TOWARDS SELF-SUPERVISED LEARNING AND EXPLAINING OF DEEP MODELS

A Dissertation

by

YAOCHEN XIE

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Shuiwang Ji
Committee Members,	Yu Ding
	Ruihong Huang
	Bobak Mortazavi
Head of Department,	Scott Schaefer

August 2023

Major Subject: Computer Science

ABSTRACT

Deep learning approaches have demonstrated impressive performance on a variety of data and tasks, where deep models take some data as inputs and are trained to output desired predictions. While the capability of expressiveness of advanced deep models has been improved greatly, their training requires a huge amount of data. A common way to train a deep model is to use the supervised mode in which a sufficient amount of input data and label pairs are given. However, since a large number of labels are required, the supervised training becomes inapplicable in many real-world scenarios, where labels are expensive, limited, imbalanced, or even unavailable. In such cases, self-supervised learning (SSL) enables the training of deep models on unlabeled data, removing the need for excessively annotated labels. When no labeled data is available, SSL can serve as an promising approach to learning representations from and enabling explainability for unlabeled data. In this dissertation, we study and develop multiple theoretically grounded approaches of using self-supervision to perform both learning and explanation under multiple scenarios with image and graph data.

The general goal of learning is to learn representations that are both informative and robust to noise from unlabeled data. In contrast to supervised learning, it is more challenging for SSL to learn deep models that are robust to the noise in given data. This is because the self-supervision from data itself may include noise. To achieve such a goal with SSL, we start by studying and investigating the denoising capability of SSL approaches. In particular, we study SSL approaches in the image denoising problems under the scenarios where clean image are unavailable. Self-supervised frameworks that learn denoising models with merely individual noisy images have shown strong capability and promising performance in various image denoising tasks. Existing self-supervised denoising frameworks are mostly built upon the same theoretical foundation inspired by denoising autoencoder, where the denoising models are required to be $\mathcal{J}$-invariant. However, our analyses indicate that the current theory and the $\mathcal{J}$-invariance may lead to denoising models with reduced performance. In this dissertation, we first introduce Noise2Same, a novel

self-supervised denoising framework. In Noise2Same, a new self-supervised loss is proposed by deriving a self-supervised upper bound of the typical supervised loss. In particular, Noise2Same requires neither $\mathcal{J}$-invariance nor extra information about the noise model and can be used in a wider range of denoising applications. We analyze our proposed Noise2Same both theoretically and experimentally. The experimental results show that our Noise2Same remarkably outperforms previous self-supervised denoising methods in terms of denoising performance and training efficiency.

Given the promising capability of denoising, we further generalize above theoretical framework for SSL into even more challenging data and problems. Specifically, we propose self-supervised approaches to learn representations with graph neural networks (GNNs) on graph data. SSL of GNNs is emerging as a promising way of leveraging unlabeled graph data. Currently, most methods are based on contrastive learning adapted from the image domain, which requires view generation and a sufficient number of negative samples. In contrast, existing predictive models do not require negative sampling, but lack theoretical guidance on the design of pretext training tasks. In this dissertation, we then propose the LaGraph, a predictive SSL framework grounded by the above denoising theory and by formulating the SSL task as the latent graph prediction problem. Learning objectives of LaGraph are derived as self-supervised upper bounds to objectives for predicting unobserved latent graphs. In addition to its improved performance, LaGraph provides explanations for recent successes of predictive models that include invariance-based objectives. We provide theoretical analysis comparing LaGraph to related methods in different domains. Our experimental results demonstrate the superiority of LaGraph in performance and the robustness to the decreasing training sample size on both graph-level and node-level tasks.

To ensure reliable deep models are learned under self-supervision, one approach is to enable the explainability of self-supervisely trained models. However, without given downstream tasks and labels, the explanation become infeasible with existing learning-based explanation pipelines and approaches. Specifically, they are incapable of producing explanations for a multitask prediction model with a single explainer. They are also unable to provide explanations in cases where

the model is trained in a self-supervised manner, and the resulting representations are used in future downstream tasks. In this dissertation, we further demonstrate with graph data that self-supervision can further be used to learn to explain self-supervisely trained deep models. Specifically, we propose a Task-Agnostic GNN Explainer (TAGE) that is independent of downstream models and trained under self-supervision with no knowledge of downstream tasks. TAGE enables the explanation of GNN embedding models with unseen downstream tasks and allows the efficient explanation of multitask models. Our extensive experiments show that TAGE can significantly speed up the explanation efficiency by using the same model to explain predictions for multiple downstream tasks while achieving an explanation quality as good as or even better than the current state-of-the-art GNN explanation approaches.

Finally, given the success in natural images and graph data, we further investigate the capability of self-supervised representation learning to advance scientific discoveries in the scenario of genome-wide association studies (GWAS), which are used to identify relationships between genetic variations and specific traits. When applying GWAS to high-dimensional medical imaging data, a key step is to extract lower-dimensional, yet informative representations of the data as traits. Representation learning for imaging genetics is largely under-explored due to the unique challenges posed by GWAS in comparison to typical visual representation learning. We tackle this problem from the mutual information (MI) perspective by identifying key limitations of existing SSL methods. We introduce a trans-modal SSL framework Genetic InfoMax (GIM), including a regularized MI estimator and a novel genetics-informed transformer to address the specific challenges of GWAS. We evaluate GIM on human brain 3D MRI data and establish standardized evaluation protocols to compare it to existing approaches. Our results demonstrate the effectiveness of GIM and a significantly improved performance on GWAS.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my deepest gratitude to the individuals who have supported and guided me throughout my dissertation journey. As the power of self-supervised learning is still limited, my achievements in the journey wouldn't have been possible without their supports.

First and foremost, I would like to extend my sincere appreciation to my advisor, Dr. Shuiwang Ji. His unwavering commitment to my research, his invaluable guidance, and expertise in the field have been instrumental in shaping the outcome of this dissertation. His mentorship and constructive feedback have challenged me to think critically and strive for excellence. I am truly grateful for the opportunity to work under his supervision.

Equally importantly, I would like to acknowledge the support and encouragement of my incredible spouse, Lin Gui, and all our family members. Their unwavering belief in me, patience, and understanding during the ups and downs of this challenging process have been my greatest source of strength. Their love and support have made this journey all the more meaningful, and I am forever grateful.

Moreover, I would like to acknowledge and thank my advisory committee members, Dr. Yu Ding, Dr. Ruihong Huang, and Dr. Bobak Mortazavi, for their valuable insights and support during my research. I would also like to thank my former collaborators and colleagues in my lab for their collaborative spirit and willingness to share their expertise have enriched my work significantly. The collective effort and the exchange of ideas have been invaluable in shaping the outcomes of this dissertation. Additionally, I would like to extend my heartfelt thanks to all my friends who have stood by me throughout this journey. Their friendship, encouragement, and occasional distractions have provided much-needed support.

Finally, I am also grateful to the staff and faculty at Texas A&M University for providing the resources, facilities, and a stimulating academic environment. The opportunities and support from the university have played a crucial role in my intellectual growth and development.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Shuiwang Ji, Ruihong Huang, and Bobak Mortazavi of the Department of Computer Science & Engineering and Professor Yu Ding of the Department of Industrial & Systems Engineering.

This dissertation is based on my publications and manuscripts collaborated with Zhao Xu and Jingtun Zhang from Texas A&M University, Dr. Zhengyang Wang , Dr. Sumeet Katariya, Dr. Xianfeng Tang, Dr. Edward W. Huang, Dr. Nikhil Rao, and Dr. Karthik Subbian from Amazon.com LLC, Dr. Ziqian Xie, Sheikh Muhammad Saiful Islam, and Dr. Degui Zhi from University of Texas Health Science Center. In those works, co-authors assisted to conduct or provided key feedback in certain parts of the experiments and writing. I would like to thank all the co-authors and collaborators for their contributions.

Funding Sources

This dissertation was supported in part by National Science Foundation grant IIS-2006861, IIS-1955189, IIS-1908220, IIS-1908198, IIS-1908166, DBI-2028361, and DBI-1922969, National Institutes of Health grant U01AG070112 and 1R21NS102828, Defense Advanced Research Projects Agency grant N66001-17-2-4031, and Amazon.com Inc.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Background and Preliminaries*

A deep model takes some data as its inputs and is trained to output desired predictions. A common way to train a deep model is to use the supervised mode in which a sufficient amount of input data and label pairs are given. However, since a large number of labels are required, the supervised training becomes inapplicable in many real-world scenarios, where labels are expensive, limited, imbalanced [5], or even unavailable. In such cases, self-supervised learning (SSL) enables the training of deep models on unlabeled data, removing the need of excessive annotated labels. When no labeled data is available, SSL serves as an approach to learn representations from unlabeled data itself. When a limited number of labeled data is available, SSL from unlabeled data can be used either as a pre-training process after which labeled data are used to fine-tune the pre-trained deep models for downstream tasks, or as an auxiliary training task that contributes to the performance of main tasks.

Recently, SSL has shown its promising capability in data restoration tasks, such as image super-resolution [6], image denoising [7, 1, 8], and single-cell analysis [2]. It has also achieved remarkable progress in representation learning for different data types, including language sequences [9, 10, 11], images [12, 13, 14, 15], and graphs with sequence models [16, 17] or spectral models [18]. The key idea of these methods is to define pretext training tasks to capture and use the dependencies among different dimensions of the input data, *e.g.*, the spatial, temporal, or channel dimensions, with robustness and smoothness. Taking the image domain as an example, [19, 20], and [21] design different pretext tasks to train convolutional neural networks (CNNs) to capture relationships between different crops from an image. [13] and [22] train CNNs to capture dependencies between different augmentations of an image.

Based on how the pretext training tasks are designed, SSL methods can be divided into two cat-

*Partially reprinted with permission from "*Self-supervised learning of graph neural networks: A unified review*" by Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, Shuiwang Ji, 2022. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45, 2412 - 2429, Copyright 2023 by IEEE.

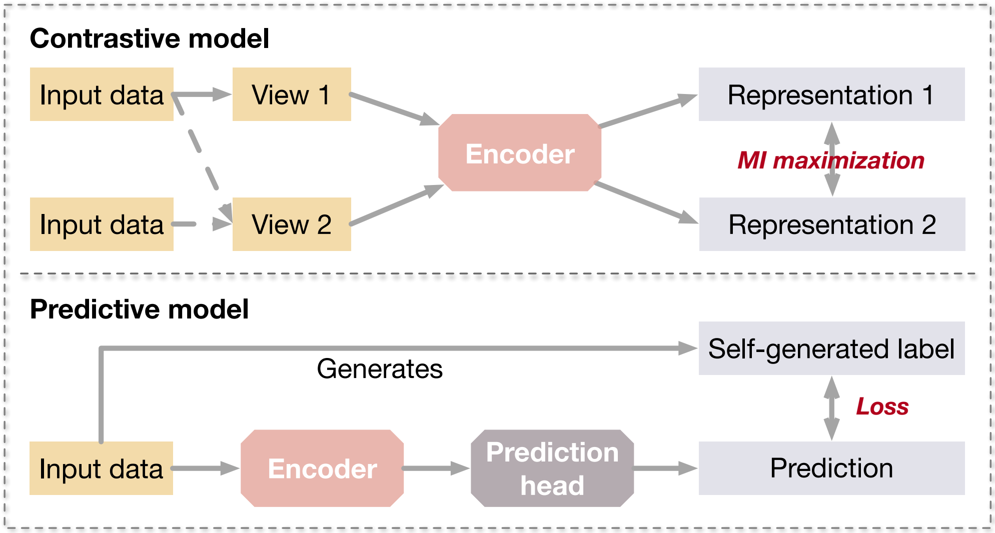


Figure 1.1: A comparison between the contrastive model and the predictive model in general.

egories; namely contrastive models and predictive models. The major difference between the two categories is that contrastive models require data-data pairs for training, while predictive models require data-label pairs, where the labels are self-generated from the data, as illustrated in Figure 1.1. Contrastive models usually utilize self-supervision to learn data representation or perform pre-training for downstream tasks. Given the data-data pairs, contrastive models perform discrimination between positive pairs and negative pairs. On the other hand, predictive models are trained in a supervised fashion, where the labels are generated based on certain properties of the input data or by selecting certain parts of the data. Predictive models usually consist of an encoder and one or more prediction heads. When applied as a representation learning or pre-training method, the prediction heads of a predictive model are removed in the downstream task.

In graph data analysis, SSL can potentially be of great importance to make use of a massive amount of unlabeled graphs such as molecular graphs [23, 24]. With the rapid development of graph neural networks (GNNs) [25, 26, 27, 28, 29, 30, 31], basic components of GNNs [32, 33, 34, 35, 36, 37] and other related fields [38, 39] have been well studied and made substantial progress. In comparison, applying SSL on GNNs is still an emerging field. Due to the similarity in data structure, many SSL methods for GNNs are inspired by methods in the image domain, such as

DGI [40] and graph autoencoders [41]. However, there are several key challenges in applying SSL on GNNs due to the uniqueness of the graph-structured data. To obtain good representations of graphs and perform effective pre-training, self-supervised models are supposed to capture essential information from both nodes attributes and structural topology of graphs [42].

For contrastive models, The performance of the learned representation or pre-trained model on downstream tasks heavily depends on the specific contrastive objective and the selection of transformations to generate views. Recent works study advanced theory-guided contrastive objectives [43] and view generation approaches [44, 45] that optimize the downstream performance of contrastive models. Although promising performance can be achieved, contrastive learning approaches usually suffer from computational cost and memory issue because its training requires the contrast among a sufficient number of examples at the same time. The drawback prevent its application to extremely large graphs which are very common in industrial applications.

For predictive models, it becomes essential that what labels should be generated so that the non-trivial representations are learned to capture information in both node attributes and graph structures. Unlike contrastive methods grounded by the problem of mutual information maximization, the predictive methods, especially the graph property prediction and invariance regularization-based methods, utilize different pretext learning tasks motivated by individual hypotheses and based on empirical studies. However, they lack guidance from unified theoretical frameworks to design specific pretext tasks for different downstream tasks. The information bottleneck principle [46, 47] may be used to interpret the effectiveness of several predictive methods but further study and investigation are desired.

1.2 Dissertation Outline

In this dissertation, we aim to study reliable self-supervised learning approaches for image and graph data. Specifically, we propose and develop theory-guided self-supervised frameworks and study the explainability of deep models trained under self-supervision without given downstream tasks.

The general goal of learning is to learn representations that are both informative and robust to noise from unlabeled data. The robustness of deep models to noise in data is especially critical for its reliability. However, in contrast to supervised learning, it is more challenging for SSL to learn deep models that are robust to the noise in given data. This is because the self-supervision from data itself may include noise. To achieve such a goal with SSL, in Chapter 2, we start by studying and investigating the denoising capability of SSL approaches. In particular, we study SSL approaches in the image denoising problems under the scenarios where clean image are unavailable. We first conduct analysis on existing self-supervised image denoising approaches and show their limitations due to the inconsistency between the assumption of their theorems and empirical conditions in Section 2.3. Next, in Section 2.4, we introduce the proposed self-supervised objectives as a strict upper bound to the supervised objective for image denoising. We then conduct theoretical and empirical analyses to show the connection to existing approaches. Finally, in Section 2.5, the experimental results show that our Noise2Same remarkably outperforms previous self-supervised denoising methods in terms of denoising performance and training efficiency.

Given the promising capability of denoising, we further generalize above theoretical framework for SSL into even more challenging data and problems. In Chapter 3, we propose self-supervised approaches to learn representations with graph neural networks (GNNs) on graph data. In Section 3.2, we propose the LaGraph, a predictive SSL framework grounded by the above denoising theory and by formulating the SSL task as the latent graph prediction problem. Learning objectives of LaGraph are derived as self-supervised upper bounds to objectives for predicting unobserved latent graphs. In Section 3.3, we provide theoretical analysis comparing LaGraph to related methods in different domains. Finally, in Section 3.4, our experimental results demonstrate the superiority of LaGraph in performance and the robustness to the decreasing training sample size on both graph-level and node-level tasks.

To ensure reliable deep models are learned under self-supervision, one approach is to enable the explainability of self-supervisely trained models. However, without a given downstream task, the explanation become infeasible with existing learning-based explanation pipelines and approaches. Specifically, they are incapable of producing explanations for a multitask prediction model with a single explainer. They are also unable to provide explanations in cases where the model is trained in a self-supervised manner, and the resulting representations are used in future downstream tasks. In Chapter 4, we further demonstrate with graph data that self-supervision can further be used to learn to explain self-supervisely trained deep models. In Section 4.2, we formulate the task-agnostic explanation problem and introduce a task-agnostic pipeline. Following the pipeline, in Section 4.3, we propose a Task-Agnostic GNN Explainer (TAGE) that is independent of downstream models and trained under self-supervision with no knowledge of downstream tasks. In Section 4.4, our extensive experiments show that TAGE can significantly speed up the explanation efficiency by using the same model to explain predictions for multiple downstream tasks while achieving explanation quality as good as or even better than current state-of-the-art GNN explanation approaches.

Finally, in Chapter 5, we formulate the GWAS problem into a representation learning task, identify, and address the challenges of applying self-supervised learning approaches in the scenario of GWAS with high-dimensional imaging data. Specifically, in Section 5.2, we distinguish the key differences between representation learning for natural images and for GWAS and demonstrate the differences lead to the failure of typical SSL methods. Based on the analysis, we propose the learning framework Genetic Infomax that is well-suited for learning representations for the GWAS purpose in Sections 5.3 and 5.4. Our experiments in Section 5.6 demonstrate a significantly improved performance in terms of the number of brain-gene associations discovered from the learned representations.

2. NOISE2SAME: OPTIMIZING A SELF-SUPERVISED BOUND FOR IMAGE DENOISING [*]

2.1 Introduction

The quality of deep learning methods for signal reconstruction from noisy images, also known as deep image denoising, has benefited from the advanced neural network architectures such as ResNet [48], U-Net [49] and their variants [50, 51, 52, 53, 54, 55]. While more powerful deep image denoising models are developed over time, the problem of data availability becomes more critical.

Most deep image denoising algorithms are supervised methods that require matched pairs of noisy and clean images for training [56, 50, 57, 58]. The problem of these supervised methods is that, in many denoising applications, clean images are hard to obtain due to instrument or cost limitations. To overcome this problem, *Noise2Noise* [59] explores an alternative training framework, where pairs of noisy images are used for training. Here, each pair of noisy images should correspond to the same but unknown clean image. Note that *Noise2Noise* is basically still a supervised method, just with noisy supervision.

Despite the success of *Noise2Noise*, its application scenarios are still limited as pairs of noisy images are not available in some cases and may have registration problems. Recently, various of denoising frameworks that can be trained on individual noisy images [6, 60, 61, 8, 2, 1] have been developed. These studies can be divided into two categories according to the amount of extra information required. Methods in the first category require the noise model to be known. For example, the simulation-based methods [60, 61] use the noise model to generate simulated noises and make individual noisy images noisier. Then a framework similar to *Noise2Noise* can be applied to train the model with pairs of noisier images and the original noisy image. The limitation is obvious as the noise model may be too complicated or even not available.

[*] Reprinted with permission from "*Noise2Same: Optimizing a self-supervised bound for image denoising.*" by Yaochen Xie, Zhengyang Wang, and Shuiwang Ji, 2020. Advances in Neural Information Processing Systems, Copyright 2020 by the authors.

On the other hand, algorithms in the second category target at more general cases where only individual noisy images are available without any extra information [6, 8, 2, 1]. In this category, self-supervised learning [62, 19, 63] has been widely explored, such as *Noise2Void* [8], *Noise2Self* [2], and the *convolutional blind-spot neural network* [1]. Note that these self-supervised models can be improved as well if information about the noise model is given. For example, Laine et al. [1] and Krull et al. [64] propose the Bayesian post-processing to utilize the noise model. However, with the proposed post-processing, these methods fall into the first category where applicability is limited.

In this chapter, we stick to the most general cases where only individual noisy images are provided and focus on the self-supervised framework itself without any post-processing step. We note that all of these existing self-supervised denoising frameworks are built upon the same theoretical background, where the denoising models are required to be $\mathcal{J}$-invariant (Section 2.2). We perform in-depth analyses on the $\mathcal{J}$-invariance property and argue that it may lead to denoising models with reduced performance. Based on this insight, we propose *Noise2Same*, a novel self-supervised denoising framework, with a new theoretical foundation. *Noise2Same* comes with a new self-supervised loss by deriving a self-supervised upper bound of the typical supervised loss. In particular, *Noise2Same* requires neither $\mathcal{J}$-invariance nor extra information about the noise model. We analyze the effect of the new loss theoretically and conduct thorough experiments to evaluate *Noise2Same*. Result show that our *Noise2Same* consistently outperforms previous self-supervised denoising methods.

2.2 Background and Related Studies

2.2.1 Self-Supervised Denoising with $\mathcal{J}$Invariant Functions

We consider the reconstruction of a noisy image $\boldsymbol{x} \in \mathbb{R}^m$, where $m = (d\times)h \times w \times c$ depends on the spatial and channel dimensions. Let $\boldsymbol{y} \in \mathbb{R}^m$ denotes the clean image. Given noisy and clean image pairs $(\boldsymbol{x}, \boldsymbol{y})$, supervised methods learn a denoising function $f : \mathbb{R}^m \to \mathbb{R}^m$ by minimizing the supervised loss $\mathcal{L}(f) = \mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2$.

Previous self-supervised denoising methods have been developed [8, 2, 1] by assuming that the noise is zero-mean and independent among all dimensions. These methods are trained on individual noisy images to minimize the self-supervised loss $\mathcal{L}(f) = \mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2$. Particularly, in order to prevent the self-supervised training from collapsing into leaning the identity function, Batson et al. [2] point out that the denoising function f should be $\mathcal{J}$-invariant, as defined below.

Definition 1. *For a given partition $\mathcal{J} = \{J_1, \cdots, J_k\}$ ($|J_1| + \cdots + |J_k| = m$) of the dimensions of an image $\boldsymbol{x} \in \mathbb{R}^m$, a function $f : \mathbb{R}^m \to \mathbb{R}^m$ is **$\mathcal{J}$-invariant** if $f(\boldsymbol{x})_J$ does not depend on $\boldsymbol{x}_J$ for all $J \in \mathcal{J}$, where $f(\boldsymbol{x})_J$ and $\boldsymbol{x}_J$ denotes the values of $f(\boldsymbol{x})$ and $\boldsymbol{x}$ on J, respectively.*

Intuitively, $\mathcal{J}$-invariance means that, when denoising $\boldsymbol{x}_J$, f only uses its context $\boldsymbol{x}_{J^c}$, where J^c denotes the complement of J. With a $\mathcal{J}$-invariant function f, we have

$$\begin{aligned} \mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2 &= \mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 + \mathbb{E}_{x,y} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 - 2 \left\langle f(\boldsymbol{x}) - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \right\rangle && (2.1) \\ &= \mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 + \mathbb{E}_{x,y} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 . && (2.2) \end{aligned}$$

Here, the third term in Equation 2.1 becomes zero when f is $\mathcal{J}$-invariant and the zero-mean assumption about the noise holds [2]. We can see from Equation 2.2 that when f is $\mathcal{J}$-invariant, minimizing the self-supervised loss $\mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2$ indirectly minimizes the supervised loss $\mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2$.

All existing self-supervised denoising methods [8, 2, 1] compute the $\mathcal{J}$-invariant denoising function f through a blind-spot network. Concretely, a subset J of the dimensions are sampled from the noisy image $\boldsymbol{x}$ as "blind spots". The blind-spot network f is asked to predict the values of these "blind spots" based on the context $\boldsymbol{x}_{J^c}$. In other words, f is blind on J. In previous studies, the blindness on J is achieved in two ways. Specifically, *Noise2Void* [8] and *Noise2Self* [2] use masking, while the *convolutional blind-spot neural network* [1] shifts the receptive field. With the blind-spot network, the self-supervised loss $\mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2$ can be written as

$$\mathcal{L}(f) = \mathbb{E}_J \mathbb{E}_{\boldsymbol{x}} \left\| f(\boldsymbol{x}_{J^c})_J - \boldsymbol{x}_J \right\|^2 . \tag{2.3}$$

While these methods have achieved good performance, our analysis in this chapter indicates that minimizing the self-supervised loss in Equation 2.3 with $\mathcal{J}$-invariant f is not optimal for self-supervised denoising. Based on this insight, we propose a novel self-supervised denoising framework, known as *Noise2Same*. In particular, our *Noise2Same* minimizes a new self-supervised loss without requiring the denoising function f to be $\mathcal{J}$-invariant.

2.2.2 Bayesian Post-Processing

From the probabilistic view, the blind-spot network f attempts to model $p(\boldsymbol{y}_J|\boldsymbol{x}_{J^c})$, where the information from $\boldsymbol{x}_J$ is not utilized thus limiting the performance. This limitation can be overcome through the Bayesian deep learning [65] if the noise model $p(\boldsymbol{x}|\boldsymbol{y})$ is known, as proposed by [1, 64]. Specifically, they propose to compute the posterior by

$$p(\boldsymbol{y}_J|\boldsymbol{x}_J, \boldsymbol{x}_{J^c}) \propto p(\boldsymbol{x}_J|\boldsymbol{y}_J)\; p(\boldsymbol{y}_J|\boldsymbol{x}_{J^c}),\ \forall J \in \mathcal{J}. \tag{2.4}$$

Here, the prior $p(\boldsymbol{y}_J|\boldsymbol{x}_{J^c})$ is Gaussian, whose the mean comes from the original outputs of the blind-spot network f and the variance is estimated by extra outputs added to f. The computation of the posterior is a post-processing step, which takes information from $\boldsymbol{x}_J$ into consideration.

Despite the improved performance, the Bayesian post-processing has limited applicability as it requires the noise model $p(\boldsymbol{x}_J|\boldsymbol{y}_J)$ to be knwon. Besides, it assumes that a single type of noise is present for all dimensions. In practice, it is common to have unknown noise models, inconsistent noises, or combined noises with different types, where the Bayesian post-processing is no longer applicable.

In contrast, our proposed *Noise2Same* can make use of the entire input image without any post-processing. Most importantly, *Noise2Same* does not require the noise model to be known and thus can be used in a much wider range of denoising applications.

2.3 Analysis of the $\mathcal{J}$Invariance Property

In this section, we analyze the $\mathcal{J}$-invariance property and motivate our work. In section 2.3.1, we experimentally show that the denoising functions trained through mask-based blind-spot methods are not strictly $\mathcal{J}$-invariant. Next, in Section 2.3.2, we argue that minimizing $\mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2$ with $\mathcal{J}$-invariant f is not optimal for self-supervised denoising.

2.3.1 Mask-Based Blind-Spot Denoising: Is the Optimal Function $\mathcal{J}$Invariant?

We show that, in mask-based blind-spot approaches, the optimal denoising function obtained through training is not strictly $\mathcal{J}$-invariant, which contradicts the theory behind these methods. As introduced in Section 2.2, mask-based blind-spot methods implement blindness on J through masking. Original values on J are masked out and replaced by other values. Concretely, in Equation 2.3, $\boldsymbol{x}_{J^c}$ becomes $\mathbb{1}_{J^c} \cdot \boldsymbol{x} + \mathbb{1}_J \cdot \boldsymbol{r}$, where $\boldsymbol{r}$ denotes the new values on the masked locations (J). As introduced in Section 2.2.1, *Noise2Void* [8] and *Noise2Self* [2] are current mask-based blind-spot methods. The main difference between them is the choice of the replacement strategy, *i.e.*, how to select $\boldsymbol{r}$. Specifically, *Noise2Void* applies the Uniform Pixel Selection (UPS) to randomly select $\boldsymbol{r}$ from local neighbors of the masked locations, while *Noise2Self* directly uses a random value.

Although the masking prevents f from accessing the original values on J during training, we point out that, during inference, f still shows a weak dependency on values on J, and thus does not strictly satisfy the $\mathcal{J}$-invariance property. In other words, mask-based blind-spot methods do not guarantee the learning of a $\mathcal{J}$-invariant function f. We conduct experiments to verify the above statement. Concretely, given a denoising function f trained through mask-based blind-spot methods, we quantify the strictness of $\mathcal{J}$-invariance by computing the following metric:

$$\mathcal{D}(f) = \mathbb{E}_J \mathbb{E}_x \left\| f(\boldsymbol{x}_{J^c})_J - f(\boldsymbol{x})_J \right\|^2 / |J|, \tag{2.5}$$

where x is the raw noisy image and $\boldsymbol{x}_{J^c}$ denotes the image whose values on J are replaced with random Gaussian noises (σ_m=0.5). Note that the replacement here is irrelevant to the the replacement strategy used in mask-based blind-spot methods. If the function f is strictly $\mathcal{J}$-invariant, $\mathcal{D}(f)$ should be close to 0 for all $\boldsymbol{x}$. Smaller $\mathcal{D}(f)$ indicates more $\mathcal{J}$-invariant f. To mitigate mutual influences among the locations within J, we use saturate sampling [8] to sample J and

Table 2.1: $\mathcal{D}(f)$ and PSNR of f trained through mask-based blind-spot methods with different replacement strategies on BSD68. The last column corresponds to a strictly $\mathcal{J}$-invariant model.

Replacement Strategy	Gaussian (σ=0.2)	Gaussian (σ=0.5)	Gaussian (σ=0.8)	Gaussian (σ=1.0)	UPS (5×5)	Shifting RF
$\mathcal{D}(f)$ ($\times 10^{-3}$)	4.326	10.91	2.141	1.569	18.31	0.105
PSNR	26.14	26.83	26.85	26.98	27.71	27.15

Table 2.2: $\mathcal{D}(f)$ and PSNR of f on trained through mask-based blind-spot methods with the same replacement strategy on different datasets.

Datasets	BSD68	HanZi	ImageNet
$\mathcal{D}(f)$ ($\times 10^{-3}$)	10.91	0.249	17.67
PSNR	26.83	13.94	20.38

make the sampling sparse enough (at a portion of 0.01%). $\mathcal{D}(f)$ is computed on the output of f before re-scaling back to [0,255]. In our experiments, we compare $\mathcal{D}(f)$ and the testing PSNR for f trained with different replacement strategies and on different datasets.

Table 2.1 provides the comparison results between f trained with different replacement strategies on the BSD68 dataset [66]. We also include the scores of the *convolutional blind-spot neural network* [1] for reference, which guarantees the strict $\mathcal{J}$-invariance through shifting receptive field, as discussed in Section 2.3.2. As expected, it has a close-to-zero $\mathcal{D}(f)$, where the non-zero value comes from mutual influences among the locations within J and the numerical precision. The large $\mathcal{D}(f)$ for all the mask-based blind-spot methods indicate that the $\mathcal{J}$-invariance is not strictly guaranteed and the strictness varies significantly over different replacement strategies.

We also compare results on different datasets when we fix the replacement strategy, as shown in Table 2.2. We can see that different datasets have strong influences on the strictness of $\mathcal{J}$-invariance as well. Note that such influences are not under the control of the denoising approach itself. In addition, although the shown results in Tables 2.1 and 2.2 are computed on testing dataset at the end of training, similar trends with $\mathcal{D}(f) \gg 0$ is observed during training.

Given the results in Tables 2.1 and 2.2, we draw our conclusions from two aspects. We first consider the mask together with the network f as a $\mathcal{J}$-invariant function g, *i.e.*, $g(x) := f(\mathbb{1}_{J^c} \cdot \boldsymbol{x} + \mathbb{1}_J \cdot \boldsymbol{r})$. In this case, the function g is guaranteed to be $\mathcal{J}$-invariant during training, and thus Equation 2.2 is valid. However, during testing, the mask is removed and a different non-$\mathcal{J}$-invariant function f is used because f achieves better performance than g, according to [2]. This contradicts the theoretical results of [2]. On the other hand, we consider the network f and the mask separately and perform training and testing with the same function f. In this case, the use of mask aims to help f learn to be $\mathcal{J}$-invariant during training so that Equation 2.2 becomes valid. However, our experiments show that f is neither strictly $\mathcal{J}$-invariant during training nor till the end of training, indicating that Equation 2.2 is not valid. With findings interpreted from both aspects, we ask whether minimizing $\mathbb{E}_x \|f(\boldsymbol{x}) - \boldsymbol{x}\|^2$ with $\mathcal{J}$-invariant f yields optimal performance for self-supervised denoising.

2.3.2 Shifting Receptive Field: How do the Strictly $\mathcal{J}$Invariant Models Perform?

We directly show that, with a strictly $\mathcal{J}$-invariant f, minimizing $\mathbb{E}_x \|f(\boldsymbol{x}) - \boldsymbol{x}\|^2$ does not necessarily lead to the best performance. Different from mask-based blind-spot methods, Laine et al. [1] propose the *convolutional blind-spot neural network*, which achieves the blindness on J by shifting receptive field (RF). Specifically, each pixel in the output image excludes its corresponding pixel in the input image from its receptive field. As values outside the receptive field cannot affect the output, the *convolutional blind-spot neural network* is strictly $\mathcal{J}$-invariant by design.

According to Table 2.1, the shift RF method outperforms all the mask-based blind-spot approaches with Gaussian replacement strategies, indicating the advantage of the strict $\mathcal{J}$-invariance. However, we notice that the UPS replacement strategy shows a different result. Here, a denoising function with less strict $\mathcal{J}$-invariance performs the best. One possible explanation is that the UPS replacement has a certain probability to replace a masked location by its original value. It weakens the $\mathcal{J}$-invariance of the mask-based denoising model but boosts the performance by yielding a result that is equivalent to computing a linear combination of the noisy input and the output of a strictly $\mathcal{J}$-invariant blind-spot network [2]. This result shows that minimizing $\mathbb{E}_x \|f(\boldsymbol{x}) - \boldsymbol{x}\|^2$ with a strictly $\mathcal{J}$-invariant f does not necessarily give the best performance. Another evidence

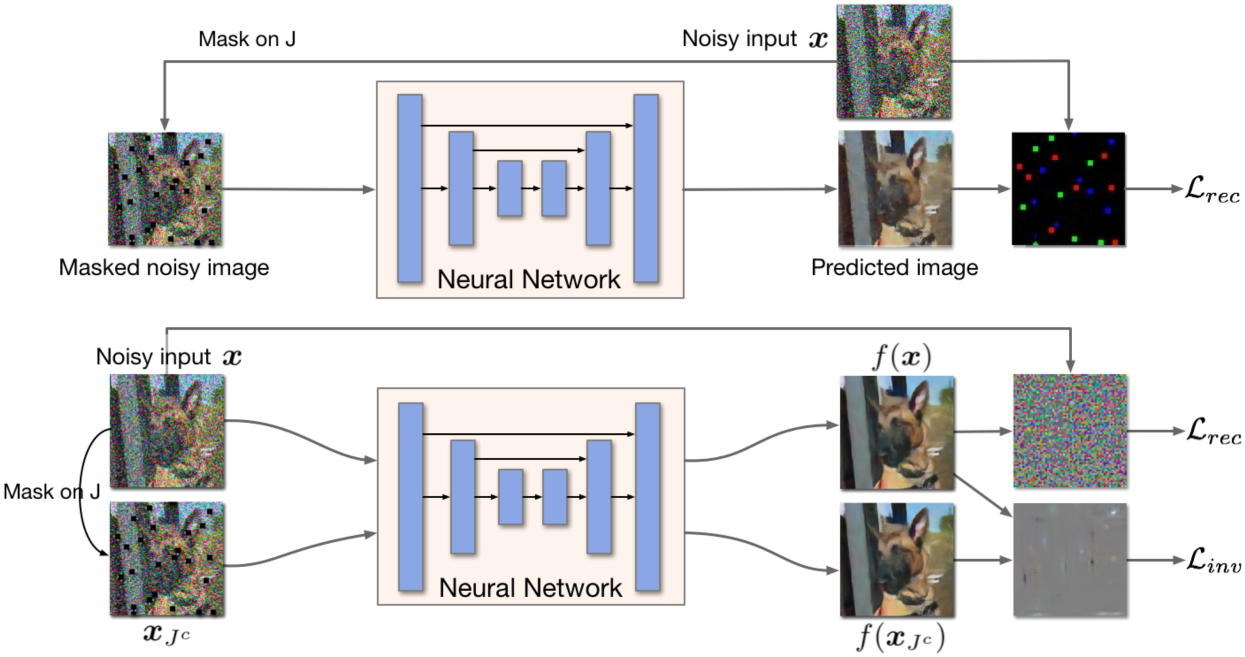


Figure 2.1: **Top**: The framework of the mask-based blind-spot denoising methods. The neural network takes the masked noisy image and predicts the masked value. The reconstruction loss is only computed on the masked dimensions. **Bottom**: The *Noise2Same* framework. The neural network takes both the full noisy image and the masked image as inputs and produces two outputs. The reconstruction loss is computed between the full noisy image and its corresponding output. The invariance loss is computed between the two outputs.

is the Bayesian post-processing introduced in Section 2.2.2, which also make the final denoising function not strictly $\mathcal{J}$-invariant while boosting the performance.

To conclude, we argue that minimizing $\mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2$ with $\mathcal{J}$-invariant f can lead to reduction in performance for self-supervised denoising. In this work, we propose a new self-supervised loss. Our loss does not require the $\mathcal{J}$-invariance. In addition, our proposed method can take advantage of the information from the entire noisy input without any post-processing step or extra assumption about the noise.

2.4 The Proposed Noise2Same Method

In this section, we introduce *Noise2Same*, a novel self-supervised denoising framework. *Noise2Same* comes with a new self-supervised loss. In particular, *Noise2Same* requires neither $\mathcal{J}$-invariant denoising functions nor the noise models.

2.4.1 Noise2Same: A Self-Supervised Upper Bound without the $\mathcal{J}$Invariance Requirement

As introduced in Section 2.2.1, the $\mathcal{J}$-invariance requirement sets the inner product term $\langle f(\boldsymbol{x}) - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle$ in Equation 2.1 to zero. The resulting Equation 2.2 shows that minimizing $\mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2$ with $\mathcal{J}$-invariant f indirectly minimizes the supervised loss, leading to the current self-supervised denoising framework. However, we have pointed out that this framework yields reduced performance.

In order to overcome this limitation, we propose to control the right side of Equation 2.2 with a self-supervised upper bound, instead of approximating $\langle f(\boldsymbol{x}) - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle$ to zero. The upper bound holds without requiring the denoising function f to be $\mathcal{J}$-invariant.

Theorem 1. *Consider a normalized noisy image $\boldsymbol{x} \in \mathbb{R}^m$ (obtained by subtracting the mean and dividing by the standard deviation) and its ground truth signal $\boldsymbol{y} \in \mathbb{R}^m$. Assume the noise is zero-mean and i.i.d among all the dimensions, and let J be a subset of m dimensions uniformly sampled from the image $\boldsymbol{x}$. For any $f : \mathbb{R}^m \to \mathbb{R}^m$, we have*

$$\mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 + \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 \leq \mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2 + 2m \, \mathbb{E}_J \left[\frac{\mathbb{E}_x \left\| f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^c})_J \right\|^2}{|J|} \right]^{1/2} \tag{2.6}$$

Proof. We consider the third term on the right-hand side of Equation 2.1. Instead of reducing the third term $2 \langle f(\boldsymbol{x}) - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle$ to 0 under the $\mathcal{J}$-invariant assumption, we control this term with its upper bound with the only assumption that $E[\boldsymbol{x}|\boldsymbol{y}] = \boldsymbol{y}$. Formally, we have

$$\mathbb{E}_{x,y} \langle f(\boldsymbol{x}) - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle = \mathbb{E}_y \mathbb{E}_{x|y} \sum_j (f(\boldsymbol{x})_j - y_j)(x_j - y_j) \tag{2.7}$$

$$= \sum_j \mathbb{E}_y \left[\mathbb{E}_{x|y} (f(\boldsymbol{x})_j - y_j)(x_j - y_j) - \mathbb{E}_{x|y} (f(\boldsymbol{x})_j - y_j) \mathbb{E}_{x|y} (x_j - y_j) \right] \tag{2.8}$$

$$= \sum_j \mathbb{E}_y \left[\mathrm{Cov}(f(\boldsymbol{x})_j - y_j, x_j - y_j | \boldsymbol{y}) \right] \tag{2.9}$$

$$= \sum_j \mathbb{E}_y \left[\mathrm{Cov}(f(\boldsymbol{x})_j, x_j | \boldsymbol{y}) \right]. \tag{2.10}$$

Equation 2.8 holds due to the zero-mean assumption, where $\mathbb{E}_{x|y}(x_j - y_j) = 0$. Now we let J be a uniformly sampled subset of the image dimensions $\{1, \cdots, m\}$, then we have the equation

$$\sum_j \mathbb{E}_y \left[\mathrm{Cov}(f(\boldsymbol{x})_j, x_j | \boldsymbol{y}) \right] = \frac{m}{|J|} \mathbb{E}_J \sum_{j \in J} \mathbb{E}_y \left[\mathrm{Cov}(f(\boldsymbol{x})_j, x_j | \boldsymbol{y}) \right]. \tag{2.11}$$

The right-hand side of the equation above can be controlled by applying Cauchy-Schwarz inequality while the input images are normalized. We have, for all J,

$$\frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Cov}(f(\boldsymbol{x})_j, x_j | \boldsymbol{y}) \Big] = \frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Cov}(f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j, x_j | \boldsymbol{y}) \Big] \tag{2.12}$$

$$\leq \frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Var}(f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j | \boldsymbol{y}) \cdot \mathrm{Var}(x_j | \boldsymbol{y}) \Big]^{1/2} \tag{2.13}$$

$$\leq \left(\frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Var}(f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j | \boldsymbol{y}) \cdot \mathrm{Var}(x_j | \boldsymbol{y}) \Big] \right)^{1/2} \tag{2.14}$$

$$\leq \left(\frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathbb{E}\big[[f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j]^2 \big] | \boldsymbol{y} \Big] \right)^{1/2} \tag{2.15}$$

$$= \left(\frac{1}{|J|} \sum_{j \in J} \mathbb{E} \Big[f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j \Big]^2 \right)^{1/2} \tag{2.16}$$

$$= \left(\frac{1}{|J|} \mathbb{E} \left\| f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^c})_J \right\|^2 \right)^{1/2}. \tag{2.17}$$

To be more specific, Equation 2.12 follows since $f(\boldsymbol{x}_{J^c})_J$ does not correlate to $\boldsymbol{x}_j$ due to the independent noise assumption and $j \notin J^c$, and subtracting $f(\boldsymbol{x}_{J^c})_j$ from $f(\boldsymbol{x})_j$ does not change the Covariance. Inequality 2.13 applies the Cauchy-Schwarz inequality. Inequality 2.14 holds due to $(\mathbb{E}X)^2 \leq \mathbb{E}X^2$. The derivation of Inequality 2.15 uses the fact that $\mathrm{Var}(x_j) = 1$ under normalization and $\mathrm{Var}(x_j | \boldsymbol{y}) \leq \mathrm{Var}(x_j) = 1$ for all j.

Consequently, we can control Equation (2.1) as

$$\mathbb{E}_{x,y}\left\|f(\boldsymbol{x})-\boldsymbol{y}\right\|^2+\mathbb{E}_{x,y}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2=\mathbb{E}_x\left\|f(\boldsymbol{x})-\boldsymbol{x}\right\|^2+2\,\mathbb{E}_{x,y}\langle f(\boldsymbol{x})-\boldsymbol{y},\boldsymbol{x}-\boldsymbol{y}\rangle \tag{2.18}$$

$$\leq \mathbb{E}_x\left\|f(\boldsymbol{x})-\boldsymbol{x}\right\|^2+2m\,\mathbb{E}_J\left[\frac{1}{|J|}\mathbb{E}\left\|f(\boldsymbol{x})_J-f(\boldsymbol{x}_{J^c})_J\right\|^2\right]^{1/2}. \tag{2.19}$$

This completes the proof of Theorem 1. □

With Theorem 1, we can perform self-supervised denoising by minimizing the right side of Inequality (2.6) instead. Following the theoretical result, we propose our new self-supervised denoising framework, *Noise2Same*, with the following self-supervised loss:

$$\mathcal{L}(f)=\mathbb{E}_x\left\|f(\boldsymbol{x})-\boldsymbol{x}\right\|^2/m+\lambda_{inv}\,\mathbb{E}_J\left[\mathbb{E}_x\left\|f(\boldsymbol{x})_J-f(\boldsymbol{x}_{J^c})_J\right\|^2/|J|\right]^{1/2}. \tag{2.20}$$

This new self-supervised loss consists of two terms: a reconstruction mean squared error (MSE) $\mathcal{L}_{rec}=\mathbb{E}_x\left\|f(\boldsymbol{x})-\boldsymbol{x}\right\|^2$ and a squared-root of invariance MSE $\mathcal{L}_{inv}=\mathbb{E}_J(\mathbb{E}_x\left\|f(\boldsymbol{x})_J-f(\boldsymbol{x}_{J^c})_J\right\|^2/|J|)^{1/2}$. Intuitively, $\mathcal{L}_{inv}$ prevents our model from learning the identity function when minimizing $\mathcal{L}_{rec}$ without any requirement on f. In fact, by comparing $\mathcal{L}_{inv}$ with $\mathcal{D}(f)$ in Equation 2.5, we can see that $\mathcal{L}_{inv}$ implicitly controls how strictly f should be $\mathcal{J}$-invariant, avoiding the explicit $\mathcal{J}$-invariance requirement. We balance $\mathcal{L}_{rec}$ and $\mathcal{L}_{inv}$ with a positive scalar weight λ_{inv}.

By default, we set $\lambda_{inv}=2$ according to Theorem 1. In some cases, setting λ_{inv} to different values according to the scale of observed $\mathcal{L}_{inv}$ during training could help achieve a better denoising performance.

Figure 2.1 compares our proposed *Noise2Same* with mask-based blind-spot denoising methods. Mask-based blind-spot denoising methods employ the self-supervised loss in Equation 2.3, where the reconstruction MSE $\mathcal{L}_{rec}$ is computed only on J. In contrast, our proposed *Noise2Same* computes $\mathcal{L}_{rec}$ between the entire noisy image $\boldsymbol{x}$ and the output of the neural network $f(\boldsymbol{x})$. To compute the invariance term $\mathcal{L}_{inv}$, we still feed the masked noisy image $\boldsymbol{x}_{J^c}$ to the neural network and compute MSE between $f(\boldsymbol{x})$ and $f(\boldsymbol{x}_{J^c})$ on J, *i.e.*, $f(\boldsymbol{x})_J$ and $f(\boldsymbol{x}_{J^c})_J$. Note that, while *Noise2Same* also samples J from $\boldsymbol{x}$, it does not require f to be $\mathcal{J}$-invariant.

2.4.2 Analysis of the Invariance Term

The invariance term $\mathcal{L}_{inv}$ is a unique and important part in our proposed self-supervised loss. In this section, we further analyze the effect of this term. To make the analysis concrete, we perform analysis based on an example case, where the noise model is given as the additive Gaussian noise $N(0, \sigma)$. Note that the example is for analysis purpose only, and the application of our proposed *Noise2Same* does not require the noise model to be known.

Theorem 2. *Consider a noisy image $\boldsymbol{x} \in \mathbb{R}^m$ and its ground truth signal $\boldsymbol{y} \in \mathbb{R}^m$. Assume the noise is i.i.d among all the dimensions, and let J be a subset of m dimensions uniformly sampled from the image $\boldsymbol{x}$. If the noise is additive Gaussian with zero-mean and standard deviation σ, we have*

$$\mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 + \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 \leq \mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2 + 2m\sigma \, \mathbb{E}_J \left[\frac{\mathbb{E} \left\| f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^c})_J \right\|^2}{|J|} \right]^{1/2} \tag{2.21}$$

Proof. We start from Equation (13) in the proof of Theorem 1. Since we have a stronger assumption that the noise model is known to be additive with standard deviation σ and zero-mean, we have $\mathrm{Var}(\boldsymbol{x}_j - \boldsymbol{y}_j) = \sigma^2$ for all j. Due to that the additive noise is orthogonal to the signal $\boldsymbol{y}$, we futher have the conditional variance $\mathrm{Var}(\boldsymbol{x}_j - \boldsymbol{y}_j | \boldsymbol{y}) = \sigma^2$. Then, similar to the proof of Theorem 1, we have,

$$\begin{aligned}
\frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Cov}(f(\boldsymbol{x})_j, x_j | \boldsymbol{y}) \Big] &= \frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Cov}(f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j, x_j - y_j | \boldsymbol{y}) \Big] && (2.22) \\
&\leq \frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Var}(f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j | \boldsymbol{y}) \cdot \mathrm{Var}(x_j - y_j | \boldsymbol{y}) \Big]^{1/2} && (2.23)
\end{aligned}$$

$$\leq \left(\frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathrm{Var}(f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j | \boldsymbol{y}) \cdot \mathrm{Var}(x_j - y_j | \boldsymbol{y}) \Big] \right)^{1/2} \tag{2.24}$$

$$= \left(\frac{1}{|J|} \sum_{j \in J} \mathbb{E}_y \Big[\mathbb{E}\big[[f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j]^2 | \boldsymbol{y} \big] \cdot \sigma^2 \Big] \right)^{1/2} \tag{2.25}$$

$$= \sigma \left(\frac{1}{|J|} \sum_{j \in J} \mathbb{E} \big[f(\boldsymbol{x})_j - f(\boldsymbol{x}_{J^c})_j \big]^2 \right)^{1/2} \tag{2.26}$$

$$= \sigma \left(\frac{1}{|J|} \mathbb{E} \left\| f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^c})_J \right\|^2 \right)^{1/2}. \tag{2.27}$$

Consequently, we can control Equation (2.1) as

$$\mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 + \mathbb{E}_{x,y} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 = \mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2 + 2\, \mathbb{E}_{x,y} \langle f(\boldsymbol{x}) - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle \tag{2.28}$$

$$\leq \mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2 + 2m\sigma\, \mathbb{E}_J \left(\frac{1}{|J|} \mathbb{E} \left\| f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^c})_J \right\|^2 \right)^{1/2}. \tag{2.29}$$

This completes the proof of Theorem 2. □

Note that the noisy image $\boldsymbol{x}$ here **does not require normalization** as in Theorem 1. Compared to Theorem 1, the σ from the noise model is added to balance the invariance term. As introduced in Section 2.4.1, the invariance term controls how strictly f should be $\mathcal{J}$-invariant and a higher weight of the invariance term pushes the model to learn a more strictly $\mathcal{J}$-invariant f. Therefore, Theorem 2 indicates that, when the noise is stronger with a larger σ, f should be more strictly $\mathcal{J}$-invariant. Based on the definition of $\mathcal{J}$-invariance, a more strictly $\mathcal{J}$-invariant f will depend more on the context $\boldsymbol{x}_{J^c}$ and less on the noisy input $\boldsymbol{x}_J$.

This result is consistent with the findings in previous studies. Batson et al. [2] propose to compute the linear combination of the noisy image and the output of the blind-spot network as a post-processing step, leading to better performance. The weights in the linear combination are determined by the variance of noise. And a higher weight is given to the output of the blind-spot network with larger noise variance. Laine et al. [1] derive a similar result through the Bayesian post-processing. This explains how the invariance term in our proposed *Noise2Same* improves denoising performance.

However, a critical difference between our *Noise2Same* and previous studies is that, the post-processing in [2, 1] cannot be performed when the noise model is unknown. To the contrary, *Noise2Same* is able to control how strictly f should be $\mathcal{J}$-invariant through the invariance term without any assumption about the noise or requirement on f. This property allows *Noise2Same* to be used in a much wider range of denoising tasks with unknown noise models, inconsistent noise, or combined noises with different types.

2.5 Experiments

We evaluate our *Noise2Same* on four datasets, including RGB natural images (ImageNet ILSVRC 2012 Val [67]), generated hand-written Chinese character images (HànZì [2]), physically captured 3D microscopy data (Planaria [56]) and grey-scale natural images (BSD68 [66]). The four datasets have different noise types and levels.

2.5.1 Comparisons with Baselines

The baselines include traditional denoising algorithms (*NLM* [68], *BM3D* [69]), supervised methods (*Noise2True*, *Noise2Noise* [59]), and previous self-supervised methods (*Noise2Void* [8], *Noise2Self* [2], the *convolutional blind-spot neural network* [1]). Note that we consider *Noise2Noise* as a supervised model since it requires pairs of noisy images, where the supervision is noisy. While *Noise2Void* and *Noise2Self* are similar methods following the blind-spot approach, they mainly differ in the strategy of mask replacement. To be more specific, *Noise2Void* proposes to use Uniform Pixel Selection (UPS), and *Noise2Self* proposes to exclude the information of the masked pixel and uses a random value on the range of given image data. As an additional mask strategy using the local average excluding the center pixel (donut) is mentioned in [2], we also include it for comparison. We use the same neural network architecture for all deep learning methods.

Note that ImageNet and HànZì have combined noises and Planaria has unknown noise models.

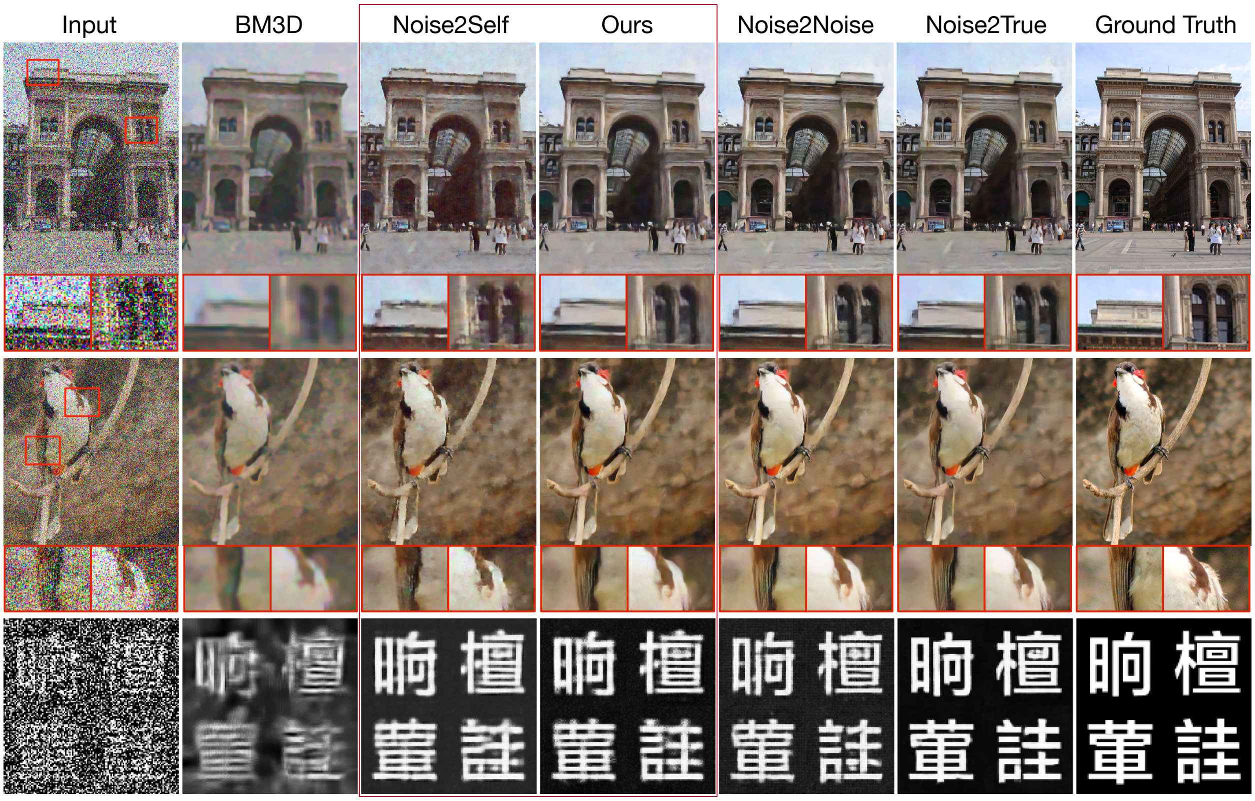


Figure 2.2: **RGB natural images and hand-written Chinese character images**: Visualizations of testing results on ImageNet dataset (first two rows) and the HànZì Dataset (the third row). We compare the denoising quality among the traditional method *BM3D*, supervised methods *Noise2True* and *Noise2Noise*, self-supervised approaches *Noise2Self* and our *Noise2Same*. From the left to the right, the columns are in the ascending order in terms of the denoising quality.

Table 2.3: Comparisons among denoising methods on different datasets, in terms of Peak Signal-to-Noise Ratio (PSNR). The post-processing of Laine et al. [1] that requires information about the noise model is included under the *Self-Supervised + noise model* category and is excluded under the *Self-Supervised* category. Noise2Self-Noise and Noise2Self-Donut refer to two masking strategies mentioned in [2], where the original results presented in [2] are produced by the noise masking. Bold numbers indicate the best performance among self-supervised methods.

		Datasets			
	Methods	ImageNet	HànZì	Planaria	BSD68
Traditional	Input	9.69	6.45	21.52 / 21.09 / 20.82	20.19
	NLM [68]	18.04	8.41	25.80 / 24.03 / 21.62	22.73
	BM3D [69]	18.74	10.90	-	28.59
Supervised	Noise2True	23.39	15.66	31.57 / 30.15 / 28.13	29.06
	Noise2Noise [59]	23.27	14.30	-	28.86
Self-Supervised + noise model	Laine et al. [1]	-	-	-	28.84
Self-Supervised	Laine et al. [1]	20.89	10.70	-	27.15
	Noise2Void [8]	21.36	13.72	25.84 / 23.57 / 21.60	27.71
	Noise2Self-Noise [2]	20.38	13.94	27.58 / 24.83 / 21.83	26.98
	Noise2Self-Donut [2]	8.62	13.29	27.63 / 24.72 / 21.73	**28.20**
	Noise2Same	**22.26**	**14.38**	**29.48 / 26.93 / 22.41**	27.95

As a result, the post-processing steps in *Noise2Self* [2] and the *convolutional blind-spot neural network* [1] are not applicable, as explained in Section 2.2. In order to make fair comparisons under the self-supervised category, we train and evaluate all models only using the images, without extra information about the noise. In this case, among self-supervised methods, only our *Noise2Same* and *Noise2Void* with the UPS replacement strategy can make use of information from the entire input image, as demonstrated in Section 2.3.2. We also include the complete version of the *convolutional blind-spot neural network* with post-processing, who is only available on *BSD68*, where the noise is not combined and the noise type is known.

Following previous studies, we use Peak Signal-to-Noise Ratio (PSNR) as the evaluation metric. The comparison results between our *Noise2Same* and the baselines in terms of PSNR on the four datasets are summarized in Table 2.3 and visualized in Figure 2.2. The results show that our *Noise2Same* achieve remarkable improvements over previous self-supervised baselines on Ima-

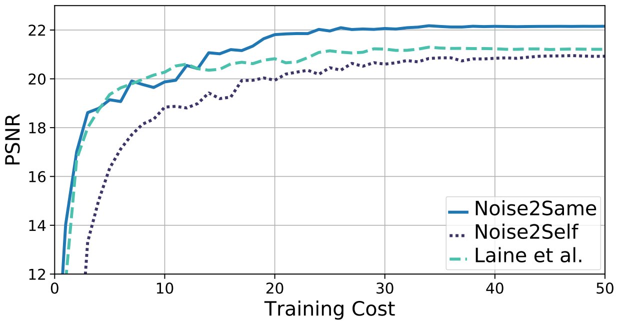


Figure 2.3: **Training efficiency**. For a fair comparison, we adjust the batch sizes for each method to fill the memory of a single GPU, namely, 128 for *Noise2Self*, 64 for *Noise2Same* and 32 for *Laine et al*. One unit of training cost represents 50 minibatch steps.

geNet, HànZì and CARE. In particular, on the ImageNet and the HànZì Datasets, our *Noise2Same* and *Noise2Void* demonstrate the advantage of utilizing information from the entire input image. Although the using of donut masking can achieve better performance on the BSD68 Dataset, it leads to model collapsing on the ImageNet Dataset and hence can be unstable. On the other hand, the *convolutional blind-spot neural network* [1] suffers from significant performance losses without the Bayesian post-processing, which requires information about the noise models that are unknown.

We note that, in our fair settings, supervised methods still have better performance over self-supervised models, especially on the Planaria and BSD68 datasets. One explanation is that the supervision usually carries extra information implicitly, such as information about the noise model. Here, we draw a conclusion different from Batson et al. [2]. That is, there are still performance gaps between self-supervised and supervised denoising methods. Our *Noise2Same* moves one step towards closing the gap by proposing a new self-supervised denoising framework.

In addition to the performance, we compares the training efficiency among self-supervised methods as well. Specifically, we plot how the PSNR changes during training on the ImageNet dataset. We compare *Noise2Same* with *Noise2Self* and the *convolutional blind-spot neural network*. The plot shows that our *Noise2Same* has similar convergence speed to the *convolutional blind-spot neural network*. On the other hand, as the mask-based method *Noise2Self* uses only

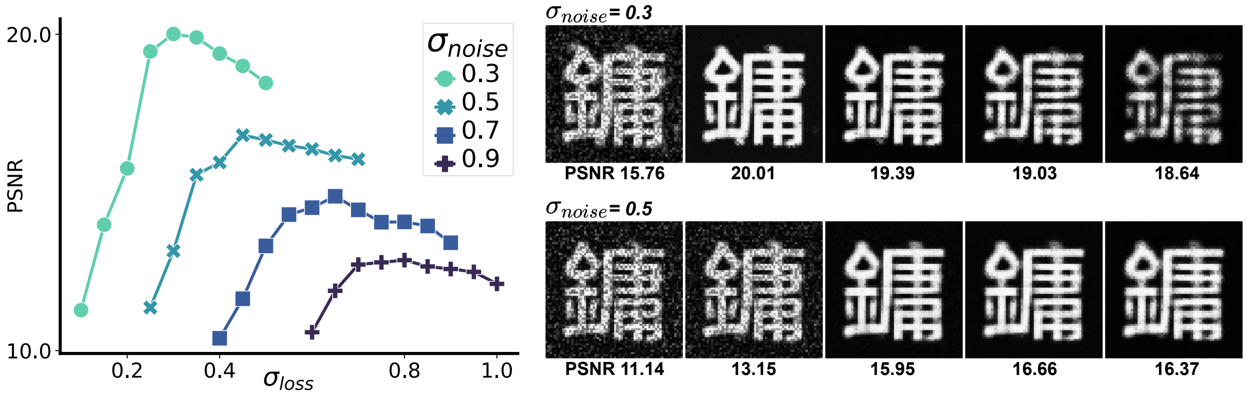


Figure 2.4: **Effect of the invariance term**. **Left**: Given additive Gaussian noise with certain σ_{noise}, how the performance of our *Noise2Same* varies over different σ_{loss}. **Right**: We visualize some denoising examples from noisy images with $\sigma_{noise} = 0.3, 0.5$. From left to right, the columns correspond to setting σ_{loss} to $0.2, 0.3, 0.4, 0.5, 0.6$, respectively.

a subset of output pixels to compute the loss function in each step, the training is expected to be slower [1].

2.5.2 Effect of the Invariance Term

In Section 2.4.2, we analyzed the effect of the invariance term using an example, where the noise model is given as the additive Gaussian noise. In this example, the variance of the noise controls how the strictness of the optimal f through the coefficient λ_{inv} of the invariance term.

Here, we conduct experiments to verify this insight. Specifically, we construct four noisy dataset from the HànZì dataset with only additive Gaussian noise at different levels ($\sigma_{noise} = 0.9, 0.7, 0.5, 0.3$). Then we train *Noise2Same* with $\lambda_{inv} = 2\sigma_{loss}$ by varying σ_{loss} from 0.1 to 1.0 for each dataset. According to Theorem 2, the best performance on each dataset should be achieved when σ_{loss} is close to σ_{noise}. The results, as reported Figure 2.4, are consistent with our theoretical results in Theorem 2.

2.6 Conclusion and Future Directions

We analyzed the existing blind-spot-based denoising methods and introduced *Noise2Same*, a novel self-supervised denoising method, which removes the assumption and over-restriction on the neural network as a $\mathcal{J}$-invariant function. We provided further analysis on *Noise2Same* and experimentally demonstrated the denoising capability of *Noise2Same*. As an orthogonal work, the combination of self-supervised denoising result and the noise model has be shown to provide additional performance gain. We would like to further explore noise model-augmented *Noise2Same* in future works.

3. SELF-SUPERVISED REPRESENTATION LEARNING VIA LATENT GRAPH PREDICTION [*]

3.1 Introduction

Self-supervised learning (SSL) methods seek to use supervisions provided by data itself and design effective pretext learning tasks. These methods allow deep models to learn from a massive amount of unlabeled data and have achieved promising successes in natural language processing [9, 10, 11] and image tasks [2, 70, 21, 13]. To use unlabeled graph data, earlier studies [16, 71] adapt sequence-based SSL methods [72, 73] to learn node representations. Inspired by the recent success of SSL in the image domain, a variety of SSL methods based on graph neural networks (GNNs) have been proposed in different learning paradigms. In particular, recent studies [40, 74, 3, 12, 75] construct SSL tasks as unsupervised approaches to learn representations from graph data at either node-level or graph-level; Hu et al. [76] propose SSL strategies to pre-train GNNs for downstream tasks; and other studies [77, 78] employ SSL as auxiliary tasks to boost the performance of main learning tasks.

Common taxonomies in recent survey works [79, 80] consider two categories of SSL methods to train GNNs; namely, contrastive methods and predictive methods. Contrastive methods employ pair-wise discrimination as their pretext learning tasks. It performs transformations or augmentations to obtain multiple views from a graph and trains GNNs to discriminate between jointly sampled view pairs and independently sampled view pairs. In contrast, predictive methods [81, 82, 83] train GNNs to predict certain labels obtained from the input graph, such as node reconstruction, connectivity reconstruction, graph statistical properties, and domain knowledge-based targets.

Adapted from the image domain, current state-of-the-art SSL methods for graphs are mostly contrastive. As a drawback, they usually depend on a large training sample size to include a sufficient number of negative samples. With limited computing resources, contrastive methods

[*] Reprinted with permission from "*Self-supervised representation learning via latent graph prediction.*" Yaochen Xie, Zhao Xu, and Shuiwang Ji. 2022. International Conference on Machine Learning, Copyright 2022 by the authors.

may not be applicable to large-scale graphs without suffering from performance loss. To address the drawback, BGRL [3] adapts BYOL [22] to the graph domain. BGRL still obtains different views from each given graph, but it eliminates the requirement of negative samples by replacing contrastive objectives with the prediction of offline embedding. BGRL has achieved competitive performance to the contrastive methods. However, unlike contrastive methods grounded by mutual information estimation and maximization, BYOL and BGRL lack theoretical guidance and require implementation measures to prevent collapsing to trivial representations, such as stop gradient, EMA, and normalization layers.

In this chapter, we propose *LaGraph*, a predictive SSL framework for representation learning of graph data, based on self-supervised latent graph prediction. In particular, we describe the notion of the latent graph and introduce the latent graph prediction as a pretext learning task. We adapt the supervised objective of latent graph prediction into a self-supervised setting by deriving its self-supervised upper bounds, according to which we present the learning framework of *LaGraph*. We provide further justifications of *LaGraph* by comparing it with theoretically sound methods in different domains. Our experimental results demonstrate the effectiveness of *LaGraph* on both graph-level and node-level representation learning, where a remarkable performance boost is achieved on a majority of datasets with higher stability to smaller batch sizes or training on subsets of nodes. Our code is available under the DIG library [*] [84].

Relations with Prior Work: Both *LaGraph* and some existing contrastive methods [75, 74, 76] apply node masking. While those contrastive methods use node masking as an augmentation to obtain different views for contrast, *LaGraph* employs it for the computation of the invariance term in its predictive objective. In addition, the objective of BGRL has a similar formulation to the invariance regularization term in our objective. The objectives of LaGraph and BGRL are from different grounding and have essential differences in their computing and effects. While the objective of BGRL is designed and engineered as a variant of contrastive methods, the LaGraph objectives are derived as a whole from the latent graph prediction. Our derived theorems associated

[*] `https://github.com/divelab/DIG`.

with *LaGraph* objectives can explain the success of BGRL to some extent and provide guidance on better-adopting objectives related to the invariance regularization on graphs.

3.2 Methods

3.2.1 Notations and Problem Formulation

We consider an undirected graph $G = (V, E)$ with a set of attributed nodes V and a set of edges E. We formulate the graph data as a tuple of matrices $(\boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{A} \in \mathbb{R}^{|V| \times |V|}$ denotes the adjacency matrix and $\boldsymbol{X} \in \mathbb{R}^{|V| \times d}$ denotes the node features of dimension d. We employ a graph encoder $\mathcal{E}$ based on graph neural networks (GNNs) to encode each node or graph into a corresponding representation. Namely, we compute the node-level representations or node embedding by $\boldsymbol{H} = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X}) \in \mathbb{R}^{|V| \times q}$ and the graph-level representation or graph embedding by $\boldsymbol{z} = \mathcal{R}(\boldsymbol{H}) \in \mathbb{R}^{1 \times q}$, where q denotes the embedding dimension and $\mathcal{R} : \mathbb{R}^{|V| \times q} \to \mathbb{R}^{1 \times q}$ is a readout function.

Self-supervised representation learning is employed to train the graph encoder $\mathcal{E}$ on a set of K graphs $\{G_i\}_{i=1}^{K}$ without labels from downstream tasks. In particular, we seek to design effective pre-text learning tasks, whose labels are obtained by task designation or from given data, to train the graph encoder $\mathcal{E}$ and produce informative representations for downstream tasks. Depending on the pre-text learning tasks, the encoder $\mathcal{E}$ is usually trained together with some prediction head $\mathcal{D}$ for predictive SSL or a discriminator for contrastive SSL.

3.2.2 Latent Graph Prediction

Our method considers latent graph prediction as a pretext task to train graph neural networks. In this subsection, we introduce the general notion of latent data, followed by its specific definition for graph data, and the construction of the learning task. For any observed data instance $\boldsymbol{x}$, we assume that there exists a corresponding latent data $\boldsymbol{x}_{\mathcal{I}}$, determining the semantic of $\boldsymbol{x}$, such that the latent data $\boldsymbol{x}_{\mathcal{I}}$ is generated from a prior $p(\boldsymbol{x}_{\mathcal{I}})$ and the observed data instance is further generated from a certain distribution conditioned on the latent data, *i.e.*, $p(\boldsymbol{x}|\boldsymbol{x}_{\mathcal{I}})$. The most common case for the pair of observed data and latent data is the noisy data and its clean version.

When it comes to graph data, we consider the case that an observed graph data $G = (\boldsymbol{A}, \boldsymbol{X})$ is (noisily) generated from its latent graph $G_\ell = (\boldsymbol{A}, \boldsymbol{F})$ with the same node set and edge set, where node feature matrices $\boldsymbol{X}$ and $\boldsymbol{F}$ for the two graphs have the same dimensionality. We make two assumptions about the graphs without loss of generality. First, we assume that the observed feature vector $\boldsymbol{x}_v$ of each node v in an observed graph is independently generated from a certain distribution conditioned on the corresponding latent graph. In other words, how $\boldsymbol{x}_v$ is generated from the latent feature $\boldsymbol{f}_v$ is not affected by the generation of other observed feature vectors. Second, we assume that the conditional distribution of the observed graph is centered at the latent graph, *i.e.*, $\mathbb{E}[\boldsymbol{X}|G_\ell] = \boldsymbol{F}$. The above assumptions are natural when we have little knowledge about the generation process and are commonly used in other types of data such as the non-structural and zero-mean noise in images. In cases where the generation processes of different nodes are related or the distribution is not centered at F, we can still consider the related or biased components into the latent feature and therefore have the assumptions satisfied.

As the latent data usually determine the semantic meaning of observed data, we believe the prediction of the latent graph can provide informative supervision for the learning of both graph-level and node-level representations. We are hence interested in constructing the learning task of latent graph prediction. To perform latent graph prediction, it is straightforward to employ a graph neural network $f : \{0, 1\}^{|V|\times|V|} \times \mathbb{R}^{|V|\times d} \to \mathbb{R}^{|V|\times d}$ that takes an observed graph $G = (\boldsymbol{A}, \boldsymbol{X})$ as inputs and predicts the feature matrix of its latent graph $G_{\mathcal{I}} = (\boldsymbol{A}, \boldsymbol{F})$. When the ground truth of the latent feature matrix $\boldsymbol{F}$ is known, the learning objective can be designed as

$$f^* = \arg\min_f \mathbb{E} \left\| f(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{F} \right\|^2 . \tag{3.1}$$

Intuitively, the latent graph prediction can be considered as a generalized task from noisy data reconstruction that predicts the signal from the noisy data with the objective $\arg\min_f \mathbb{E} \left\| f(\boldsymbol{x}) - \boldsymbol{s} \right\|^2$, where the mapping from the signal to the noisy data $p(\boldsymbol{x}|\boldsymbol{s})$ can usually be explicitly modeled and samples of signal (ground truth) can usually be captured. In the data reconstruction case, pairs of $(\boldsymbol{x}, \boldsymbol{s})$ can be therefore directly captured or synthetically generated given a certain noise model $p(\boldsymbol{x}|\boldsymbol{s})$. However, when the task is generalized to latent graph prediction, there is a key challenge preventing us from directly applying the prediction task. That is, whereas there are natural supervisions for noisy data reconstruction, the latent graph is not observed and we are unable to explicitly model the mapping from latent graphs to observed graphs, *i.e.*, the conditional distribution $p(G|G_{\mathcal{I}})$.

3.2.3 Self-Supervised Upper Bounds for Latent Graph Prediction

As discussed in the previous subsection, unlike typical noisy data reconstruction tasks, the latent graph is not observed and $p(G|G_{\mathcal{I}})$ cannot be modeled explicitly. This makes it difficult to construct a direct learning task for latent graph prediction using the objective in Equation (3.1). We therefore seek to optimize an alternative objective that approximately optimizes the objective in Equation (3.1) without requiring the distribution $p(G|G_{\mathcal{I}})$, nor features $\boldsymbol{F}$ of the latent graph. We now introduce the proposed self-supervised objective for latent graph prediction.

We derive our self-supervised objective without involving $\boldsymbol{F}$ by constructing an upper bound of the objective in Equation (3.1). Specifically, we let $J \subset \{0, \cdots, |V|-1\}$ be an arbitrary subset of node indices, J^c denote the complement of set J, and $\boldsymbol{X}_{J^c} := \mathbb{1}_{J^c} \odot \boldsymbol{X} + \mathbb{1}_J \odot \boldsymbol{M}$ be the feature matrix with features of nodes in V_J masked, where $\odot$ denotes element-wise multiplication, $\boldsymbol{M} \in \mathbb{R}^{|V| \times d}$ denotes a matrix consisting of independent random noise or zeros as masking values, and $\mathbb{1}_J \in \mathbb{R}^{|V| \times d}$ denotes an indicator matrix such that $\mathbb{1}_J[i,:] = \mathbf{1}, \forall i \in J$ and $\mathbb{1}_J[i,:] = \mathbf{0}, \forall i \notin J$. We describe the self-supervised upper bound in Theorem 3.

Theorem 3. *Consider a graph $G = (\boldsymbol{A}, \boldsymbol{X})$ and its latent graph $G_{\mathcal{I}} = (\boldsymbol{A}, \boldsymbol{F})$. We let the variance of any elements in $\boldsymbol{X}$ be bounded by σ^2 and J be a subset of nodes V in the graph G. For any* *graph neural network $f : \{0,1\}^{|V|\times|V|} \times \mathbb{R}^{|V|\times d} \to \mathbb{R}^{|V|\times d}$, we have the following inequality*

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{A},\boldsymbol{X},\boldsymbol{F}}\left[\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{F}\|^2+\|\boldsymbol{X}-\boldsymbol{F}\|^2\right] \\
&\qquad\le \mathbb{E}_{\boldsymbol{A},\boldsymbol{X}}\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{X}\|^2+ \\
2\sigma|V|\,&\mathbb{E}_J\left[\frac{\mathbb{E}_{\boldsymbol{A},\boldsymbol{X}}\|f_J(\boldsymbol{A},\boldsymbol{X})-f_J(\boldsymbol{A},\boldsymbol{X}_{J^c})\|^2}{|J|}\right]^{1/2}.
\end{aligned} \tag{3.2}$$

Intuitively, the first component in the upper bound derived in Theorem 3 measures the reconstruction error on the feature matrix $\boldsymbol{X}$ of the given observed graph G, enforcing the intermediate representations to be informative. The second component controls how much information is accessible from the input feature of a node v_i when reconstructing the feature of v_i, by encouraging the output of a node to be invariant to the missing of its features in the input graph. We then call the first component a reconstruction term and the second component an invariance regularization term. Note that the invariance regularization is only computed on masked nodes in contrast to the BGRL objective, based on different theoretical grounding and leading to a different effect. A more detailed discussion is provided in Section 3.

In tasks of self-supervised representation learning, we are more interested in graph-level or node-level representations than predicted latent graphs. In these cases, we expect the representations also hold the invariance property held by the final outputs. We, therefore, seek to apply the invariance regularization to the representations, since a regularization applied to the output does not necessarily control the information accessibility of representations produced intermediately in the graph neural network. To do so, we separately consider the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ in the graph neural network f. We introduce certain assumptions to the decoder network $\mathcal{D}$ and the readout function $\mathcal{R}$, and derive two additional upper bounds for node-level and graph-level representation learning, respectively in the following corollaries.

Corollary 1. *Let $G = (\boldsymbol{A},\boldsymbol{X})$ be a given graph, $G_{\mathcal{I}} = (\boldsymbol{A},\boldsymbol{F})$ be its latent graph, $\mathcal{E}$ and $\mathcal{D}$ be a graph encoder and a prediction head (decoder) consisting of fully-connected layers. If the prediction head $\mathcal{D}$ is ℓ-Lipschitz continuous with respect to l_2-norm, we further have the following* *inequality,*

$$\begin{aligned}\mathbb{E}\big[\, \|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{F}\|^2 + \|\boldsymbol{X} - \boldsymbol{F}\|^2 \,\big] &\leq \mathbb{E}\, \|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{X}\|^2 \\ &+ 2\sigma |V| \ell \, \mathbb{E}_J \left[\frac{\mathbb{E}\, \|\boldsymbol{H}_J - \boldsymbol{H}'_J\|^2}{|J|} \right]^{1/2}, \end{aligned} \tag{3.3}$$

where $\boldsymbol{H} = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X})$ and $\boldsymbol{H}' = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X}_{J^c})$ denote the node embedding of the given graph and the masked graph, respectively, and $\boldsymbol{H}_J := \boldsymbol{H}[J, :]$ selects rows with indices in J.

Corollary 2. *Let $G = (\boldsymbol{A}, \boldsymbol{X})$ be a given graph, $G_{\mathcal{I}} = (\boldsymbol{A}, \boldsymbol{F})$ be its hidden latent graph, $\mathcal{E}$ be a graph encoder, $\mathcal{R}$ be a readout function satisfying k-Bilipschitz continuity with respect to l_2-norm, and $\mathcal{D}$ be a prediction head (decoder). If the prediction head $\mathcal{D}$ is ℓ-Lipschitz continuous with respect to l_2-norm, we have the following inequality,*

$$\begin{aligned}\mathbb{E}\big[\, \|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{F}\|^2 + \|\boldsymbol{X} - \boldsymbol{F}\|^2 \,\big] &\leq \mathbb{E}\, \|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{X}\|^2 \\ &+ 2\sigma |V| k \ell \, \mathbb{E}_J \left[\frac{\mathbb{E}\, \|\boldsymbol{z} - \boldsymbol{z}'\|^2}{|J|} \right]^{1/2}, \end{aligned} \tag{3.4}$$

where $\boldsymbol{z} = \mathcal{R}(\boldsymbol{H})$ and $\boldsymbol{z}' = \mathcal{R}(\boldsymbol{H}')$ denote the graph-level representations of the given graph and the masked graph, respectively.

Proof. We first prove Corollary 1. Consider an ℓ-Lipschitz continuous prediction head with respect to l_2-norm consists of fully connected layers. We have

$$\|f_J(\boldsymbol{A}, \boldsymbol{X}) - f_J(\boldsymbol{A}, \boldsymbol{X}_{J^c})\|_2 = \|\mathcal{D}(\boldsymbol{H}_J) - \mathcal{D}(\boldsymbol{H}'_J)\|_2 \leq \ell \, \|\boldsymbol{H}_J - \boldsymbol{H}'_J\|_2 \,. \tag{3.5}$$

We therefore have the following inequality

$$\mathbb{E}\, \|f_J(\boldsymbol{A}, \boldsymbol{X}) - f_J(\boldsymbol{A}, \boldsymbol{X}_{J^c})\|_2^2 \leq \mathbb{E}\Big[\ell^2 \, \|\boldsymbol{H}_J - \boldsymbol{H}'_J\|_2^2 \Big]. \tag{3.6}$$

We apply the above inequality to Theorem 1 and obtain the following inequality

$$
\begin{aligned}
&\mathbb{E}\big[\, \|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{F}\|^2 + \|\boldsymbol{X}-\boldsymbol{F}\|^2 \,\big] \\
&\qquad \le \mathbb{E}\,\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{X}\|^2 + 2\sigma|V|\mathbb{E}_J\bigg(\frac{1}{|J|}\mathbb{E}\,\|f_J(\boldsymbol{A},\boldsymbol{X})-f_J(\boldsymbol{A},\boldsymbol{X}_{J^c})\|^2\bigg)^{1/2} && (3.7)\\
&\qquad \le \mathbb{E}\,\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{X}\|^2 + 2\sigma|V|\mathbb{E}_J\bigg(\frac{1}{|J|}\mathbb{E}\Big[\ell^2\,\|\boldsymbol{H}_J-\boldsymbol{H}'_J\|_2^2\Big]\bigg)^{1/2} && (3.8)\\
&\qquad = \mathbb{E}\,\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{X}\|^2 + 2\sigma|V|\ell\mathbb{E}_J\bigg(\mathbb{E}\,\|\boldsymbol{H}_J-\boldsymbol{H}'_J\|_2^2/|J|\bigg)^{1/2}, && (3.9)
\end{aligned}
$$

which completes the proof of Corollay 1.

Similarly, for Corollay 2, we have

$$
\|f_J(\boldsymbol{A},\boldsymbol{X})-f_J(\boldsymbol{A},\boldsymbol{X}_{J^c})\|_2 = \|\mathcal{D}(\boldsymbol{H}_J)-\mathcal{D}(\boldsymbol{H}'_J)\|_2 \le \ell\,\|\boldsymbol{H}_J-\boldsymbol{H}'_J\|_2\,. \tag{3.10}
$$

Given an ℓ_r-Bilipschitz continuous readout function $\mathcal{R}$, the following inequalities hold,

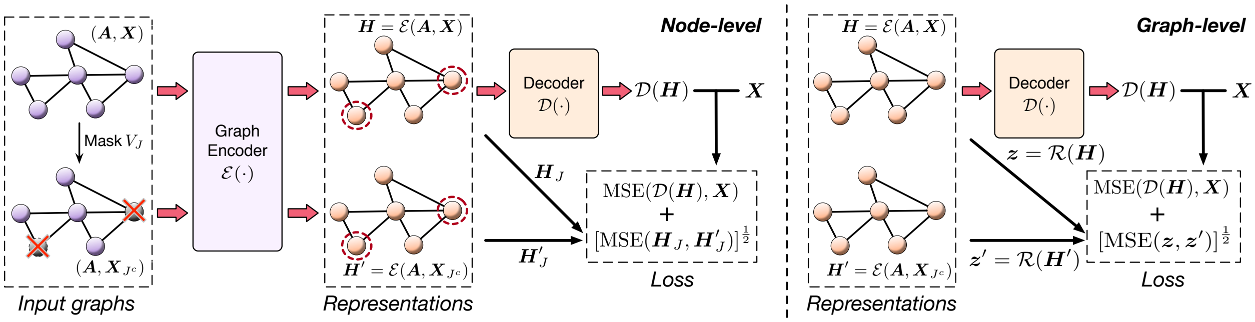
$$
\frac{1}{\ell_r}\,\|\boldsymbol{H}_J-\boldsymbol{H}'_J\|_2 \le \|\mathcal{R}(\boldsymbol{H}_J)-\mathcal{R}(\boldsymbol{H}'_J)\|_2 \le \ell_r\,\|\boldsymbol{H}_J-\boldsymbol{H}'_J\|_2\,. \tag{3.11}
$$

We therefore have

$$
\begin{aligned}
&\mathbb{E}\big[\, \|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{F}\|^2 + \|\boldsymbol{X}-\boldsymbol{F}\|^2 \,\big] \\
&\qquad \le \mathbb{E}\,\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{X}\|^2 + 2\sigma|V|\ell\mathbb{E}_J\bigg(\mathbb{E}\,\|\boldsymbol{H}_J-\boldsymbol{H}'_J\|_2^2/|J|\bigg)^{1/2} && (3.12)\\
&\qquad \le \mathbb{E}\,\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{X}\|^2 + 2\sigma|V|\ell\ell_r\mathbb{E}_J\bigg(\mathbb{E}\,\|\mathcal{R}(\boldsymbol{H}_J)-\mathcal{R}(\boldsymbol{H}'_J)\|_2^2/|J|\bigg)^{1/2} && (3.13)\\
&\qquad = \mathbb{E}\,\|f(\boldsymbol{A},\boldsymbol{X})-\boldsymbol{X}\|^2 + 2\sigma|V|k\ell\mathbb{E}_J\bigg(\mathbb{E}\,\|\boldsymbol{z}-\boldsymbol{z}'\|_2^2/|J|\bigg)^{1/2}, && (3.14)
\end{aligned}
$$

which completes the proof of Corollay 2. □

We note that the assumptions and restrictions are natural or practically satisfiable. The assumption that the variance of each element in $\boldsymbol{X}$ is bounded by σ holds when node features are from



Node-level
Graph-level
$(\boldsymbol{A}, \boldsymbol{X})$
Mask V_J
$(\boldsymbol{A}, \boldsymbol{X}_{J^c})$
Input graphs
Graph Encoder $\mathcal{E}(\cdot)$
$\boldsymbol{H} = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X})$
$\boldsymbol{H}' = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X}_{J^c})$
Representations
Decoder $\mathcal{D}(\cdot)$
$\mathcal{D}(\boldsymbol{H})$
$\boldsymbol{X}$
$\boldsymbol{H}_J$
$\boldsymbol{H}'_J$
$\text{MSE}(\mathcal{D}(\boldsymbol{H}), \boldsymbol{X})$ + $[\text{MSE}(\boldsymbol{H}_J, \boldsymbol{H}'_J)]^{\frac{1}{2}}$
Loss
$\boldsymbol{H} = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X})$
$\boldsymbol{H}' = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X}_{J^c})$
Representations
Decoder $\mathcal{D}(\cdot)$
$\mathcal{D}(\boldsymbol{H})$
$\boldsymbol{X}$
$\boldsymbol{z} = \mathcal{R}(\boldsymbol{H})$
$\boldsymbol{z}' = \mathcal{R}(\boldsymbol{H}')$
$\text{MSE}(\mathcal{D}(\boldsymbol{H}), \boldsymbol{X})$ + $[\text{MSE}(\boldsymbol{z}, \boldsymbol{z}')]^{\frac{1}{2}}$
Loss

Figure 3.1: Overview of the *LaGraph* framework. Given a training graph, we randomly mask a small portion $V_J \in V$ of its nodes and input both the original graph and masked graph to the encoder $\mathcal{E}$. Crossed nodes in the figure have all their attributes masked but topology preserved. The final loss consists of a reconstruction loss on node features and an invariance loss between representations of the original graph and the masked graph. We omit the encoding part of the graph-level framework as frameworks for the two levels mainly differ in whether the invariance term is computed on representations of masked nodes or graph-level representations obtained by $\mathcal{R}$.

$\{0, 1\}^d$ or when feature normalization is applied. The ℓ-Lipschitz continuous property is common for neural networks. And the k-Bilipschitz continuity can be satisfied by applying an injective readout function such as global sum pooling, which is commonly used in graph-level tasks.

3.2.4 The *LaGraph* Framework

We design our self-supervised learning framework according to upper bounds derived in Corollary 1 and Corollary 2. To train encoder $\mathcal{E}$ together with decoder $\mathcal{D}$ under self-supervision, we input to the encoder both the given graph $(\boldsymbol{A}, \boldsymbol{X})$ and its variation $(\boldsymbol{A}, \boldsymbol{X}_{J^c})$ with a random subset J of node indices for nodes to be masked and obtain node-level representations $\boldsymbol{H} = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X})$ and $\boldsymbol{H}' = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X}_{J^c})$ for the two graphs respectively. The self-supervised losses are computed on input node features, reconstructed node features, and representations, as demonstrated in Figure 3.1.

In particular, we consider a mini-batch of N graphs $\{(\boldsymbol{A}_i, \boldsymbol{X}_i)\}_{i=1}^N$ and their corresponding masked variation $\{(\boldsymbol{A}_i, \boldsymbol{X}_{(i, {J_i}^c)})\}_{i=1}^N$ where J_i denotes the node indices subset for the i-th graph. The self-supervised loss for node-level representation learning follows Corollary 1 and is computed as

$$
\begin{aligned}
L_{node}(\mathcal{E},\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} \left\|\mathcal{D}(\boldsymbol{A}_i,\boldsymbol{H}_i) - \boldsymbol{X}_i\right\|^2/|V_i| \\
+\alpha\left[\frac{\sum_i \left\|\mathbb{1}_{J_i}\odot\boldsymbol{H}_i - \mathbb{1}_{J_i}\odot\boldsymbol{H}_i'\right\|^2}{\sum_i |J_i|}\right]^{1/2},
\end{aligned}
\tag{3.15}
$$

where α is a hyper-parameter corresponding to the multiplier $2\sigma\ell$ in Corollary 1. To fulfill the conditions in Corollary 1, we employ fully-connected layers instead of graph convolutional layers in the decoder $\mathcal{D}$.

Similarly, using the same notations above, the self-supervised loss for graph-level representation learning follows Corollary 2 and is computed as

$$
\begin{aligned}
L_{graph}(\mathcal{E},\mathcal{D}) =& \frac{1}{N}\sum_{i=1}^{N} \left\|\mathcal{D}(\boldsymbol{A}_i,\boldsymbol{H}_i) - \boldsymbol{X}_i\right\|^2/|V_i| \\
&+\alpha'\left[\sum_i \left\|\boldsymbol{z}_i - \boldsymbol{z}_i'\right\|^2/\sum_i |J_i|\right]^{1/2},
\end{aligned}
\tag{3.16}
$$

where $\boldsymbol{z}_i = \mathcal{R}(\boldsymbol{H}_i)$ and $\boldsymbol{z}_i' = \mathcal{R}(\boldsymbol{H}_i')$ denote the graph-level representations obtained by applying readout function $\mathcal{R}$ to the node-level representations, respectively, and α' is a hyper-parameter corresponding to the multiplier $2\sigma k\ell$ in Corollary 2. To fulfill the conditions in Corollary 2, we employ global sum pooling as the readout function $\mathcal{R}$, where as the decoder $\mathcal{D}$ here can consists of either fully-connected layers or graph convolutional layers.

3.3 Theoretical Analysis and Relations with Prior Work

In this section, we further theoretically justify and motivate *LaGraph* by providing comparisons and connections between our method and existing related methods, including denoising autoencoders [63, 85], information bottleneck principle [46], and contrastive methods based on local-global mutual information maximization [40, 86, 12]. We also discuss the relation and difference to BGRL [3] and Barlow-Twin [87].

3.3.1 Denoising Autoencoders

Denoising autoencoders employ an encoder-decoder network architecture and perform self-supervised training by masking or corrupting a portion of dimensions of the given data and reconstructing the masked or corrupted value given their context. Such an approach has been also applied for self-supervised image denoising [2], known as blind-spot denoising. Similar to our method, the denoising autoencoder can be also viewed as an approximation of the latent graph prediction. Using the same notation in Section 3.2, we formulate the connection between latent graph prediction and the graph denoising autoencoder in the following theorem.

Theorem 4. *Let J be a uniformly sampled subset of node indices of the given graph $(\boldsymbol{A}, \boldsymbol{X})$, $\mathcal{F}$ be the class of all graph neural networks, and $\mathcal{F}^*$ be the class of graph neural networks such that $f^*_J(\boldsymbol{A}, \boldsymbol{X})$ does not depend on $\boldsymbol{X}_J$, for any J and $f^* \in \mathcal{F}^*$. Given any graph neural network $f \in \mathcal{F}$, there exist $f^* \in \mathcal{F}^*$ and $f' \in \mathcal{F}$ such that*

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{A},\boldsymbol{X},\boldsymbol{F}}\left[\|f(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{F}\|^2 + \|\boldsymbol{X} - \boldsymbol{F}\|^2\right] && (3.17)\\
=&\mathbb{E}_{\boldsymbol{A},\boldsymbol{X}}\|f(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{X}\|^2 + \\
&\qquad \mathbb{E}_{\boldsymbol{A},\boldsymbol{X},\boldsymbol{F}}\left[2\langle f(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{F}, \boldsymbol{X} - \boldsymbol{F}\rangle\right] && (3.18)\\
\approx&\mathbb{E}_{\boldsymbol{A},\boldsymbol{X}}\|f^*(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{X}\|^2 && (3.19)\\
=&|V|\mathbb{E}_J\mathbb{E}_{\boldsymbol{A},\boldsymbol{X}}\|f'_J(\boldsymbol{A}, \boldsymbol{X}_{J^c}) - \boldsymbol{X}_J\|^2/|J|. && (3.20)
\end{aligned}$$

Equation (7) is proved in the proof of Theorem 1. It can be verified that the second term, *i.e.*, the expectation of the inner product, in Equation (7) reduces to zero when the neural network f satisfies that $f_J(\boldsymbol{A}, \boldsymbol{X})$ does not depend on $\boldsymbol{X}_J$, for any J, according to Batson and Royer [2]. The objective can be therefore approximated by Equation (8) with the neural network f^* satisfying such a property. To let any graph neural network f satisfy the property, one can apply masks to a portion of nodes indexed by J so that their original value is inaccessible by f when predicting $f_J(\boldsymbol{A}, \boldsymbol{X})$. Therefore, the latent graph prediction objective under supervision can be further approximated by

Equation (9), which describes the objective of a graph denoising autoencoder.

A substantial difference between our method and the denoising autoencoder lies in how to handle the inner product term in Equation (7). In particular, the denoising autoencoder forces the term to be zero by assuming certain properties of the graph neural network, whereas our method derives an upper bound, *i.e.*, the invariance term, for the inner product. Theoretically, the graph denoising autoencoder is equivalent to our framework with an infinite weight scalar for the invariance term. As a drawback, when $f_J(\boldsymbol{A}, \boldsymbol{X})$ does not depend on $\boldsymbol{X}_J$, the learned representations can be less informative as representations of nodes in V_J do not include the information of $\boldsymbol{X}_J$, for any J, leading to performance loss. Our proposed upper bounds allow an encoder to access a certain level of information of the masked nodes, whose representations can be as good as ones from supervised learning. In fact, our method can be viewed as an autoencoder with an invariance regularization.

3.3.2 The Information Bottleneck Principle

The information bottleneck principle [46] is a technique for data compression and signal processing in the field of information theory, and has been widely applied in deep learning problems [47, 88]. Let $\boldsymbol{X}$ be a random variable to be compressed, $\tilde{\boldsymbol{X}}$ be an observed relevant variable, and $\boldsymbol{Z}$ denote the compressed representation of $\boldsymbol{X}$. The information bottleneck principle seeks to optimize the following problem

$$\boldsymbol{T}^* = \arg\min_{\boldsymbol{T}} I(\boldsymbol{T}; \boldsymbol{Y}) - \beta I(\boldsymbol{T}; \boldsymbol{X}), \tag{3.21}$$

where $I(\cdot;\cdot)$ denotes the mutual information and $\beta > 1$ is a Lagrange multiplier. The work Barlow Twin [87] has discussed a connection between the information bottleneck principle and self-supervised learning. In particular, to apply information bottleneck to SSL, one usually obtain $\tilde{\boldsymbol{X}}$ by performing augmentations or distortions on the given data $\boldsymbol{X}$. And Equation (3.21) can be rewritten into

$$\boldsymbol{T}^* = \arg\min_{\boldsymbol{T}} \left[H(\boldsymbol{T}) - H(\boldsymbol{T}|\boldsymbol{Y}) \right] \tag{3.22}$$

$$- \beta\big[H(\boldsymbol{T}) - H(\boldsymbol{T}|\boldsymbol{X})\big] \tag{3.23}$$

$$= \arg\min_{\boldsymbol{T}} H(\boldsymbol{T}|\boldsymbol{X}) - \lambda H(\boldsymbol{T}), \tag{3.24}$$

where $\lambda = \frac{\beta-1}{\beta} > 0$ is a weight scalar. Intuitively, the conditional entropy $H(\boldsymbol{Z}|\boldsymbol{X})$ is to be minimized, indicating that the distortion should add no additional information to the representation $\boldsymbol{Z}$. In other words, the representation $\boldsymbol{Z}$ should be as invariant as possible to distortions applied to $\boldsymbol{X}$. In addition, the entropy $H(\boldsymbol{Z})$ is to be maximized, indicating that the representation $\boldsymbol{Z}$ itself should be as informative as possible.

The two terms in objectives of *LaGraph* correspond to the terms in Equation (3.24). In particular, the invariance term corresponding to $H(\boldsymbol{Z}|\boldsymbol{X})$ and the reconstruction term aims to ensure informative representations, *i.e.*, to maximize $H(\boldsymbol{Z})$. Objectives in existing SSL methods such as BYOL [22], its variation BGRL [3] in graph domain, and Barlow Twin [87] also include invariance terms corresponding to $H(\boldsymbol{Z}|\boldsymbol{X})$. To encourage informative representations, Barlow Twin further includes a redundancy reduction term to minimize the cross-correlation between different dimensions of the representation, as a proxy of the maximization of $H(\boldsymbol{Z})$. In addition, the InfoNCE (NT-XENT) loss employed in some contrastive learning methods [75, 74] induces a similar effect, according to Zbontar et al. [87]. Both Equation (3.24) and the derivation of *LaGraph* objectives indicate the importance of the invariance term in SSL objectives. In addition, compared to the redundancy reduction term in Barlow Twin and the noise contrast in InfoNCE, *LaGraph* objectives can directly guarantee the learning of informative representations measured by the reconstruction capability.

3.3.3 Contrastive Learning by Maximizing Local-Global Mutual Information

Motivated by Deep InfoMax [89], recent graph self-supervised learning methods [40, 86, 12] constructs their learning tasks by maximizing the mutual information between local (node-level) representations and a global (graph-level) summary of the graph. Practically, as a k-layer encoder $\mathcal{E}$ has the receptive field of at most k-hop neighborhood, the goal becomes the maximization of the mutual information between local representations and their k-hop neighborhood, formulated as

$$\mathcal{E}^* = \arg\max_{\mathcal{E}} \sum_{i=1}^{|V|} I(\boldsymbol{X}_i^{(k)}; \mathcal{E}_i(\boldsymbol{A}, \boldsymbol{X})), \tag{3.25}$$

where I denotes the mutual information, $\boldsymbol{X}_i^{(k)}$ is the k-hop neighborhood of node i, $\mathcal{E}$ is a graph encoder with k GNN layers, and $\mathcal{E}_i(\boldsymbol{A}, \boldsymbol{X})$ denotes the local representation of node i. The learning objective is motivated by the goal that the local representations should contain as much the global information of the entire graph (or the k-hop neighborhood) as possible.

As for *LaGraph*, the reconstruction term encourages representations to contain sufficient information to reconstruct the input features while the invariance term limits the information accessibility from a local node when reconstructing its features. The two terms in the objective jointly promote node representations to learn limited local information and as much contextual information from the neighborhood as possible for reconstruction. It hence has a similar effect to the local-global mutual information maximization.

3.3.4 Other Invariance-Based Objectives

Recent self-supervised learning objectives such as BGRL, Barlow-Twin, and the consistency regularization [90] have similar invariance terms as one in the LaGraph objective. Specifically, BGRL minimizes the difference between representations of two augmented views. In spite of the similarity, the invariance terms in LaGraph and other objectives have different grounding and effects.

Regarding how the objectives are computed, the invariance term in the LaGraph objective for node-level representation learning is computed only on masked nodes, in contrast to BGRL and Barlow-Twins objectives where invariance of all nodes are computed. It is worth noting that the proposed objective is an upper bound to the latent graph prediction only if the invariance is computed on the masked nodes, according to the derivation in the proof of Theorem 1. Intuitively, during the computation of a node representation, the invariance term in LaGraph enforces the encoder to capture less information from the node itself and more contextual information. Computing the invariance regularization term on unmasked nodes could lead to a contradicted effect, i.e., discouraging encoders to capture information from contextual nodes, as it lets the representation remain consistent when its masked neighbor nodes are changed. We believe the derivation and the intuition of the proposed objective can provide insights on adopting the invariance regularization into graph self-supervised learning studies.

3.4 Experiments

We conduct experiments on both node-level and graph-level self-supervised representation learning tasks with datasets used in two most recent state-of-the-art methods for SSL [75, 3]. For graph-level tasks, we follow GraphCL [75] to perform evaluations on eight graph classification datasets [91, 92, 93, 94, 95] from TUDataset [96]. For node-level tasks, as the citation network datasets [97, 98, 99] are recognized to be saturated and unreliable for GNN evaluation [100, 3], we follow Thakoor et al. [3] to include four transductive node classification datasets from Shchur et al. [100], including Amazon Computers, Amazon Photos from the Amazon Co-purchase Graph [101], Coauthor CS, and Coauthor Physics from the Microsoft Academic Graph [102]. We further include three larger-scale inductive datasets, PPI, Reddit, and Flickr, for node-level classification used in SUBG-CON [4].

We follow You et al. [75] and Zhu et al. [74] for the standard linear evaluation protocols at graph-level and node-level, respectively. In particular, for both levels, we first train the graph encoder on unlabeled graph datasets with the corresponding self-supervised objective. We then compute and freeze the corresponding representations and train a linear classification model on top of the fixed representations with their corresponding labels. Linear SVM and the regularized logistic regression are employed as linear classifiers for graph-level datasets and node-level datasets, according to You et al. [75] and Zhu et al. [74], respectively. For inductive node-level datasets, the self-supervised training is only performed on graphs in the training datasets whereas the test graphs are unavailable during the self-supervised training.

Table 3.1: Performance on graph-level classification tasks, scores are averaged over 5 runs. Bold and underlined numbers highlight the top-2 performance. OOM indicates running out-of-memory on a 56GB Nvidia A6000 GPU.

	NCI1	PROTEINS	DD	MUTAG	COLLAB	RDT-B	RDT-M5K	IMDB-B
GL	–	–	–	81.7±2.1	–	77.3±0.2	41.0±0.2	65.9±1.0
WL	80.0±0.5	72.9±0.6	–	80.7±3.0	–	68.8±0.4	46.1±0.2	72.3±3.4
DGK	80.3±0.5	73.3±0.8	–	87.4±2.7	–	78.0±0.4	41.3±0.2	67.0±0.6
Node2Vec	54.9±1.6	57.5±3.6	75.1±0.5	72.6±10.2	55.7±0.2	73.8±0.5	34.1±0.4	50.0±0.8
Sub2Vec	52.8±1.5	53.0±5.6	73.6±1.5	61.1±15.8	62.1±1.4	71.5±0.4	36.7±0.4	55.3±1.5
Graph2Vec	73.2±1.8	73.3±2.1	76.2±0.1	83.2±9.3	59.9±0.0	75.8±1.0	47.9±0.3	71.1±0.5
GAE	73.3±0.6	74.1±0.5	77.9±0.5	84.0±0.6	56.3±0.1	74.8±0.2	37.6±1.6	52.1±0.2
VGAE	73.7±0.3	74.0±0.5	77.6±0.4	84.4±0.6	56.3±0.0	74.8±0.2	39.1±1.6	52.1±0.2
InfoGraph	76.2±1.1	74.4±0.3	72.9±1.8	89.0±1.1	70.7±1.1	82.5±1.4	53.5±1.0	73.0±0.9
GraphCL	77.9±0.4	74.4±0.5	**78.6±0.4**	86.8±1.3	71.4±1.2	89.5±0.8	56.0±0.3	71.1±0.4
MVGRL	75.1±0.5	71.5±0.3	OOM	89.7±1.1	OOM	84.5±0.6	OOM	**74.2±0.7**
LaGraph	**79.9±0.5**	**75.2±0.4**	78.1±0.4	**90.2±1.1**	**77.6±0.2**	**90.4±0.8**	**56.4±0.4**	73.7±0.9

3.4.1 Comparisons with Baselines

We perform experiments on both graph-level and node-level datasets to demonstrate the effectiveness of *LaGraph*. We construct our model and losses according to Section 3.2.4.

Graph-level Datasets. We evaluate the performance of *LaGraph* in terms of the linear classification accuracy and compare it with three kernel-based methods including graphlet kernel (GL) [103], Weisfeiler-Lehman kernel (WL) [104], and deep graph kernel (DGK) [95], together with five unsupervised methods including Node2Vec [71], Sub2Vec [105], Graph2Vec [17], GAE and VGAE [41]. We further compare the results with recent SOTA SSL methods based on contrastive learning, including InfoGraph [86] , MVGRL [12], and GraphCL [75]. Results in Table 3.1 show that *LaGraph* outperforms the current SOTA methods on a majority of datasets and is on par with the best performance on the rest of datasets.

Node-level Datasets. We perform node-level experiments on both transductive and inductive learning tasks. Transductive self-supervised learning of node representation allows utilization of all data at hand to pre-train GNNs for downstream tasks. Although labels of nodes are not visible during pre-training, patterns and information present in all nodes are observed. In contrast to transductive learning, inductive self-supervised learning only allows using a portion of data to pre-train GNNs, while holding out a certain amount of data for downstream tasks. Our inductive tasks include two cases. First, the PPI dataset consists of 24 graphs, and the training and testing nodes are split by graphs. In this case, the inductive task is considered across multiple graphs. In other words, node representations are learned from training graphs, and the encoder is evaluated on testing graphs. Second, Flickr and Reddit each consist of only one graph, the training and testing nodes are from the same graph. During self-supervised training, all test nodes are masked-out. During evaluation, all training nodes are masked-out, i.e., test nodes are unseen nodes of the graph during train. For both cases of inductive learning, data used during the self-supervised training stage and data used during evaluation stage are distinct, but the feature dimensionality should be the same for data used in both stages.

For the evaluation of transductive learning, we compare the performance of *LaGraph* in terms of linear classification accuracy with DeepWalk [106], GAE, VGAE, and six contrastive learning methods including Deep Graph InfoMax (DGI) [40], GMI [107], MVGRL [12], GRACE [74], GCA [45], and BGRL [3], where BGRL is the current state-of-the-art SSL method for node-level representation learning. We further include the results of directly performing linear classification on raw node features (raw features) and by supervised training for references. To be consistent with Thakoor et al. [3], we have ensured that the GPU memory consumption of *LaGraph* is under 16GB for the four transductive datasets. We then perform additional experiments on the larger-scale inductive datasets [108, 109, 81] and compare our results in terms of micro-averaged F1-score with DeepWalk, unsupervised GraphSAGE [81], DGI, GMI, SUBG-CON [4] and BGRL. Results for both transductive datasets and inductive datasets shown in Table 3.2. As there is no official BGRL implementation available at the time our experiments are conducted, results with $*$ are obtained from an unofficial public implementation[†]. Results suggest competitive performance of *LaGraph* compared to the existing SOTA methods. Moreover, LaGraph consumes even less memory than BGRL, which requires twice the memory for its GNN encoders for the EMA parameter update.

Experiment Environment Details. We train graph-level datasets on a 11GB GeForce RTX

[†] `https://github.com/namkyeong/bgrl_pytorch`.

Table 3.2: Performance on node-level datasets, 20 runs averaged. Results of SSL methods with the best performance are highlighted in bold numbers. *Left*: Mean classification accuracy on transductive datasets, with baseline results from Thakoor et al. [3]. *Right*: Micro-averaged F1 scores on larger-scale inductive datasets, with baseline results from Thakoor et al. [3] and Jiao et al. [4].

Transductive	Am.Comp.	Am.Pht.	Co.CS	Co.Phy
Raw features	73.8±0.0	78.5±0.0	90.4±0.0	93.6±0.0
DeepWalk	85.7±0.1	89.4±0.1	84.6±0.2	91.8±0.2
GAE	87.7±0.3	92.7±0.3	92.4±0.2	95.3±0.1
VGAE	88.1±0.3	92.8±0.3	92.5±0.2	95.3±0.1
Supervised	86.5±0.5	92.4±0.2	93.0±0.3	95.7±0.2
DGI	84.0±0.5	91.6±0.2	92.2±0.6	94.5±0.5
GMI	82.2±0.3	90.7±0.2	OOM	OOM
MVGRL	87.5±0.1	91.7±0.1	92.1±0.1	95.3±0.0
GRACE	87.5±0.2	92.2±0.2	92.9±0.0	95.3±0.0
GCA	88.9±0.2	92.5±0.2	93.1±0.0	95.7±0.0
BGRL	**89.7±0.3**	92.9±0.3	93.2±0.2	95.6±0.1
LaGraph	88.0±0.3	**93.5±0.4**	**93.3±0.2**	**95.8±0.1**

Inductive	PPI	Flickr	Reddit
Raw feat.	42.5±0.3	20.3±0.2	58.5±0.1
GAE	75.7±0.0	50.7±0.2	OOM
VGAE	75.8±0.0	50.4±0.2	OOM
Super-GCN	51.5±0.6	48.7±0.3	93.3±0.1
Super-GAT	97.3±0.2	OOM	OOM
GraphSAGE	46.5±0.7	36.5±1.0	90.8±1.1
DGI	63.8±0.2	42.9±0.1	94.0±0.1
GMI	65.0±0.0	44.5±0.2	95.0±0.0
SUBG-CON	66.9±0.2	48.8±0.1	**95.2±0.0**
BGRL-GCN	69.6±0.2	50.0±0.3*	OOM*
BGRL-GAT	70.5±0.1	44.2±0.1*	OOM*
LaGraph	**74.6±0.0**	**51.3±0.1**	**95.2±0.0**

2080 Ti GPU, and node-level datasets on a 56GB Nvidia RTX A6000 GPU. Our experiments are implemented with PyTorch 1.7.0 and PyTorch Geometric 1.7.0. All neural networks employ batch normalization [110], and are optimized with Adam optimizer [111]. We initialize GNNs with Xavier initialization [112].

3.4.2 Ablation Study

We further conduct three ablation studies to explore model robustness to smaller batch sizes on graph-level data and to the training with sub-graphs on large-scale node-level datasets.

Robustness to Batch Sizes. Different from contrastive learning methods, *LaGraph* does not require negative samples to perform noise contrast or pair-wise discrimination. Therefore, an advantage of *LaGraph* is that the performance is robust to the batch size as it does not depend on large batch sizes with sufficient negative samples. To verify the statement, we perform an ablation study on how model performance changes when decreasing the batch size from 128 to 8 for graph-level datasets. We include corresponding results of GraphCL which uses InfoNCE for references and show the comparisons in Figure 3.2. The results indicate while contrastive methods based on InfoNCE suffer from significant performance loss with a small batch size, *LaGraph* are more

Table 3.3: Model performance when trained on a subset of nodes.

	# nodes sampled	100	1,000	2,500	5,000	10,000	all
Flickr	% nodes sampled	0.22%	2.24%	5.60%	11.20%	22.41%	100.00%
	F1-score - *LaGraph*	6.07	51.12	51.12	51.27	51.29	51.26
	Memory - *LaGraph*	1389MB	1465MB	1553MB	1725MB	2065MB	4211MB
	F1-score - GraphCL	45.27	45.27	45.27	45.38	45.45	45.48
	Memory - GraphCL	1647MB	2599MB	4137MB	6741MB	11905MB	47939MB
Reddit	% nodes sampled	0.07%	0.65%	1.63%	3.25%	6.50%	100.00%
	F1-score - *LaGraph*	5.76	95.05	95.06	95.08	95.09	95.22
	Memory - *LaGraph*	1403MB	1475MB	1585MB	1783MB	2161MB	16933MB
	F1-score - GraphCL	93.24	93.24	93.25	93.31	93.32	OOM
	Memory - GraphCL	4199MB	6117MB	6687MB	9297MB	14495MB	OOM

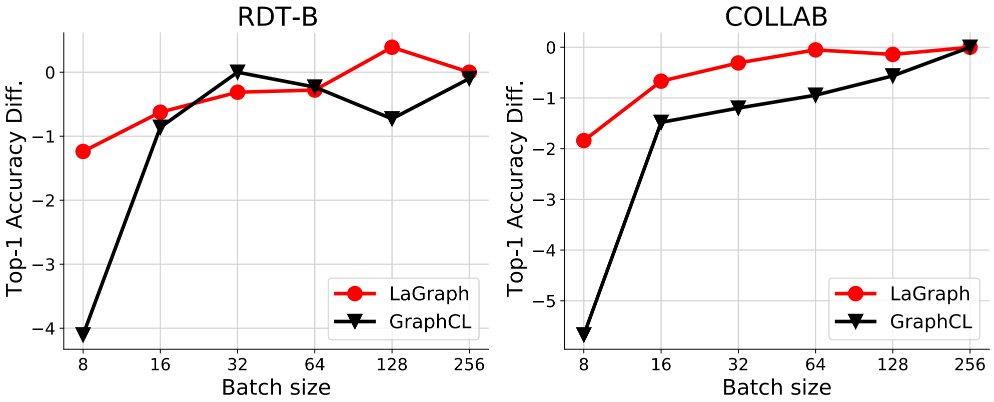


Figure 3.2: Model robustness to small batch sizes on RDT-B and COLLAB. Shown are relative changes in accuracy over different batch sizes compared to the batch size of 256.

robust to the batch size.

Training on Sub-graphs for Large-scale Datasets. Training graph encoders on all nodes for some large-scale graphs can be heavily expensive in computation. We hence conduct an ablation study on how training graph encoders on a portion of sampled nodes instead of the entire graph affects the effectiveness of training. Results in Table 3.3 suggest that the model performance remains stable when decreasing the number of nodes until the number becomes extremely small. The collapse is due to the very sparse connectivity and *LaGraph* fails to reconstruct a node from its neighbor nodes as there are no neighbors at all. In contrast, though GraphCL does not collapse at extremely small subsets, it suffers more from performance loss above 1,000 nodes and consumes significantly more GPU memory.

3.5 Conclusions and Future Directions

We introduced *LaGraph*, a state-of-the-art predictive SSL framework whose objectives are based on self-supervised latent graph prediction. We provided theoretical analysis and discussed the relationship between *LaGraph* and theories in different related domains. Experimental results demonstrate the strong effectiveness of the proposed framework and the stability to the training scale for both graph-level and node-level tasks. Currently, our framework mainly considers the latent graph regarding its node features. Further investigation into a latent graph prediction framework that includes richer information such as edge features and latent connectivity into self-supervision can potentially bring additional improvement to the performance.

4. TASK-AGNOSTIC GRAPH EXPLANATIONS [*]

4.1 Introduction

Graph neural networks (GNNs) [25, 113, 26] have achieved remarkable success in learning from real-world graph-structured data due to their unique ability to capture both feature-wise and topological information. Extending their success, GNNs are widely applied in various research fields and industrial applications including quantum chemistry [114, 115], drug discovery [116, 117, 23], large-scale social networks [118, 119], and recommender systems [120, 121]. While multiple approaches have been proposed and studied to improve GNN performance, GNN explainability is an emerging area and has a smaller body of research behind it. Recently, explainability has gained more attention due to an increasing desire for GNNs with more security, fairness, and reliability. Being able to provide a good explanation to a GNN prediction increases model reliability and reduces the risk of incorrect predictions, which is crucial in fields such as molecular biology, chemistry, fraud detection, etc.

Existing methods adapting the explanation methods for convolutional neural networks (CNNs) or specifically designed for GNNs have shown promising explanations on multiple types of graph data. A recent survey [122] categorizes existing explanation methods into gradient-based, perturbation, decomposition, and surrogate methods. In particular, perturbation methods involve learning or optimization [123, 124, 125, 126, 127] and, while bearing higher computational costs, generally achieve state-of-the-art performance in terms of explanation quality. These methods train *post-hoc* explanation models on top of the prediction model to be explained. Earlier approaches like GNNExplainer [126] require training or optimizing an individual explainer for each data instance, i.e., a graph or a node to be explained. In contrast, PGExplainer [125] performs inductive learning, i.e., it only requires a one-time training, and the explainer can be generalized to explain all data instances without individual optimization. Compared to other optimization-based explanation

[*] Reprinted with permission from "*Task-agnostic graph explanations.*" Yaochen Xie, Sumeet Katariya, Xianfeng Tang, Edward W. Huang, Nikhil Rao, Karthik Subbian, and Shuiwang Ji, 2022. International Conference on Machine Learning, Copyright 2022 by the authors.

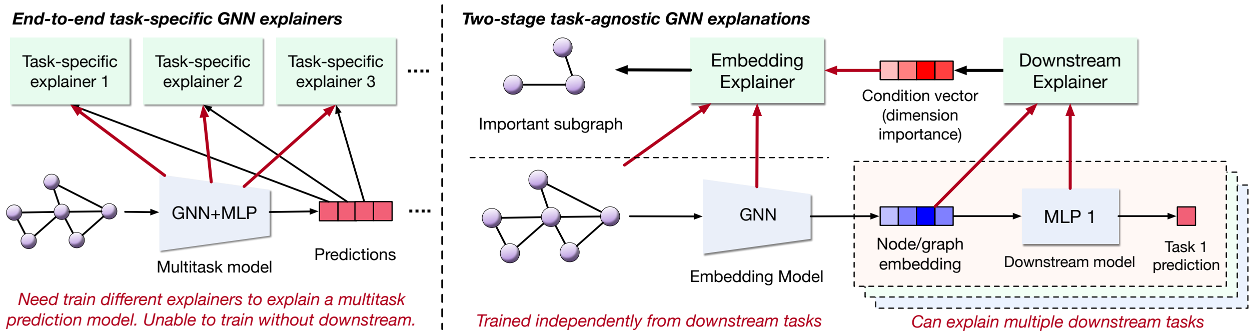

End-to-end task-specific GNN explainers
Task-specific explainer 1
Task-specific explainer 2
Task-specific explainer 3
GNN+MLP
Multitask model
Predictions
Need train different explainers to explain a multitask prediction model. Unable to train without downstream.
Two-stage task-agnostic GNN explanations
Important subgraph
Embedding Explainer
Condition vector (dimension importance)
Downstream Explainer
GNN
Embedding Model
Node/graph embedding
MLP 1
Downstream model
Task 1 prediction
Trained independently from downstream tasks
Can explain multiple downstream tasks

Figure 4.1: A comparison between typical end-to-end task-specific GNN explainers and the proposed task-agnostic explanation pipeline. To explain a multitask model, typical explanation pipelines need to optimize multiple explainers, whereas the two-stage explanation pipeline only learns one embedding explainer that can cooperate with multiple lightweight downstream explainers.

methods, PGExplainer significantly improves efficiency in terms of time cost without performance loss by learning. Following a similar inductive learning paradigm, more recent work ReFine [128] and GSAT [129] aim to provide multi-grained explanations and jointly learned explanations with GNNs, respectively.

However, even state-of-the-art explanation methods like PGExplainer are still task-specific at training and hence suffer from two crucial drawbacks. First, current methods are inefficient in explaining multitask prediction for graph-structured data. For example, one may need to predict multiple chemical properties in drug discovery for a molecular graph. In particular, ToxCast from MoleculeNet has 167 prediction tasks. In these cases, it is common to apply a single GNN model with multiple output dimensions to make predictions for all tasks. However, one is unable to employ a single explainer to explain the above model, since current explainers are trained specifically to explain one prediction task. As a result, in the case of ToxCast, one must train 167 explainers to explain the GNN model. Second, in industry settings, it is common to train GNN models in a two-stage fashion due to scaling, latency, and label sparsity issues. The first stage trains a GNN-based embedding model with a massive amount of unlabeled data in an unsupervised manner to learn embeddings for nodes or graphs. The second stage trains lightweight models such as multilayer perceptrons (MLPs) using the frozen embeddings as input to predict the downstream tasks. In the first stage, the downstream tasks are usually unknown or undefined, and existing task-specific explainers cannot be applied. Also, there can be tens to hundreds of downstream tasks trained on these GNN embeddings, and training a separate explainer for each task is undesirable and downright impossible.

To address the above limitations, we present a new task-agnostic explanation pipeline, where the learned explainer is independent from downstream tasks and can take downstream models as input conditions, as shown in Figure 4.1. Specifically, we decompose a prediction model into a GNN embedding model and a downstream model, designing separate explainers for each component. We design the downstream explainers to cooperate with the embedding explainer. The embedding explainer is trained using a self-supervised training framework, which we dub Task-Agnostic GNN Explainer (TAGE), with no knowledge of downstream tasks, models, or labels. In contrast to existing explainers, the learning objective for TAGE is computed at the graph or node embeddings without involving task-related predictions. In addition to eliminating the need for downstream tasks in TAGE, we argue that the self-supervision performed on the embeddings can bring an additional performance boost in terms of the explanation quality compared to existing task-specific baselines such as GNNExplainer and PGExplainer.

We summarize our contributions as follows: 1) We introduce the task-agnostic explanation problem and propose a two-stage explanation pipeline involving an embedding explainer and a downstream explainer. This enables the explanation of multiple downstream tasks with a single embedding explainer. 2) We propose a self-supervised training framework TAGE, which is based on conditioned contrastive learning to train the embedding explainer. The training of TAGE requires no knowledge of downstream tasks. 3) We perform experiments on real-world datasets and observe that TAGE outperforms existing learning-based explanation baselines in terms of explanation quality, universal explanation ability, and the time required for training and inference.

Relations with Prior Work Our work studies a novel explanation problem under the two-stage and multi-task settings. The settings are important in both industrial and academic scenarios but have not been studied by prior work. Whereas existing studies focus on designing optimization

Table 4.1: Comparisons on properties of common GNN explainers. Inductivity and task-agnosticism are inapplicable for gradient/rule-based methods as they do not require learning. In the last column, we show the number of required explainers for a dataset with N samples and M tasks.

	Learning	Inductive	Task-agnostic	# explainers required
Gradient- & Rule-based	No	-	-	1
GNNExplainer [126]	Yes	No	No	$M * N$
SubgraphX [127]	Yes	No	No	$M * N$
PGExplainer [125]	Yes	Yes	No	M
Task-agnostic explainers	Yes	Yes	Yes	1

approaches [126, 127] and explainer architectures [125] under the typical task-specific setting, our work focuses on an orthogonal problem to enable task-agnostic explanations with the proposed framework including the universal embedding explainer and conditioned learning objectives.

4.2 Task-Agnostic Explanations

4.2.1 Notations and Learning-Based GNN explanation

Our study considers the attributed graph G with node set V and edge set E. We formulate the attributed graph as a tuple of matrices $(\boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{A} \in \{0,1\}^{|V|\times|V|}$ denotes the adjacency matrix and $\boldsymbol{X} \in \mathbb{R}^{|V|\times d_f}$ denotes the feature matrix with feature dimension of d_f. We assume that the prediction model F that is to be explained operates on graph-structured data through two components: a GNN-based embedding model and lighter downstream models. Denoting the input space by $\mathcal{G}$, a node-level embedding model $\mathcal{E}_n : \mathcal{G} \to \mathbb{R}^{|V|\times d}$ takes a graph as input and computes embeddings of dimension d for all nodes in the graph, whereas a graph-level embedding model $\mathcal{E}_g : \mathcal{G} \to \mathbb{R}^{1\times d}$ computes an embedding for the input graph. Subsequently, the downstream model $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}$ computes predictions for the downstream task based on the embeddings.

Typical GNN explainers consider a task-specific GNN-based model as a complete unit, *i.e.*, $F := \mathcal{D} \circ \mathcal{E}$. Given a graph G and the GNN-based model F to be explained , our goal is to identify the subgraph G_{sub} that contributes the most to the final prediction made by F. In other words, we claim that a given prediction is made because F captures crucial information provided by some subgraph G_{sub}. The learning-based (or optimization-based) GNN explanation employs a parametric explainer $\mathcal{T}_\theta$ associated with the GNN model F to compute the subgraph G_{sub} of the given graph data. Concretely, the explainer $\mathcal{T}_\theta$ computes the importance score for each node or edge, denoted as w_i or w_{ij}, or masks for node attributes denoted as m. It then selects the subgraph G_{sub} induced by important nodes and edges, *i.e.*, whose scores exceed a threshold t, and by masking the unimportant attributes. In our study, we follow [125], focusing on the importance of edges to provide explanations to GNNs. Formally, we have $G_{sub} := (V, E_{sub}) = \mathcal{T}_\theta(G)$, where $E_{sub} = \{(v_i, v_j) : (v_i, v_j) \in E, w_{ij} \geq t\}$.

4.2.2 Task-Agnostic Explanations

As introduced in Section 4.1, all existing learning-based or optimization-based explanation approaches are task-specific and hence suffer from infeasibility or inefficiency in many real-application scenarios. In particular, they are of limited use when downstream tasks are unknown or undefined, and fail to employ a single explainer to explain a multitask prediction model.

To enable the explanation of GNNs in two-stage training and multitask scenarios, we introduce a new explanation paradigm called the task-agnostic explanation. The task-agnostic explanation considers a whole prediction model as an embedding model followed by any number of downstream models. It focuses on explaining the embedding model regardless of the number or the existence of downstream models. In particular, the task-agnostic explanation trains only one explainer $\mathcal{T}_\theta^{(tag)}$ to explain the embedding model $\mathcal{E}$, which should satisfy the following features. First, given an input graph G, the explainer $\mathcal{T}_\theta^{(tag)}$ should be able to provide different explanations according to specific downstream tasks being studied. Table 1 compares the properties of common GNN explanation methods and the desired task-agnostic explainers in multitask scenarios. Second, the explainer $\mathcal{T}_\theta^{(tag)}$ can be trained when only the embedding model is available, *e.g.*, at the first stage of a two-stage training paradigm, regardless of the presence of downstream tasks. When downstream tasks and models are unknown, $\mathcal{T}_\theta^{(tag)}$ can still identify which components of the input graph are important for certain embedding dimensions of interest.

4.3 The TAGE Framework

Our explanation framework TAGE follows the typical scheme of GNN explanation introduced in the previous section. It provides explanations by identifying important edges in a given graph and removing the edges that lead to significant changes in the final prediction. Specifically, the goal of the TAGE is to predict the importance score for each edge in a given graph. Different from existing methods, the proposed TAGE breaks down typical end-to-end GNN explainers into two components. We now provide general descriptions and detailed formulations to the proposed framework.

4.3.1 Task-Agnostic Explanation Pipeline

Following the principle of the desired task-agnostic explanations, we introduce the task-agnostic explanation pipeline, where a typical explanation procedure is performed in two steps. In particular, we decompose the typical end-to-end learning-based GNN explainer into two parts: the embedding explainer $\mathcal{T}_{\mathcal{E}}$ and the downstream explainer $\mathcal{T}_{down}$, corresponding to the two components in the two-stage training and prediction procedure. We compare the typical explanation pipeline and the two-stage explanation pipeline in Figure 4.1. The embedding explainer and downstream explainers can be trained or constructed independently from each other. In addition, the embedding explainer can cooperate with any downstream explainers to perform end-to-end explanations on input graphs.

The downstream explainer aims to explain task-specific downstream models. As downstream models are usually lightweight MLPs, we simply adopt gradient-based explainers for downstream explainers without training. The downstream explainer takes a downstream model and the graph or node embedding vector as inputs and computes the importance score of each dimension on the embedding vector. The importance scores then serve as a condition vector input to the embedding explainer. Given the condition vector, the embedding explainer explains the GNN-based embedding model by identifying an important subgraph from the input graph data. In other words, given different condition vectors associated with different downstream tasks or models, the embedding explainer can provide corresponding explanations for the same embedding model. Formally, we denote the downstream explainer for models from $\mathscr{D}$ by $\mathcal{T}_{down} : \mathscr{D} \times \mathbb{R}^d \to \mathbb{R}^d$, which maps input models and embeddings into importance scores $\boldsymbol{m}$ for all embedding dimensions. We denote the embedding explainer associated with the embedding model $\mathcal{E}$ by $\mathcal{T}_{\mathcal{E}} : \mathbb{R}^d \times \mathcal{G} \to \mathcal{G}$, which maps a given graph into a subgraph of higher importance, conditioned on the embedding dimension importance $\boldsymbol{m} \in \mathbb{R}^d$.

The training procedures of the embedding explainer are independent of downstream tasks or downstream explainers. In particular, the downstream explainer is obtained from the downstream model only, and the training of the embedding explainer only requires the embedding model and the input graphs. As downstream models are usually constructed as stacked fully connected (FC) layers and the explanation of FC layers has been well studied, our study mainly focuses on the non-trivial training procedure and design of the embedding explainer.

4.3.2 Training Embedding Explainer under Self-Supervision

A straightforward idea of explaining an embedding model with no knowledge of downstream tasks is to employ existing explainers and perform explanation on the pretext task, such as graph reconstruction [41] or context prediction [76], used during the pre-training of GNNs. However, such explanations cannot be generalized to future downstream tasks as there are limited dependencies between the pretext task and downstream tasks. Therefore, training an embedding explainer without downstream models or labels is challenging, and it is desirable to develop a generalizable training approach for the embedding explainer. To this end, we propose a self-supervised learning framework for the embedding explainer.

The learning objective of the proposed framework seeks to maximize restricted mutual information (MI) between two embeddings, i.e., one of the given graph and one of the corresponding subgraph of high importance induced by the explainer, in a conditioned subspace. We introduce a masking vector $\boldsymbol{p} \in \mathbb{R}^d$ as the condition to indicate specific dimensions of embeddings on which to maximize the MI. During the explanation, we obtain the masking vector from the importance vector computed by any downstream explainer $\mathcal{T}_{down}$. As no downstream importance vector is

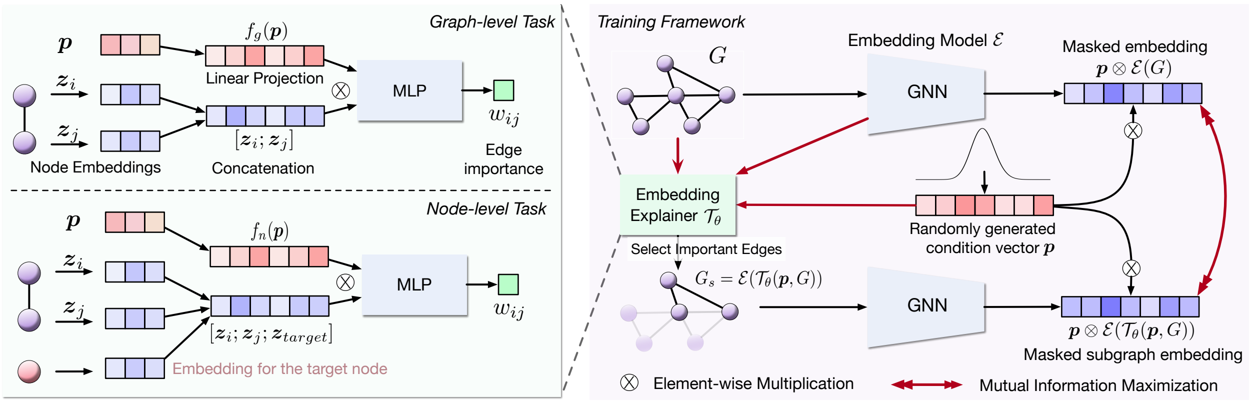

Graph-level Task
$\boldsymbol{p}$
$f_g(\boldsymbol{p})$
$\boldsymbol{z}_i$
Linear Projection
MLP
w_{ij}
$\boldsymbol{z}_j$
$[\boldsymbol{z}_i; \boldsymbol{z}_j]$
Edge importance
Node Embeddings
Concatenation
Node-level Task
$\boldsymbol{p}$
$f_n(\boldsymbol{p})$
$\boldsymbol{z}_i$
MLP
$\boldsymbol{z}_j$
w_{ij}
$[\boldsymbol{z}_i; \boldsymbol{z}_j; \boldsymbol{z}_{target}]$
Embedding for the target node
Training Framework
G
Embedding Model $\mathcal{E}$
Masked embedding
$\boldsymbol{p} \otimes \mathcal{E}(G)$
GNN
Embedding Explainer $\mathcal{T}_\theta$
Randomly generated condition vector $\boldsymbol{p}$
Select Important Edges
$G_s = \mathcal{E}(\mathcal{T}_\theta(\boldsymbol{p}, G))$
GNN
$\boldsymbol{p} \otimes \mathcal{E}(\mathcal{T}_\theta(\boldsymbol{p}, G))$
Masked subgraph embedding
⊗ Element-wise Multiplication
Mutual Information Maximization

Figure 4.2: Overviews of the self-supervised training framework for the embedding model (right) and the architecture of the parametric explainers (left). During training, we generate random condition vectors $\boldsymbol{p}$ as an input to the embedding explainer and mask the embeddings. The learning objective seeks to maximize the mutual information between two embeddings on certain dimensions.

available at training, we sample the masking vector $\boldsymbol{p}$ from a multivariate Laplace distribution due to the sparse gradient, *i.e.*, only a few dimensions are of high importance, assuming embeddings from well-trained models are informative with low dimension redundancy. Empirically, the Laplacian assumption holds on all datasets we work with as we observe that gradients follow a Laplace distribution. Formally, the learning objective based on the restricted MI is

$$\max_{\theta} \mathrm{E}_{\boldsymbol{p}}[\mathbf{MI}(\boldsymbol{p} \otimes \mathcal{E}(G), \boldsymbol{p} \otimes \mathcal{E}(\mathcal{T}_\theta(\boldsymbol{p}, G)))], \tag{4.1}$$

where $\mathbf{MI}(\cdot, \cdot)$ computes the mutual information between two random vectors, $\boldsymbol{p}$ denotes the random masking vector sampled from a certain distribution, $\mathcal{T}_\theta(\boldsymbol{p}, G)$ computes the subgraph of high importance, and $\otimes$ denotes the element-wise multiplication, which applies masking to the embeddings $\mathcal{E}(\cdot)$. Figure 4.2 outlines the training framework and objective. Intuitively, given an input graph and the desired embedding dimensions to be explained, the explainer $\mathcal{T}_\theta$ predicts the subgraph whose embedding shares the maximum mutual information with the original embedding on the desired dimensions.

Practically, the mutual information is intractable and is hence hard to directly compute. A common approach to achieving efficient computation and optimization is to adopt the upper bound estimations of mutual information [79], namely, the Jenson-Shannon Estimator (JSE) [130] and the InfoNCE [131]. These upper bound estimations are also referred to as contrastive loss and are widely applied in self-supervised representation learning [89, 86, 40] for both images and graphs. Adopting these estimators, the objectives are efficiently computed as

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \log \left[\sigma\big((\boldsymbol{p} \otimes \boldsymbol{z}_i)^T (\boldsymbol{p} \otimes \boldsymbol{z}_{i,\theta}) \big) \right] + \frac{1}{N^2 - N} \sum_{i \neq j} \log \left[1 - \sigma\left((\boldsymbol{p} \otimes \boldsymbol{z}_i)^T (\boldsymbol{p} \otimes \boldsymbol{z}_{j,\theta}) \right) \right], \tag{4.2}$$

$$\min_{\theta} -\frac{1}{N} \sum_{i=1}^{N} \left[\log \frac{\exp\{(\boldsymbol{p} \otimes \boldsymbol{z}_i)^T (\boldsymbol{p} \otimes \boldsymbol{z}_{i,\theta})\}}{\sum_{j \neq i} \exp\{(\boldsymbol{p} \otimes \boldsymbol{z}_i)^T (\boldsymbol{p} \otimes \boldsymbol{z}_{j,\theta})\}} \right], \tag{4.3}$$

for JSE and InfoNCE, respectively, where N denotes the number of samples in a mini-batch, σ denotes Sigmoid function, $\boldsymbol{z}_i$ and $\boldsymbol{z}_{i,\theta}$ are embeddings of the original graph G_i and its subgraph $\mathcal{T}_\theta(G_i)$, or target nodes of the two graphs. Our objective involves condition vectors as masks on the embeddings, which differs from typical contrastive loss used in self-supervised representation learning. We hence call the proposed objective the conditioned contrastive loss.

To restrict the size of subgraphs given by the explainer, we follow previous studies [125] to add a size regularization term R, computed as the averaged importance score, to the above objectives. In the case where edge importance scores $w_{ij} \in [0, 1]$ are computed, the regularization term is computed as

$$R(G) = \sum_{(v_i, v_j) \in E} \lambda_s |w_{ij}| - \lambda_e \left[w_{ij} \log w_{ij} - (1 - w_{ij}) \log(1 - w_{ij}) \right], \tag{4.4}$$

where λ_s and λ_e are hyper-parameters controlling the size and the entropy of edge importance scores, respectively.

4.3.3 Explainer Architectures

Embedding explainers. To be consistent with PGExplainer, we adopt the multilayer perceptron (MLP) as the base architecture to predict the importance score w_{ij} for each edge $(u_i, u_j) \in E$, on top of learned embeddings $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ of the two nodes connected by the edge. Edges with scores higher than a threshold are considered important edges that remain in the selected subgraph. In order for the embedding explainer to cooperate with different downstream explainers and provide diverse explanations for different tasks, it additionally requires a condition vector as input indicating the specific downstream task to be explained. Formally, the graph-level embedding explainer takes the embeddings, $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, and the condition vector $\boldsymbol{p}$ as inputs and computes the importance score by

$$w_{ij} = \text{MLP}_g\big([\boldsymbol{z}_i; \boldsymbol{z}_j] \otimes \sigma(f_g(\boldsymbol{p}))\big), \tag{4.5}$$

where $[\cdot;\cdot]$ denotes the concatenation along the feature dimension, $\otimes$ denotes the element-wise multiplication, σ denotes the activation function, and $f_g : \mathbb{R}^d \to \mathbb{R}^{2d}$ is a linear projection. The node-level embedding explainer takes an additional node embedding as its input, as the explainers are expected to predict different scores for the same edge when explaining different target nodes. The formulation of computing the importance score is as follows,

$$w_{ij} = \text{MLP}_n\big([\boldsymbol{z}_i; \boldsymbol{z}_j; \boldsymbol{z}_{target}] \otimes \sigma(f_n(\boldsymbol{p}))\big), \tag{4.6}$$

where $f_g : \mathbb{R}^d \to \mathbb{R}^{3d}$ is a linear projection, and z_{target} denotes the embedding of the target node whose prediction is to be explained.

Downstream explainers. We use the following gradient-based explainer to compute condition vectors for different downstream models. Formally, given an input embedding $\boldsymbol{z}$ and its prediction probabilities $\mathcal{D}(\boldsymbol{z}) \in [0,1]^C$ among all C classes, we compute the gradient of the maximal probability w.r.t. the input embedding:

$$\boldsymbol{g} = \frac{\partial \max_{c \le C} \mathcal{D}(\boldsymbol{z})[c]}{\partial \boldsymbol{z}} \in \mathbb{R}^{1\times d},$$

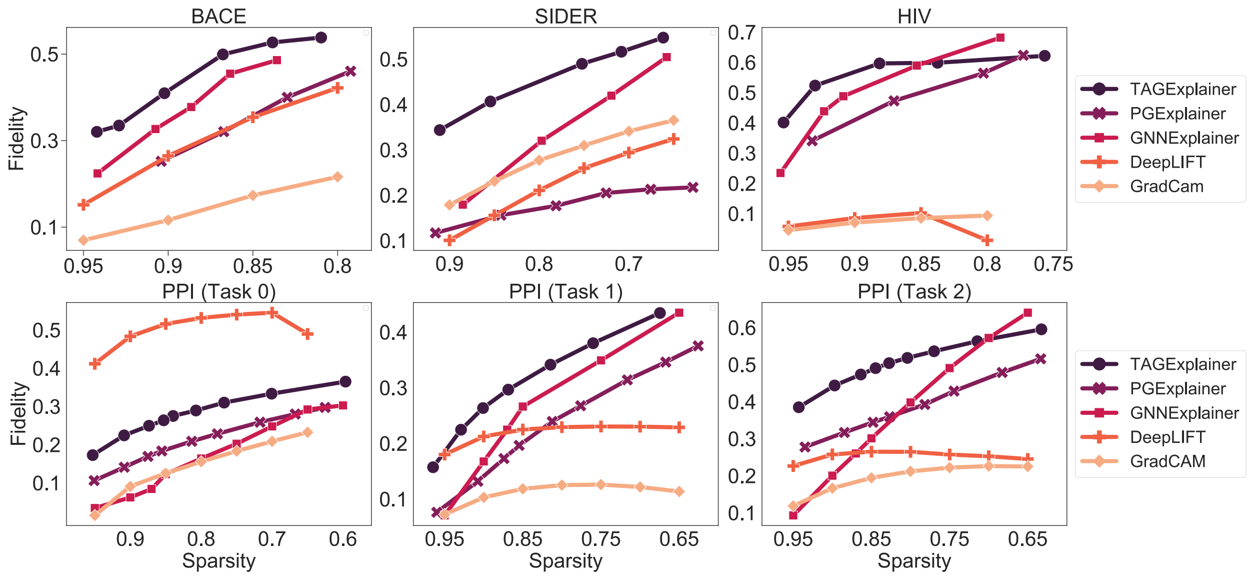


Figure 4.3: Quantitative performance comparisons on six tasks from MoleculeNet (top row) and PPI (bottom row). The curves are obtained by varying the threshold for selecting important edges.

where $\mathcal{D}(\boldsymbol{z})[c]$ denotes the probability for class c. To convert the gradient into the condition vector, we further perform normalization and only take positive values reflecting only positive influence to the predicted class probability, *i.e.*, $\boldsymbol{p} = \text{ReLU}(\text{norm}(\boldsymbol{g}^T))$.

4.4 Experimental Studies

We conduct two groups of quantitative studies evaluating the explanation quality and the universal explanation ability, *i.e.*, training a single explainer to explain all downstream tasks, of TAGE. We then compare the efficiency of multiple learning-based GNN explainers in terms of training and explanation time costs. We further provide visualizations to demonstrate the explanation quality as well as the ability to explain GNN models without downstream tasks.

4.4.1 Datasets

To demonstrate the effectiveness of the proposed TAGE on both node-level and graph-level tasks, we evaluate TAGE on three groups of real-world datasets that contain potentially multiple tasks.

MoleculeNet. The MoleculeNet [23] library provides a collection of molecular graph datasets for the prediction of different molecule properties. In a molecular graph, each atom in the molecule is considered a node, and each bond is considered an edge. The prediction of molecule properties is a graph-level task. We include three graph classification tasks from MoleculeNet to evaluate the explanation of graph-level tasks: HIV, SIDER, and BACE.

Protein-Protein Interaction. The Protein-Protein Interaction (PPI) [108] dataset documents the physical interactions between proteins in 24 different human tissues. In PPI graphs, each protein is considered as a node with its motif and immunological features, and there is an edge between two proteins if they interact with each other. Each node in the graphs has 121 binary labels associated with different protein functions. As different protein functions are not exclusive to each other, the prediction of each protein function is considered an individual task instead of a multi-class classification. And hence typical approaches require individual explainers for the 121 tasks. We utilize the first five out of 121 tasks to evaluate the explanation of node-level tasks.

E-commerce Product Network. The E-commerce Product Network (EPN)* is constructed with subsampled, anonymized logs from an e-commerce store, where entities including buyers, products, merchants, and reviews are considered as nodes, and interactions between entities are considered as edges. We subsample the data for the sake of experimental evaluations and the dataset characteristics do not mirror actual production traffic. We study the explanation of the classification of fraudulent entities (nodes), where the predictions for different types of entities are considered individual tasks. We evaluate our framework specifically on classifications of the buyer, merchant, and review nodes.

4.4.2 Experiment Settings and Evaluation Metrics

For each real-world dataset, we evaluate explainers on multiple downstream tasks that share a single embedding model. For consistency with industrial use cases, we perform the two-stage training paradigm to obtain GNN models to be explained. In particular, we first use unlabeled graphs to train the GNN-based embedding model in an unsupervised fashion. We then freeze the embedding model and use the learned embeddings to train individual downstream models struc-

*Proprietary dataset

tured as 2-layer MLPs. Specifically, for graph-level classification tasks in MoleculeNet, we employ the GNN pretraining strategy context prediction [76] to train a 5-layer GIN [26] as the embedding model on ZINC-2M [132] containing 2 million unlabeled molecules. For the node-level classification on PPI, we employ the self-supervised training method GRACE [74] to train a 2-layer GCN [25] on all 21 graphs from PPI without using labels. For the larger-scale node-level classification on EPN, we use graph autoencoder (GAE) [41] to train the embedding model on sampled subgraphs of EPN.

As the involved real-world datasets do not have ground truth for explanations, we follow previous studies [133, 122, 127] to adopt a fidelity score and a sparsity score to quantitatively evaluate the explanations. Intuitively, the fidelity score measures the level of change in the probability of the predicted class when removing important nodes or edges, whereas the sparsity score measures the relative amount of important nodes or nodes associated with important edges. A formulation of the scores is provided in Appendix B. Note that compared to explanation evaluation with ground truths, the fidelity score is considered more faithful to the model, especially when the model makes incorrect predictions, in which case the explanation ground truths become inconsistent with the evidence of making the wrong predictions. In practice, one needs to trade-off between the fidelity score and the sparsity score by selecting the proper threshold for the importance.

Table 4.2: Fidelity scores with controlled sparsity on graph-level molecule property prediction tasks. Each column corresponds to an explainer model trained on (or without) a specific downstream task. Underlines highlight the best explanation quality in terms of fidelity, on the same level of sparsity.

	PGExplainer (trained on)				TAGE
Eval on	BACE	HIV	BBBP	SIDER	w/o downstream
BACE	**0.252 ±0.340**	0.007 ±0.251	0.026 ±0.022	-0.151 ±0.330	**0.378 ±0.293**
HIV	-0.001 ±0.197	**0.473 ±0.404**	0.013 ±0.029	-0.060 ±0.356	**0.595 ±0.321**
BBBP	0.001 ±0.237	-0.056 ±0.226	**0.182 ±0.169**	-0.252 ±0.440	**0.193 ±0.161**
SIDER	0.012 ±0.219	-0.009 ±0.212	0.003 ±0.029	**0.444 ±0.391**	**0.521 ±0.278**

Table 4.3: Fidelity scores with controlled sparsity on the node-level classification dataset PPI. Each column corresponds to an explainer model trained on (or without) a specific downstream task. Underlines highlight the best explanation quality in terms of fidelity, on the same level of sparsity.

	PGExplainer (trained on)					TAGE
Eval on	Task 0	Task 1	Task 2	Task 3	Task 4	w/o downstream
Task 0	**0.184 ±0.3443**	-0.005 ±0.268	0.033 ±0.335	0.034 ±0.310	0.018 ±0.194	**0.271 ±0.385**
Task 1	0.046 ±0.447	**0.197 ±0.380**	0.043 ±0.314	0.008 ±0.297	0.021 ±0.183	**0.300 ±0.415**
Task 2	0.028 ±0.434	0.001 ±0.283	**0.345 ±0.458**	0.024 ±0.320	0.097 ±0.320	**0.499 ±0.480**
Task 3	0.075 ±0.364	-0.015 ±0.219	0.036 ±0.317	**0.262 ±0.418**	0.040 ±0.221	**0.289 ±0.427**
Task 4	0.035 ±0.413	-0.021 ±0.238	0.223 ±0.438	0.075 ±0.374	**0.242 ±0.373**	**0.330 ±0.442**

Table 4.4: Fidelity scores with controlled sparsity on the E-commerce product dataset. Each column corresponds to one explainer model trained on different tasks or without downstream tasks. Underlines highlight the best explanation quality in terms of fidelity, on the same level of sparsity.

	PGExplainer (trained on)			TAGE
Eval on	Buyers	Sellers	Reviews	w/o downstream
Buyers	**0.2009 ±0.2233**	0.1731 ±0.3774	0.1740 ±0.4463	**0.2713 ±0.1834**
Sellers	0.5465 ±0.4773	**0.3246 ±0.4026**	0.1128 ±0.3019	**0.6515 ±0.3426**
Reviews	0.4178 ±0.3683	0.1258 ±0.3492	**0.2310 ±0.4178**	**0.5692 ±0.4214**

4.4.3 Quantitative Studies

We conduct two groups of quantitatively experimental comparisons. We first demonstrate the explanation quality of individual tasks in terms of the fidelity score and the sparsity score. We do this by comparing TAGE with multiple baseline methods including non-learning-based methods GradCAM [133] and DeepLIFT [134], as well as learning-based methods GNNExplainer [126]

Table 4.5: Comparison of computational time cost among three learning-based GNN explainers on the PPI dataset. The left two columns record time cost breakdown for T downstream tasks. The fourth column estimates the total time cost for explaining all 121 tasks of PPI. The last row shows the speedup times compared to GNNExplainer and PGExplainer, respectively.

Time cost	Training (s)	Inference (s)	Total time (T=1) (s)	Est. total for 121 tasks
GNNExplainer	20040.1*T	–	20040.1	28 d
PGExplainer	7117.0*T	**427.2*T**	7604.2	10.7 d
TAGE	**1405.3**	582.7*T	**1988.0**	**0.83 d**
Speedup	**14.3*T× / 5.1*T×**	– / 0.73×	**10.1× / 3.8×**	**33.7× / 12.9×**

and PGExplainer [125]. We do not include other optimization or search-based methods such as Monte-Carlo tree search [135] due to the significant time cost on real-world datasets. Note that to show the effectiveness of universal explanations over different downstream tasks, we only train one embedding explainer for all tasks in a dataset, on top of which a gradient-based downstream explainer is applied to explain multiple downstream tasks. In contrast, for existing learning-based methods, we need to train multiple explainers to explain downstream tasks individually. For all methods, we vary the threshold for selecting important nodes or edges and compare how fidelity scores change over sparsity scores on each task and dataset. The results are shown in Figure 4.3. In particular, TAGE outperforms other learning-based explainers on BACE, SIDER, and PPI (tasks 0 and 1). For HIV and PPI (task 2), TAGE is more effective at higher sparsity levels, *i.e.*, when fewer nodes are considered important and masked.

To justify the necessity of task-agnostic explanation and demonstrate the universal explanation ability of TAGE, we include PGExplainer as our baseline and compare the explanation quality when adopting a single explainer to explain multiple downstream tasks. For PGExplainer, we train multiple explainers on different downstream tasks and evaluate each explainer on different downstream tasks. For TAGE, we train one explainer without downstream tasks and evaluate it on different downstream tasks. Results shown in Table 4.2 (MoleculeNet), Table 4.3 (PPI), and Table 4.4 (EPN) indicate that task-specific explainers fail to generalize to different downstream tasks and hence are unable to provide universal explanations. On the other hand, the task-agnostic explainer, although trained without downstream tasks, can provide explanations with even higher quality for downstream tasks.

GNNExplainer and PGExplainer should generally outperform task-agnostic explainers, as they are specific to data examples or tasks. This should especially be true when TAGE and PGExplainer have the same level of parameters. However, we find that TAGE outperforms the learning-based baselines. We believe that the underperformance of baselines is due to the non-injective characteristic of the downstream MLPs, where different embeddings can produce similar downstream prediction results. In other words, a similar downstream prediction are not necessarily produced

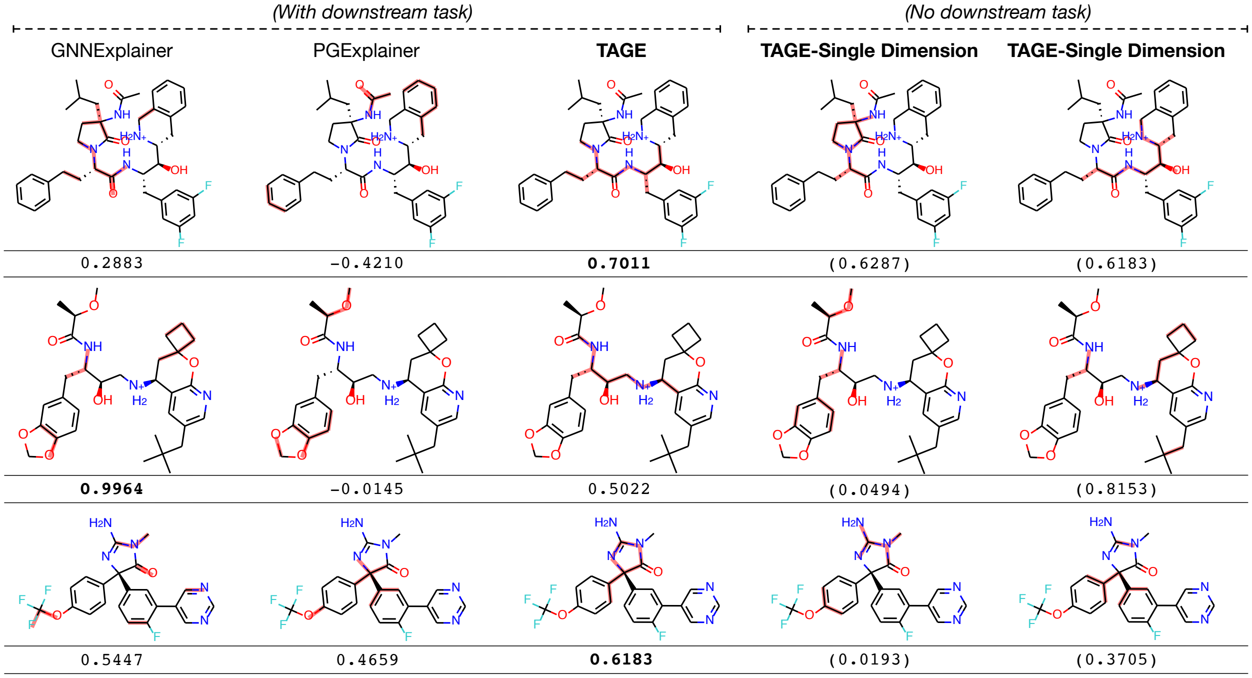


Figure 4.4: Visualizations on explanations to the GNN model for the BACE task. The top 10% important edges are highlighted with red shadow. The numbers below molecules are fidelity scores when masking out the top 10% important edges. The right two columns are explanations for two certain embedding dimensions without downstream tasks. Fidelity scores in the right two columns explaining two embedding dimensions are still computed for the BACE task but are just for reference.

by embeddings that share high mutual information. Due to this characteristic, the learning objective of TAGE computed between embeddings brings stronger supervision than the objective computed between final predictions, as the latter objective does not guarantee consistency between embeddings or between input graphs and subgraphs.

Multitask Explanation Efficiency A major advantage of the task-agnostic explanation is that it removes the need for training individual explainers, which consumes the majority of the total time cost to explain a model on a dataset. We hence evaluate the efficiency of TAGE in terms of time cost for explanation and compare it to the two learning-based explainer baselines. We record the time cost for the training and inference of different explanation methods on the same dataset and device, shown in Table 4.5. All results are obtained from running the explanation on the PPI dataset with 121 node classification tasks with a single Nvidia Tesla V100 GPU. Although the inference time cost of TAGE is slightly higher than that of PGExplainer, the results show TAGE costs significantly less time than GNNExplainer and PGExplainer, especially in the multitask cases ($T > 1$). TAGE allows the explanation of many downstream tasks within a reasonable time duration.

4.4.4 Visualization of Explanations

We visualize the explanations of the three learning-based explanation methods on the BACE task, which aims to predict the inhibitor effect of molecules on human β-secretase 1 (BACE-1) [23]. The visualization results are shown in Figure 4.4. Each molecule visualization shows the top 10% important edges (bonds) predicted by an explainer marked in red, together with the fidelity score on the molecule. The left three columns are explanation results with the BACE downstream task. The right columns are explanations by TAGE to two specific graph embedding dimensions, without downstream models. Embedding dimensions with greater values among all are selected in the visualizations. To obtain explanations of certain embedding dimensions, we input the one-hot vectors to the embedding explainer as condition vectors. The visualization results indicate that while baseline methods select scattered edges as important, TAGE tends to select edges that form a connected substructure, which is more reasonable when explaining molecule property predictions where a certain functional group is important for the property.

While there are no ground-truth explanations for the molecular datasets, the validity of results produced by TAGE can be evidenced by multiple domain research. Take BACE for example, [136] study multiple BACE-1 inhibitors that are similar to one presented in our results (Figure 4.4 - line 3). Inhibitors in Table 1–3 and 8 of [1] share the common "2-imidazoline" structure as explained by TAGE, whereas structures such as =O and -OCF_3 as explained by GNNE and PGE are not necessarily in an inhibitor. Moreover, inhibitors studied by [137] share the common "-C(=O)-C-N(H)-C(OH)-" chain structure as present in the explanation results by TAGE (Figure 4.4 - lines 1 and 2), whereas structures explained by other explainers are not necessarily for a molecule to be a BACE-1 inhibitor. Nevertheless, it's still fidelity scores that give the most reliable evaluation. In addition, the right three columns indicate that dimensions in the embedding correspond to different substructures and TAGE is able to provide explanations to the dimensions without downstream tasks.

4.5 Conclusions

Existing task-specific learning-based explainers become inapplicable under real scenarios when downstream tasks or models are unavailable and suffer from inefficiency when explaining real-world graph datasets with multiple downstream tasks. We introduced TAGE, including the task-agnostic GNN explanation pipeline and the self-supervised training framework to train the embedding explainer without knowing downstream tasks or models. Our experiments demonstrate that the TAGE generally achieves higher explanation quality in terms of fidelity and sparsity with a significantly reduced explanation time cost.

5. GENETIC INFOMAX: EXPLORING MUTUAL INFORMATION MAXIMIZATION IN HIGH-DIMENSIONAL IMAGING GENETICS STUDIES

5.1 Introduction

Genome-wide association studies (GWAS) have been an effective approach driving genetic discovery in the past 15 years [138]. Given a phenotype of interest and a cohort of individuals with both the measurements of the phenotype and the genotypes over markers across the genome, linear or linear mixed model are built to test for the association of each marker to the phenotype and thus pinpoint the relevant gene loci. However, the typical GWAS studies are focused on well-established phenotypes, typically the risks of diseases, or well-established macro-level measurements such as heights, BMI, or molecular-level measurements such as protein and metabolomic biomarkers. When the phenotype of interest is a high-dimensional complex data modality such as imaging data, there is a lack of sophisticated approaches for deriving phenotypes for GWAS. Taking brain imaging as an example, existing approaches mostly used traditional non-learning software to derive brain region-based volumetric or surface features. These approaches carry human preconceptions and biases, and thus limited the expressiveness of these phenotypes and the power of genetic discovery.

Recently deep learning approaches [139, 140, 141, 142] derive phenotypes from medical images by learning a latent representation that captures the inherent content of the input image. However, approaches learning from imaging data alone fail to utilize the accompanying genetic data. Those approaches tend to capture patterns that are not related to genes and common patterns shared by multiple individuals. For example, [139] found that representations learned by an image autoencoder are unable to fully reconstruct fine details that are individually specific. To overcome these limitations, a solution is to incorporate trans-modal learning strategies that utilize the pairwise relationship between imaging and genetic data, such as trans-modal contrastive learning [143, 144, 142]. Unfortunately, the use of genetic data, including the encoding of data and capturing image-genetic relationships, still poses significant challenges. Results by [142] suggest that, despite the promising results for downstream classification of disease risk, multi-modal contrastive approaches still underperform compared to typical image-only approaches [140] on GWAS tasks. This underperformance becomes even more pronounced for higher-dimensional 3D data.

In this chapter, we formulate the problem as learning the representation of imaging data that shares the maximum mutual information with genetic data. By using mutual information as a perspective, we are able to examine the key reasons for the failure of typical trans-modal contrastive learning for GWAS on high-dimensional imaging data. To push the limits of existing learning approaches, we propose **Genetic InfoMax** (**GIM**), a trans-modal learning framework that includes a regularized mutual information estimator and a novel transformer-based genetic encoder. The framework addresses the issues of dimensional collapse and non-generalizable associations in representation learning for GWAS, and fully utilizes the genetic data with physical and genetic position information. Our experiments demonstrate that GIM significantly improves performance on all four evaluation metrics.

5.2 Problem Formulation

We study the problem of genome-wide association studies (GWAS) on high-dimensional data. The GWAS aims to identify associations between specific genetic variants, known as single nucleotide polymorphisms (SNPs), in the genome and certain traits of interest such as the risk of disease and other biological characteristics of organisms. In particular, each individual genetic data is denoted by $\boldsymbol{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_L\}$, and traits of interest denoted by $\boldsymbol{y}$, the GWAS process involves statistical tests on the sample pairs $\{(\boldsymbol{G}_i, \boldsymbol{y}_i)\}_{i=1,\cdots,M}$ from a large number of M individuals to identify the specific subset $\boldsymbol{G}^{g \to y} \subset \boldsymbol{G}$ of SNPs that are associated with the target traits $\boldsymbol{y}$. Here L is the number of SNPs, each $\boldsymbol{g}_i \in \{0, 1, 2\}$, representing the number of carried variants for each individual. The values of nearby SNPs are often correlated due to their common inheritance from a shared ancestor. To account for this, it is necessary to select an independent subset of genetic information $\boldsymbol{G}^{\text{ind}} \subset \boldsymbol{G}^{g \to y}$, which can be achieved by clustering and selecting the SNPs with the lowest p-value from each cluster after the statistical test, which is important for accurate analysis and interpretation of the genetic data. Practically, when conducting GWAS on high-dimensional data $\boldsymbol{Y}$ such as medical imaging, an additional step is required before performing statistical tests. This step involves reducing the number of traits from the high-dimensional data $\boldsymbol{Y}$ to a smaller number $\boldsymbol{y}$ through experts' diagnosis or computational approaches.

Traits Computing as Representation Learning. To enable GWAS on high-dimensional data, we are interested in computationally obtaining informative lower-dimensional traits, termed GWAS representation learning, from the high-dimensional data. Specifically, for any high-dimensional data $\boldsymbol{Y}$ to be studied, the problem is formulated as to learn lower-dimensional representations of $\boldsymbol{Y}$ with a corresponding encoder f_θ in a self-supervised manner, such that a larger number of independent SNPs $|G^{\text{ind}}|$ can be identified from the pairs $\{(\boldsymbol{G}_i, f_\theta(\boldsymbol{Y}_i))\}_{i=1,\cdots,M}$. The goal of identifying more independent SNPs requires that more information related to genetic variants is captured by the representation of $\boldsymbol{Y}$. We focus on learning d-dimensional representations $\boldsymbol{y} = f_\theta(\boldsymbol{Y}) \in \mathbb{R}^q$ with the following optimization objective

$$\theta^* = \arg\max_\theta \mathcal{I}(f_\theta(\boldsymbol{Y}), \boldsymbol{G}), \tag{5.1}$$

where $\mathcal{I}(\cdot,\cdot)$ denotes the mutual information (MI) between two random variables. As computing the true value of mutual information is intractable, it becomes critical to develop an appropriate mutual information estimation under the GWAS problem setting.

Notations of Data. We instantiate our problem specifically with the 3D human brain magnetic resonance imaging (MRI) data and SNPs from the human genome. We denote the 3D brain MRI by $\boldsymbol{Y} \in \mathbb{R}^{H\times W\times D\times 1}$, where H, W, D denote the three spatial dimensions, and 1 denotes the single channel of the MRI. The human genetic data $\boldsymbol{G}$ consists of N positions on the human genome with frequent variants (SNPs). Each SNP $\boldsymbol{g}_i$ is represented by a four-tuple $\left(d_i, c_i, p_i^{\text{phy}}, p_i^{\text{gen}}\right)$. In the four-tuple, $d_i \in \{0, 1, 2\}$ denotes the genotype of the SNP, the number of copies of the mutant allele, $c_i \in \{1, \cdots, 22\}$ denotes the index of chromosome the SNP belongs to, $p_i^{\text{phy}} \in \mathbb{N}$ denotes the physical position in terms of base pair (bp) of the SNP in the chromosome, and $p_i^{\text{gen}} \in \mathbb{R}^+$ denotes the genetic position of the SNP in terms of centimorgan (cM). Note that the genetic data is not a sequence but an array since two neighbor SNPs $\boldsymbol{g}_i$ and $\boldsymbol{g}_{i+1}$ are not necessarily to be consecutive on the original genome and the physical (and genetic) distance between them $\left|p_i^{\text{phy}} - p_{i+1}^{\text{phy}}\right|$ is meaningful. We further denote arrays consisting of all genotypes, chromosomes, and positions sorted on chromosome id and physical position by $\boldsymbol{d}$, $\boldsymbol{c}$, $\boldsymbol{p}^{\text{phy}}$, and $\boldsymbol{p}^{\text{gen}}$, respectively.

5.3 What Makes Appropriate MI Estimators for GWAS?

With the goal of learning representations that capture as much information about the genetic variations as possible, our objective is to maximize the MI between the representation and genetic data with an appropriate MI estimator. One commonly used approach for estimating the mutual information between multi-dimensional variables is the Jensen-Shannon Estimator (JSE) [130]. The JSE involves a discriminator to distinguish whether samples of the two variables belong to the same individual or are independently sampled. Specifically, under our problem setting, the JSE-based training loss is computed as

$$\begin{aligned}\mathcal{L}_{\text{JSE}}(\boldsymbol{B};\theta,\phi) = &-\frac{1}{|B|}\sum_{(\boldsymbol{Y},\boldsymbol{G})\in\boldsymbol{B}}\log\Big(\mathcal{D}_\phi\big(f_\theta(\boldsymbol{Y}),\,\boldsymbol{G}\big)\Big)\\ &-\frac{1}{|B|(|B|-1)}\sum_{(\boldsymbol{Y},\boldsymbol{G})\in\boldsymbol{B}}\left[\sum_{(\boldsymbol{Y}',\boldsymbol{G}')\in\boldsymbol{B}\setminus\{(\boldsymbol{Y},\boldsymbol{G})\}}\log\Big(1-\mathcal{D}_\phi\big(f_\theta(\boldsymbol{Y}),\,\boldsymbol{G}'\big)\Big)\right],\end{aligned} \tag{5.2}$$

where $\boldsymbol{B}$ is a mini-batch of paired MRI and genetic data and $\mathcal{D}_\phi : \mathbb{R}^q \times \mathcal{G} \to (0,1)$ is a learnable discriminator to determine whether $f_\theta(\boldsymbol{Y})$ and $\boldsymbol{G}$ are from the same individual. Together with the Noise Contrastive Estimation (InfoNCE) [145], these learning processes are also known to be in the contrastive manner across two modalities; namely, the MRI and genetic data.

To achieve desirable performance in maximizing MI and discovering genetic associations, the discriminator should meet certain requirements. First, the learnable discriminator $\mathcal{D}_\phi$ should be able to take as inputs the genetic data $\boldsymbol{G}$ and encode all the useful information from $\boldsymbol{G}$. A well-designed genetic encoder is thus a critical component of $\mathcal{D}_\phi$. It should be able to efficiently and effectively use not only the genotypes, but also their corresponding chromosome, physical posi-

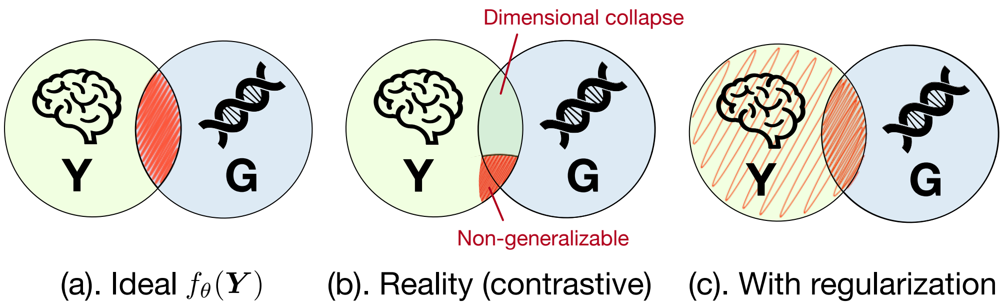


Figure 5.1: An illustration showing how the representation $f_\theta(\boldsymbol{Y})$ captures the mutual information between $\boldsymbol{Y}$ and $\boldsymbol{G}$ in different cases. The circles are the entropy of G and Y, respectively, and their intersection is the mutual information $\mathcal{I}(G, Y)$. Areas indicated by squiggles in red represent the information contained in $f_\theta(\boldsymbol{Y})$.

tion, and genetic position information. In Section 5.4, we propose a novel transformer-based genetic encoder, dubbed genetic transformer, that fully utilizes all this information based on genetic intuitions.

Second, the discriminator should make predictions based on all generalizably associated patterns, rather than memorizing noise or focusing on a small portion of associated patterns that can be easily learned. However, due to the nature of contrastive learning and several differences between typical visual representation learning and GWAS representation learning, we will argue below that it is challenging to meet this requirement. The typical contrastive loss can lead to degenerated results for GWAS as shown in Figure 5.1.

5.3.1 Uniqueness of GWAS Representation Learning and Limitations of Contrastive Losses

To understand the limitations of typical contrastive losses in the GWAS setting, we first identify key differences between the visual representation learning problem for natural images and the GWAS representation learning on high-dimensional data. We explain how each difference can contribute to limitations or failures of typical contrastive losses in the GWAS setting, and provide empirical evidence to support our arguments.

Difference 1: Goals of learning representations. Typical visual representations of natural images aim to capture the key semantics or class information about major objects in images. With this

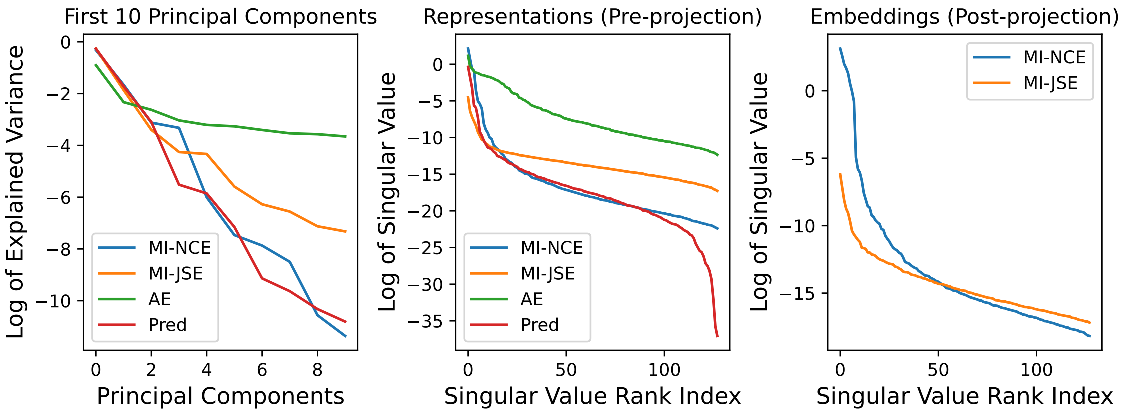


Figure 5.2: The logarithm of the explained variance for the first 10 principal components (**left**), the singular value spectrum of learned representations (**middle**) and embeddings after projection (**right**). Comparisons are among contrastive MI estimators with InfoNCE (MI-NCE), with JSE (MI-JSE), Autoencoder (AE), and genetic data prediction (Pred).

goal, it is acceptable for the representation to capture only semantic or class-related information, or even required that representations are invariant to elements such as context [146, 147] and transformations [148, 149]. In this case, a good representation for downstream tasks does not necessarily maximize the MI during contrastive learning. In contrast, a good representation for the GWAS purpose should capture every detail or pattern in the high-dimensional MRI data that is associated with the genes, since there is no such key semantics or class information. In this case, the downstream GWAS performance is closely associated with $\mathcal{I}(f_\theta(\boldsymbol{Y}), \boldsymbol{G})$.

Limitation 1: Dimensional collapse. A recent study on visual representation learning [150] identifies and empirically shows that typical contrastive approaches suffer from the dimensional collapse issue, where the learned representations occupy a lower-dimensional subspace than their designated dimensions. The dimensional collapse results in high redundancy, limits the information captured by representations, and therefore leads to reduced performance in downstream tasks. Indeed, our analyses show that the dimensional collapse issue also presents in the cross-modal contrastive setting. We compare the singular values of representations learned by predictive methods and contrastive methods in Figure 5.2. Results indicate that the contrastive estimators NCE and JSE suffer from dimensional collapse with a dramatic drop in explained ratios and singular values.

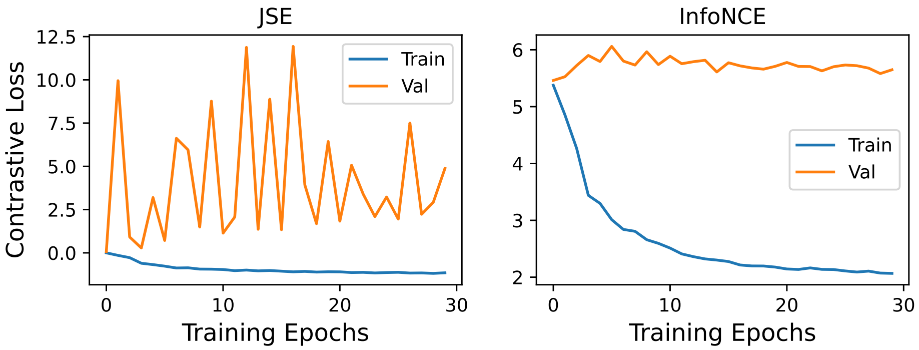


Figure 5.3: Generalization capability of contrastive MI estimators JSE (**left**) and InfoNCE (**right**). The learned discriminators fail to generalize to unseen pairs, leading to a large discrepancy between training and validation losses.

Even worse, the GWAS performance suffers more from the dimensional collapse issue due to its nature described in **Difference 1**, as both the information of $f_\theta(\boldsymbol{Y})$ and the mutual information $\mathcal{I}(f_\theta(\boldsymbol{Y}), \boldsymbol{G})$ is limited.

Difference 2: Augmentation approaches and data dimensions. Contrastive learning relies on a large number of samples to more accurately estimate and maximize the mutual information between different views or modalities. Previous studies [13, 44] have shown that augmentations are crucial for contrastive learning, as they prevent representations from focusing on patterns that are irrelevant to downstream tasks and multiply the number of training samples. For higher-dimensional 3D MRI data, more samples and diverse augmentations are necessary [151]. However, the availability of medical imaging data is limited, and most augmentations used for typical visual representation learning are not applicable to medical imaging. For example, since MRIs are single-channel, color space augmentations are not possible. Augmentations based on rotation and flipping are not suitable for brain MRI data due to their asymmetric nature. Random linear transform or non-linear morph may change the shape of the elements of the image and thus are discouraged. In the case of 3D MRI, the applicable augmentations are very limited.

Limitation 2: Non-generalizable associations. According to [152], augmentations play a critical role in the generalization capability of contrastive learning approaches. However, due to the limited number of applicable augmentation techniques and the dimensionality of 3D data, the discriminator tends to capture non-generalizable or false associations from the training samples such as memorizing the shape, the layout of the brain in the MRI, or specific noise in the data to identify individuals. In Figure 5.3, we evaluate the generalizability of models trained with contrastive loss by comparing the MI estimation on training and validation pairs. The remarkable discrepancy in losses between the training and validation sets suggests that the discriminator used by contrastive loss is unable to generalize to new samples, indicating that contrastive loss is a poor estimation of MI in our case.

From the perspective of mutual information, due to the dimensional collapse, a limited amount of training samples, sufficient means of augmentations, empirical results show that the JSE is not an optimal estimator of mutual information. The true mutual information is hence not maximized by the learned representations, leading to degraded performance in GWAS. Figure 5.1 (a–b) illustrates the relationship among the brain MRI, genetic data, and the learned representation. In the ideal case shown in (a), the representation should perfectly cover the mutual information between $\boldsymbol{Y}$ and $\boldsymbol{G}$, so that the learning objective achieves its maximal with

$$\mathcal{I}(f_\theta(\boldsymbol{Y}), \boldsymbol{G}) = \mathcal{I}(\boldsymbol{Y}, \boldsymbol{G}) \geq \mathcal{I}(f_{\theta'}(\boldsymbol{Y}), \boldsymbol{G}),\ \forall f_{\theta'}. \tag{5.3}$$

In practice with brain MRI data, the contrastive loss results in representations that only capture a small portion of $\mathcal{I}(\boldsymbol{Y}, \boldsymbol{G})$ due to the two limitations described above, as shown in (b).

5.3.2 MI Estimator with Regularizations

Given the issues and limitations outlined above, our goal is to improve the representation by incorporating more generalizably associated patterns in addition to those identified by the contrastive MI estimator. However, unlike in the case of natural images where many elements are known to be non-generalizable, our limited knowledge of undiscovered genetic associations makes it difficult to determine which patterns are generalizable and which are not. As a result, it is challenging to develop targeted augmentations that make the representation invariant to unwanted patterns.

To achieve our goal without requiring further knowledge, we propose to uniformly increase the total information contained in the representation by including an entropy term in the learning objective. The objective is formulated as

$$\max_{\theta}\left[\hat{\mathcal{I}}(f_\theta(\boldsymbol{Y}),\boldsymbol{G})+\lambda\mathrm{H}(f_\theta(\boldsymbol{Y}))\right], \tag{5.4}$$

where $\hat{\mathcal{I}}$ is the contrastive MI estimation JSE, H denotes the entropy of a random variable, and λ is a weight scalar. The entropy term encourages the representation to capture more information about $\boldsymbol{Y}$ and reduces its redundancy. A certain portion of the information can contribute to the generalizable associations, as illustrated in Figure 5.1-(c). The entropy term serves as a regularization to the estimated MI to improve its generalizability. From the optimization aspect of view, the objective is considered as adding a Lagrange multiplier to maximize the entropy $\mathrm{H}(f_\theta(\boldsymbol{Y}))$, subject to the constraint that the estimated mutual information $\hat{\mathcal{I}}(f_\theta(\boldsymbol{Y}),\boldsymbol{G})$ achieves its maximum. When multiple patterns can be used to identify individuals, the entropy term encourages the model to capture as many of them as possible, instead of capturing the easiest ones.

There are various methods to estimate and optimize the entropy, such as minimizing the off-diagonal values in the covariance matrix of the representation [87]. However, these estimations require a large mini-batch size, which is not suitable in our case due to memory constraints caused by the 3D data and MRI encoder. As an alternative, we use the reconstruction of MRI data as a proxy to maximize the entropy. The loss is then computed as

$$\mathcal{L}(\boldsymbol{B};\theta,\phi,\psi)=\mathcal{L}_{\mathrm{JSE}}(\boldsymbol{B};\theta,\phi)+\frac{\lambda}{|B|}\sum_{(\boldsymbol{Y},\boldsymbol{G})\in\boldsymbol{B}}\|\boldsymbol{Y}-h_\psi(f_\theta(\boldsymbol{Y}))\|^2, \tag{5.5}$$

where h_ψ is a deterministic decoding head used to reconstruct the MRI from the representation $f_\theta(\boldsymbol{Y})$. Compared to other proxies discussed by [87], the reconstruction term is less sensitive to small mini-batch sizes. To justify the reconstruction term, we have

$$H(f_\theta(\boldsymbol{Y}))=\mathcal{I}(f_\theta(\boldsymbol{Y}),\boldsymbol{Y})+\cancelto{0}{H(f_\theta(\boldsymbol{Y})|\boldsymbol{Y})}, \tag{5.6}$$

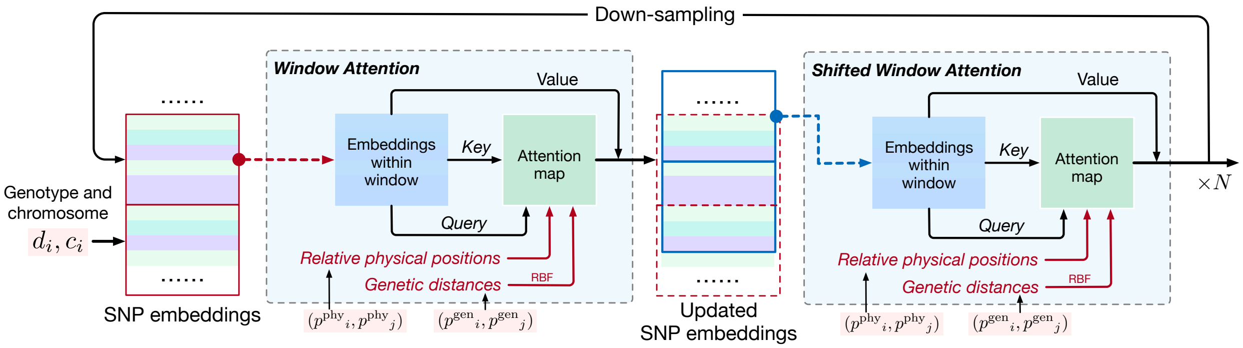


Figure 5.4: A swin-transformer block in the proposed genetic encoder. Red and blue boxes represent the windows and shifted windows.

and the reconstruction term is to maximize the log-likelihood that $f_\theta(\boldsymbol{Y})$ and $\boldsymbol{Y}$ in a positive pair belong to the same individual, similarly to the first term in the right-hand side (RHS) of Eq. (5.2). From the perspective of [153], the two terms in $\mathcal{L}_{\text{JSE}}(\boldsymbol{B}; \theta, \phi)$ and the reconstruction term aim at three properties of $f_\theta(\boldsymbol{Y})$; namely, the alignment to $\boldsymbol{G}$, the uniformity, and the alignment to $\boldsymbol{Y}$, respectively. As the uniformity of $f_\theta(\boldsymbol{Y})$ is encouraged by the estimation of $\mathcal{I}(f_\theta(\boldsymbol{Y}), \boldsymbol{G})$, we omit the corresponding term in the estimation of $\mathcal{I}(f_\theta(\boldsymbol{Y}), \boldsymbol{Y})$ for memory efficiency.

5.4 Genetics-Informed Transformer

A typical current genotyping microarray of the human genome includes more than 650k SNPs, with the physical spacing between any two consecutive SNPs being inconsistent. Genetic encoders developed in existing studies [154, 142] based on convolutional neural networks (CNNs) and multilayer perceptrons (MLPs) are incapable of handling the unstructured genetic data with extremely large sizes. To address this, we develop an effective genetics-informed transformer to encode genetic data in accordance with the following objectives:

1. Significant optimized computational cost, in recognition of certain biological assumptions.

2. Information aggregation among SNPs from arbitrary positions in the genome, considering multiple genetics dependency measurements.

3. Flexibility to accept any segments or subsets of the genetic data as input, thereby facilitating

cropping or downsampling-based augmentations on the genetic data.

An overview of the proposed transformer block is shown in Fig. 5.4, where attention operators with shifting windows [155] are used to enable efficient computation, and the aggregation is specialized with both physical and genetic distances of SNPs. In the transformer, two blocks are connected by down-sampling with attention-based pooling operators, and the initial SNP embeddings are computed based on both the genotypes and the chromosome each SNP belongs to. Finally, an attention-based readout is used to compute the global representation.

Window attention in swin-transformer. The 1D swin-transformer performs self-attention operations within each window split from the entire genetic array to enable efficient computing. It contains two components; those are, window attention and shifted window attention, as shown in Fig. 5.4. Given input SNP embeddings $\boldsymbol{H} \in \mathbb{R}^{L\times q}$ where L denotes the number of SNPs and q denotes the embedding dimension, window attention first splits $\boldsymbol{H}$ into a set of windows $\{\boldsymbol{H}_i \in \mathbb{R}^{w\times q}\}_{i=1,\cdots,\lfloor L/w\rfloor}$ where each window has a size w. A self-attention block is then applied to each $\boldsymbol{H}_i$ to update the SNP embeddings by aggregating information within that window. The updated windows are then merged back following the order when splitting $\boldsymbol{H}$, forming $\boldsymbol{H}'$ as the final output of the window attention component. In the following shifted window attention component, $\boldsymbol{H}'$ is first shifted by a length of $\lfloor w/2\rfloor$. Then similar splitting and self-attention are performed as in window attention to update SNP embeddings from each individual window. Finally, the updated windows are merged back, and the merged sequence is also shifted back by a length of $\lfloor w/2\rfloor$. There exists a biological assumption that strong and informative dependencies between SNPs exist only when they are within a certain distance. Hence, compared with performing global attention on all 650k marker positions, performing attention within windows in our proposed methods aggregates similar information, but largely reduces the computing cost.

Aggregation based on multiple dependencies. Multiple attention heads are computed to capture various types of dependencies among markers. Specifically, the computing of attention scores captures dependencies from three perspectives; those are, the SNP embeddings reflecting potential co-mutation, the encodings of the physical positions of SNPs on a chromosome indicating local dependencies among SNPs, and the encodings of genetic positions of SNPs measuring genetic linkages. Formally, the attention score $\alpha_{i,j}^k$ between the i-th and j-th SNPs for the k-th attention head is computed as

$$\alpha_{i,j}^k = f_\alpha^k(\boldsymbol{g}_i, \boldsymbol{g}_j) = f_{\alpha_g}^k(\boldsymbol{h}_i, \boldsymbol{h}_j) + f_{\alpha_p}^k(p_i^{\text{phy}}, p_j^{\text{phy}}) + f_{\text{RBF}}^k(p_i^{\text{gen}} - p_j^{\text{gen}}), \tag{5.7}$$

where $\boldsymbol{h}_i = \boldsymbol{H}[i] \in \mathbb{R}^q$ denote the SNP embedding. We compute the three individual components of attention scores that can capture different genetic dependencies as

$$\begin{gathered} f_h^k(\boldsymbol{h}_i, \boldsymbol{h}_j) = \left(\boldsymbol{h}_i^T \boldsymbol{W}_h^{k,l}\right)\left(\boldsymbol{h}_j^T \boldsymbol{W}_h^{k,r}\right)^T, f_{\text{pe}}^k(p_i^{\text{phy}}, p_j^{\text{phy}}) = \left[\boldsymbol{e}_{\text{p}}(p_i^{\text{phy}})^T \boldsymbol{W}_{\text{pe}}^{k,l}\right]\left[\boldsymbol{e}_{\text{p}}(p_i^{\text{phy}})^T \boldsymbol{W}_{\text{pe}}^{k,r}\right]^T, \\ f_{\text{RBF}}^k(p_i^{\text{gen}} - p_j^{\text{gen}}) = \left[\boldsymbol{r}\left(\left|p_i^{\text{gen}} - p_j^{\text{gen}}\right|\right)\right]^T \left[\mathbb{1}_{(p_i^{\text{gen}} - p_j^{\text{gen}}) \geq 0}^T \boldsymbol{W}_{\text{rbf}}^{k,+} + \mathbb{1}_{(p_i^{\text{gen}} - p_j^{\text{gen}}) < 0}^T \boldsymbol{W}_{\text{rbf}}^{k,-}\right], \end{gathered} \tag{5.8}$$

where $\boldsymbol{e}_p(\cdot)$ denotes the position encoding [9], $\boldsymbol{W}_g^{k,l}, \boldsymbol{W}_g^{k,r}, \boldsymbol{W}_{\text{pe}}^{k,l}, \boldsymbol{W}_{\text{pe}}^{k,r}$, and $\boldsymbol{W}_{\text{rbf}}^{k,+}, \boldsymbol{W}_{\text{rbf}}^{k,-}$ are trainable projections. $\mathbb{1}_{condition}^T$ is an indicator vector where all elements are 1s if the condition holds, and are 0s otherwise. The function $\boldsymbol{r}$ denotes a distance expansion with radial basis functions (RBF) [156]. Denoting $s := \left|p_i^{\text{gen}} - p_j^{\text{gen}}\right|$, the term $\boldsymbol{r}\left(\left|p_i^{\text{gen}} - p_j^{\text{gen}}\right|\right)$ in the above equation is computed as

$$\boldsymbol{r}(s) = \left[\exp\left\{(s-t)^2/\sigma^2\right\}\right]_{t \in \{t_0, \cdots, t_c\}} \in \mathbb{R}^c, \tag{5.9}$$

where $\{t_0, \cdots, t_c\}$ is a set of non-negative real numbers ranging from 0 and a preset threshold. The asymmetric projections in all f_h, f_{pe} and f_{RBF} functions indicate that $\alpha_{i,j}$ does not necessarily equal to $\alpha_{j,i}$, leading to more expressive models to capture dependencies. In addition, the computing of each attention head is based on a combination of three types of dependencies, which enables information aggregation among different SNPs based on genetic dependencies. By doing this, the complicated genetic dependencies of the input genome data can be captured.

5.5 Related Work

Dimension reduction for GWAS. When doing genome-wide association studies, people oftentimes find themselves dealing with high-dimensional quantitative traits. In order to reduce computational cost and redundancy, and in the hope of finding meaningful underlying patterns, many works perform dimension reduction of the high dimensional traits before doing GWAS, including principal component analysis [157, 158, 159], independent component analysis [160, 161] and non-negative matrix factorization [162]. These approaches are effective in capturing linear dependencies but are less capable of identifying complicated traits from imaging data.

Deep learning-based approaches. Recently, several works used unsupervised learning to characterize high-dimensional medical data. iGWAS [140] applied contrastive learning between multiple images of the same person to reveal potential genetic signals, ContIG [142] applied contrastive learning between medical imaging data and genetic data to learn the feature representation. DeepEndo autoencoder [139] used a convolutional autoencoder to reduce the dimensionality of the imaging data and found genetic associations of these extracted phenotypes. TransferGWAS [141] used both supervised task and reconstruction task to learn the feature representation. Specifically, ContIG [142] is the first to use contrastive learning between images and genetics on the GWAS problem. However, there are distinguishable differences between the work and ours. First, ContIG aims to learn general representation for multiple downstream tasks, such as classifications of the risk of several diseases. With this goal, ContIG treats the problem as a typical visual representation learning task. On the contrary, our study focuses on the representation learning specifically for GWAS. Second, our approaches are built upon the grounding of mutual information maximization, whereas ContIG is grounded by contrastive learning for unlabelled data. Third, our work focuses on a more challenging setting with 3D MRI data, where typical contrastive approaches may fail.

Mutual Information Maximization. Previous research has employed mutual information maximization as a pretext task for representation learning on various data types, including images [89], videos [163], and graphs [40, 164]. However, these studies primarily focus on classification or regression as downstream tasks. Our work presents unique challenges and goals of mutual information maximization under the GWAS setting and we are the first to examine GWAS from a mutual information perspective.

5.6 Experiments on GWAS Representation Learning

We use the brain imaging dataset from UK Biobank [165] in this study, it is currently the largest public brain imaging dataset. Specifically, we use T1-weighted MRI imaging data accessed on October 15, 2021. We register and pre-process the MRI data into the shape of 182×218×182. For representation learning, we split the MRI-genetic data pairs into 4,597 training and 1,533 validation pairs based on ethnicity.

5.6.1 Data Processing and Split

All MRIs were linearly registered (affine registration with 12 DOF) to standard MNI152 space using the UKBiobank-provided transformation matrix with FSL FLIRT [166] and all the outputs are of shape 182×218×182. A large portion of the UKBiobank population is white British. In order to maximize the power of genetic discovery and avoid the complication of population stratification, the genetic association study was only done on the white British (UKBiobank data field 21000 and 22006) cohort. So we selected 6,130 images from subjects of mixed ethnicities (all non-white British samples plus a small number of random white British samples) not overlapped with the samples for the genetic discovery to do the training and validation, among which 4,597 was randomly selected for training and 1,533 for validation. We used two quality metrics "inverted contrast-to-noise ratio" (UKBiobank data field 25735) and "Discrepancy between T2 FLAIR brain image and T1 brain image" (UKBiobank data field 25736) to ensure the quality of the training data.

Evaluation Metrics We involve three metrics to evaluate the representations learned by different models; namely, the **number of loci** discovered by GWAS, the **estimated mutual information**, and the **heritability** of the representations. Among the three metrics, the number of loci is primary as it indicates All three metrics are computed on a testing dataset that is unseen during the representation learning process and measures the quality of representation for GWAS purposes. To enable efficient evaluation, we obtain the first 10 principal components of representations and compute all metrics on the 10-dimensional vectors. Details about the evaluation metrics are provided below.

Number of Loci We perform genome-wide scans over 658,720 directly genotyped SNPs * and on 28,489 white British participants unseen during training. We use BOLT-LMM (Version 2.3.4) [167] for running GWAS. Age, gender, and the first 10 ancestral principal components are used as covariates. We use Bonferroni corrected p-value threshold of $5\mathrm{e}{-9}$ and a minor allele frequency threshold of 1% to get the significant SNPs and filter out the rare variants. We then cluster the significant SNPs into loci using a 250 kb window, which is approximately 0.25 cM [160]. The number of loci indicates the amount of genetic contribution to the learned features.

Heritability measures the proportion of variation of the feature explained by the genetic factors. It provides insight into the genetic basis of a feature. A higher heritability indicates that the representation is better associated with the genetic data. The heritability is computed using LDSC v1.0.1 [168].

Mutual Information We estimate the mutual information between MRI representations and genetic data on the test set to explicitly demonstrate that the proposed objective adds to the generalizability of captured associations to unseen pairs. We train individual JSE-based mutual information estimators with the same architecture for different methods. We train the MI estimator until the contrastive loss converges and take the opposite of the converged value as the MI estimation for each model.

5.6.2 Implementation Details

We compare our approach with multiple baseline approaches in four groups; namely, MRI encoders that are randomly initialized, trained by predictive approaches, contrastive approaches with MRI data only, and trans-modal contrastive approaches that involves genetic data. The baselines include existing or straightforward training schemes Autoencoder [141, 139], `Gen Prediction` that uses genetic data prediction as a pretext task, Barlow-Twins [87], SimCLR [13], and ContIG [142]. We additionally include their variants `Autoencoder-attention` that uses the

* Applied Biosystems UK BiLEVE Axiom Array, UKBiobank data field 22438

same MRI encoder as ours, `SimCLR-JSE`, where the contrastive objective in SimCLR is replaced by JSE, and `Decorrelated InfoNCE`, where a decorrelation term is added to the contrastive loss. The implementation details of both our methods and baselines are described below.

MRI encoders The MRI encoder is constructed as a 3D convolutional network consisting of three residual blocks [169] connected by two downsampling operators with stride convolutions. The numbers of channel maps are 32, 64, and 128, respectively for the three blocks. The final representations are 128-dimensional computed by a dense layer upon flattened feature maps. When computing the reconstruction loss, we include additional 128-dimensional vectors computed from a multi-head attentive readout from feature maps, and the reconstruction is performed on the 256-dimensional representation after concatenation. However, dimensions from attention are not used in GWAS computation. This is to further prevent the encoder from learning too detailed patterns, possibly noise, that are non-generalizable to the test set.

Genetic encoders Our genetic encoder consists of three 1D swin-transformer blocks connected by two down-sampling operators with a down-sampling rate of 10. The positional encoding for SNP physical positions is 128. The embedding dimensions are 32, 64, and 128 for the three blocks, respectively. The window size to perform attention is 10 and the number of heads is 4 for all self-attention operators. The downsampling operator computes the attention with learnable queries within each window, where the window size is equal to the downsampling size. The global pooling operators compute attention with learnable queries among all positions at multiple scales and resolutions.

Training The models are implemented with PyTorch [170] and are trained on a single Nvidia A100 GPU. The training is performed with the Adam optimizer [111], cosine annealing scheduler [171] with a starting learning rate of 0.001 and the mini-batch size of 12. We simply set λ in the objective to 1 and do not exhaustively tune it. During training, we randomly crop the 3D MRI into smaller patches of size $[160, 160, 160]$. We first train the models with the genetic encoder frozen for 200 epochs, then include the augmentation on genetic data and continue training for additional 100 epochs, and finally co-train both MRI and genetic encoders and projection heads for 50 epochs with augmentation on genetic data. To perform augmentation, the genetic data has a probability of 0.2 to be randomly cropped and a probability of 0.8 to be evenly down-sampled into a length of 65,000.

Baseline approaches For Gen Prediction, we apply a linear layer to the output representation as a prediction head $h^g : \mathbb{R}^q \rightarrow |G|$ to predict the class of each genotype in the genetic data and optimize the following loss.

$$\mathcal{L}_{\text{GenPred}} = \text{Cross-Entropy}\Big(h^g\big(f_\theta(\boldsymbol{Y})\big), \boldsymbol{d}\Big).$$

For baseline trans-modal contrastive methods, we follow the architecture, training loss, and training settings in [142]. For the MRI augmentations, we perform the random flipping and rotation on the x-z plane, along with the random 3D patching. However, we found the flipping and rotation do not help on the GWAS performance in the 3D MRI case. For correlated InfoNCE, we compute the covariance matrix of learned MRI representations and minimize its difference with the identity matrix,

$$\mathcal{L}_{\text{decor}} = ||\hat{\boldsymbol{z}}^T\hat{\boldsymbol{z}} - \boldsymbol{I}||^2,$$

where $\hat{\boldsymbol{z}}$ is the normalized MRI representations in the mini-batch. Since the mini-batch size is small due to memory constrain, the covariance estimation can be less accurate, still leading to reduced performance.

5.6.3 Quantitative Results on T1-Weighted MRI

The comparisons among representations learned by different methods in terms of the three metrics are shown in Table 5.1. The results indicate that the proposed learning framework with regularized MI estimator and genetic transformer significantly improves the quality of learned representation in terms of the number of discovered loci and the heritability. The improved MI of our methods on test pairs also suggests a stronger generalization capability. Additionally, we have the following observations.

The level of mutual information on test pairs agrees with # loci and heritability. The results

Table 5.1: Comparisons of quantitative evaluation results on the test set. “Unique” refers to the number of loci discovered by a method that is NOT discovered by any methods in other groups. All metrics are the higher the better.

		# Independent Loci			
	Methods	All	Unique	MI (JSE)	Heritability h^2
	Random Init	14	1	1.2165	0.0756 ± 0.0656
Predictive	Autoencoder [139]	26	1	1.3120	0.3121 ± 0.0769
	Autoencoder-attention	23	4	1.3124	0.2984 ± 0.0773
	Gen Prediction	10	0	1.2412	0.0918 ± 0.1110
Contrastive	Barlow Twins [87]	11	1	1.2996	0.0814 ± 0.0636
	SimCLR [13]	15	1	1.2397	0.1448 ± 0.1128
	SimCLR-JSE	17	7	1.3044	0.1604 ± 0.1151
Trans-Modal Contrastive	InfoNCE (ContIG, [142])	11	0	1.2299	0.1334 ± 0.0588
	Decorrelated InfoNCE	13	3	1.2382	0.0527 ± 0.0349
	GIM (Ours)	**40**	**15**	**1.3681**	**0.3723 ± 0.0305**

suggest that higher mutual information on the test set implies a higher heritability and more loci discovered. It justifies our formulation of learning representation for GWAS as the problem of maximizing mutual information.

Typical trans-modal contrastive approaches fail for MRI data. Trans-modal contrastive learning with typical contrastive loss performs fairly well on 2D retina imaging [142] but suffers more from the performance reduction on the higher-dimensional 3D data. In the 3D MRI case, we found that the simplest Autoencoder approach performs even better than contrastive and typical trans-modal contrastive approaches. Moreover, [140] suggests that the contrastive learning between the retina of the left and right eyes can also result in better performance than the typical trans-modal contrastive approaches. These further strengthen our analyses on the limitations due to dimensional collapse and non-generalizable associations described in Section 5.3.1.

5.6.4 Additional Results and Ablations

Additional results on T2-weighted MRI. We additionally apply GIM to a second modality, namely the T2-weighted MRI. Similarly, we compute GWAS on the first 10 principle components of the learned representation on the test set. In contrast to the results for T1, we observe that

Table 5.2: Results for T2-weighted MRI.

Methods	# All Indp Loci
Autoencoder	29
Barlow-Twin	7
SimCLR	21
SimCLR-JSE	15
ContIG	22
GIM (Ours)	**38**

learning informative representations is less challenging for contrastive methods in the T2 case. In addition, contrastive methods equipped with NCE generally perform better than their JSE counterparts. This result is consistent with those presented in [142]. Nevertheless, results in Table 5.2 show a consistent out-performance of GIM over baselines, indicating generalizable effectiveness.

Effectiveness of individual proposed components. To show the effectiveness and necessity of both the proposed learning objective and the genetic transformer, we track the change in the number of all loci, unique loci, the heritability score, and newly discovered genes with the highest significance when incrementally adding each component. Table 5.3 shows the results of adding the regularization to the objective, replacing the MLP encoder with the genetics-informed transformer, and performing random cropping on the genetic data. The results suggest that adding each component generally increases the useful information carried by the representations, leading to more loci discovered. We mapped significant SNPs to genes using Plink v1.9 [172], and we presented the genes that are associated with the most significant new SNP of each model in the ablation study in Table 5.3. *CENPW* is known to associated with neurogenesis [173] and cortical morphology [174], *WNT16* with skull and brain shape [175, 176], *ITPR3* with neuropathy [177] and many psychiatric disorders [178] and *MSRB3* with Alzheimer's [179]. We also queried each locus in the result of the Big40 study [180, 160], which uses thousands of conventional image-derived phenotypes to do GWAS and we found a locus not presented in the Big40 study in Chromosome 2, base pair 218466221 to 218604356 (in hg19 coordinate). This locus is mapped to *DIRC3*, which has been shown to be associated with Alzheimer's disease [181, 182]. This showcases the potential of our

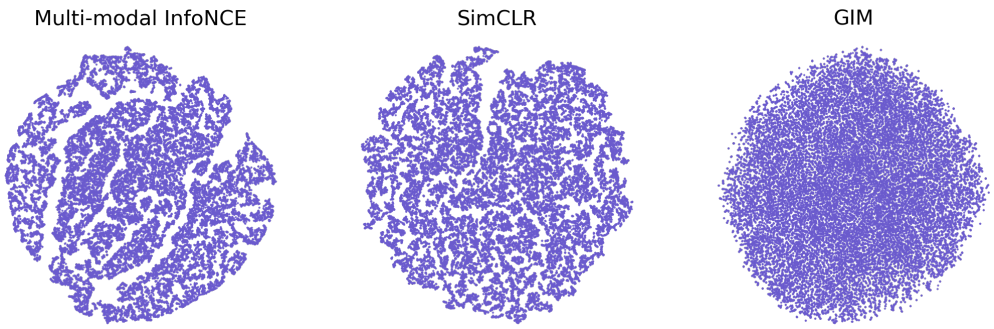


Figure 5.5: Visualization of learned representations with t-SNE. Forming clusters is not desired in the GWAS setting.

method in capturing features missed by the traditional expert-defined pipelines.

Table 5.3: Change in the number of loci and heritability when incrementally adding components to the models. The positive and negative numbers are the counts of newly discovered and missing loci when a component is added. The last column shows the most significant gene newly discovered.

Methods	# Loci	Change	# Unique Loci	Change	h^2	Significant Gene
Base-Contrastive	11	-	0	-	0.1334	-
+ Regularization	29	+19 / -1	6	+6 / -0	0.3390	*CENPW*
+ Genetic Transformer	32	+14 / -9	10	+7 / -3	0.3773	*WNT16*
+ Random Cropping	36	+17 / -13	13	+10 / -7	0.3807	*ITPR3*
+ Co-training	40	+20 / -16	15	+11 / -9	0.3723	*MSRB3*

Distribution of learned representations. We visualize the distribution of representations learned by trans-modal InfoNCE, SimCLR, and GIM, respectively, with t-SNE in Figure 5.5. Compared to baseline approaches, GIM learns representations that are more uniformly distributed in the space. According to the discussion on the difference between learning goals, the goal of our representation learning is not to form clusters for downstream classification purposes but to uniformly encode as much information about the genetic data as possible [153]. Under this setting, clusters of representations are not desired and may harm the GWAS performance due to reduced capacity for other characteristics.

5.7 Conditional MRI Generation: A Followup Task

Recently, diffusion models [183] have shown promising performance in multiple generative tasks such as text-controlled image generation [184], image editing [185, 186], and protein generation [187]. Specifically, [184] have studied the unconditional generation of T1-weighted brain MRI. In this section, we extend the success of the proposed learning framework and genetics-informed transformer by incorporating the diffusion models with the pre-trained genetics-informed transformer under GIM to enable MRI data generation conditioned on a given gene. We consider the conditional generation as an additional task to evaluate the pre-trained genetic encoder.

The diffusion model considers the generation process as a reverse process of gradually adding noise to the data $\boldsymbol{x}_0$ until it becomes fully noisy $\boldsymbol{x}_T \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ within T steps. It generates data from pure noise by performing step-by-step denoising for T steps. Specifically, the training of diffusion model is to learn a noise estimator f_θ^{diff} that recovers $\boldsymbol{x}_0$ from the noisy data $\boldsymbol{x}_t$ for any time step t with the objective

$$\min_\theta \mathbb{E}_{t\sim[1,T],\boldsymbol{x}_0,\boldsymbol{\epsilon}_t} \|\boldsymbol{\epsilon}_t - f_\theta^{\text{diff}}(\boldsymbol{x}_t, t)\|^2, \tag{5.10}$$

where $\boldsymbol{\epsilon}_t$ is the true noise contained in $\boldsymbol{x}_t$. When conditioned generation is performed, the noise estimator f_θ^{cond} additionally takes the condition embedding $\boldsymbol{c}$ as an input and the following objective is optimized

$$\min_\theta \mathbb{E}_{t\sim[1,T],(\boldsymbol{x}_0,\boldsymbol{c}),\boldsymbol{\epsilon}_t} \|\boldsymbol{\epsilon}_t - f_\theta^{\text{cond}}(\boldsymbol{x}_t, \boldsymbol{c}, t)\|^2, \tag{5.11}$$

where $\boldsymbol{x}_0$ and $\boldsymbol{c}$ are sampled from a joint distribution (*i.e.*, using paired data). In the case of MRI generation, we let $\boldsymbol{c} \in \mathbb{R}^{N_{tk}\times q}$, equivalent to N_{tk} tokens, be the down-sampled and updated SNP embeddings at the last transformer block before global readout. The flexible input SNP positions to the genetics-informed transformer allows for conditioning the generation with a segment or a down-sampled version of the genetic data. In the noise estimator f_θ^{cond}, the condition is incorporated with cross-attention blocks between latent feature maps $\boldsymbol{X}_{t,\ell} \in \mathbb{R}^{DHW\times d_q}$ and $\boldsymbol{c}$ at each

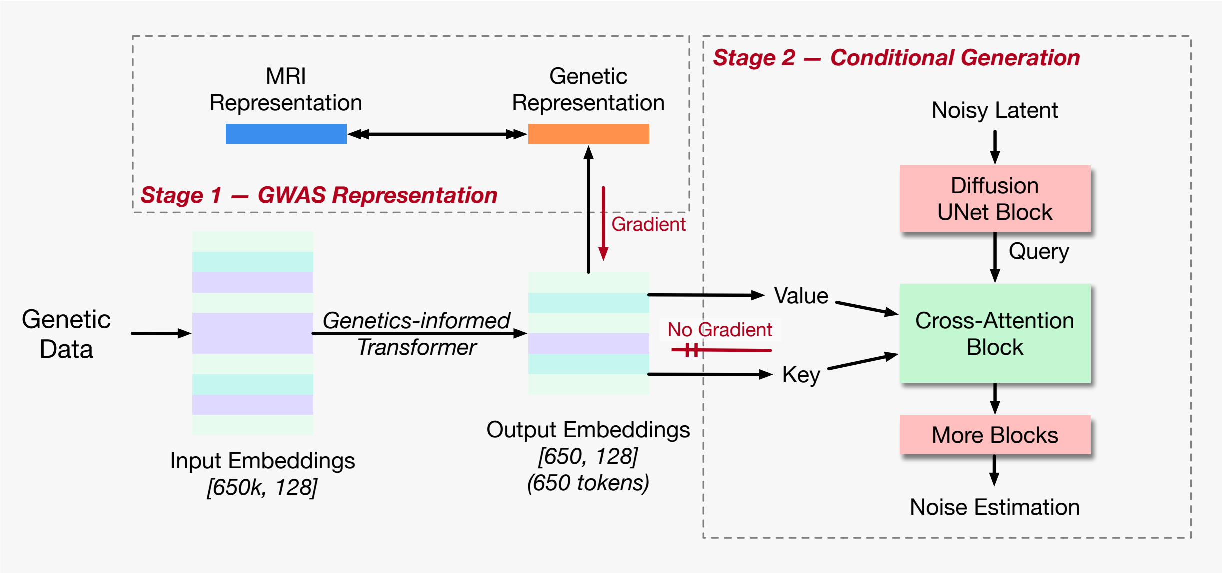


Figure 5.6: Framework overview for conditional MRI generation. Given the genetics-informed transformer pre-trained using GIM, we freeze its parameter and compute the condition embedding for conditional generation. In the cross-attention block, each voxel in the 3D feature map will attend and aggregation information from the condition embedding based on the relavance.

convolutional block ℓ. Concretely, we have the output of each cross-attention block as

$$\boldsymbol{X}'_{t,\ell} = \text{Normalize}\Big(\mathcal{Q}\big(\boldsymbol{X}_{t,\ell}\big)\mathcal{K}\big(h(\boldsymbol{c})\big)^T\Big)\mathcal{V}\big(h(\boldsymbol{c})\big), \tag{5.12}$$

where $\mathcal{Q}, \mathcal{K}, \mathcal{V}$ are linear projections in a cross-attention operator, h is a parametric projection head upon condition embeddings. The framework is illustrated in Figure 5.6.

Generation Performance as Genetic Encoder Evaluation Since the quality of MRI generation is closely associated with the performance of genetic encoder, in terms of how much genetic information is captured, we consider the quality of generated MRI data as an additional metric to evaluate the genetic encoder. Specifically, we measure the conditional generation quality using the mean squared error between the generated MRI from the gene of a certain individual and its physically captured counterpart. The baselines include the mean squared error between a unconditionally generated MRI by [184] and real MRI data, together with the averaged squared difference between any pair of real MRI data. For the genetics-informed transformer, we fix its parameters in all cases but add different learnable projection heads h, and vary the input genetic data with downsampling. The comparison shown in Table 5.4 suggests that including the pre-trained genetic encoder can consistently enable effective conditional generation with improved generation quality, even when no parameter is further tuned or added to the genetic encoder. Additional parameters included in the projection heads can further improve conditional generation performance. This indicates that useful information associated with the brain MRI can be effectively captured from the genetic data by the proposed genetics-informed transformer. Moreover, downsampling the input genetic data does not significantly reduce the generation quality, suggesting a flexibility of condition inputs.

Table 5.4: Conditional generation performance in terms of mean squared error (MSE). `Readout` indicate a global readout is applied in the projection head and a single global embedding is used as the condition. `Downsampled` indicates a 10 times down-sampling is applied to the input genetic data.

Models	h	Downsampled	# Tokens	MSE
Avg. dist.	–	–	–	0.3505
Diffusion - Unconditional	–	–	–	0.2928
Diffusion - Conditional	Readout	No	1	0.2858
	Readout + MLP	No	1	0.2809
	MLP	Yes	60	0.2747
	MLP	No	650	**0.2720**

Potential application scenarios Existing genetics studies enable the effective and efficient discovery of SNPs that are associated with certain human tissues. However, the complicated nature of the deep phenotype encoders makes it challenging to interpret which specific substructures, such as subarea in the brain MRI, is associated with each SNP identified by GWAS. This lack of interpretability prevents further exploration of AD's cause and highlights the need for appropriate trans-modal explanation approaches for GWAS. The successful conditional generation allows for multiple application scenarios to bridge the gap in genetics studies such as counterfactual explanations and phenotype simulation for gene editing. Specifically, when given different genetic data as conditions, the generative model tends to produce different MRI data that accurately match the given condition, even from the same initial noise. This allows us to track the different in the generated MRI data when perturbing certain SNP locations or segment of interest. The difference can be further used as an explanation outcome in genome-wide associations.

5.8 Conclusions

In this chapter, we have investigated the differences and limitations of GWAS representation learning to compare to typical visual representation learning and have presented Genetic InfoMax, a GWAS representation learning framework. We have established standardized evaluation protocols to benchmark existing and our approaches. Our experiments demonstrate a significant boost in GWAS performance by GIM.

REFERENCES

[1] S. Laine, T. Karras, J. Lehtinen, and T. Aila, “High-quality self-supervised deep image denoising,” *Advances in Neural Information Processing Systems*, pp. 6970–6980, 2019.

[2] J. Batson and L. Royer, “Noise2Self: Blind denoising by self-supervision,” in *Proceedings of the 36th International Conference on Machine Learning*, pp. 524–533, 2019.

[3] S. Thakoor, C. Tallec, M. G. Azar, R. Munos, P. Veličković, and M. Valko, “Bootstrapped representation learning on graphs,” *arXiv preprint arXiv:2102.06514*, 2021.

[4] Y. Jiao, Y. Xiong, J. Zhang, Y. Zhang, T. Zhang, and Y. Zhu, “Sub-graph contrast for scalable self-supervised graph representation learning,” *arXiv preprint arXiv:2009.10273*, 2020.

[5] Y. Yang and Z. Xu, “Rethinking the value of labels for improving class-imbalanced learning,” in *Advances in Neural Information Processing Systems*, 2020.

[6] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[7] Y. Xie, Z. Wang, and S. Ji, “Noise2Same: Optimizing a self-supervised bound for image denoising,” in *Advances in Neural Information Processing Systems*, pp. 20320–20330, 2020.

[8] A. Krull, T.-O. Buchholz, and F. Jug, “Noise2Void-learning denoising from single noisy images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2129–2137, 2019.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[10] J. Wu, X. Wang, and W. Y. Wang, "Self-supervised dialogue learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3857–3867, 2019.

[11] H. Wang, X. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, and W. Y. Wang, "Self-supervised learning for contextualized extractive summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2221–2227, 2019.

[12] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning*, 2020.

[14] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[15] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," *arXiv preprint arXiv:1906.05849*, 2019.

[16] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

[17] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.

[18] A. Tsitsulin, D. Mottin, P. Karras, A. Bronstein, and E. Müller, "Sgr: Self-supervised spectral graph representation learning," *arXiv preprint arXiv:1811.06237*, 2018.

[19] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

[20] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*, pp. 69–84, Springer, 2016.

[21] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[22] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, pp. 21271–21284, 2020.

[23] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: A benchmark for molecular machine learning," *Chemical Science*, vol. 9, no. 2, pp. 513–530, 2018.

[24] Z. Wang, M. Liu, Y. Luo, Z. Xu, Y. Xie, L. Wang, L. Cai, and S. Ji, "Advanced graph and sequence neural networks for molecular property prediction and drug discovery," *arXiv preprint arXiv:2012.01981*, 2020.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[26] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *International Conference on Learning Representations*, 2019.

[27] H. Gao and S. Ji, "Graph U-Nets," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 2083–2092, 2019.

[28] M. Liu, Z. Wang, and S. Ji, "Non-local graph neural networks," *arXiv preprint arXiv:2005.14612*, 2020.

[29] L. Cai, J. Li, J. Wang, and S. Ji, "Line graph neural networks for link prediction," *arXiv preprint arXiv:2010.10046*, 2020.

[30] M. Liu, H. Gao, and S. Ji, “Towards deeper graph neural networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 338–348, 2020.

[31] L. Cai and S. Ji, “A multi-scale approach for graph link prediction,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 3308–3315, 2020.

[32] H. Gao, Z. Wang, and S. Ji, “Large-scale learnable graph convolutional networks,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1416–1424, 2018.

[33] H. Gao, Y. Chen, and S. Ji, “Learning graph pooling and hybrid convolutional operations for text representations,” in *Proceedings of the Web Conference*, pp. 2743–2749, 2019.

[34] H. Gao, Y. Liu, and S. Ji, “Topology-aware graph pooling networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[35] Z. Wang and S. Ji, “Second-order pooling for graph neural networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[36] H. Yuan and S. Ji, “StructPool: Structured graph pooling via conditional random fields,” in *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[37] H. Gao and S. Ji, “Graph representation learning via hard and channel-wise attention networks,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 741–749, ACM, 2019.

[38] H. Yuan, J. Tang, X. Hu, and S. Ji, “XGNN: Towards model-level explanations of graph neural networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 430–438, 2020.

[39] Y. Liu, H. Yuan, L. Cai, and S. Ji, “Deep learning of high-order interactions for protein interface prediction,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 679–687, 2020.

[40] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and D. Hjelm, “Deep graph infomax,” in *International Conference on Learning Representations*, 2019.

[41] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *arXiv preprint arXiv:1611.07308*, 2016.

[42] Y. Ma and J. Tang, *Deep Learning on Graphs*. Cambridge University Press, 2020.

[43] X. Xu, C. Deng, Y. Xie, and S. Ji, “Group contrastive self-supervised learning on graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[44] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What makes for good views for contrastive learning?,” *arXiv preprint arXiv:2005.10243*, 2020.

[45] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Graph contrastive learning with adaptive augmentation,” in *Proceedings of The Web Conference 2021*, WWW ’21, 2021.

[46] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in *The 37th annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377, 1999.

[47] N. Tishby and N. Zaslavsky, “Deep learning and the information bottleneck principle,” in *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2015.

[48] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[49] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[50] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[51] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, IEEE, 2016.

[52] Z. Wang, N. Zou, D. Shen, and S. Ji, “Non-local u-nets for biomedical image segmentation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6315–6322, 2020.

[53] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image restoration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[54] Z. Wang, Y. Xie, and S. Ji, “Global voxel transformer networks for augmented microscopy,” *Nature Machine Intelligence*, vol. 3, pp. 161–171, 2021.

[55] Y. Liu, H. Yuan, Z. Wang, and S. Ji, “Global pixel transformers for virtual staining of microscopy images,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 2256–2266, 2020.

[56] M. Weigert, U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, *et al.*, “Content-aware image restoration: pushing the limits of fluorescence microscopy,” *Nature methods*, vol. 15, no. 12, pp. 1090–1097, 2018.

[57] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, “Unprocessing images for learned raw denoising,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11036–11045, 2019.

[58] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, “Toward convolutional blind denoising of real photographs,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1712–1722, 2019.

[59] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, T. Aila, *et al.*, “Noise2noise,” in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 2971–2980, 2018.

[60] N. Moran, D. Schmidt, Y. Zhong, and P. Coady, “Noisier2noise: Learning to denoise from unpaired noisy data,” *arXiv preprint arXiv:1910.11908*, 2019.

[61] J. Xu, Y. Huang, L. Liu, F. Zhu, X. Hou, and L. Shao, "Noisy-as-clean: learning unsupervised denoising from the corrupted image," *arXiv preprint arXiv:1906.06878*, 2019.

[62] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1058–1067, 2017.

[63] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.," *Journal of Machine Learning Research*, vol. 11, no. 12, 2010.

[64] A. Krull, T. Vicar, and F. Jug, "Probabilistic noise2void: Unsupervised content-aware denoising," *arXiv preprint arXiv:1906.00651*, 2019.

[65] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in Neural Information Processing Systems*, pp. 5574–5584, 2017.

[66] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th IEEE International Conference on Computer Vision*, vol. 2, pp. 416–423, July 2001.

[67] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[68] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 60–65, IEEE, 2005.

[69] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," in *Image Processing: Algorithms and Systems, Neural Networks, and*

Machine Learning, vol. 6064, p. 606414, International Society for Optics and Photonics, 2006.

[70] Y. Xie, Z. Wang, and S. Ji, “Noise2Same: Optimizing a self-supervised bound for image denoising,” in *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pp. 20320–20330, 2020.

[71] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[72] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, 2013.

[73] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *International Conference on Learning Representations*, 2013.

[74] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Deep graph contrastive representation learning,” in *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

[75] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph contrastive learning with augmentations,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 5812–5823, 2020.

[76] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, “Strategies for pretraining graph neural networks,” in *International Conference on Learning Representations*, 2020.

[77] W. Jin, T. Derr, H. Liu, Y. Wang, S. Wang, Z. Liu, and J. Tang, “Self-supervised learning on graphs: Deep insights and new direction,” *arXiv preprint arXiv:2006.10141*, 2020.

[78] D. Kim and A. Oh, “How to find your friendly neighborhood: Graph attention design with self-supervision,” in *International Conference on Learning Representations*, 2021.

[79] Y. Xie, Z. Xu, J. Zhang, Z. Wang, and S. Ji, “Self-supervised learning of graph neural networks: A unified review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[80] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang, “Self-supervised learning: Generative or contrastive,” *arXiv preprint arXiv:2006.08218*, vol. 1, no. 2, 2020.

[81] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

[82] D. Hwang, J. Park, S. Kwon, K.-M. Kim, J.-W. Ha, and H. J. Kim, “Self-supervised auxiliary learning with meta-paths for heterogeneous graphs,” *arXiv preprint arXiv:2007.08294*, 2020.

[83] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang, “Self-supervised graph transformer on large-scale molecular data,” in *Advances in Neural Information Processing Systems*, 2020.

[84] M. Liu, Y. Luo, L. Wang, Y. Xie, H. Yuan, S. Gui, H. Yu, Z. Xu, J. Zhang, Y. Liu, K. Yan, H. Liu, C. Fu, B. M. Oztekin, X. Zhang, and S. Ji, “DIG: A turnkey library for diving into graph deep learning research,” *Journal of Machine Learning Research*, vol. 22, no. 240, pp. 1–9, 2021.

[85] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, “Mgae: Marginalized graph autoencoder for graph clustering,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 889–898, 2017.

[86] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, “Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization,” in *International Conference on Learning Representations*, 2019.

[87] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” *arXiv preprint arXiv:2103.03230*, 2021.

[88] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, "On the information bottleneck theory of deep learning," in *International Conference on Learning Representations*, 2018.

[89] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference on Learning Representations*, 2019.

[90] C. Wei, K. Shen, Y. Chen, and T. Ma, "Theoretical analysis of self-training with deep networks on unlabeled data," in *International Conference on Learning Representations*, 2021.

[91] N. Wale and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," in *Sixth International Conference on Data Mining*, pp. 678–689, 2006.

[92] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, pp. i47–i56, 06 2005.

[93] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of Molecular Biology*, vol. 330, no. 4, pp. 771–783, 2003.

[94] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds correlation with molecular orbital energies and hydrophobicity," *Journal of Medicinal Chemistry*, vol. 34, pp. 786–797, 02 1991.

[95] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

[96] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

[97] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the construction of internet portals with machine learning,” *Information Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.

[98] C. L. Giles, K. D. Bollacker, and S. Lawrence, “Citeseer: An automatic citation indexing system,” in *Proceedings of the Third ACM Conference on Digital Libraries*, p. 89–98, Association for Computing Machinery, 1998.

[99] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[100] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, “Pitfalls of graph neural network evaluation,” *Relational Representation Learning Workshop, NeurIPS 2018*, 2018.

[101] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, “Image-based recommendations on styles and substitutes,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, 2015.

[102] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, and K. Wang, “An overview of microsoft academic service (mas) and applications,” in *Proceedings of the 24th International Conference on World Wide Web*, pp. 243–246, 2015.

[103] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, “Efficient graphlet kernels for large graph comparison,” in *Artificial intelligence and statistics*, pp. 488–495, PMLR, 2009.

[104] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, “Weisfeiler-lehman graph kernels.,” *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.

[105] B. Adhikari, Y. Zhang, N. Ramakrishnan, and B. A. Prakash, “Sub2vec: Feature learning for subgraphs,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 170–182, Springer, 2018.

[106] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

[107] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, “Graph representation learning via graphical mutual information maximization,” in *Proceedings of The Web Conference 2020*, pp. 259–270, 2020.

[108] M. Zitnik and J. Leskovec, “Predicting multicellular function through multi-layer tissue networks,” *Bioinformatics*, vol. 33, no. 14, pp. i190–i198, 2017.

[109] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, “Graphsaint: Graph sampling based inductive learning method,” *arXiv preprint arXiv:1907.04931*, 2019.

[110] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[111] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[112] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, pp. 249–256, 2010.

[113] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *International Conference on Learning Representations*, 2018.

[114] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning*, pp. 1263–1272, PMLR, 2017.

[115] L. Wang, Y. Liu, Y. Lin, H. Liu, and S. Ji, “ComENet: Towards complete and efficient message passing for 3D molecular graphs,” in *Advances in Neural Information Processing Systems*, 2022.

[116] Y. Luo, K. Yan, and S. Ji, “GraphDF: A discrete flow model for molecular graph generation,” *arXiv preprint arXiv:2102.01189*, 2021.

[117] Z. Wang, M. Liu, Y. Luo, Z. Xu, Y. Xie, L. Wang, L. Cai, and S. Ji, “Advanced graph and sequence neural networks for molecular property prediction and drug discovery,” *arXiv preprint arXiv:2012.01981*, 2020.

[118] Y. Xie, Z. Xu, and S. Ji, “Self-supervised representation learning via latent graph prediction,” in *Proceedings of the 39th International Conference on Machine Learning*, pp. 24460–24477, PMLR, 2022.

[119] H. Yu, L. Wang, B. Wang, M. Liu, T. Yang, and S. Ji, “GraphFM: Improving large-scale GNN training via feature momentum,” in *Proceedings of the 39th International Conference on Machine Learning*, pp. 25684–25701, PMLR, 2022.

[120] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, “Graph neural networks for social recommendation,” in *The World Wide Web Conference*, pp. 417–426, 2019.

[121] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983, 2018.

[122] H. Yuan, H. Yu, S. Gui, and S. Ji, “Explainability in graph neural networks: A taxonomic survey,” *arXiv preprint arXiv:2012.15445*, 2020.

[123] W. Lin, H. Lan, and B. Li, “Generative causal explanations for graph neural networks,” in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[124] W. Lin, H. Lan, H. Wang, and B. Li, “Orphicx: A causality-inspired latent variable model for interpreting graph neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13729–13738, 2022.

[125] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, “Parameterized explainer for graph neural network,” in *Advances in Neural Information Processing Systems*, pp. 19620–19631, 2020.

[126] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, “GNNExplainer: Generating explanations for graph neural networks,” in *Advances in Neural Information Processing Systems*, pp. 9244–9255, 2019.

[127] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, “On explainability of graph neural networks via subgraph explorations,” *arXiv preprint arXiv:2102.05152*, 2021.

[128] X. Wang, Y. Wu, A. Zhang, X. He, and T.-S. Chua, “Towards multi-grained explainability for graph neural networks,” in *Advances in Neural Information Processing Systems*, pp. 18446–18458, 2021.

[129] S. Miao, M. Liu, and P. Li, “Interpretable and generalizable graph learning via stochastic attention mechanism,” in *Proceedings of the 39th International Conference on Machine Learning*, pp. 15524–15543, PMLR, 2022.

[130] S. Nowozin, B. Cseke, and R. Tomioka, “f-GAN: Training generative neural samplers using variational divergence minimization,” in *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

[131] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

[132] T. Sterling and J. J. Irwin, “ZINC 15 – Ligand discovery for everyone,” *Journal of Chemical Information and Modeling*, vol. 55, no. 11, pp. 2324–2337, 2015.

[133] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, “Explainability methods for graph convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10772–10781, 2019.

[134] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *Proceedings of the 34th International Conference on Machine Learning*, pp. 3145–3153, 2017.

[135] W. Jin, R. Barzilay, and T. Jaakkola, “Multi-objective molecule generation using interpretable substructures,” in *Proceedings of the 37th International Conference on Machine Learning*, pp. 4849–4859, PMLR, 2020.

[136] P. Jain and H. R. Jadhav, “Quantitative structure activity relationship analysis of aminoimidazoles as bace-i inhibitors,” *Medicinal Chemistry Research*, vol. 22, no. 4, pp. 1740–1746, 2013.

[137] D. Huang, Y. Liu, B. Shi, Y. Li, G. Wang, and G. Liang, “Comprehensive 3D-QSAR and binding mode of BACE-1 inhibitors using R-group search and molecular docking,” *Journal of Molecular Graphics and Modelling*, vol. 45, pp. 65–83, 2013.

[138] A. Abdellaoui, L. Yengo, K. J. Verweij, and P. M. Visscher, “15 years of gwas discovery: Realizing the promise,” *The American Journal of Human Genetics*, 2023.

[139] K. Patel, Z. Xie, H. Yuan, S. M. S. Islam, W. Zhang, A. Gottlieb, H. Chen, L. Giancardo, A. Knaack, E. Fletcher, *et al.*, “New phenotype discovery method by unsupervised deep representation learning empowers genetic association studies of brain imaging,” *medRxiv*, pp. 2022–12, 2022.

[140] Z. Xie, T. Zhang, S. Kim, J. Lu, W. Zhang, C.-H. Lin, M.-R. Wu, A. Davis, R. Channa, L. Giancarlo, *et al.*, “igwas: image-based genome-wide association of self-supervised deep phenotyping of human medical images,” *medRxiv*, 2022.

[141] M. Kirchler, S. Konigorski, M. Norden, C. Meltendorf, M. Kloft, C. Schurmann, and C. Lippert, “transfergwas: Gwas of images using deep transfer learning,” *Bioinformatics*, vol. 38, no. 14, pp. 3621–3628, 2022.

[142] A. Taleb, M. Kirchler, R. Monti, and C. Lippert, “ContIG: Self-supervised multimodal contrastive learning for medical imaging with genetics,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20908–20921, 2022.

[143] W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Zou, “Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning,” in *Advances in Neural Information Processing Systems*, 2022.

[144] M. Zolfaghari, Y. Zhu, P. Gehler, and T. Brox, “Crossclr: Cross-modal contrastive learning for multi-modal video representations,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1450–1459, October 2021.

[145] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” *Advances in neural information processing systems*, vol. 32, 2019.

[146] M. Zhang, N. S. Sohoni, H. R. Zhang, C. Finn, and C. Ré, “Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations,” in *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[147] C.-H. Chang, G. A. Adam, and A. Goldenberg, “Towards robust classification model by counterfactual and invariant data generation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15212–15221, 2021.

[148] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, “What should not be contrastive in contrastive learning,” in *International Conference on Learning Representations*, 2020.

[149] A. Foster, R. Pukdee, and T. Rainforth, “Improving transformation invariance in contrastive representation learning,” in *International Conference on Learning Representations*, 2020.

[150] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, “Understanding dimensional collapse in contrastive self-supervised learning,” in *International Conference on Learning Representations*, 2021.

[151] V. Vapnik, “An overview of statistical learning theory,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[152] W. Huang, M. Yi, and X. Zhao, “Towards the generalization of contrastive self-supervised learning,” *arXiv preprint arXiv:2111.00743*, 2021.

[153] T. Wang and P. Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” in *International Conference on Machine Learning*, pp. 9929–9939, PMLR, 2020.

[154] A. van Hilten, S. A. Kushner, M. Kayser, M. A. Ikram, H. H. Adams, C. C. Klaver, W. J. Niessen, and G. V. Roshchupkin, “Gennet framework: interpretable deep learning for predicting phenotypes from genetic data,” *Communications Biology*, vol. 4, no. 1, pp. 1–9, 2021.

[155] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[156] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, “Schnet: A continuous-filter convolutional neural network for modeling quantum interactions,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 992—1002, 2017.

[157] B. Zhao, T. Li, Y. Yang, X. Wang, T. Luo, Y. Shan, Z. Zhu, D. Xiong, M. E. Hauberg, J. Bendl, *et al.*, “Common genetic variation influencing human white matter microstructure,” *Science*, vol. 372, no. 6548, p. eabf3736, 2021.

[158] K. Yano, Y. Morinaka, F. Wang, P. Huang, S. Takehara, T. Hirai, A. Ito, E. Koketsu, M. Kawamura, K. Kotake, *et al.*, “Gwas with principal component analysis identifies a gene comprehensively controlling rice architecture,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 42, pp. 21262–21267, 2019.

[159] L. Ma, C. Qing, M. Zhang, C. Zou, G. Pan, and Y. Shen, “Gwas with a pca uncovers candidate genes for accumulations of microelements in maize seedlings,” *Physiologia Plantarum*, vol. 172, no. 4, pp. 2170–2180, 2021.

[160] L. T. Elliott, K. Sharp, F. Alfaro-Almagro, S. Shi, K. L. Miller, G. Douaud, J. Marchini, and S. M. Smith, “Genome-wide association studies of brain imaging phenotypes in uk biobank,” *Nature*, vol. 562, no. 7726, pp. 210–216, 2018.

[161] G. D. Pearlson, J. Liu, and V. D. Calhoun, “An introductory review of parallel independent component analysis (p-ica) and a guide to applying p-ica to genetic data and imaging phenotypes to identify disease-associated biological pathways and systems in common complex disorders,” *Frontiers in genetics*, vol. 6, p. 276, 2015.

[162] J. Wen, I. Nasrallah, A. Abdulkadir, T. Satterthwaite, G. Erus, T. Robert-Fitzgerald, A. Singh, A. Sotiras, A. Boquet-Pujadas, Z. Yang, *et al.*, “Mega-analysis of brain structural covariance, genetics, and clinical phenotypes,” *Research Square*, 2022.

[163] R. D. Hjelm and P. Bachman, “Representation learning with video deep infomax,” *arXiv preprint arXiv:2007.13278*, 2020.

[164] H. Stärk, D. Beaini, G. Corso, P. Tossou, C. Dallago, S. Günnemann, and P. Liò, “3d infomax improves gnns for molecular property prediction,” in *International Conference on Machine Learning*, pp. 20479–20502, PMLR, 2022.

[165] C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray, *et al.*, “Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age,” *PLoS medicine*, vol. 12, no. 3, p. e1001779, 2015.

[166] M. Jenkinson, P. Bannister, M. Brady, and S. Smith, “Improved optimization for the robust and accurate linear registration and motion correction of brain images,” *Neuroimage*, vol. 17, no. 2, pp. 825–841, 2002.

[167] P.-R. Loh, G. Tucker, B. K. Bulik-Sullivan, B. J. Vilhjalmsson, H. K. Finucane, R. M. Salem, D. I. Chasman, P. M. Ridker, B. M. Neale, B. Berger, *et al.*, “Efficient bayesian mixed-model analysis increases association power in large cohorts,” *Nature genetics*, vol. 47, no. 3, pp. 284–290, 2015.

[168] B. Bulik-Sullivan, H. K. Finucane, V. Anttila, A. Gusev, F. R. Day, P.-R. Loh, L. Duncan, J. R. Perry, N. Patterson, E. B. Robinson, *et al.*, “An atlas of genetic correlations across human diseases and traits,” *Nature genetics*, vol. 47, no. 11, pp. 1236–1241, 2015.

[169] Z. Wang, Y. Xie, and S. Ji, “Global voxel transformer networks for augmented microscopy,” *Nature Machine Intelligence*, vol. 3, pp. 161–171, 2021.

[170] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

[171] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.

[172] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly, *et al.*, “Plink: a tool set for whole-genome association and population-based linkage analyses,” *The American journal of human genetics*, vol. 81, no. 3, pp. 559–575, 2007.

[173] N. Aygün, A. L. Elwell, D. Liang, M. J. Lafferty, K. E. Cheek, K. P. Courtney, J. Mory, E. Hadden-Ford, O. Krupa, L. de la Torre-Ubieta, *et al.*, “Brain-trait-associated variants impact cell-type-specific gene regulation during neurogenesis,” *The American Journal of Human Genetics*, vol. 108, no. 9, pp. 1647–1668, 2021.

[174] B. B. Sun, S. J. Loomis, F. Pizzagalli, N. Shatokhina, J. N. Painter, C. N. Foley, M. E. Jensen, D. G. McLaren, S. S. Chintapalli, *et al.*, “Genetic map of regional sulcal morphology in the

human brain from uk biobank data," *Nature Communications*, vol. 13, no. 1, p. 6071, 2022.

[175] F. Gori, U. Lerner, C. Ohlsson, and R. Baron, "A new wnt on the bone: Wnt16, cortical bone thickness, porosity and fractures," *BoneKEy reports*, vol. 4, p. 669, 2015.

[176] C. Medina-Gomez, B. H. Mullin, A. Chesi, V. Prijatelj, J. P. Kemp, C. Shochat-Carvalho, K. Trajanoska, C. Wang, R. Joro, T. E. Evans, *et al.*, "Genome wide association metanalysis of skull bone mineral density identifies loci relevant for osteoporosis and craniosynostosis," *MedRxiv*, pp. 2021–11, 2021.

[177] J. Rönkkö, S. Molchanova, A. Revah-Politi, E. M. Pereira, M. Auranen, J. Toppila, J. Kvist, A. Ludwig, J. Neumann, G. Bultynck, *et al.*, "Dominant mutations in itpr3 cause charcot-marie-tooth disease," *Annals of Clinical and Translational Neurology*, vol. 7, no. 10, pp. 1962–1972, 2020.

[178] J. Cabana-Domínguez, B. Torrico, A. Reif, N. Fernàndez-Castillo, and B. Cormand, "Comprehensive exploration of the genetic contribution of the dopaminergic and serotonergic pathways to psychiatric disorders," *Translational Psychiatry*, vol. 12, no. 1, p. 11, 2022.

[179] S. L. Adams, L. Benayoun, K. Tilton, O. R. Chavez, J. J. Himali, J. K. Blusztajn, S. Seshadri, and I. Delalle, "Methionine sulfoxide reductase-b3 (msrb3) protein associates with synaptic vesicles and its expression changes in the hippocampi of alzheimer's disease patients," *Journal of Alzheimer's Disease*, vol. 60, no. 1, pp. 43–56, 2017.

[180] S. M. Smith, G. Douaud, W. Chen, T. Hanayik, F. Alfaro-Almagro, K. Sharp, and L. T. Elliott, "An expanded set of genome-wide association studies of brain imaging phenotypes in uk biobank," *Nature neuroscience*, vol. 24, no. 5, pp. 737–745, 2021.

[181] A. C. Naj, C. Reitz, F. Rajabli, G. R. Jun, P. Benchek, G. Tosto, J. Sha, C. Zhu, N. A. Kushch, W.-P. Lee, *et al.*, "Multi-ancestry genome-wide association analysis of late-onset alzheimer's disease (load) in 60,941 individuals identifies a novel cross-ancestry association in lrrc4c," *Alzheimer's & Dementia*, vol. 18, p. e065822, 2022.

[182] X.-L. Wang and L. Li, “Cell type-specific potential pathogenic genes and functional pathways in alzheimer’s disease,” *BMC neurology*, vol. 21, no. 1, pp. 1–18, 2021.

[183] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[184] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

[185] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, “Sdedit: Guided image synthesis and editing with stochastic differential equations,” in *International Conference on Learning Representations*, 2021.

[186] G. Couairon, J. Verbeek, H. Schwenk, and M. Cord, “Diffedit: Diffusion-based semantic image editing with mask guidance,” *arXiv preprint arXiv:2210.11427*, 2022.

[187] J. Ingraham, M. Baranov, Z. Costello, V. Frappier, A. Ismail, S. Tie, W. Wang, V. Xue, F. Obermeyer, A. Beam, *et al.*, “Illuminating protein space with a programmable generative model,” *bioRxiv*, pp. 2022–12, 2022.