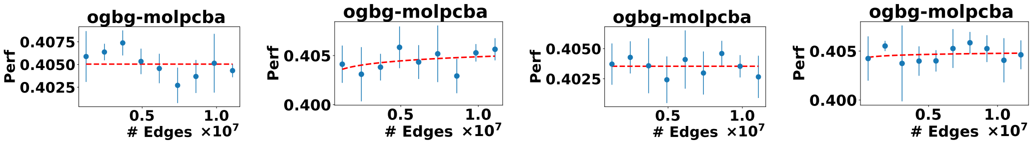
In link prediction, constructing subsets to explore data scaling may present new challenges. During the testing stage, edges used for evaluation are removed, as the model's goal is to predict their existence. Successful link prediction depends on generating negative samples i.e. node pairs without links, to train the model in distinguishing true links from non-existent ones. However, as we reduce nodes and edges to examine scaling, some negative samples might reflect non-links absent in the original graph, introducing noise. This distribution shift in negative samples can impact test performance.

Therefore, we focus on the graph classification task as the inductive graph classification setting helps to construct clear-control data scaling settings.

E More Results on Data Scaling

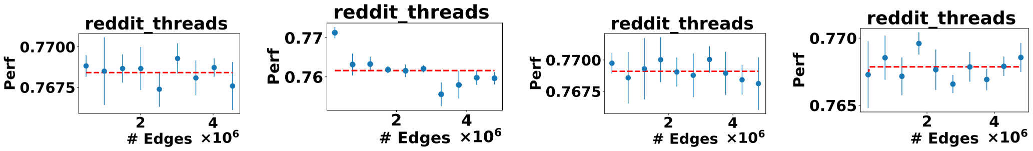
E.1 Data Scaling on Downstream Performance Using different metrics

As illustrated in [21], a proper metric may better reflect the scaling behavior. Therefore, we provide the following Figures showing the results on ogbg-molpcba and reddit-threads datasets with the number of edges or nodes as the metrics. As the average numbers of nodes and edges are stable across different pre-training sets with different ratios, the overall conclusions from Figure 1, 2, 3, 4 remain the same, which is that there is no scaling behavior can be observed on the downstream performance.



(a) GraphCL R^2=0.0 (b) GraphMAE R^2=0.16 (c) InfoGraph R^2=0.0 (d) JOAO R^2=0.03

Figure 14: Data Scaling of Performance on ogbg-molpcba with standard deviation. x-axis indicates the number of edges used for pre-training and y-axis indicates the downstream performance. No obvious scaling behavior can be observed.



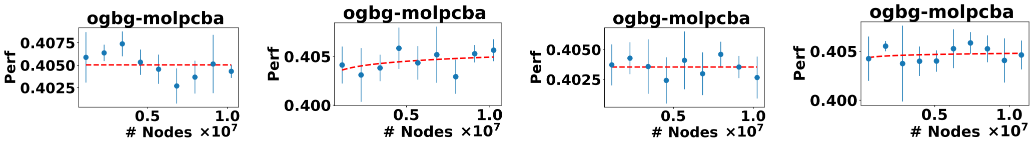
(a) GraphCL $R^2 = 0.0$ (b) GraphMAE $R^2 = 0.0$ (c) InfoGraph $R^2 = 0.0$ (d) JOAO $R^2 = 0.009$

Figure 15: Data Scaling of Performance on reddit-threads with standard deviation. x-axis indicates the number of edges used for pre-training and y-axis indicates the downstream performance. No obvious scaling behavior can be observed.

E.2 Data Scaling on Downstream Performance

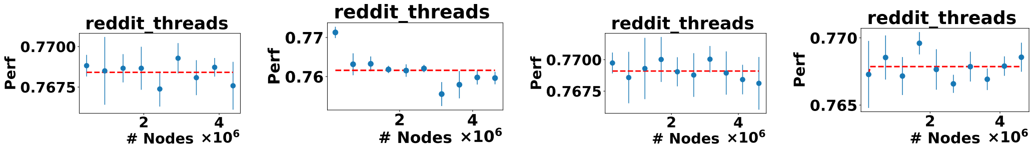
In this section, we are providing additional results on the deferred ogbg-molhiv for graph classification task, and ZINC-Full dataset for graph regression task.

Our observations from these results remain the same as that we illustrated in the main content that there is no scaling behavior can be observed on downstream performance with the gradually scaled-up pretraining data.



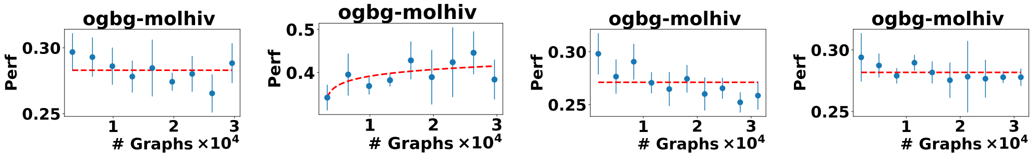
(a) GraphCL R^2=0.0 (b) GraphMAE R^2=0.16 (c) InfoGraph R^2=0.0 (d) JOAO R^2=0.03

Figure 16: Data Scaling of Performance on ogbg-molpcba with standard deviation. x-axis indicates the number of nodes used for pre-training and y-axis indicates the downstream performance. No obvious scaling behavior can be observed.



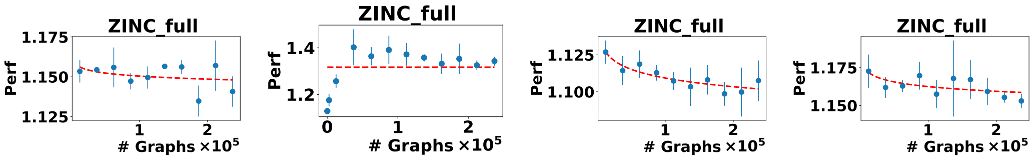
(a) GraphCL $R^2 = 0.0$ (b) GraphMAE $R^2 = 0.0$ (c) InfoGraph $R^2 = 0.0$ (d) JOAO $R^2 = 0.009$

Figure 17: Data Scaling of Performance on reddit-threads with standard deviation. x-axis indicates the number of nodes used for pre-training and y-axis indicates the downstream performance. No obvious scaling behavior can be observed.



(a) GraphCL $R^2 = 0.0$ (b) GraphMAE $R^2 = 0.31$ (c) InfoGraph $R^2 = 0.0$ (d) JOAO $R^2 = 0.0$

Figure 18: Data Scaling of Performance on ogbg-molhiv with standard deviation. x-axis indicates the number of edges used for pre-training and y-axis indicates the downstream performance. No obvious scaling behavior can be observed.



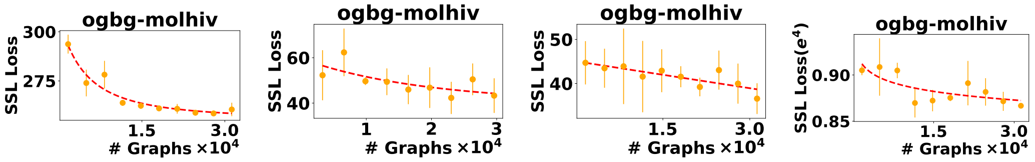
(a) GraphCL R^2=0.11 (b) GraphMAE R^2=0.0 (c) InfoGraph R^2=0.80 (d) JOAO R^2=0.39

Figure 19: Data Scaling of Performance(RMSE) on ZINC-Full with standard deviation. x-axis indicates the data amount used for pre-training and y-axis indicates the downstream performance. The scaling behavior can be observed but the range is quite narrow.

E.3 Data Scaling on SSL Loss

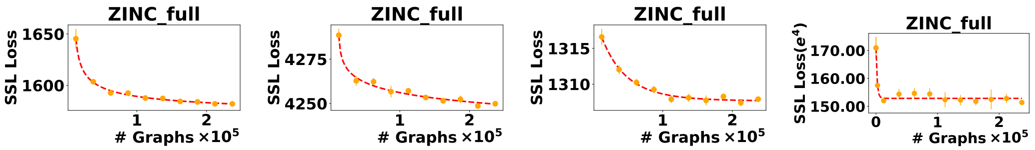
In this section, we are providing additional results of Data Scaling on SSL Loss on the deferred ogbg-molhiv dataset for graph classification and ZINC-Full dataset for graph regression task.

Our observations from these results remain the same as that we illustrated in the main content that scaling behavior can be observed on downstream SSL loss with the gradually scaled-up pre-training data.



(a) InfoGraph R^2=0.92 (b) GraphCL R^2=0.51 (c) JOAO R^2=0.66 (d) GraphMAE R^2=0.57

Figure 20: Data Scaling of SSL Loss on ogbg-molhiv with standard deviation. x-axis indicates the data amount used for pre-training and y-axis indicates the SSL Loss on the held-out test data. More obvious scaling behavior can be observed compared to performance.



(a) InfoGraph $R^2 = 0.99$ (b) GraphCL $R^2 = 0.98$ (c) JOAO $R^2 = 0.98$ (d) GraphMAE $R^2 = 0.97$

Figure 21: Data Scaling of SSL Loss on ZINC-Full with standard deviation. x-axis indicates the data amount used for pre-training and y-axis indicates the SSL Loss on the held-out test data. More obvious scaling behavior can be observed compared to performance.

F More Results on Model Scaling

F.1 Model scaling results on downstream performance with standard derivation

Here we present the plots of results on the ogbg-molpcba and reddit-threads, shown in Figure 22, 23, 24, 25 with standard deviation of the results, corresponding to Figure 5, 6, 7, 8, resepctively. With the standard deviation taken into account, our conclusion that under the two different manners of model scaling settings, there is no obvious scaling effect can be observed from the downstream performance still stands.

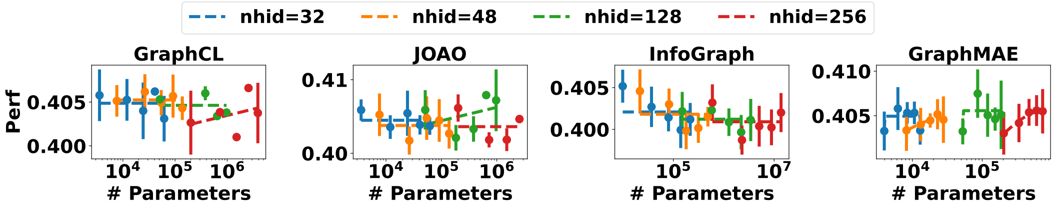


Figure 22: Performance on ogbg-molpcba with standard deviation.. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed on all methods.The R^2 values for each method are listed as follows.GraphCL:0.0,JOAO:0.0,InfoGraph:0.0,GraphMAE:0.46

F.2 Model scaling on Downstream performance

In this section, we are providing additional results on the ogbg-molhiv (Figure 26, 27), ogbg-ppa (Figure 28, 29) and ZINC-Full (Figure 30, 31) datasets respectively. ZINC-Full is evaluated by graph regression tasks while the others are evaluated by graph classification task.

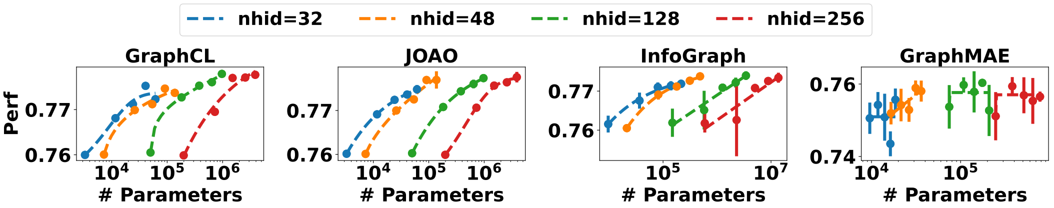


Figure 23: Performance on reddit-threads with standard deviation. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. Obvious scaling behavior can be observed on all methods except GraphMAE. The R^2 values for each method are listed as follows, GraphCL:0.99,JOAO:0.99,InfoGraph:0.95,GraphMAE:0.36.

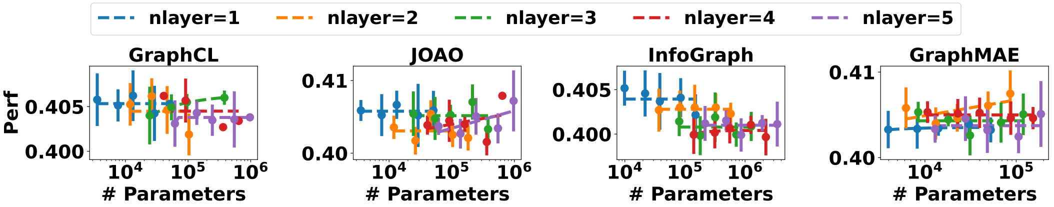


Figure 24: Performance on ogbg-molpcba grouped by layer with standard deviation. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed. The R^2 values for each method are listed as follows.GraphCL:0.0,JOAO:0.0,InfoGraph:0.11,GraphMAE:0.18.

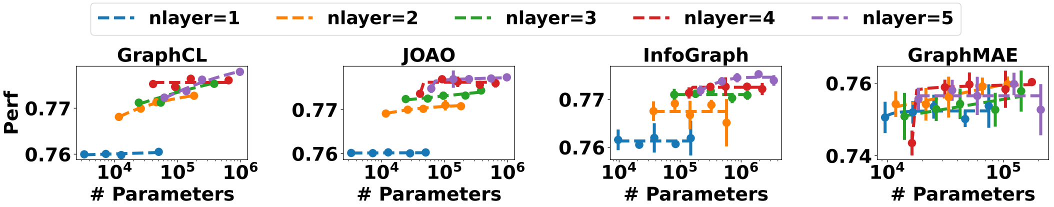


Figure 25: Performance on reddit-threads grouped by layers with standard deviation. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. The R^2 values for each method are listed as follows.GraphCL:0.56,JOAO:0.79,InfoGraph:0.49,GraphMAE:0.64.

Our observations from these results remain the same as that we illustrated in the main content that there is no obvious scaling behavior can be observed under model scaling via increasing number of layers or the hidden size of the model.

F.3 Model scaling on SSL loss

In this section, we are providing additional results of the Model scaling on SSL loss on the ogbg-molhiv (Figure 32, 33), ogbg-ppa (Figure 34, 35) and ZINC-Full (Figure 36, 37) datasets respectively.

Our observations from these results remain the same as that we illustrated in the main content that there is obvious scaling behavior on downstream SSL loss that can only be observed under model scaling via increasing number of layers of the model.

G Deferred details about GraphCL/JOAO SSL method

Compared with InfoGraph, there are two major differences between GraphCL/JOAO and InfoGraph. (1) The SSL Loss (2) The strategy for constructing the augmented view for contrastive learning.

We first investigate the influence of loss. InfoGraph is using JSD Loss while GraphCL and JOAO are using InfoNCE loss. As this is the most obvious difference between the methods, we switch the SSL Loss, which can be considered as switching a single component between two different frameworks.

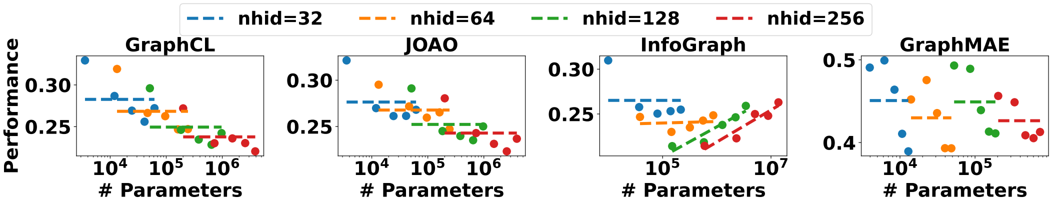


Figure 26: Performance on ogbg-molhiv. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. The R^2 values for each method are listed as follows. GraphCL:0.0,JOAO:0.0,InfoGraph:0.42,GraphMAE:0.0.

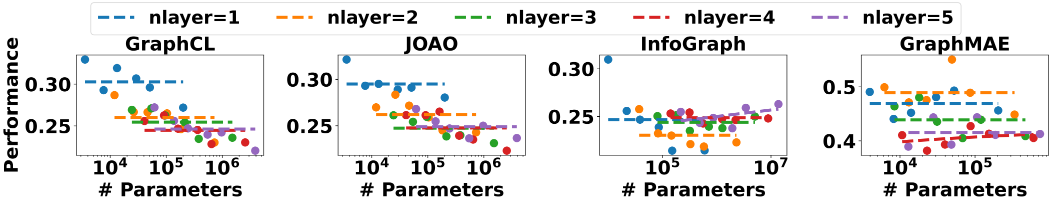


Figure 27: Performance on ogbg-molhiv grouped by layer. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed. The R^2 values for each method are listed as follows.GraphCL:0.0,JOAO:0.0,InfoGraph:0.11,GraphMAE:0.18.

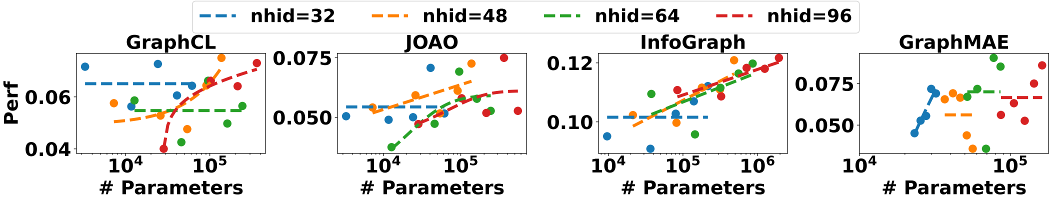


Figure 28: Performance on ogbg-ppa. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed on all methods.The R^2 values for each method are listed as follows.GraphCL:0.39,JOAO:0.33,InfoGraph:0.43,GraphMAE:0.22.

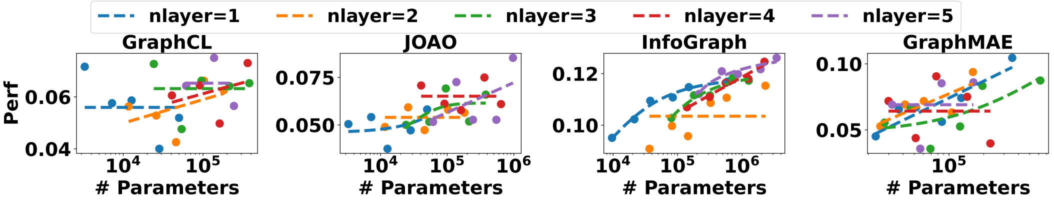


Figure 29: Performance on ogbg-ppa grouped by layers. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed.The R^2 values for each method are listed as follows,GraphCL:0.06,JOAO:0.14,InfoGraph:0.75,GraphMAE:0.38

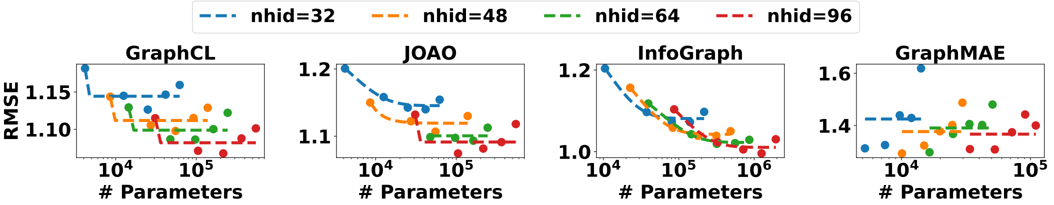


Figure 30: Performance on ZINC-Full. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. Scaling behavior can be observed but the range is narrow. The R^2 values for each method are listed as follows.GraphCL:0.0,JOAO:0.55,InfoGraph:0.97,GraphMAE:0.46

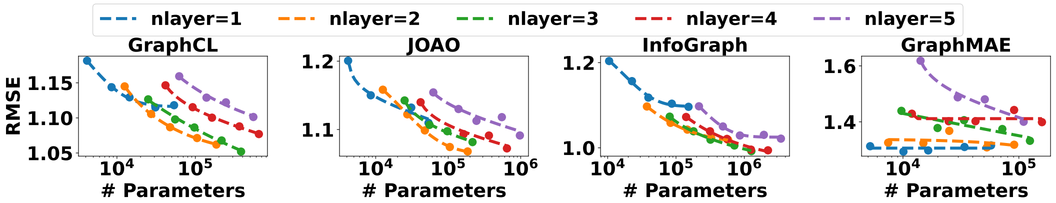


Figure 31: Performance on ZINC-Full grouped by layers. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. Scaling behavior can be observed but the range is narrow. The R^2 values for each method are listed as follows,GraphCL:0.99,JOAO:0.96,InfoGraph:0.99,GraphMAE:0.41. Kindly note that R^2 does not indicate the existence of scaling effect instead of the fitting quality.

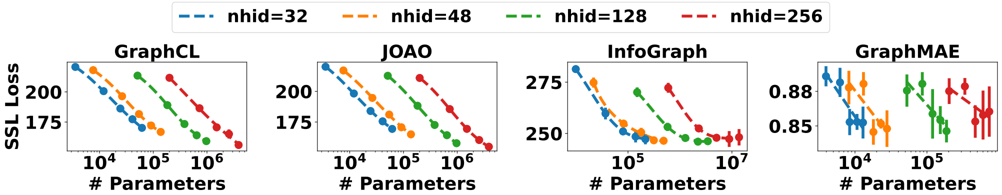


Figure 32: SSL Loss on ogbg-molhiv with standard deviation. Obvious scaling behaviour can be observed on all methods. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.99, JOAO:0.99, InfoGraph:0.99, GraphMAE:0.73.

The initial observation indicates that the instability of GraphCL and JOAO remained the same for the revised version while the revised InfoGraph was still stable. Consequently, our key conclusion from the above results is that the SSL Loss is not the factor that affects the stability.

We further investigated the difference in generating the augmented views. By fixing the randomness in utilizing augmenters to generate augmented views for contrastive learning. The rest of the settings are the same as the model scaling settings with fixed hidden size. After fixing the randomness in augmented view generation and selecting a proper contrastive strategy, GraphCL and JOAO obtain more stable results, where more obvious scaling behavior can be exhibited. Meanwhile, their downstream performances are still almost overlapped with the ones with randomly selected data augmentations. These results also support our conclusions that the gap between SSL and downstream tasks blocks the SSL methods from improving on downstream performance corresponding to SSL loss and the component design is critical for exhibiting scaling behavior for the future Graph Foundation Model design.

H The Investigation Results on PCQM4Mv2

In this section, we present the results of InfoGraph and GraphMAE on PCQM4Mv2 datasets.

H.1 Data Scaling Results

We follow the same data scaling settings outlined in Section 4 to ensure consistency in our results. It is important to note that the PCQM4Mv2 task is a regression problem, with mean absolute error (MAE) as the evaluation metric.

Therefore, the observed increase in MAE with larger amounts of pre-training data in Figure 38(a) does not indicate improved downstream performance for InfoGraph. In fact, this trend suggests that as more pre-training data is used, the downstream performance of InfoGraph actually declines. We hypothesize that the enhanced capabilities from increased pre-training data primarily benefit the SSL task but may harm downstream task performance due to the gap between Graph SSL and downstream tasks. The gain in the SSL Loss corresponds to the increasing pre-training data, as shown in Figure 39(a) can support our hypothesis.

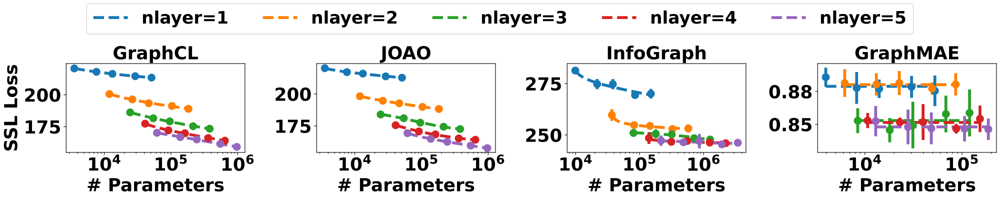


Figure 33: SSL Loss on ogbg-molhiv grouped by layer with standard deviation. No obvious scaling behavior can be observed. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.99, JOAO:0.99, InfoGraph:0.70, GraphMAE:0.26.

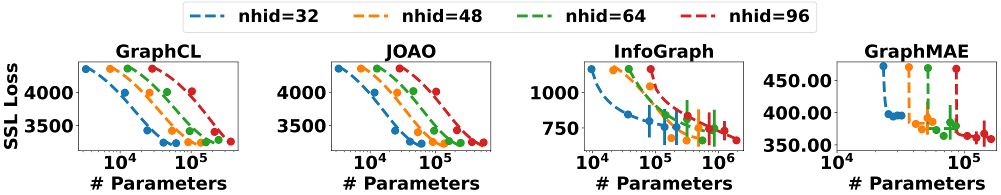


Figure 34: SSL Loss on ogbg-ppa with standard deviation. Obvious scaling behaviour can be observed. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.97, JOAO:0.97, InfoGraph:0.95, GraphMAE:0.98.

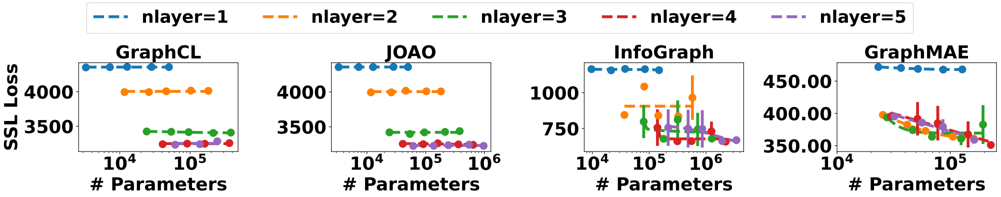


Figure 35: SSL Loss on ogbg-ppa grouped by layer with standard deviation. No obvious scaling behaviour can be observed. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.18, JOAO:0.11, InfoGraph:0.34, GraphMAE:0.92

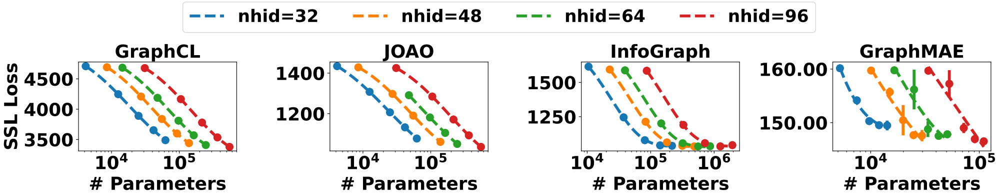


Figure 36: SSL Loss on ZINC-Full with standard deviation.Obvious scaling behaviour can be observed on all methods. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.99, JOAO:0.99, InfoGraph:0.99, GraphMAE:0.95.

For our evaluation on GraphMAE, there is no scaling behavior on the downstream performance can be observed as shown in Figure 38(b). However, the Graph SSL loss of GraphMAE, specifically the feature reconstruction loss, does not exhibit any scaling behavior neither as shown in Figure 39(b). We hypothesis this phenomenon is caused by the inherently low difficulty of the feature reconstruction task within PCQM4Mv2 dataset. The node features in PCQM4Mv2 represent atom features, which are relatively stable and straightforward to reconstruct due to the inherent nature of features in real world. This contrasts with datasets like reddit-threads, where node features are based on degree information, leading to more variability and instability in the features. The simplicity of reconstructing atom features in molecular graphs likely contributes to the absence of scaling behavior in the SSL loss. Even when trained with minimal data in Figure 39(b), GraphMAE demonstrates strong performance

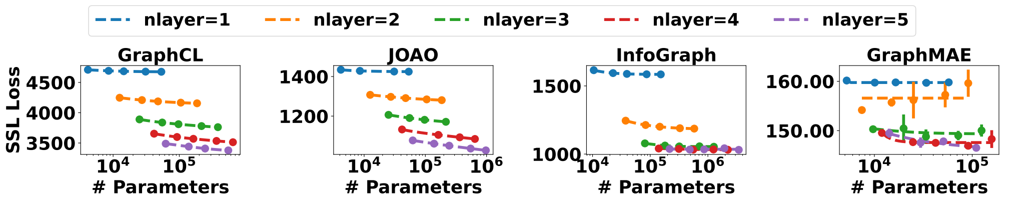
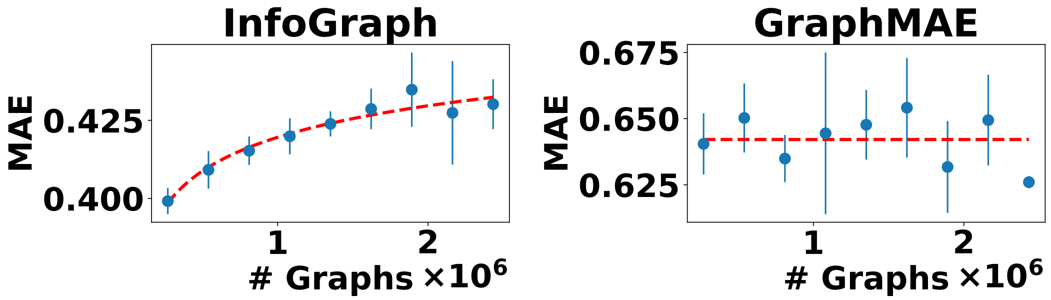


Figure 37: SSL Loss on ZINC-Full grouped by layer with standard deviation. No obvious scaling behaviour can be observed. x-axis denotes the total number of parameters and y-axis denotes SSL Loss on the held-out test set. The R^2 values for each method are listed as follows. GraphCL:0.99, JOAO:0.99, InfoGraph:0.78, GraphMAE:0.58.

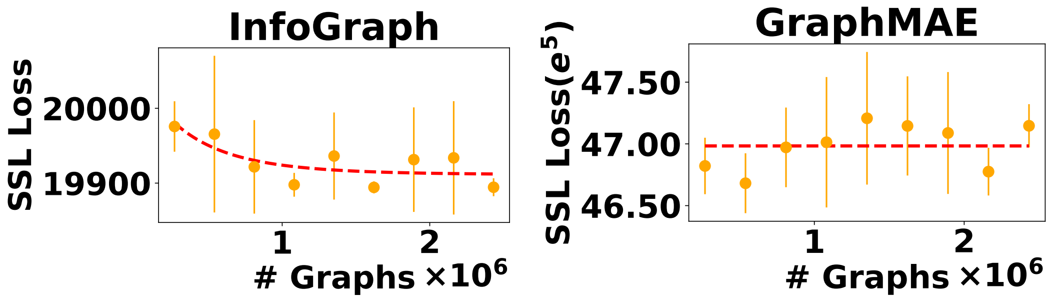
on the Graph SSL task. This suggests that the model does not rely heavily on large-scale data to achieve effective feature reconstruction, further supporting the notion that the task's ease within our specific dataset diminishes the potential scaling behavior.



(a) InfoGraph $R^2 = 0.94$

(b) GraphMAE $R^2 = 0.31$

Figure 38: Data Scaling of Performance on PCQM4Mv2 with standard deviation. x-axis indicates the number of edges used for pre-training and y-axis indicates the downstream performance. Notably, the increasing MAE(mean absolute error) does not indicate improved downstream performance.



(a) InfoGraph $R^2 = 0.59$

(b) GraphMAE $R^2 = 0.0$

Figure 39: Data Scaling of SSL Loss on PCQM4Mv2 with standard deviation. x-axis indicates the number of edges used for pre-training and y-axis indicates the SSL Loss on the held-out test set.

H.2 Model Scaling Results

We follow the same model scaling settings outlined in Section 5 to ensure consistency in our results. It is important to note that the PCQM4Mv2 task is a regression problem, with mean absolute error (MAE) as the evaluation metric. Therefore, the observed increase in MAE with more layers of backbone GIN in Figure 40 does not indicate improved downstream performance for InfoGraph or GraphMAE. In fact, this trend suggests that as stacking more layers of backbone GIN, the downstream performance actually declines. We hypothesize this phenomenon as the result of over-squashing issue, which may hinder the model's capability to capture the long-range dependency. The PCQM4Mv2 consists of small graphs with “average nodes”: 14.11, and “average edges”: 14.52. Then in this case, it is very likely that increasing more layers of the backbone GIN may cause over-squashing. Then the critical long-range interaction may not be well-captured [41], potentially leading to the loss of valuable information in the graph-level representation and thus affecting the regression performance of the model. Moreover, as shown in Figure 41, using only 1 or 2 layers of the backbone GIN achieve better results than using more layers. Meanwhile, increasing the hidden-size with the number of layer fixed can not further improve the performance. Therefore, our observation is that under two different manners of model scaling settings, the downstream performance can not improve to the increasing model parameters correspondingly.

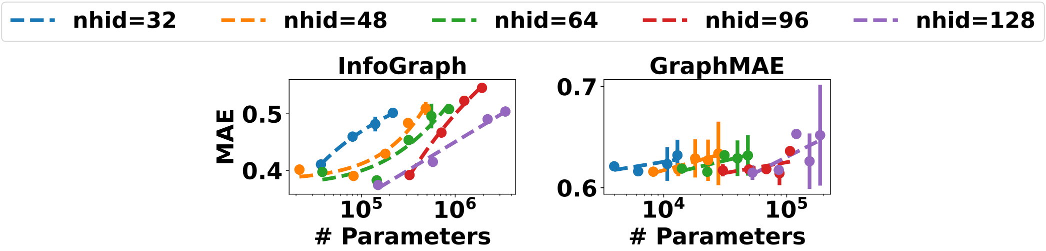


Figure 40: SSL Loss on PCQM4MV2 with standard deviation . x-axis denotes the total number of parameters and y-axis denotes the downstream performance. The R^2 values for each method are listed as follows, InfoGraph:0.96,GraphMAE:0.66.

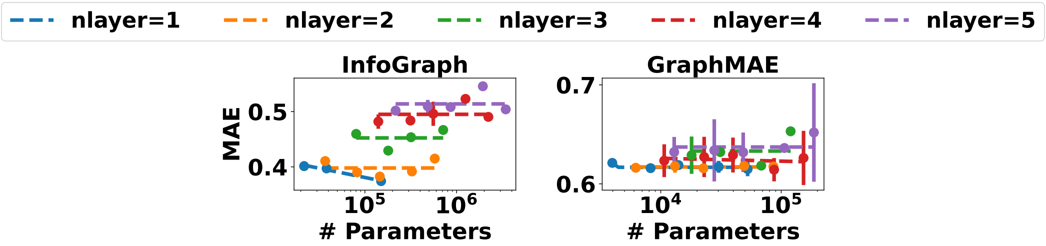


Figure 41: SSL Loss on PCQM4MV2 with standard deviation . x-axis denotes the total number of parameters and y-axis denotes the downstream performance. The R^2 values for each method are listed as follows, InfoGraph:0.19,GraphMAE:0.13.

Under the two different manners of model scaling settings, there is scaling behavior can be observed on the SSL loss of InfoGraph by increasing the number of layers of the backbone GIN, as shown in Figure 42 , while no scaling behavior can be observed for increasing the hidden size with the number of layers fixed in Figure 43. This conclusion remain the same as the conclusion in the main content.

However, for GraphMAE, there is a trend that the Graph SSL loss of GraphMAE, *i.e.,* the feature reconstruction loss will increase by stacking more layers of the backbone GIN as shown in Figure 42. Conversely, for other datasets such as ogbg-molhiv, ogbg-molpcba, and ogbg-ppa, the SSL loss consistently decreases as the number of GIN layers increases.

Here we propose two potential factors contributing to this phenomenon:

1. Inherent Difficulty of Masked Feature Reconstruction on PCQM4Mv2: The PCQM4Mv2 dataset could present a more challenging masked feature reconstruction task compared to other datasets. This difficulty may arise from the diverse atomic compositions and the stringent real-world constraints associated with quantum chemical properties. Specifically, the PCQM4Mv2 dataset encompasses a wider variety of atom types, which may make accurate prediction of masked nodes more complex. In contrast, other datasets like biological datasets(*e.g.,* ogbg-molhiv, ogbg-molpcba), and chemical compound dataset (including organic chemical compound,*e.g.,* ZINC_full) predominantly feature molecules rich in carbon atoms. The prevalence of carbon could simplify the reconstruction task, as predicting a masked carbon atom could be more likely to be accurate given its commonality in these datasets.

2. Over-Squashing Induced by Increasing GIN Layers Impairs Long-Range Dependency Capture: Long-range dependencies are crucial for accurately reconstructing masked features in molecular graphs, particularly for quantum chemical properties [41]. In the PCQM4Mv2 dataset, these dependencies are essential to satisfy the complex interactions required for chemical stability and accurate electronic structure representation. However, increasing the number of GIN layers in GraphMAE may cause the over-squashing problem, where information from distant nodes becomes compressed as it propagates through multiple layers. This compression may hinder the model's ability to capture and utilize long-range dependencies effectively, potentially leading to poorer reconstruction performance and, consequently, higher Graph SSL loss.

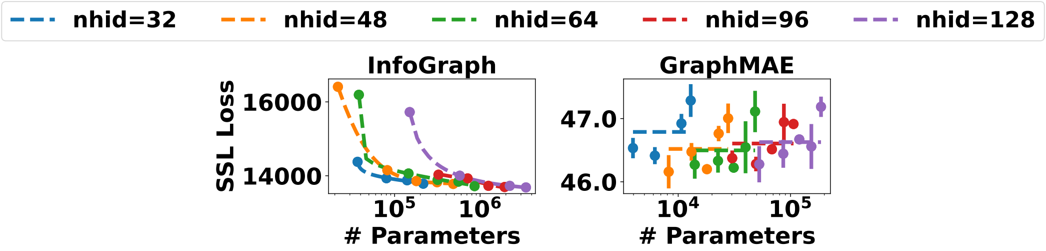


Figure 42: SSL Loss on PCQM4MV2 with standard deviation . x-axis denotes the total number of parameters and y-axis denotes the SSL Loss on the held-out test set. The R^2 values for each method are listed as follows, InfoGraph:0.96,GraphMAE:0.0.

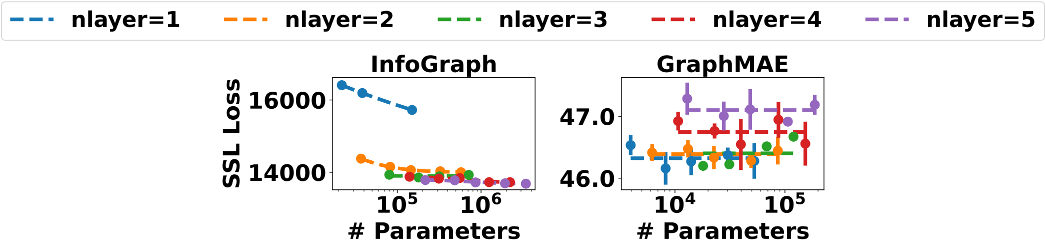


Figure 43: SSL Loss on PCQM4MV2 with standard deviation . x-axis denotes the total number of parameters and y-axis denotes the SSL Loss on the held-out test set. The R^2 values for each method are listed as follows, InfoGraph:0.76,GraphMAE:0.0.

I Results of replacing backbone for model scaling.

According to our obtained observation that the scaling behavior we observed is more related to the increasing number of aggregation, we also try to investigate by replacing the backbone with basic Graph Transformer for long-range perception.

We denote the InfoGraph using basic graph transformer as InfoGraphGTS and conducted the experiments on three datasets, reddit-threads, ogbg-molhiv and ogbg-molpcba.

The conclusion that there is no scaling behavior on the downstream performance remains the same like ogbg-molhiv and ogbg-molpcba as shown in Figure 44 and 45.

Meanwhile, for reddit-threads, the InfoGraphGTS perform worse than original InfoGraph with GIN as shwon in Figure 46. This may be resulted by the heavy rely on the graph structure. As the node features only including the degree information, it is very likely that capturing the connection among different nodes via basic Graph Transformer could be challenging.

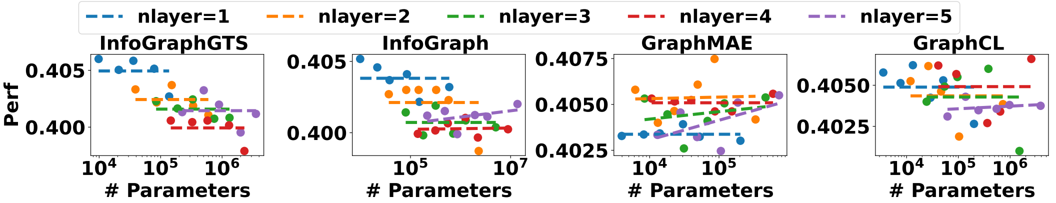


Figure 44: Performance on ogbg-molpcba grouped by layers. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed.The R^2 values for each method are listed as follows, InfoGraphGTS:0.0,InfoGraph:0.03,GraphMAE:0.07,GraphCL:0.03.

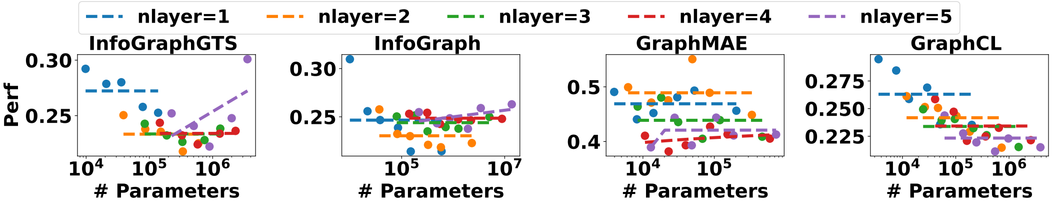


Figure 45: Performance on ogbg-molhiv grouped by layers. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed.The R^2 values for each method are listed as follows, InfoGraphGTS:0.06.InfoGraph:0.04,GraphMAE:0.07,GraphCL:0.00.

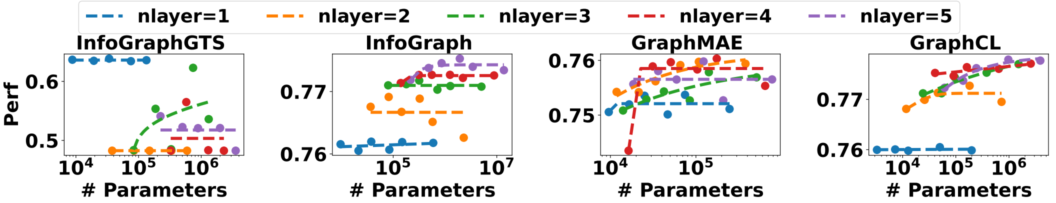


Figure 46: Performance on reddit-threads grouped by layers. x-axis denotes the total number of parameters and y-axis denotes the downstream performance. No obvious scaling behaviour can be observed.The R^2 values for each method are listed as follows, InfoGraphGTS:nan,InfoGraph:0.33,GraphMAE:0.53,GraphCL:0.61.