# DISCOVERING THE REPRESENTATION BOTTLENECK OF GRAPH NEURAL NETWORKS FROM MULTI-ORDER INTERACTIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Most graph neural networks (GNNs) rely on the message passing paradigm to propagate node features and build interactions. Recent studies point out that different graph learning tasks require different ranges of interactions between nodes. In this work, we explore the capacity of GNNs to capture multi-order interactions between nodes, and the order represents the complexity of the context where interactions take place. We study two standard graph construction methods, namely, *K-nearest neighbor* (KNN) graphs and *fully-connected* (FC) graphs, and concentrate on scientific problems in the 3D Euclidean space. We demonstrate that the inductive bias introduced by KNN-graphs and FC-graphs prevents GNNs from learning interactions of the most appropriate complexity. We found that such a phenomenon is broadly shared by several GNNs for diverse graph learning tasks, so we name it a *representation bottleneck*. To overcome that, we propose a novel graph rewiring approach based on interaction strengths of various orders to adjust the receptive fields of each node dynamically. Extensive experiments in molecular property prediction and dynamic system forecast prove the superiority of our method over state-of-the-art graph rewiring baselines. This paper provides a reasonable explanation of why subgraphs play a vital role in determining graph properties. The code is available at `https://github.com/smiles724/bottleneck`.

## 1 INTRODUCTION

Over the past decade, *graph neural networks* (GNNs) (Kipf & Welling, 2016; Hamilton et al., 2017; Dwivedi et al., 2020) have witnessed growing popularity thanks to their ability to deal with graphs that have complex relationships and interdependence between objects, ranging from social networks (Fan et al., 2019) to computer programs (Nair et al., 2020). Particularly, GNNs show promising strength in scientific research. They are used to derive insights from structures of molecules (Wu et al., 2018) and reason about relations in a group of interacting objects.

As a consequence, their success provokes the bottleneck question: "What are the common limitations of GNNs in real-world modeling applications, such as molecules and dynamic systems?" Knowing that GNNs are typically expressed as a neighborhood aggregation or message passing scheme (Gilmer et al., 2017; Veličković et al., 2017), we leverage the interactions between input variables (Deng et al., 2021) to investigate the bottleneck of GNNs. That is, we aim to analyze which types of interaction patterns (e.g., certain physical or chemical concepts) are likely to be encoded by GNNs, and which others are difficult to manipulate.

As a relevant answer, the preceding work observes the liability of CNNs to capture too complex and too simple pairwise interactions (Deng et al., 2021). In this work, we first theoretically and empirically prove that this inclination is attributed to two factors: the data distribution of image datasets and the inductive bias of locality introduced by CNNs' small kernel size. Then we refine the measurement of multi-order interactions so that the metric works for both node-level and graph-level predictions, and study two common graph construction methods in scientific domains, i.e., *K-nearest neighbor* (KNN) graphs and *fully-connected* (FC) graphs. Then with massive empirical evidence from molecular representation learning and dynamic system modeling, we discover that, as

opposed to CNNs' behavior, GNNs are more vulnerable to the improper inductive bias induced by the assumption of graph connectivity and can deviate significantly from the data distribution. This imperfect inductive bias brought by KNN-graphs and FC-graphs prohibits GNNs from encoding some particular interaction patterns, so GNNs fail to achieve the global minimum loss. Accordingly, we name this phenomenon as *a representation bottleneck* of GNNs.

In order to fully release the expressiveness of GNNs and resolve the above-mentioned obstacle, we propose a novel graph rewiring technique based on the distribution of interaction strengths, which progressively optimizes the inductive bias of GNNs via calibrating the topological structures of input graphs. Experiments on both synthetic and real-world datasets validate its efficacy over existing graph rewiring baselines for GNN interpretability and generalization.

## 2 PRELIMINARY

**Multi-order interactions.** Suppose a graph has a set of $n$ variables (a.k.a. nodes). It can represent a macroscopic physical system with $n$ celestial bodies, or a microscopic biochemical system with $n$ atoms, denoted as $N = \{1, ..., n\}$. Given a well-trained GNN model $f$, let $f(N)$ represent the model output of all input variables. For node-level tasks, the GNN forecasts a value (e.g., atomic energy) or a vector (e.g., atomic force or velocity) for each node. For graph-level tasks, $f(N) \in \mathbb{R}$ is a scalar (e.g., drug toxicity or binding affinity). GNNs make predictions by interactions between input variables instead of working individually on each variable (Qi et al., 2018; Li et al., 2019; Lu et al., 2019; Huang et al., 2020a). Previous studies (Bien et al., 2013; Tsang et al., 2017; Zhang et al., 2020; Deng et al., 2021) concentrate on pairwise interactions and use the multi-order interaction $I^{(m)}(i, j)$ to measure interactions of different complexities between two input variables $i, j \in N$.

Specifically, the $m$-th order interaction $I^{(m)}(i, j)$ measures the average interaction utility between variables $i, j$ under all possible contexts consisting of $m$ variables. Mathematically, the multi-order interaction is defined as follows:

$$I^{(m)}(i, j) = \mathbb{E}_{S\subseteq N,\{i,j\}\subseteq S,|S|=m}[\Delta f(i, j, S)], 3 \le m \le n, \tag{1}$$

where $\Delta f(i, j, S) = f(S) - f(S\backslash\{i\}) - f(S\backslash\{j\}) + f(S\backslash\{i, j\})$ and $S \subset N$ is the context consisting of $m$ variables. $f(S)$ is the output when we keep variables in $S$ unchanged but alter variables in $N\backslash S$. Since it is irrational to feed an empty graph into a GNN, we demand the context $S$ to have at least one variable with $m \ge 3$ and omit the $f(\emptyset)$ term. Note that Zhang et al. (2020) assume variables $i, j$ do not belong to the context $S$. Contrarily, we argue that it is more reasonable to interpret $m$ as the contextual complexity of the interaction of variables $i, j$ are included in the context, and provide proof in Appendix A.2 that these two cases are equivalent but from different views. An elaborate introduction of $I^{(m)}$ (e.g., the connection with existing metrics) are in Appendix A.

**Representation bottleneck.** To measure the reasoning complexity of the DNN, researchers compute the relative interaction strength $J^{(m)}$ of the encoded $m$-th order interaction as follows:

$$J^{(m)} = \frac{\mathbb{E}_{x\in\Omega}\left[\mathbb{E}_{i,j}\left[\left|I^{(m)}(i, j \mid x)\right|\right]\right]}{\sum_{m'}\left[\mathbb{E}_{x\in\Omega}\left[\mathbb{E}_{i,j}\left[\left|I^{(m')}(i, j \mid x)\right|\right]\right]\right]}, \tag{2}$$

where $\Omega$ stands for the set of all samples, and the strength $J^{(m)}$ is calculated over all pairs of input variables in all data points. Remarkably, the distribution of $J^{(m)}$ measures the distribution of the complexity of interactions encoded in DNNs. Then we normalize $J^{(m)}$ by the summation value of $I^{(m)}(i, j \mid x)$ with different orders rather than the average value in (Deng et al., 2021) to constrain $0 \le J^{(m)} \le 1$ for explicit comparison across various tasks and datasets.

According to the efficiency property of $I^{(m)}(i, j)$ (Deng et al., 2021), the change of DNN parameters $\Delta W$ can be decomposed as the sum of gradients $\frac{\partial I^{(m)}(i,j)}{\partial W}$. Mathematically, we denote $L$ as the loss function and $\eta$ as the learning rate. With $U = \sum_{i\in N} f(\{i\})$, and $R^{(m)} = -\eta\frac{\partial L}{\partial f(N)}\frac{\partial f(N)}{\partial I^{(m)}(i,j)}$, it is attained by:

$$\Delta W = -\eta\frac{\partial L}{\partial W} = -\eta\frac{\partial L}{\partial f(N)}\frac{\partial f(N)}{\partial W} = \Delta W_U + \sum_{m=3}^{n}\sum_{i,j\in N, i\neq j} R^{(m)}\frac{\partial I^{(m)}(i, j)}{\partial W}. \tag{3}$$

## 3 REVISITING REPRESENTATION BOTTLENECKS OF DNNS

**DNNs are not born to capture low-order and high-order interactions.** We first retrospect relevant findings of DNNs' representation bottleneck. Deng et al. (2021) use $\Delta W^{(m)}(i,j) = R^{(m)} \frac{\partial I^{(m)}(i,j)}{\partial W}$ in Equ. 3 to represent the compositional component of $\Delta W$ w.r.t. $\frac{\partial I^{(m)}(i,j)}{\partial W}$ and claim that it is proportional to $F^{(m)} = \frac{n-m+1}{n(n-1)} / \sqrt{\binom{n-2}{m-2}}$. Despite their delicate theoretical framework, a simple counterexample is when $\frac{m}{n} \to 0$ or $\frac{m}{n} \to 1$, $F^{(m)}$ ought to be approximately the same (see Fig. 1). This is in conflict with the experimental curves in (Deng et al., 2021), where $J^{(m)}$ of low-order (e.g., $m = 0.05n$) is much higher than that of high-order (e.g., $m = 0.95n$). There the empty set $\emptyset$ is disregarded as the input for DNNs. But in Appendix A.3, we demonstrate that even if $f(\emptyset)$ is taken into consideration and $n$ is large (e.g, $n \geq 100$), $J^{(m)}$ ought to be non-zero only when $\frac{m}{n} \to 0$. This phenomenon indicates that DNNs fail to capture any middle-order or high-order interactions, which is against the truth that DNNs perform well in tasks that require high-order interactions such as protein interface prediction (Liu et al., 2020).

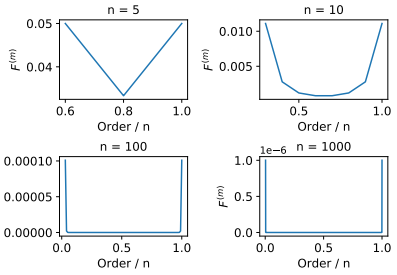



Figure 1: The theoretical distributions of $F^{(m)}$ under different $n$.

**Inductive bias and data distributions are the determinant factors.** The inaccurate statement in Deng et al. (2021) is due to their flawed assumption. The hypothesis that the derivatives of $\Delta f(i,j,S)$ over model parameters, i.e., $\frac{\partial \Delta f(i,j,S)}{\partial W}$, conform to normal distributions should be rejected (see Appendix C.3). $\frac{\partial \Delta f(i,j,S)}{\partial W}$, indeed, varies with the contextual complexities (i.e., $|S|$), and is determined by not only the data distribution of interaction strengths in particular datasets but the model architectures $f$.

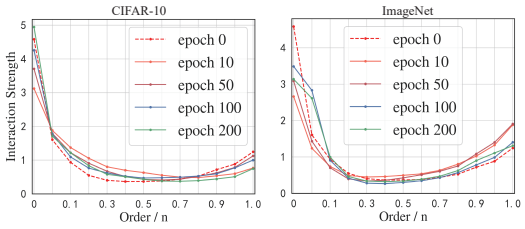



Figure 2: The change of interaction strengths for ResNet on CIFAR-10 and ImageNet, measured after various training epochs.

On the one hand, we define the data distribution of interaction strengths on dataset $D$, denoted as $J_D^{(m)}$, as the experimental distribution of interaction strengths for some model $f$ with randomly initialized parameters. We re-produce experiments in (Deng et al., 2021). Fig. 2 implies that in CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Russakovsky et al., 2015), $J_D^{(m)}$'s (referring to the epoch-0 curve) low-order and high-order interactions are much stronger than middle-order, and little difference exists between $J_D^{(m)}$ and $J^{(m)}$ (referring to non-zero epoch curves) at different epochs. This fact verifies our assertion that $J^{(m)}$ heavily depends on $J_D^{(m)}$.

On the other hand, the locality is a critical inductive bias for CNNs. It assumes that entities are in spatially close proximity with one another and isolated from distant ones (Battaglia et al., 2018), hence CNNs with small kernel sizes are bound to low-order interactions. Recent studies also demonstrate that increasing the kernel size can alleviate the local inductive bias (Ding et al., 2022). To testify our argument, we examine the change of interaction strengths for MLP using MLP-Mixer (Tolstikhin et al., 2021) in Fig. 3. Though MLP shares a similar $J_D^{(m)}$ with CNN, its $J^{(m)}$ is much smoother. This is because MLP-Mixer assumes full connection of different patches and is not constrained by the inductive bias of locality, so it can learn a more adorable $J^{(m)}$. The implementation details on visual tasks are in Appendix B.

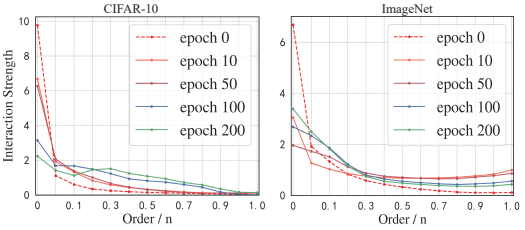



Figure 3: The change of interaction strengths for MLP-Mixer on CIFAR-10 and ImageNet.

# 4 REPRESENTATION BOTTLENECK OF GNNS

## 4.1 NODE-LEVEL MULTI-ORDER INTERACTION

$I^{(m)}(i,j)$ in Equ. 1 is designed to analyze the influence of interactions over the integral system (e.g., a molecule or a galaxy) and is therefore only suitable in the circumstance of graph-level prediction. No such metric exists to measure the effects of those interactions on each component (e.g., atom or particle) of the system. To overcome this limitation, we propose a new metric as the following:

$$I_i^{(m)}(j) = \mathbb{E}_{S\subseteq[N],\{i,j\}\subseteq S,|S|=m}[\Delta f_i(j,S)], 2 \le m \le N, \tag{4}$$

where $\Delta f_i(j,S) = \|f_i(S) - f_i(S\backslash\{j\})\|_p$, and $\|.\|_p$ is the $p$-norm. We denote $f_i(S)$ as the output for the $i$-th variable when variables in $S$ are kept unchanged. Then the corresponding node-level interaction strength is defined as $J^{(m)} = \frac{\mathbb{E}_{x\in\Omega}\left[\mathbb{E}_i\left[\mathbb{E}_j\left[\left|I_i^{(m)}(j|x)\right|\right]\right]\right]}{\sum_{m'}\left[\mathbb{E}_{x\in\Omega}\left[\mathbb{E}_i\left[\mathbb{E}_j\left[\left|I_i^{(m')}(j|x)\right|\right]\right]\right]\right]}$. Equ. 4 allows us to measure the representation capability of GNNs in node-level classification or regression tasks.

## 4.2 GRAPH CONSTRUCTIONS FOR SCIENTIFIC PROBLEMS

**Prerequisites.** How to handle variables in $[N]\backslash S$ is critical to $I^{(m)}(i,j)$. Nonetheless, the widely-used setting in Ancona et al. (2019) for sequences or pixels is not applicable there. In real-world scenarios, including molecules or dynamic systems, the most crucial feature of variables (atoms or particles) is their classes (e.g., one-hot embeddings). An average over different molecules or systems can lead to ambiguous atom or particle types. As an alternative, we consider dropping these variables in $[N]\backslash S$ instead of replacing them with a mean value. In particular, the deletion of those variables ought to satisfy two succeeding properties: (1) The subgraph must maintain connectivity, where entities can reach others freely. Otherwise, each disjoint subgraph is an entirely independent system, and breaks the fundamental assumption in nature that molecules or dynamic systems are an organic whole and indivisible. (2) No ambiguity is intrigued from both structural and feature views. For instance, an element with an invalid atomic number of 3.64 is not permitted.

**KNN vs. fully-connected graphs.** To achieve these constraints, we employ KNN to build edges based on pairwise distances in the 3D space (named KNN-graph), a common technique in macro-molecules (Fout et al., 2017; Ganea et al., 2021; Stärk et al., 2022). When we centre on $S$ and ignore other variables, subgraphs are re-constructed via KNN to ensure connectivity. We also act our analysis on fully-connected graphs (named FC-graph), where all nodes are connected to each other (Chen et al., 2019; Wu et al., 2021; Baek et al., 2021; Jumper et al., 2021). Consequently, removing any entity in FC-graphs will not influence the association of other pairs. More discussion on graph construction is in Appendix A.4.

## 4.3 GRAPH REWIRING FOR INDUCTIVE BIAS OPTIMIZATION

**The representation bottleneck of GNNs.** For modern GNNs, the loss $L$ is typically non-convex with multiple local and even global minima (Foret et al., 2020) that may yield similar values of $L$ while acquiring different capacities to learn interactions (i.e., different $J^{(m)}$). As declared in Prop. 1 (the explanation is in Appendix A.5), if $J^{(m)}$ is not equivalent to the optimal strength ${J^{(m)}}^*$, then the corresponding model $f$ must be stuck in a local minimum point of the loss surface.

**Proposition 1** *Let ${J^{(m)}}^*$ be the interaction strength of the function $f^*$ that achieves the global minimum loss $\ell^*$ on some data $D$. If another model $f'$ converges to a loss $\ell'$ after the parameters update and ${J^{(m)}}' \neq {J^{(m)}}^*$, then $\ell'$ must be a local minimum loss, i.e, $\ell' > \ell^*$.*

However, as analyzed in Section 3, $J_D^{(m)}$ and bad inductive bias can prevent DNNs from capturing appropriate orders of interactions, namely, achieving ${J^{(m)}}^*$. For GNNs, we empirically show in Section 5 that bad inductive bias has a much huger impact on $J^{(m)}$ than $J_D^{(m)}$. This is because graphs support arbitrary pairwise relational structures (Battaglia et al., 2018) and accordingly, the inductive bias of GNNs is more flexible and significant. To be specific, while the local inductive

bias of CNNs comes from their relatively small kernel size, the inductive bias of GNNs primarily depends on the graph connectivity. For example, FC-graphs consist of all pairwise relations, while in KNN-graphs, some pairs of entities possess a relation and others do not. These graph construction mechanisms can bring improper inductive bias and result in poor $J^{(m)}$, which is far from $J^{(m)^*}$.

In order to approach $J^{(m)^*}$, recent work (Deng et al., 2021) imposes two losses to encourage or penalize the learning of interactions of specific complexities. Nevertheless, they require models to make accurate predictions on subgraphs. But variable removal brings the out-of-distribution (OOD) problem (Chang et al., 2018; Frye et al., 2020; Wang et al., 2022), which can manipulate GNNs' outcome arbitrarily and produce erroneous predictions (Dai et al., 2018; Zügner et al., 2018). More importantly, these losses are based on the assumption that the image class remains regardless of pixel removal. But it is not rational to assume the stability of molecular properties if we alter their components. In this work, instead of intervening the loss, we rely on the modification of GNNs' inductive bias to capture the most informative order $m^*$ of interactions and therefore reach $J^{(m)^*}$.

**Graph rewiring to optimize the inductive bias.** Unfortunately, $m^*$ can never be known unless sufficient domain knowledge is supplied. But Fig. 2 shows that in the initial training epochs (e.g., 10 or 50 epochs), CNNs do not directly dive into low-order of interactions. Instead, they deviate from $J_D^{(m)}$ and have the inclination to learn a more informative order of interactions (e.g., middle-order) regardless of the inductive bias. Motivated by this subtle tendency, we resort to the order of interactions that increase the most during training in $J^{(m)}$ as the guidance to reconstruct graphs and estimate $J^{(m)^*}$.

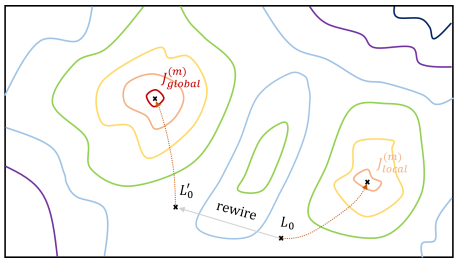



Figure 4: Transformation of the training loss with graph rewiring in the loss surface.

To this end, we dynamically adjust the reception fields of each entity within molecules or systems by establishing or destroying edges, as described in Algorithm 1. Such a method is often generically referred to as *graph rewiring* (Topping et al., 2021). By adjusting graph topology that arouses the inductive bias of GNNs, the representation bottleneck of GNNs is broken, and $J^{(m)}$ is able to gradually approximate $J^{(m)^*}$. Simultaneously, the training loss can finally reach the global minimum after gradient descent (see Fig. 4). Emphatically, our algorithm (named ISGR) is applicable for both KNN-graphs and FC-graphs, considering the latter starts with an adequately large $k_0 \geq n-1$.

**Algorithm 1** Interaction Strength-based Graph Rewiring (ISGR) Algorithm.

**Require:** nodes $V$, pairwise distance $d$, number of neighbors $k_0$, threshold $\bar{J}$, epoch interval $\Delta e$
  Construct a KNN-graph with $K = k_0$ based on $d$ and compute the initial interaction strength $J_0^{(m)}$;
  **for** each $\Delta e$ epochs **do**
    Sample a mini-batch $B$ and calculate the corresponding interaction strengths $J_B^{(m)}$;
    **if** the maximum increase of some order exceeds $\bar{J}$, i.e., $\max\left(\Delta J^{(m)}\right) \geq \bar{J}$ **then**
      Find the order whose interaction strength increases the most $m^* = \underset{m}{\operatorname{argmax}}\left(\Delta J^{(m)}\right)$;
      Increase the number of neighboring nodes $k$ if $m^* > k$, otherwise decrease $k$;
      Reconstruct a KNN-graph with $K = k$ and train the model with this new graph structure.
    **end if**
  **end for**

### 4.4 Relations to Other Types of GNN Bottlenecks

Multiple papers uncover that GNNs may perform poorly on tasks that require long-range dependencies. **Under-reaching** (Barceló et al., 2020) states the inability of a node to be aware of nodes that are farther away than the number of layers. This can be naively avoided by deepening GNNs, but universal evidence indicates that the increase of layers leads to a severe decline in prediction

capability. **Over-smoothing** (Li et al., 2018; Oono & Suzuki, 2019; Chen et al., 2020) and **over-squashing** (Alon & Yahav, 2020; Topping et al., 2021) are two mainstream accepted explanations for this decline. The former owes the failure to indistinguishable node representations when tackling short-range tasks that assume local dependency, while the latter believes that the compression of information from the exponentially receptive field is the core reason for degraded performance in long-range problems. Our work discusses the representation bottleneck of GNNs, which highly resonates with but is essentially different from over-squashing. On the one hand, over-squashing, the prime motivator for graph rewiring, is proposed by the information loss due to long-range dependencies that are not adequately captured by GNNs. These dependencies can be also regarded as middle or high-order interactions that are not fully captured due to improper graph connectivity. On the other hand, our representation bottleneck is built on the theory of multi-order interactions, while over-squashing replies on the message propagation of node features. This makes our representation bottleneck more general than over-squashing. A detailed comparison is available in Appendix D.

# 5 Experimental Results

In this section, we present four case studies where the aforementioned framework is applied to analyze the representation bottleneck of GNNs for scientific research. Among them, Newtonian dynamics and molecular dynamics simulations are node-level prediction tasks, while Hamiltonian dynamics and molecular property prediction are graph-level prediction tasks. More experimental details are elucidated in Appendix C.

## 5.1 Data and Experimental Settings

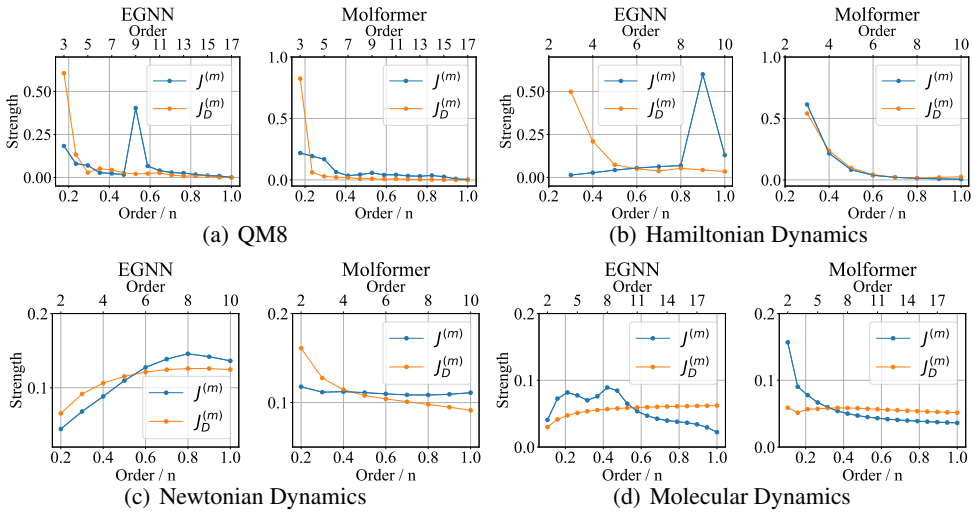
**Newtonian dynamics.** Newtonian dynamics (Whiteside, 1966) describes the dynamics of particles according to Newton's law of motion: the motion of each particle is modeled using incident forces from nearby particles, which changes its position, velocity, and acceleration. Several important forces in physics, such as the gravitational force, are defined on pairs of particles, analogous to the message function of GNNs (Cranmer et al., 2020). We adopt the N-body particle simulation dataset in (Cranmer et al., 2020). It consists of N-body particles under six different interaction laws. More details can be referred to Appendix C.1.

**Hamiltonian dynamics.** Hamiltonian dynamics (Greydanus et al., 2019) describes a system's total energy $\mathcal{H}(\mathbf{q}, \mathbf{p})$ as a function of its canonical coordinates $\mathbf{q}$ and momenta $\mathbf{p}$, e.g., each particles' position and momentum. The dynamics of the system change perpendicularly to the gradient of $\mathcal{H}$: $\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\frac{\mathrm{d}\mathcal{H}}{\mathrm{d}\mathbf{q}}$. There we take advantage of the same datasets from the Newtonian dynamics case study, and attempt to learn the scalar total energy $\mathcal{H}$ of the system.

**Molecular dynamics simulations.** Molecular dynamics (MD) (Frenkel & Smit, 2001; Karplus & McCammon, 2002; Tuckerman, 2010) has long been has long been the *de facto* choice for modeling complex atomistic systems from first principles. There we adopt the ISO17 dataset (Schütt et al., 2017, 2018), which is generated from MD simulations using the Fritz-Haber Institute *ab initio* simulation package (Blum et al., 2009). ISO17 consists of 129 molecules, each containing 5K conformational geometries and total energies with a resolution of 1 femtosecond in the trajectories. Our target is to predict the atomic forces of the molecule at different timeframes.

**Molecular property prediction.** The forecast of a broad range of molecular properties is a fundamental task in drug discovery (Drews, 2000). The properties in current molecular collections can be mainly divided into four categories: quantum mechanics, physical chemistry, biophysics, and physiology, ranging from molecular-level properties to macroscopic influences on the human body (Wu et al., 2018). We utilize two benchmark datasets. QM7 (Blum & Reymond, 2009) is a subset of GDB-13 and is composed of 7K molecules. QM8 (Ramakrishnan et al., 2015) is a subset of GDB-17 with 22K molecules. Note that QM7 and QM8 provide one and twelve properties, respectively, and we merely use the *E1-CC2* property in QM8 for simplicity.

**Baselines and backbones.** We compare ISGR to a variety of graph rewiring methods. **+FA** (Alon & Yahav, 2020) modifies the last GNN layer to be fully connected. **DIGL** (Klicpera et al., 2019)




(a) QM8 (b) Hamiltonian Dynamics

(c) Newtonian Dynamics (d) Molecular Dynamics

Figure 5: Distributions of interaction strengths of EGNN and Molformer in graph-level and node-level prediction tasks. We use double-x axes to represent the order $m$ and the ratio $\frac{m}{n}$

leverages generalized graph diffusion to smooth out the graph adjacency and promote connections among nodes at short diffusion distances. **SDRF** (Topping et al., 2021) acts by alleviating a graph's strongly negatively curved edges. Two state-of-the-art geometric GNNs are selected to perform on these two graph types. We pick up Equivariant Graph Neural Network (EGNN) (Satorras et al., 2021) for KNN-graphs, and Molformer (Wu et al., 2021) with no motifs for FC-graphs. EGNN is roto-translation and reflection equivariant without the spherical harmonics (Thomas et al., 2018). Molformer is a variant of Transformer (Vaswani et al., 2017; Hernández & Amigó, 2021), designed for molecular graph learning.

### 5.2 Investigation of GNNs' Representation Bottleneck

**The learned distribution of interaction strengths can deviate from the data distribution.** Fig. 5 reports the learned distributions $J^{(m)}$ and the data distributions $J_D^{(m)}$ for both graph-level and node-level tasks. The complementary plots for QM7 are available in Appendix C.4. From these curves, it can be drawn that unlike CNNs in Fig. 2, $J^{(m)}$ of GNNs can be divergent from $J_D^{(m)}$.

For molecular property prediction, $J_D^{(m)}$ is more intensive on low-order ($\frac{m}{n} \leq 0.3$). But after sufficient training, $J^{(m)}$ for EGNN mainly have high values for middle-order interactions ($0.5 \leq \frac{m}{n} \leq 0.8$), and the middle-order segment ($0.4 \leq \frac{m}{n} \leq 0.6$) of $J^{(m)}$ for Molformer also increases the most. This illustrates that subgraphs with a middle size are very informative substructures to reveal the biological or chemical properties of small molecules. This finding persistently accords with the fact that motifs such as functional groups play a key part in determining molecular attributes (Yu et al., 2020; Wang et al., 2021; Wu et al., 2022). While for Hamiltonian dynamic systems, $J_D^{(m)}$ is majorly intense for low-order and middle-order interactions ($\frac{m}{n} \leq 0.6$). In spite of that, $J^{(m)}$ of EGNN concentrates more on high-order ($0.7 \leq \frac{m}{n} \leq 0.9$) but neglect low-order ($\frac{m}{n} \leq 0.5$).

Regarding node-level prediction tasks, the scenery is more straightforward. Though $J_D^{(m)}$ for EGNN and Molformer are in different shapes, $J^{(m)}$ both moves towards low-order interactions ($\frac{m}{n} \leq 0.3$) for MD and high-order interactions for Newtonian dynamics ($\frac{m}{n} \geq 0.7$). All those phenomenons demonstrate considerable discrepancies between $J^{(m)}$ and $J_D^{(m)}$ for GNNs.

**The inductive bias heavily determines the change of learned distributions.** Unequivocally, the inclines of EGNN and Molformer to learn interactions of specific orders are distinct. Due to the inductive bias introduced by KNN-graphs, EGNN is more prone to pay attention to interactions of $K$th-order (e.g., $K = 8$ in our setting). Contrarily, Molformer, based on FC-graphs, assumes that all particles can affect each other directly, resulting in a more unconstrained $J^{(m)}$. For example, its

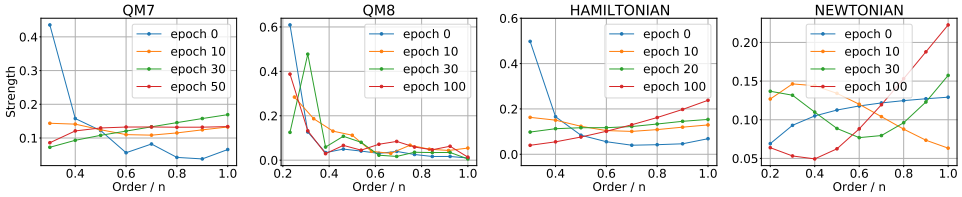



Figure 7: The change of interaction strengths with different training epochs for EGNN.

$J^{(m)}$ on Newtonian dynamics is extremely smooth like a straight line, but its $J^{(m)}$ on Hamiltonian and MD are steep curves. All these evidences bolster our proposal that the inductive bias brought by the topological structure of input graphs significantly impacts $J^{(m)}$ of GNNs.

Particularly, KNN-graphs are more susceptible to improper inductive bias, which prevents EGNN concerning interactions of orders that differ from $K$ and can lead to worse performance. However, FC-graphs (or KNN-graphs with a large $K$) are not a panacea for all tasks. Except that FC-graphs require much more computational costs and may be prohibited in the case of tremendous entities, the performance of Molformer severely depends on the sufficiency and quality of training data. As shown in Tab. 2, Molformer does not surpass EGNN on all datasets. Instead, it behaves worse than EGNN on Hamiltonian ($1.250 > 0.892$) and MD ($0.736 > 0.713$).

### 5.3 Effectiveness of ISGR Algorithm

**Graph and node regression results.** We conduct experiments to examine the efficiency of our ISGR method. Results are reported with the mean and standard deviation of three repetitions in Tab. 1 and 2, where the top two are in bold and underlined, respectively. It can be observed that our ISGR algorithm significantly improves the performance of EGNN and Molformer upon all baselines on both graph-level and node-level tasks. Particularly, the promotion of ISGR for EGNN is much higher, which confirms our assertion that GNNs based on KNN-graphs are more likely to suffer from bad inductive bias. On the other hand, the improvement for Molformer in QM7 is more considerable than in QM8. This proves that GNNs based on FC-graphs are more easily affected by inappropriate inductive bias (i.e., full connection) when the data is insufficient since the size of QM7 (7K) is far smaller than QM8 (21K). We also see that +FA outweighs DIGL and SDRF, the rewiring algorithms by edge sampling, when the graph connectivity is built on KNN. However, when encountering FC-graphs, +FA losses efficacy and SDRF achieves a larger improvement than DIGL.

Table 1: Comparison of different rewiring methods for node-level prediction tasks.

| Task | Newtonian Dynamics | | Molecular Dynamics | |
|---|---|---|---|---|
| Model | EGNN | Molformer | EGNN | Molformer |
| None | 6.951 ± 0.098 | 1.929 ± 0.051 | 1.409 ± 0.082 | 0.848 ± 0.053 |
| +FA | <u>5.348 ± 0.183</u> | – | <u>0.826 ± 0.105</u> | – |
| DIGL | 5.637 ± 0.147 | 1.902 ± 0.081 | 1.108 ± 0.131 | 0.790 ± 0.078 |
| SDRF | 5.460 ± 0.133 | <u>1.885 ± 0.068</u> | 0.942 ± 0.152 | <u>0.751 ± 0.046</u> |
| ISGR (Ours) | **4.734 ± 0.103** | **1.879 ± 0.066** | **0.713 ± 0.097** | **0.736 ± 0.048** |

**The change of $m^*$ during training.** We plot the variation tendency of $m^*$ over different epochs in Fig. 6. It shows that different tasks enjoy various optimal $K$ (denoted as $K^*$). Explicitly, Hamiltonian dynamics and Newtonian dynamics benefit from full-connection ($\frac{K^*}{n} = 1$), while the molecular property prediction including QM7 and QM8 benefits more from middle-order interactions ($\frac{K*}{n} \approx 0.5$). This phenomenon perfectly fits the physical laws, because the system in Newtonian and Hamiltonian datasets is extremely compact with close pairwise distances. Those particles are more likely to be influenced by all the other nodes.

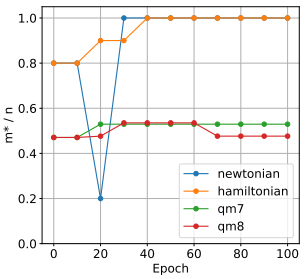



Figure 6: The change of $m^*$ over epochs for EGNN.

**The change of interaction strengths during training.** Fig. 7 depicts how $J^{(m)}$ changes when the training proceeds with our ISGR algorithm. Although for data like QM7, QM8, and Hamiltonian dynamics, $J_D^{(m)}$ mostly concentrate on low-order interactions ($\frac{m}{n} \leq 0.4$), $J^{(m)}$ progressively adjust

Table 2: Comparison of different rewiring methods for graph-level prediction tasks.

| Task | Hamiltonian Dynamics | | QM7 | | QM8 | |
|---|---|---|---|---|---|---|
| Model | EGNN | Molformer | EGNN | Molformer | EGNN | Molformer |
| None | $1.392 \pm 0.042$ | $1.545 \pm 0.036$ | $68.182 \pm 3.581$ | $51.119 \pm 2.193$ | $0.012 \pm 0.001$ | $0.012 \pm 0.001$ |
| +FA | $1.168 \pm 0.043$ | – | $55.288 \pm 3.074$ | – | $0.012 \pm 0.001$ | – |
| DIGL | $1.151 \pm 0.044$ | $1.337 \pm 0.072$ | $61.028 \pm 3.804$ | $41.188 \pm 5.329$ | $0.012 \pm 0.001$ | $0.011 \pm 0.001$ |
| SDRF | $1.033 \pm 0.790$ | $1.265 \pm 0.039$ | $59.921 \pm 3.765$ | $35.792 \pm 4.565$ | $0.011 \pm 0.001$ | $0.011 \pm 0.001$ |
| ISGR (Ours) | $\mathbf{0.892 \pm 0.051}$ | $\mathbf{1.250 \pm 0.029}$ | $\mathbf{53.134 \pm 2.711}$ | $\mathbf{34.439 \pm 4.017}$ | $\mathbf{0.011 \pm 0.000}$ | $\mathbf{0.010 \pm 0.001}$ |

to middle-order and high-order ($\frac{m}{n} \geq 0.4$). Regarding Newtonian dynamics, $J_D^{(m)}$ is very smooth, but $J^{(m)}$ at initial epochs (i.e., 10 and 20 epochs) oddly focus on low-order interactions ($\frac{m}{n} \leq 0.4$). Nevertheless, our ISGR method timely corrects the wrong tendency, and eventually, $J^{(m)}$ becomes more intensive in segments of middle-order and high-order ($\frac{m}{n} \geq 0.6$).

## 6 Related Work

**GNNs' expressiveness and bottlenecks.** It is found that GNNs captures only a tiny fragment of first-order logic (Barceló et al., 2020), which arises from the deficiency of a node's receptive field. Meanwhile, GNNs are observed not to benefit from the increase of layers due to *over-smoothing* (Li et al., 2018; Klicpera et al., 2018; Chen et al., 2020) and *over-squashing* (Alon & Yahav, 2020; Topping et al., 2021). To the best of our knowledge, none considers GNNs' capacity in encoding pairwise interactions, and we are the foremost to understand GNNs' expressiveness from interactions under different contextual complexities and link the expressive limitation with the inductive bias of graph connectivity. More elaborate related works are in Appendix E.

**GNNs' representation capacity.** It becomes an emerging area to evaluate the representation capability of DNNs (Shwartz-Ziv & Tishby, 2017; Neyshabur et al., 2017; Novak et al., 2018; Weng et al., 2018; Fort et al., 2019), where Zhang et al. (2020) and Deng et al. (2021) pioneeringly employ interactions between variables to inspect the limitation of DNNs in feature representations. Notwithstanding, prior works merely highlight the behaviors of general DNNs and examine their assertions via MLP and CNNs. In comparison, we emphasize GNNs that operate on structured graphs, distinct from images and texts.

**Graph rewiring.** Rewiring is a process of altering the graph structure to control the information flow. Among diverse existing approaches such as connectivity diffusion (Klicpera et al., 2019), bridge-node insertion (Battaglia et al., 2018), and multi-hop filters (Frasca et al., 2020), edge sampling shows great power in tackling *over-smoothing* and *over-squashing*. The sampling strategies can be random drop (Huang et al., 2020b) or based on edge relevance (Klicpera et al., 2019; Kazi et al., 2022). Recently, Alon & Yahav (2020) modify the last layer to a FC-graph to help GNNs grab long-range interactions. Taking a step further, Topping et al. (2021) prove that negatively curved edges are responsible for *over-squashing* and introduces a curvature-based rewiring method to alleviate that. Differently, our rewiring algorithm originates from a completely new motivation, i.e., reshaping graph structure to allow GNNs to learn the most informative order of interactions.

## 7 Conclusion

In this paper, we discover and strictly analyze the representation bottleneck of GNNs from the complexity of interactions encoded in networks. Remarkably, inductive bias rather than the data distribution is more dominant in the expressions of GNNs to capture pairwise interactions. This observation also motivates us to conclude that inductive biases introduced by most graph construction mechanisms such as KNN and full connection are sub-optimal. Inspired by this gap, we design a novel rewiring method based on the inclination of GNNs to encode more informative orders of interactions. We conduct experiments on four synthetic and real-world tasks, verifying that GNNs are allowed to reach the global minimum loss and break the bottleneck via our efficient algorithm. Limitations of our work are stated in Appendix F.

## REFERENCES


Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pp. 272–281. PMLR, 2019.

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

Pablo Barceló, Egor Kostylev, Mikael Monet, Jorge Pérez, Juan Reutter, and Juan-Pablo Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Jacob Bien, Jonathan Taylor, and Robert Tibshirani. A lasso for hierarchical interactions. *Annals of statistics*, 41(3):1111, 2013.

Lorenz C Blum and Jean-Louis Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society*, 131(25): 8732–8733, 2009.

Volker Blum, Ralf Gehrke, Felix Hanke, Paula Havu, Ville Havu, Xinguo Ren, Karsten Reuter, and Matthias Scheffler. Ab initio molecular simulations with numeric atom-centered orbitals. *Computer Physics Communications*, 180(11):2175–2196, 2009.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

Benson Chen, Regina Barzilay, and Tommi Jaakkola. Path-augmented graph transformer network. *arXiv preprint arXiv:1905.12712*, 2019.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3438–3445, 2020.

Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33:17429–17442, 2020.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International conference on machine learning*, pp. 1115–1124. PMLR, 2018.

Subhrangshu Das and Saikat Chakrabarti. Classification and prediction of protein–protein interaction interface using machine learning algorithm. *Scientific reports*, 11(1):1–12, 2021.

Huiqi Deng, Qihan Ren, Xu Chen, Hao Zhang, Jie Ren, and Quanshi Zhang. Discovering and explaining the representation bottleneck of dnns. *arXiv preprint arXiv:2111.06236*, 2021.

Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11963–11975, 2022.

Jurgen Drews. Drug discovery: a historical perspective. *science*, 287(5460):1960–1964, 2000.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pp. 417–426, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Stanislav Fort, Paweł Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.

Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.

Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.

Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv preprint arXiv:2111.07786*, 2021.

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

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28(4):547–565, 1999.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Barbara Hammer, Alessio Micheli, and Alessandro Sperduti. Universal approximation capability of cascade correlation for structures. *Neural Computation*, 17(5):1109–1159, 2005.

Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. *arXiv preprint arXiv:2106.04263*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Adrián Hernández and José M Amigó. Attention mechanisms and their applications to complex systems. *Entropy*, 23(3):283, 2021.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Kexin Huang, Cao Xiao, Lucas M Glass, Marinka Zitnik, and Jimeng Sun. Skipgnn: predicting molecular interactions with skip-graph networks. *Scientific reports*, 10(1):1–16, 2020a.

Wenbing Huang, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Tackling over-smoothing for general graph convolutional networks. *arXiv preprint arXiv:2008.09864*, 2020b.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021. doi: 10.1038/s41586-021-03819-2. (Accelerated article preview).

Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–652, 2002.

Anees Kazi, Luca Cosmo, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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

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

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *arXiv preprint arXiv:1911.05485*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

AA Leman and Boris Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 539–548, 2019.

Yi Liu, Hao Yuan, Lei Cai, and Shuiwang Ji. Deep learning of high-order interactions for protein interface prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 679–687, 2020.

Chengqiang Lu, Qi Liu, Chao Wang, Zhenya Huang, Peize Lin, and Lixin He. Molecular property prediction: A multilevel quantum interactions modeling perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1052–1060, 2019.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Aravind Nair, Avijit Roy, and Karl Meinke. Funcgnn: a graph neural network approach to program similarity. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–11, 2020.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 401–417, 2018.

Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza, and O Anatole Von Lilienfeld. Electronic spectra from tddft and machine learning in chemical space. *The Journal of chemical physics*, 143(8):084111, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Vıcctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pp. 9323–9332. PMLR, 2021.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. Computational capabilities of graph neural networks. *IEEE Transactions on Neural Networks*, 20 (1):81–102, 2008a.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008b.

Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8(1):1–8, 2017.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Hannes Stärk, Octavian-Eugen Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. *arXiv preprint arXiv:2202.05146*, 2022.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Peter Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems (NIPS)*, 2021.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv:2111.14522*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. *arXiv preprint arXiv:1705.04977*, 2017.

Mark Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford university press, 2010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Xiang Wang, An Zhang, Xia Hu, Fuli Feng, Xiangnan He, Tat-Seng Chua, et al. Deconfounding to explanation evaluation in graph neural networks. *arXiv preprint arXiv:2201.08802*, 2022.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

Derek Thomas Whiteside. Newtonian dynamics. *History of Science*, 5:104, 1966.

Fang Wu, Qiang Zhang, Dragomir Radev, Jiyu Cui, Wen Zhang, Huabin Xing, Ningyu Zhang, and Huajun Chen. 3d-transformer: Molecular representation with transformer in 3d space. *arXiv preprint arXiv:2110.01191*, 2021.

Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872*, 2022.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition. *arXiv preprint arXiv:2010.05563*, 2020.

Hao Zhang, Sen Li, Yinchao Ma, Mingjie Li, Yichen Xie, and Quanshi Zhang. Interpreting and boosting dropout from a game-theoretic view. *arXiv preprint arXiv:2009.11729*, 2020.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2847–2856, 2018.

# APPENDIX

The Appendix is structured as follows:

- In Appendix A we provide supplementary materials of the multi-order interaction tool for readers who are not familiar with this field. Moreover, we prove that our reformed multi-order interaction is equivalent to the original definition, and also offer a straightforward explanation of the proposition that appeared in the main text.
- In Appendix B we introduce the details of our implementation in exploring the change of interaction strengths for CNNs and MLP-Mixer on visual problems.
- In Appendix C we describe the formulation of datasets, training details, and some other additional experimental results.
- In Appendix E we give more descriptions of prior studies in regards to the expressiveness of GNNs.
- In Appendix D we systematically compare our representation bottleneck with exiting bottlenecks of GNNs.
- In Appendix F we describe the limitations and potential negative social impact of our work. We also point out the future direction to enrich the content of our paper.

## A INTRODUCTION AND THEORETICAL ANALYSIS OF MULTI-ORDER INTERACTIONS

### A.1 INTRODUCTION OF MULTI-ORDER INTERACTIONS

In this subsection, we give a more detailed introduction of the multi-order interaction, which is employed to analyze the representation ability of GNNs and CNNs in the main body, in case some audiences may find it to follow. This introduction largely utilizes Zhang et al. (2020) and Deng et al. (2021) for reference, and we strongly recommend interesting readers to take a glance at these articles. Notably, the original multi-order interaction between input variables is defined as follows:

$$I^{(m)}(i,j) = \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\Delta f(i,j,S)], 3 \le m \le n, \tag{5}$$

where $\Delta f(i,j,S) = f(S\cup\{i,j\}) - f(S\cup\{i\}) - f(S\cup\{j\}) + f(S)$ and $S \subset N$ represents the context with $m$ variables. $I^{(m)}(i,j)$ denotes the interaction between variables $i,j \in N$ of the $m$-th order, which measures the average interaction utility between $i,j$ under contexts of $m$ variables. There are five desirable properties that $I^{(m)}(i,j)$ satisfies:

**- Linear property.** If two independent games $f_1$ and $f_2$ are combined, obtaining $g(S) = f_1(S) + f_2(S)$, then the multi-order interaction of the combined game is equivalent to the sum of multi-order interactions derived from $f_1$ and $f_2$, i.e.,$I_g^{(m)}(i,j) = I_{f_1}^{(m)}(i,j) + I_{f_2}^{(m)}(i,j)$.

**- Nullity property.** If a dummy variable $i \in N$ satisfies $\forall S \subseteq N\setminus\{i\}, f(S\cup\{i\}) = f(S) + f(\{i\})$, then variable $i$ has no interactions with other variables, i.e., $\forall m, \forall j \in N\setminus\{i\}, I^{(m)}(i,j) = 0$.

**- Commutativity property.** Intuitively, $\forall i,j \in N, I^{(m)}(i,j) = I^{(m)}(j,i)$.

**- Symmetry property.** Suppose two variables $i,j$ are equal in the sense that $i,j$ have same co-operations with other variables, i.e., $\forall S \subseteq N\setminus\{i,j\}, f(S\cup\{i\}) = f(S\cup\{j\})$, then we have $\forall k \in N, I^{(m)}(i,k) = I^{(m)}(j,k)$.

**- Efficiency property (Deng et al., 2021).** The output of a DNN can be decomposed into the sum of interactions of different orders between different pairs of variables as:

$$f(N) - f(\emptyset) = \sum_{i\in N}\mu_i + \sum_{i,j\in N, i\neq j}\sum_{m=0}^{n-2} w^{(m)} I^{(m)}(i,j), \tag{6}$$

where $\mu_i = f(\{i\}) - f(\emptyset)$ represents the independent effect of variable $i$, and $w^{(m)} = \frac{n-1-m}{n(n-1)}$.

**Connection with Shapley value and Shapley interaction index.** Shapley value is introduced to measure the numerical importance of each player to the total reward in a cooperative game, which has been widely accepted to interpret the decision of DNNs in recent years (Lundberg & Lee, 2017; Ancona et al., 2019). For a given DNN and an input sample with a set of input variables $N = \{1, \ldots, n\}$, we use $2^N = \{S \mid S \subseteq N\}$ to denote all possible variable subsets of $N$. Then, DNN $f$ can be considered as $f : 2^N \to \mathbb{R}$ that calculates the output $f(S)$ of each specific subset $S \subseteq N$. Each input variable $i$ is regarded as a player, and the network output $f(N)$ of all input variables can be considered as the total reward of the game. The Shapley value aims to fairly distribute the network output to each individual variables as follows:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [f(S \cup \{i\}) - f(S)], \tag{7}$$

where $f(S)$ denotes the network output when we keep variables in $S$ unchanged while masking variables in $N \setminus S$ by following the setting in Ancona et al. (2019). It has been proven that the Shapely value is a unique method to fairly allocate overall reward to each player that satisfies *linearity*, *nullity*, *symmetry*, and *efficiency* properties.

**Connections between the Shapley interaction index and the Shapely value.** Input variables of a DNN usually interact with each other, instead of working individually. Based on the Shapley value, Grabisch & Roubens (1999) further proposes the Shapley interaction index to measure the interaction utility between input variables. The Shapley interaction index is the only axiomatic extension of the Shapley value, which satisfies *linearity*, *nullity*, *symmetry*, and *recursive* properties. For two variables $i, j \in N$, the Shapley interaction index $I(i, j)$ can be considered as the change of the numerical importance of variable $i$ by the presence or absence of variable $j$.

$$I(i, j) = \tilde{\phi}(i)_{j \text{ always present}} - \tilde{\phi}(i)_{j \text{ always absent}}, \tag{8}$$

where $\tilde{\phi}(i)_{j\text{always present}}$ denotes the Shapley value of the variable $i$ computed under the specific condition that variable $j$ is always present. $\tilde{\phi}(i)_{j \text{ always absent}}$ is computed under the specific condition that $j$ is always absent.

**Connections between the multi-order interaction and the Shapley interaction index.** Based on the Shapley interaction index, Zhang et al. (2020) further defines the order of interaction, which represents the contextual complexity of interactions. It has been proven that the above Shapley interaction index $I(i, j)$ between variables $i, j$ can be decomposed into multi-order interactions as follows:

$$I(i, j) = \frac{1}{n - 1} \sum_{m=0}^{n-2} I^{(m)}(i, j). \tag{9}$$

### A.2 Proof of Multi-order Interactions

There we explain why $I^{(m)}(i, j)$ has no difference whether we include variables $\{i, j\}$ in $S$ or not. In the setting of (Zhang et al., 2020; Deng et al., 2021), $I^{(m)}(i, j)$ takes the following form:

$$I^{(m)}(i, j) = \mathbb{E}_{S \subseteq N \setminus \{i,j\}, |S|=m}[\Delta f(i, j, S)], \tag{10}$$

where $\Delta f(i, j, S) = f(S \cup \{i, j\}) - f(S \cup \{i\}) - f(S \cup \{j\}) + f(S)$ and $i, j \notin S$. While in our formulation, the order $m' = m + 2$ corresponds to the context $S' = S \cup \{i, j\}$. Now we denote our version of the multi-order interaction as $I'^{(m')}(i, j)$ with $\Delta' f(i, j, S)$ and aim to show that $I^{(m)}(i, j) = I'^{(m')}(i, j)$.

It is trivial to obtain that $f(S \cup \{i, j\}) - f(S \cup \{i\}) - f(S \cup \{j\}) + f(S) = f(S') - f(S' \setminus \{i\}) - f(S' \setminus \{j\}) + f(S' \setminus \{i, j\})$, which indicates that $\Delta f(i, j, S) = \Delta' f(i, j, S')$. Therefor, we can get $I^{(m+2)}(i, j) = I'^{(m')}(i, j)$.

### A.3 Theoretical Distributions of $F^{(m)}$

Fig. 8 depicts the theoretical distributions of $F^{(m)}$ for different $n$. Unlike Fig. 1, the empty set $\emptyset$ is allowed as the input for DNNs. Apparently, when the number of variables $n$ is very large ($n \geq 100$),

$F^{(m)}$ is only positive for $\frac{m}{n} \to 0$. For macromolecules such as proteins, the number of atoms is usually more than ten thousand. If the theorem in (Deng et al., 2021) that the strengths $\Delta W^{(m)}(i,j)$ of learning the $m$-order interaction is strictly proportional to $F^{(m)}$ holds, DNNs would be impossible to put attention to any middle-order interactions, which is proven to be critical for modeling protein-protein interactions (Liu et al., 2020; Das & Chakrabarti, 2021).

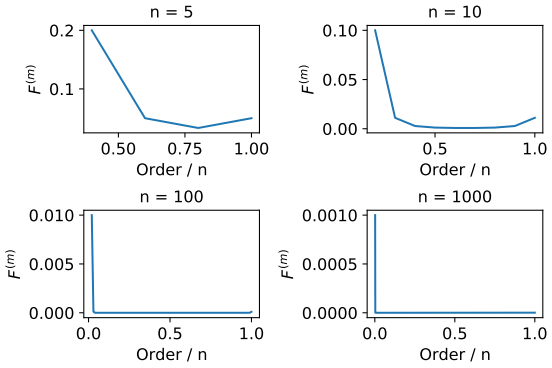
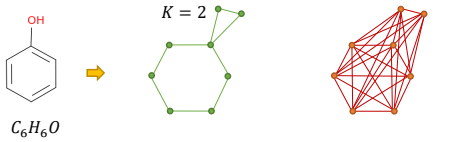



Figure 8: Distributions of $F^{(m)}$ with different numbers of variables $n$ where $f(\emptyset)$ is taken into consideration.

### A.4 Graph Construction Approaches

Unlike social networks or knowledge graphs, there are, indeed, no explicit edges in graphs of most scientific problems. So in order to represent molecules or systems as graphs, *KNN-graphs*, *fully-connected graphs*, and *r-ball graphs* are the three most broadly used mechanisms to build the connectivity. In *r*-ball graphs, an edge between any atoms exists as long as their inter-atomic distance is shorter than a threshold value. But in order to better utilize the multi-order interaction theory, the way of graph construction must satisfy two properties: (1) The subgraph maintains the connectivity. (2) No ambiguity should be intrigued from neither the structural nor feature view. Consequently, *r*-ball graphs do not satisfy the first property. For instance, a node $a$ (e.g., hydrogen) can be connected to only another node $b$. If we remove $b$ from the system, $a$ would be an isolated particle, which should be forbidden.

Apart from that, KNN-graphs (see Fig. 9 (a)) and FC-graphs (see Fig. 9 (b)) are the other two broad practices to establish connections between entities. In fact, FC-graphs are a special type of KNN-graphs, where $K \geq n-1$. However, FC-graphs or KNN-graphs with a large $K$ suffer from high computational expenditure and are usually infeasible with thousands of entities. In addition, they are sometimes unnecessary since the impact from distant nodes is so minute to be ignored.




(a) KNN-graphs (b) FC-graphs

Figure 9: Different graph constructions of the compound $C_6H_6O$.

### A.5 Explanation of Proposition 1

Prop. 1 illustrates an intuitive necessary condition for a model $f$ to achieve the global minimum loss. That is, the learned strength $J^{(m)}$ must match the optimal strength $J^{(m)*}$. Otherwise, $f$ must be inferior to the best predictor $f^*$. This proposition is very straightforward. The global minimum

loss $L^*$ has its corresponding model weight $f^*$ and thereby a unique learned strength $J^{(m)*}$. Given another model $f$, if its strength $J^{(m)}$ is different from $J^{(m)*}$, then $f$ is different from $f^*$. As a consequence, the loss of $f$ must be larger than $L^*$. Notably, there we consider the loss function to be non-convex. In contrast, if the loss function is convex, standard optimization techniques like gradient descent will easily find parameters that converge towards global minima.

# B CNNs AND MLP-MIXER ON VISUAL TASKS

To investigate the change of interaction strengths during the training process in image classification, we train ResNet-50 (He et al., 2016) and MLP-Mixer (*Small*) (Tolstikhin et al., 2021) and calculate the interaction strength by the official implementation provided by Deng et al. (2021). MLP-mixer is an architecture based exclusively on multi-layer perceptrons (MLPs). It contains two types of layers: one with MLPs applied independently to image patches (i.e. "mixing" the per-location features), and one with MLPs applied across patches (i.e. "mixing" spatial information). We discuss MLP-mixer to compare it with the traditional CNNs. Notably, CNNs assume the local inductive bias, while MLP-mixer instead connects each patch with other patches (e.g., no constraint of locality).

We follow the training settings of DeiT (Touvron et al., 2021) and train 200 epochs with the input resolution of $224 \times 224$ on CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Russakovsky et al., 2015) datasets. Fig. 2 plots the corresponding strengths at different epochs, where the dotted line denotes the initial interaction strength without training (referring to epoch 0), i.e., the data distribution of strengths $J_D^{(m)}$.

Through visualization, it can be easily found that $J_D^{(m)}$ in both CIFAR-10 and ImageNet have already obeyed a mode that the low-order ($\frac{m}{n} \leq 0.2$) and high-order ($\frac{m}{n} \geq 0.8$) interaction strengths are much higher than middle-order ($0.2 \leq \frac{m}{n} \leq 0.8$). The variation of interaction strengths is very slight with the training proceeding, which validates our statement that the data distribution has a strong impact on the learned distribution. More importantly, we challenge the argument in Deng et al. (2021), who believe it is difficult for DNNs to encode middle-order interaction. But in our experiments on GNNs, we document that DL-based models are capable of capturing middle-order interactions.

The capability of CNNs to capture desirable levels of interactions are constrained by the improper inductive bias. Remarkably, some preceding work (Han et al., 2021; Ding et al., 2022) has proved the effectiveness of enlarging kernel size to resolve the local inductive bias, where an adequately large kernel size is able to improve the performance of CNNs comparable to ViT and MLP-Mixer. Nevertheless, how to determine the scale of convolutional kernels is still under-explored. Our ISGR algorithm provides a promising way to seek the optimal kernel size based on the interaction strength, abandoning the exhaustive search.

# C EXPERIMENTAL DETAILS AND ADDITIONAL RESULTS

## C.1 NEWTONIAN DYNAMICS DATASET

The following six forces are utilized in the dataset of Newtonian dynamics: (1) $1/r$ orbital force: $-m_1 m_2 \hat{r}/r$; (2) $1/r^2$ orbital force $-m_1 m_2 \hat{r}/r^2$; (3) charged particles force $q_1 q_2 \hat{r}/r^2$ (4) damped springs with $|r-1|^2$ potential and damping proportional and opposite to speed; (5) discontinuous forces, $-\left\{0, r^2\right\}\hat{r}$, switching to 0 force for $r < 2$; and (6) springs between all particles, a $(r-1)^2$ potential. There we only use the spring force for our experiments.

## C.2 TRAINING DETAILS

All experiments are implemented by Pytorch (Paszke et al., 2019) on an A100 GPU. An Adam (Kingma & Ba, 2014) optimizer is used without weight decay, and a ReduceLROnPlateau scheduler is enforced to adjust it with a factor of 0.6 and patience of 10. The initial learning rate is 1e-4, and the minimum learning rate is 5e-6. The batch size is 512 for the sake of a fast training speed. Each model is trained for 1200 epochs, and early stopping is used if the validation error fails

to decrease for 30 successive epochs. We randomly split each dataset into training, validation, and test sets with a ratio of 80/10/10.

For both EGNN and Molformer, the numbers of layers (i.e., depths) are 3, and the dimensions of the input feature are 32. Besides, Molformer has 4 attention heads and a dropout rate of 0.1. Its dimension of the feed-forward network is 128. It is worth noting that we employ multi-scale self-attention with a distance bar of $[0.8, 1.6, 3]$ to achieve better performance. This multi-scale mechanism helps Molformer to concentrate more on local contexts. However, it does not harm FC-graphs, and the connections between all pairs of entities remain. We also discover that the multi-scale mechanism has little impact on the distribution of $J^{(m)}$ and $J_D^{(m)}$. Regarding the setup of the ISGR algorithm, the threshold $\bar{J}$ to adjust the number of neighbors is tuned via a grid search. The interval of epochs $\Delta e$ is 10, and the initial $k_0 = 8$. Concerning baselines, we follow Klicpera et al. (2019) and Topping et al. (2021) and optimize hyperparameters by random search. Table 3 documents $\alpha$, $k$, and $\epsilon$ for DIGL, whose descriptions can be found in Klicpera et al. (2019). Table 4 reports the maximum iterations, $\tau$ and $C^+$ for SDRF, whose descriptions is available in Topping et al. (2021).

Table 3: Hyperparameters for DIGL.

| Task | Newtonian Dynamics | | Molecular Dynamics | | Hamiltonian Dynamics | | QM7 | | QM8 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | EGNN | Molformer | EGNN | Molformer | EGNN | Molformer | EGNN | Molformer | EGNN | Molformer |
| $\alpha$ | 0.0259 | 0.1284 | 0.0732 | 0.1041 | 0.1561 | 0.3712 | 0.0655 | 0.2181 | 0.1033 | 0.1892 |
| $k$ | 32 | 32 | 32 | 32 | 64 | 64 | - | - | - | - |
| $\epsilon$ | - | - | - | - | 0.0001 | - | - | - | - | 0.0002 |

Table 4: Hyperparameters for SDRF.

| Task | Newtonian Dynamics | | Molecular Dynamics | | Hamiltonian Dynamics | | QM7 | | QM8 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | EGNN | Molformer | EGNN | Molformer | EGNN | Molformer | EGNN | Molformer | EGNN | Molformer |
| Max Iter. | 15 | 11 | 39 | 34 | 16 | 13 | 22 | 17 | 25 | 12 |
| $tau$ | 120 | 163 | 54 | 72 | 114 | 186 | 33 | 35 | 48 | 60 |
| $C^+$ | 0.73 | 1.28 | 1.44 | 1.06 | 0.96 | 0.88 | 0.53 | 0.70 | 0.69 | 0.97 |

We create a simulated system with 10 identical particles with a unit weight for Hamiltonian and Newtonian cases. For QM7, QM8, and ISO17 datasets, we sample 10 molecules that have the lowest MAE. For Hamiltonian and Newtonian datasets, we sample 100 timeframes that have the lowest prediction errors. Then for each molecule or dynamic system, we compute all pairs of entities $i, j \in [N]$ without any sampling strategy. Moreover, we limit the number of atoms between 10 and 18 to compute the interaction strengths for QM7 and MQ8.

### C.3 Examination of the Normal Distribution Hypothesis

We use *scipy.stats.normaltest* in the Scipy package (Virtanen et al., 2020) to test the null hypothesis that $\frac{\partial \Delta f(i,j,S)}{\partial W}$ comes from a normal distribution, i.e, $\frac{\partial \Delta f(i,j,S)}{\partial W} \sim \mathcal{N}\left(0, \sigma^2\right)$. This test is based on D'Agostino and Pearson's examination that combines skew and kurtosis to produce an omnibus test of normality. The $p$-values of well-trained EGNN and Molformer on the Hamiltonian dynamics dataset are 1.97147e-11 and 2.38755e-10, respectively. The $p$-values of randomly initialized EGNN and Molformer on the Hamiltonian dynamics dataset are 2.41749e-12 and 9.78953e-07, separately. Therefore, we are highly confident in rejecting the null hypothesis (e.g., $\alpha = 0.01$) and insist that $\frac{\partial \Delta f(i,j,S)}{\partial W}$ depends on the data distributions of downstream tasks and the backbone model architectures.

### C.4 Distributions of Strengths in QM7

Due to the limitation of space, we move Fig. 10 to the Appendix, which shows the learned distribution and data distribution of EGNN and Molformer in the QM7 dataset. The conclusions are very similar to the discovery in QM8. Although the data distribution of strengths $J_D^{(m)}$ concentrate on low-order ($\frac{m}{n} \leq 0.4$), the learned distribution of strengths $J^{(m)}$ are mainly allocated on

middle-order interactions ($0.5 \leq \frac{m}{n} \leq 0.7$). Especially for EGNN, its spike of $J^{(m)}$ is at $m = 9$. While concerning Molformer, the segment of its $J^{(m)}$ that increases most is dispersive between $0.3 \leq \frac{m}{n} \leq 0.5$.

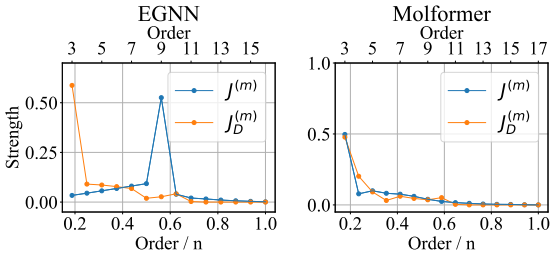



Figure 10: Learned and data distributions of the interaction strength of EGNN and Molformer in the QM7 dataset.

## D Comprehensive Comparison to Existing Bottlenecks of GNNs

GNNs based on the message passing diagram show extraordinary results with a small number of layers. Nevertheless, such GNNs fail to capture information that depends on the entire structure of the graph and prevent the information flow from reaching distant nodes. This phenomenon is called **under-reaching** (Barceló et al., 2020). To overcome this limitation, an intuitive resolution is to increase the layers. But unfortunately, GNNs with many layers tend to suffer from the **over-smoothing** (Oono & Suzuki, 2019) or **over-squashing** (Alon & Yahav, 2020) problems.

*Over-smoothing* takes place when node embeddings become indistinguishable. It occurs in GNNs that are used to tackle short-range tasks, i.e., the accurate prediction majorly depends on the local neighborhood. On the contrary, long-range tasks require as many layers as the range of interactions between nodes. But this would contribute to the exponential increase of the node's receptive field and compress the information flow, which is named *over-squashing*. In our study, we do not specify which category of problems to be addressed (i.e., long-range or short-range). Instead, we aim to explore which sort of interactions that GNNs are more likely to encode (i.e., too simple, intermediately complex, and too complex). It is also worth noting that different tasks require different levels of interactions. For instance, Newtonian and Hamiltonian dynamics demand too complex interactions, while molecular property prediction prefers interactions of intermediate complexity. Then based on both theoretical and empirical evidence, we discover that improper inductive bias introduced by the way to construct graph connectivity prevents GNNs from capturing the desirable interactions, resulting in the representation bottleneck of GNNs.

So what is the significant difference between our representation bottleneck and *over-squashing*? Most foundationally and importantly, our representation bottleneck is based on the theory of multi-order interactions, while *over-squashing* relies on the message propagation of node representations. To be specific, it is demonstrated in Alon & Yahav (2020) that the propagation of messages is controlled by a suitable power of $\hat{A}$ (the normalized augmented adjacency matrix), which relates *over-squashing* to the graph topology. In contrast, we show that the graph topology strongly determines the distribution of interaction strengths $J^{(m)}$, i.e., whether GNNs are inclined to capture too simple or too complex interactions. This difference in theoretical basis leads to the following different behaviors of our representation bottleneck and *over-squashing*:

- The multi-order interaction technique focuses on interactions under a certain context, whose complexity is measured as the number of its variables (i.e., nodes) $m$ divided by the total number of variables of the environment (i.e., the graph) $n$. Thus, the complexity of interactions is, indeed, a relative quantity. Conversely, *over-squashing* (as well as *under-reaching*) concerns about the absolute distance between nodes. Given a pair of nodes $i$ and $j$, if the shortest path between them is $r$, then at least $r$ layers are required for $i$ to reach out to $j$. More generally, long-range or short-range tasks discussed in most GNN studies are referring to this $r$-distance. *Over-squashing*, therefore, follows this $r$-distance metric and argues that the information aggregated across a long path is compressed, which causes the degradation of GNNs' performance.

As a result, our representation bottleneck can occur in both short-range and long-range tasks, but *over-squashing* mainly exists in long-range problems. For short-range tasks, if we assume a KNN-graph with a large $K$ or even fully-connected graphs (i.e., nodes can have immediate interactions with distant nodes), then the receptive field of each node is very large and GNNs intend to concentrate on too complex interactions but fail to capture interactions within local neighbors. For long-range tasks, if we assume a KNN-graph with a small $K$ (i.e., nodes only interact with nearby nodes), then the receptive field of each node is relatively small compared to the size of the entire graph. Consequently, GNNs prefer to capture too simple interactions but are incapable of seizing more informative complex interactions.

- More essentially, the multi-order interaction theory of our representation bottleneck is model-agnostic, but the starting point of *over-squashing* is message passing, the characteristic of most GNN architectures. To make it more clear, the calculation of multi-order interactions (see Equ. 1 and 2) is completely independent of the network (e.g., CNNs, GNNs, RNNs). However, the theory of *over-squashing* is founded on the message passing procedure. This hypothesis makes *over-squashing* limited to the group of GNNs that are built on message passing. But other kinds of GNNs such as the invariants of Transformers may not suffer from this catastrophe. Instead, the analysis of multi-order interactions in our representation bottleneck can be utilized to any GNN architecture, even if it abandons the traditional message passing mechanism.

To summarize, our representation bottleneck is more universal than *over-squashing* which is built upon the absolute distance and merely talks about long-range tasks. This is due to the fact that our representation bottleneck is given birth to by the theory of multi-order interactions rather than the property of message propagation.

## E More Related Work on Expressiveness of GNNs

It is well-known that MLP can approximate any Borel measurable function (Hornik et al., 1989), but few study the universal approximation capability of GNNs (Wu et al., 2020). Hammer et al. (2005) demonstrates that cascade correlation can approximate functions with structured outputs. Scarselli et al. (2008a) prove that a RecGNN (Scarselli et al., 2008b) can approximate any function that preserves unfolding equivalence up to any degree of precision. Maron et al. (2018) show that an invariant GNN can approximate an arbitrary invariant function defined on graphs. Xu et al. (2018) show that common GNNs including GCN (Kipf & Welling, 2016) and GraphSage (Hamilton et al., 2017) are incapable of differentiating different graph structures. They further prove if the aggregation functions and the readout functions of a GNN are injective, it is at most as powerful as the Weisfieler-Lehman (WL) test (Leman & Weisfeiler, 1968) in distinguishing different graphs.

## F Limitations and Future Directions

Despite the reported success of our proposed ISGR, GNNs still face a few limitations. In real-world applications such as social networks, improper modeling of interactions may lead to the failure of classifying fake users and posts. In addition, since GNNs have been extensively deployed in assisting biologists with the discovery of new drugs, inappropriate modeling can postpone the screening progress and impose negative consequences on the drug design. Another limitation of our work is that results are only demonstrated on small systems with few particles and two sorts of GNNs. The generalization of our claim remains investigated on larger datasets and more architectures.