GeoSSL: Molecular Geometry Self-Supervised Learning with SE(3)-Invariant Denoising Distance Matching

Anonymous Author(s)
Affiliation
Address
email

Abstract

Pretraining molecular representations is critical in a variety of applications for drug and material discovery due to the limited number of labeled molecules, yet most existing work focuses on pretraining on 2D molecular graphs. The power of pretraining on 3D geometric structures, however, has been less explored. This is owning to the difficulty of finding a sufficient proxy task that can empower the pretraining to effectively extract essential features from the geometric structures. Motivated by the dynamic nature of 3D molecules, where the continuous motion of a molecule in the 3D Euclidean space forms a smooth potential energy surface, we propose a 3D coordinate denoising pretraining framework to model such an energy landscape. Leveraging an SE(3)-invariant score matching method, we propose GeoSSL in which the coordinate denoising proxy task is effectively boiled down to the denoising of the pairwise atomic distances in a molecule. Our comprehensive experiments confirm the effectiveness and robustness of our proposed method.

1 Introduction

Learning effective molecular representations is critical in a variety of tasks in drug and material discovery, such as molecular property prediction [12, 18, 64] and *de novo* molecular design and optimization [6, 45, 66]. Recent work based on graph neural networks (GNNs) [18] has shown superior performance thanks to the simplicity and effectiveness of GNNs in modeling graph-structured data. Recently, there is growing interest in developing pretraining or self-supervised learning methods for learning molecular representations by leveraging the huge amount of unlabeled molecule data [25, 32, 54, 65]. These methods have shown superior performance on many tasks, especially when the number of labeled molecules is insufficient. One limitation of these approaches is that they represent molecules as topological graphs and molecular representations are learned through pretraining 2D topological structures (*i.e.*, based on the covalent bonds). But intrinsically, for molecules, a more natural representation is their 3D geometric structures, which largely determine the corresponding physical and chemical properties. Recent work [18, 33] has empirically verified the importance of applying 3D geometric information for molecular property prediction tasks. Therefore, a more promising direction is to pretrain molecular representations based on their 3D geometric structures.

The main challenge for molecule geometric pretraining arises from discovering an effective proxy task to empower the pretraining to extract essential features from the 3D geometric structures. Our proxy task is motivated by the following observations. Studies [42] have shown that molecules are not static but in a continuous motion in the 3D Euclidean space, forming a potential energy surface (PES). As shown in Figure 1, it is desirable to study the molecule in the local minima of the PES, called *conformer*. However, such stable state conformer often comes with different noises for the following reasons. First, the statistical and systematic errors on conformation estimation are unavoidable [10]. Second, it has been well-acknowledged that a conformer can have vibrations around the local minima in PES. Thus we want to denoise the molecular coordinates to mimic these errors.

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

To achieve the aforementioned goal, we propose GeoSSL, an SE(3)-invariant denoising distance matching pretraining algorithm. In a nutshell, to capture the smooth energy surface around the local minima, we aim to maximize the mutual information (MI) between a given *stable geometry* and its *perturbed version* (*i.e.*, $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ in Figure 1). In practice, it is difficult to directly maximize the mutual information between two random variables. Thus, we propose to maximize a lower-bound of the above mutual information, which in turn amounts to denoising a geometric structure. Moreover, directly denoising such noisy coordinates, nevertheless, remains challenging because one may need to effectively constrain the pairwise atomic distances while changing the atomic coordinates. To cope with this obstacle, we further leverage an SE(3)-invariant score matching method to successfully transform the coordinate denoising desire to the denoising of pairwise atomic distances.

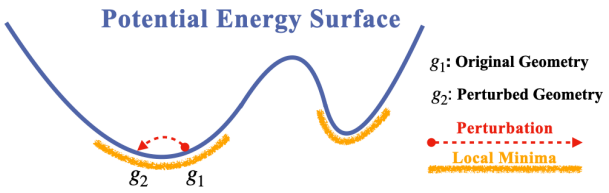

Figure 1: Illustration on coordinate geometry of molecules. The molecule is in a continuous motion, forming a potential energy surface (PES), where each 3D coordinate (x-axis) corresponds to an energy value (y-axis). The provided molecules, *i.e.*, conformers, are in the local minima ($\boldsymbol{g}_1$). It often comes with noises around the minima (*e.g.*, statistical and systematic errors or vibrations), which can be captured using the perturbed geometry ($\boldsymbol{g}_2$).

Our main contributions are summarized as follows. (1) We propose a novel coordinate denoising method for molecular geometry pretraining, which to the best of our knowledge is the first to only utilize 3D molecular data for pretraining. (2) To overcome the challenge of attaining the coordinate denoising objective, we introduce an SE(3)-invariant score matching strategy to successfully transform such objective into the denoising of pairwise atomic distances, which can be effectively computed. (3) We empirically demonstrate the effectiveness and robustness of our proposed method, GeoSSL.

2 Method

We denote each molecule 3D position (conformer) as $\boldsymbol{g} = (X, R)$. Here $X \in \mathbb{R}^{n\times d}$ is the atom attribute matrix and $R \in \mathbb{R}^{n\times 3}$ is the atom 3D-coordinate matrix, where n is the number of atoms and d is the feature dimension. The representations for the i-th node and whole molecule are:

$$h_i = \text{GNN-3D}(T(\boldsymbol{g}))_i = \text{GNN-3D}(T(X,R))_i, \qquad h = \text{READOUT}\big(h_0, \ldots, h_{n-1}\big), \tag{1}$$

where T is the transformation function like atom masking, and READOUT is the readout function. In this work, we take the mean over all the node representations as the readout function.

2.1 Coordinate Perturbation for Geometric Data

The mainstream self-supervised learning community designs the pretraining task by defining multiple views from the data, and these views share common information to some degree. Thus, by designing generative or contrastive task to maximize the mutual information (MI) between these views, the pretrained representation can encode certain key information. This will make the representation more robust and can be more generalizable to downstream tasks. In our work, we propose GeoSSL, an SE(3)-invariant self-supervised learning (SSL) method for molecule geometric representation learning.The 3D geometric information, or the atomic coordinates are critical to molecular properties. We carry out an additional ablation study to verify this in Appendix C. Then based on this acknowledgement, we propose a geometry perturbation, which adds small noises to the atom coordinates. For notation, following Appendix B, we define the original geometry graph and an augmented geometry graph as two views, denoted as $\boldsymbol{g}_1 = (X_1, R_1)$ and $\boldsymbol{g}_2 = (X_2, R_2)$ respectively. The augmented geometry graph can be seen as a coordinate perturbation to the original graph with the same atom types, *i.e.*, $X_2 = X_1$ and $R_2 = R_1 + \epsilon$, where ϵ is drawn from a normal distribution.

2.2 Coordinate Denoising with Mutual Information Maximization

The two views defined above share certain common information. To maximize the MI, we turn to maximizing the following lower bound on the two geometry views:

$$I(G_1; G_2) = \mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log \frac{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}{p(\boldsymbol{g}_1)p(\boldsymbol{g}_2)}\Big] \geq \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log p(\boldsymbol{g}_1|\boldsymbol{g}_2) + \log p(\boldsymbol{g}_2|\boldsymbol{g}_1)\Big] \triangleq \mathcal{L}_{\text{MI}}. \tag{2}$$

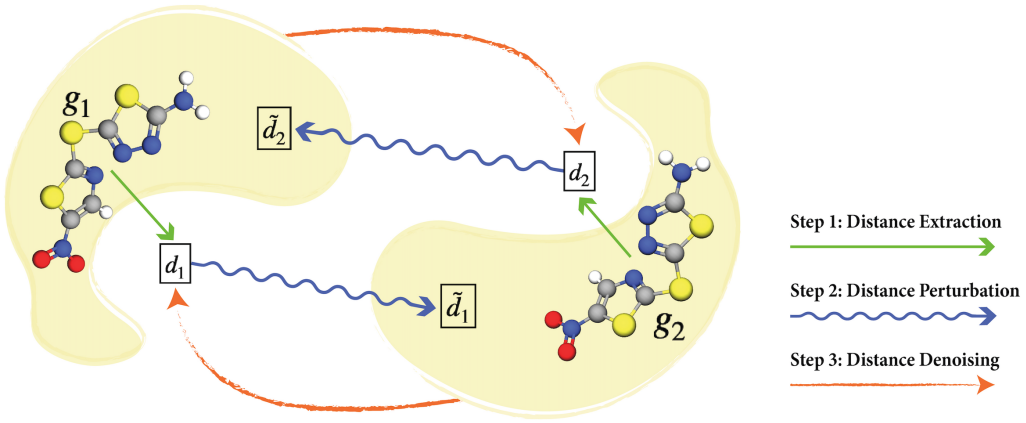

Figure 2: Pipeline for GeoSSL (GeoSSL). The $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are around the same local minima, yet with coordinate noises perturbation. Originally we want to conduct coordinate denoising between these two views. Then as proposed in GeoSSL, we transform it to an equivalent problem, *i.e.*, distance denoising. This figure shows the three key steps: extract the distances from the two geometric views, then perform distance perturbation, and finally denoise the perturbed distances. Notice that the covalent bonds are added for illustration only.

To solve Equation (2), we introduce the energy-based model (EBM) for estimation. EBM has been acknowledged as a flexible framework for its powerful usage in modeling distribution over highly-structured data, like molecules [24, 31]. The lower bound can be turned into:

$$\mathcal{L}_{\text{Coor-MI}} = \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log\frac{\exp(f(R_1,\boldsymbol{g}_2))}{A_{R_1|\boldsymbol{g}_2}}\Big] + \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_2,\boldsymbol{g}_1)}\Big[\log\frac{\exp(f(R_2,\boldsymbol{g}_1))}{A_{R_2|\boldsymbol{g}_1}}\Big], \tag{3}$$

where the $f(\cdot)$ are the negative of energy functions, and $A_{R_1|\boldsymbol{g}_2}$ and $A_{R_2|\boldsymbol{g}_1}$ are the intractable partition functions. This equation can be treated as denoising the atom coordinates of one view from the geometry of the other view.

2.3 From Coordinate Denoising to Distance Denoising: GeoSSL

2.3.1 Denoising Distance Matching

Score. The score is defined as the gradient of the negative energy function w.r.t. the atom coordinates:

$$s(R_1,\boldsymbol{g}_2) \triangleq \nabla_{R_1}\log p(R_1|\boldsymbol{g}_2) = \nabla_{R_1} f(R_1,\boldsymbol{g}_2). \tag{4}$$

Score Decomposition: From Coordinates To Distances. Through back-propagation [46], the score on atom coordinates can be further decomposed into the scores attached to pairwise distances:

$$s(R_1,\boldsymbol{g}_2)_i = \sum_{j\neq i}\frac{\partial f(R_1,\boldsymbol{g}_2)}{\partial d_{1,ij}}\cdot\frac{\partial d_{1,ij}}{\partial r_{1,i}} = \sum_{j\neq i}\frac{1}{d_{1,ij}}\cdot s(\boldsymbol{d}_1,\boldsymbol{g}_2)_{ij}\cdot(r_{1,i}-r_{1,j}), \tag{5}$$

where $s(\boldsymbol{d}_1,\boldsymbol{g}_2)_{ij} \triangleq \frac{\partial f(R_1,\boldsymbol{g}_2)}{\partial d_{1,ij}}$. Such decomposition has a nice underlying intuition from the pseudo-force perspective: the pseudo-force on each atom can be further decomposed as the summation of pseudo-forces attached to the pairwise distances between this atom and all its neighbors. Note that here the pairwise atoms are connected in the 3D Euclidean space, not by the covalent bonds.

Denoising Distance Matching (DDM). Then we adopt the denoising score matching (DSM) [59] to our task. To be more concrete, we take the Gaussian kernel as the perturbed noise distribution on each pairwise distance, *i.e.*, $q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{g}_2) = \mathbb{E}_{p_{\text{data}}(\boldsymbol{d}_1|\boldsymbol{g}_2)}[q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1)]$, where σ is the deviation in Gaussian perturbation. One main advantage of using the Gaussian kernel is that the following gradient of conditional log-likelihood has a closed-form formulation: $\nabla_{\tilde{\boldsymbol{d}}_1}\log q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1,\boldsymbol{g}_2) = (\boldsymbol{d}_1-\tilde{\boldsymbol{d}}_1)/\sigma^2$, and the objective function of DSM is to train a score network to match it.

To adapt to our setting, by taking the Fisher divergence as the discrepancy metric and the trick mentioned above, the estimation objective can be simplified to:

$$D_F(q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{g}_2)||p_\theta(\tilde{\boldsymbol{d}}_1|\boldsymbol{g}_2)) = \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\boldsymbol{d}_1|\boldsymbol{g}_2)}\mathbb{E}_{q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1,\boldsymbol{g}_2)}\big[\|s_\theta(\tilde{\boldsymbol{d}}_1,\boldsymbol{g}_2) - \frac{\boldsymbol{d}_1-\tilde{\boldsymbol{d}}_1}{\sigma^2}\|^2\big] + C. \tag{6}$$

For more detailed derivations, please refer to Appendix D.

Table 1: Downstream results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
–	0.048	44.50	26.00	21.11	0.016	0.025	8.31	7.67	0.132	7.77	7.89	1.322
Supervised	0.049	45.33	26.61	21.77	0.016	0.026	8.97	8.59	0.170	8.35	8.19	1.346
Type Prediction	0.050	47.28	30.56	23.18	0.016	**0.024**	9.32	9.10	0.163	8.94	8.60	1.357
Distance Prediction	0.063	47.62	29.18	22.40	0.019	0.045	12.02	12.31	0.636	11.76	12.22	1.840
Angle Prediction	0.056	47.36	29.53	22.61	0.018	0.027	10.23	10.13	0.143	9.95	9.70	1.643
3D InfoGraph	0.053	44.79	27.09	21.66	0.016	0.027	9.22	8.78	0.143	8.94	9.11	1.465
RR	0.048	44.85	25.42	20.82	**0.015**	0.025	8.56	8.20	0.133	7.89	7.62	1.329
InfoNCE	0.052	45.65	26.70	21.87	0.016	0.027	9.17	9.62	0.130	8.77	8.63	1.519
EBM-NCE	0.049	44.18	26.29	21.46	**0.015**	0.026	8.56	8.13	0.126	8.01	7.96	1.447
GeoSSL (ours)	**0.046**	**40.22**	**23.48**	**19.42**	**0.015**	**0.024**	**7.65**	**7.09**	**0.122**	**6.99**	**6.92**	**1.307**

Table 2: Downstream results on 8 force prediction tasks from MD17. We take 1K for training, 1K for validation, and the number of molecules for test are varied among different tasks, ranging from 48K to 991K. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
–	0.556	0.052	0.213	0.338	0.138	0.288	0.155	0.194
Supervised	0.478	0.145	0.318	0.434	0.460	0.527	0.251	0.404
Type Prediction	1.656	0.349	0.414	0.886	1.684	1.807	0.660	1.020
Distance Prediction	1.434	0.090	0.378	1.017	0.631	1.569	0.350	0.415
Angle Prediction	0.839	0.105	0.337	0.517	0.772	0.931	0.274	0.676
3D InfoGraph	0.844	0.114	0.344	0.741	1.062	0.945	0.373	0.812
RR	0.502	0.052	0.219	0.334	0.130	0.312	0.152	0.192
InfoNCE	0.881	0.066	0.275	0.550	0.356	0.607	0.186	0.559
EBM-NCE	0.598	0.073	0.237	0.518	0.246	0.416	0.178	0.475
GeoSSL (ours)	**0.453**	**0.051**	**0.166**	**0.288**	**0.129**	**0.266**	**0.122**	**0.183**

3 Experiments

In this section, we compare our method with nine 3D geometric pretraining baselines, including one randomly-initialized, one supervised, and seven self-supervised approaches. For the downstream tasks, we adopt 22 tasks covering quantum mechanics prediction, force prediction and binding affinity prediction. Our proposed GeoSSL is model-agnostic, and here we evaluate our method using one of the state-of-the-art geometric graph neural networks, PaiNN [44]. Due to space limit, pther experiment details, *e.g.*, the pretraining dataset and downstream datasets, and pretraining baselines are also provided in Appendix E.

QM9 [40] and MD17 [9] are two datasets on the quantum mechanics and force prediction. The results are displayed in Tables 1 and 2 respectively. From Tables 1 and 2, we can observe that most the pretraining baselines tested perform on par with or even worse than the randomly-initialized baseline. Promisingly, our proposed GeoSSL achieves consistently improved performance on all the 20 tasks in QM9 and MD17. All these observations empirically verify the effectiveness of the distance denoising, which models the most determinant factor in molecule geometric data. The ligand binding affinity (LBA) and ligand efficacy prediction (LEP) are two binding affinity prediction tasks proposed in Atom3D [56]. Results in Table 11 (Appendix) indicate that, for the LBA task, one pretraining method fails to generalize to LBA (the loss gets too large), and all the other pretraining baselines cannot beat the randomly-initialized baseline. For the LEP task, the supervised and two contrastive learning pretraining baselines stand out for both ROC and PR metrics. Meaningfully, for both tasks, GeoSSL is able to achieve promising improvement, revealing that modeling the local region around conformer with distance denoising can also benefit for binding affinity downstream tasks.

4 Conclusions and Future Directions

We proposed a novel coordinate denoising method, coined GeoSSL, for molecular geometry pretraining, and showed its superior performance to state-of-the-art pretraining baselines.

Our work opens up venues for multiple promising directions. First from the machine learning perspective, we propose a general pipeline on using Energy Based Model (EBM) for MI maximization. Yet, there are more explorations on the success of EBM, like GFlowNet [3], and it would be interesting to explore how to combine it with molecular geometric data along this systematic path. In addition, GeoSSL does not utilize the 2D structure (*i.e.*, covalent bonds for molecules), and it would be desirable to consider how to utilize the distance denoising together with the 2D topology information.

References

[1] Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, pp. 1–10, 2021. 9

[2] Simon Axelrod and Rafael Gomez-Bombarelli. Geom: Energy-annotated molecular conformations for property prediction and molecular generation. *arXiv preprint arXiv:2006.05531*, 2020. 10

[3] Yoshua Bengio, Tristan Deleu, Edward J Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *arXiv preprint arXiv:2111.09266*, 2021. 4

[4] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998. 14

[5] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik Bekkers, and Max Welling. Geometric and physical quantities improve e(3) equivariant message passing. *arXiv preprint arXiv:2110.02905*, 2021. 9

[6] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019. 1

[7] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 392–401, 2020. 9

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021. 15

[9] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017. 4, 15

[10] John D Chodera and Frank Noé. Markov state models of biomolecular conformational dynamics. *Current opinion in structural biology*, 25:135–144, 2014. 1, 11

[11] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020. 10

[12] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015. 1

[13] Thomas Engel and Johann Gasteiger. *Applied chemoinformatics: achievements and future opportunities*. John Wiley & Sons, 2018. 10

[14] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Chemrl-gem: Geometry enhanced molecular representation learning for property prediction. *arXiv preprint arXiv:2106.06130*, 2021. 9

[15] Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020. 9

[16] Siddhant Garg and Yingyu Liang. Functional regularization for representation learning: A unified theoretical perspective. *Advances in Neural Information Processing Systems*, 33:17187–17199, 2020. 14

[17] Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022. 9

[18] 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. 1

[19] Jonathan Godwin, Michael Schaarschmidt, Alexander L Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple GNN regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1wVvweK3oIb. 11

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 15

[21] Yuzhi Guo, Jiaxiang Wu, Hehuan Ma, and Junzhou Huang. Self-supervised pre-training for protein embeddings using tertiary structures. 2022. 10

[22] Michael Gutmann and Aapo 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. JMLR Workshop and Conference Proceedings, 2010. 10, 12, 14

[23] Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996. 11

[24] Ryuichiro Hataya, Hideki Nakayama, and Kazuki Yoshizoe. Graph energy-based model for molecular graph generation. In *Energy Based Models Workshop-ICLR 2021*, 2021. 3

[25] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations, ICLR*, 2020. 1, 9

[26] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, pp. 1857–1867, 2020. 9

[27] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 10

[28] Johannes Klicpera, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020. 9

[29] Johannes Klicpera, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 9

[30] Greg Landrum et al. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013. 11

[31] Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. Graphebm: Molecular graph generation with energy-based models. *arXiv preprint arXiv:2102.00546*, 2021. 3

[32] Shengchao Liu, Mehmet Furkan Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *arXiv preprint arXiv:1806.09206*, 2018. 1, 9

[33] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xQUe1pOKPam. 1, 9, 14, 15

[34] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 9

[35] Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021. 9, 14, 15

[36] Yixin Liu, Shirui Pan, Ming Jin, Chuan Zhou, Feng Xia, and Philip S Yu. Graph self-supervised learning: A survey. *arXiv preprint arXiv:2103.00111*, 2021. 9

[37] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 9, 16

[38] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. Learning graph-convolutional representations for point cloud denoising. In *European conference on computer vision*, pp. 103–118. Springer, 2020. 9

[39] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 9

[40] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014. 4, 15

[41] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021. 9

[42] H Bernhard Schlegel. Exploring potential energy surfaces for chemical reactions: an overview of some practical methods. *Journal of computational chemistry*, 24(12):1514–1527, 2003. 1

[43] 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. 9, 14, 15

[44] Kristof T Schütt, Oliver T Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021. 4, 9, 14, 15

[45] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020. 1

[46] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, pp. 9558–9568. PMLR, 2021. 3, 13

[47] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1711–1719, 2020. 9

[48] Muhammed Shuaibi, Adeesh Kolluru, Abhishek Das, Aditya Grover, Anuroop Sriram, Zachary Ulissi, and C Lawrence Zitnick. Rotation invariant graph neural networks using spin convolutions. *arXiv preprint arXiv:2106.09575*, 2021. 9

[49] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 13, 14

[50] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 13, 14

[51] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021. 10, 12

[52] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 10, 12, 14

[53] Antonia Stank, Daria B Kokh, Jonathan C Fuller, and Rebecca C Wade. Protein binding pocket dynamics. *Accounts of chemical research*, 49(5):809–815, 2016. 15

[54] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations, ICLR*, 2020. 1, 9

[55] 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. 9

[56] Raphael JL Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Brandon Anderson, Stephan Eismann, Risi Kondor, et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020. 4, 15, 16

[57] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019. 9

[58] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pp. arXiv–1807, 2018. 15

[59] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 3, 10, 13

[60] Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005. 16

[61] Lirong Wu, Haitao Lin, Zhangyang Gao, Cheng Tan, Stan Li, et al. Self-supervised on graphs: Contrastive, generative, or predictive. *arXiv preprint arXiv:2105.07342*, 2021. 9

[62] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021. 9

[63] Zhao Xu, Youzhi Luo, Xuan Zhang, Xinyi Xu, Yaochen Xie, Meng Liu, Kaleb Dickerson, Cheng Deng, Maho Nakata, and Shuiwang Ji. Molecule3d: A benchmark for predicting 3d geometries from molecular graphs. *arXiv preprint arXiv:2110.01717*, 2021. 15

[64] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019. 1

[65] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020. 1, 9

[66] Chengxi Zang and Fei Wang. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 617–626, 2020. 1

A Benchmarks and Related Work

A.1 Equivariant Geometric Molecule Representation Learning

Geometric Representation Learning. Recently, 3D geometric representation learning has been widely explored in the machine learning community, including but not limited to 3D point clouds [7, 38, 47, 57], N-body particle [39, 41], and 3D molecular conformation [5, 28, 29, 35, 43, 44, 48], amongst many others. The learned representation should satisfy the physical constraints, *e.g.*, it should be equivariant to the transition on the Euclidean space. Such constraints can be depicted with the group symmetry as introduced below.

SE(3)-Invariance Energy. Constrained by the physical nature of 3D data, a key principle we need to follow is to learn an SE(3)-equivariant representation function. The SE(3) is the special Euclidean group consisting of rigid transformations in the 3D Cartesian space, where the transformations include all the combinations of translations and rotations. Namely, the learned representation should be equivariant to translations and rotations for molecule geometries. We also note that for some specific tasks like molecular chirality [1], the representation needlessly satisfy the reflection equivariance. For more rigorous discussion, please check [15, 17, 55]. In this work, we will design an SE(3)-invariant energy function based on an SE(3)-equivariant representation backbone model.

A.2 Self-Supervised Learning for Molecule Representation Learning

In general, there are two categories of self-supervised learning (SSL) [34, 36, 61, 62]: contrastive and generative, and the main difference is if the supervised signals are constructed in an inter-data or intra-data manner. To be more concrete, contrastive SSL extracts two views from the data and designs the supervised signals by detecting if the sampled view pairs are from the same data, and generative SSL learns structural information by reconstructing partial information from the data itself.

2D Molecular Graph (Topology) Self-Supervised Learning. Currently, one of the mainstream research lines for molecule pretraining is on the 2D molecular graph. It treats the molecules as 2D graph, where atoms and bonds are nodes and edges respectively. It then carries out a pretraining task by either detecting if the two augmentations (*e.g.*, neighborhood extraction, node dropping, edge dropping, etc) correspond to the same molecular graph [25, 54, 65] or if the representation can successfully reconstruct the masked subgraph in an auto-encoding manner [25, 26, 32].

3D Molecular Graph (Geometry) Self-Supervised Learning. As the increasing interest on the 3D geometric representation learning, there has been some initial explorations [14, 33] involving the geometric SSL for molecules. GraphMVP [33] introduces an extra 2D topology and employs detection and reconstruction tasks simultaneously between 2D and 3D graphs, yet it focuses on 2D downstream tasks. ChemRL-GEM [14] designs a novel model using both the 2D and 3D molecular graphs. In terms of SSL, it utilizes the geometry information by taking the distance prediction and angle prediction as the generative pretraining tasks. Some of their geometric SSL tasks will be used as baselines in our work, yet we want to highlight that our work is focusing on the pure 3D geometric data without the covalent bonds (2D topology). To the best of our knowledge, our work is the first to explicitly do SSL on pure 3D geometry along the molecule representation learning research line.

A.3 Benchmark on QM9

Current work is using different optimization strategies and different data split (in terms of the splitting size). Originally there are 133,885 molecules in QM9, where 3,054 are filtered out, leading to 130,831 molecules. During the benchmark, we find that:

- The performance on QM9 is very robust to either using (1) 110K for training, 10K for val, 10,831 for test or using (2) 100K for training, 13,083 for val and 17,748 for test.
- The optimization, especially the learning rate scheduler is very critical. During the benchmarking, we find that using cosine annealing learning rate schedule [37] is generally the most robust.

For more detailed discussion on QM9, please refer to Appendix E. We show the benchmark results on QM9 in Table 3.

Table 3: Benchmark results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation is mean absolute error (MAE).

	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
SchNet	0.070	50.38	31.81	25.76	0.029	0.031	14.60	14.24	0.131	13.99	14.12	1.686
SE(3)-Trans	0.136	58.27	35.95	35.41	0.052	0.068	68.50	70.22	1.828	70.14	72.28	5.302
EGNN	0.067	48.77	28.98	24.44	0.032	0.031	11.02	11.07	0.078	10.83	10.70	1.578
DimeNet++	0.046	38.14	21.23	17.57	0.029	0.022	7.98	7.19	0.306	6.86	6.93	1.204
SphereNet	0.050	39.54	21.88	18.66	0.026	0.025	8.65	7.43	0.262	8.28	8.01	1.390
SEGNN	0.057	41.08	22.46	21.46	0.025	0.028	13.07	13.94	0.472	14.64	13.89	1.662
PaiNN	0.048	44.50	26.00	21.11	0.016	0.025	8.31	7.67	0.132	7.77	7.89	1.322

A.4 Related Work

We acknowledge that there is a parallel work called Protein Tertiary SSL (PTSSL) [21] working on the geometric self-supervised learning. Yet, there are some fundamental differences between theirs and ours, as listed below: **(1) Key notion on pseudo-force.** PTSSL directly applies the denoised score matching method into protein tertiary structures, yet our focus is on how the notion of pseudo-force can come into the play, which possess better generalization ability. **(2) Task setting.** PTSSL works on protein and utilize both the 2D and 3D information, and our work is purely working on the 3D geometric information. **(3) Technical novelty.** PTSSL designs the DSM objective for SSL, and what we propose is a systematic tool: using energy-based model and score matching to solve the geometric SSL problem opens a new venue in this field. **(4) Objective.** PTSSL directly designs one objective function, which is denoising from one view to the other. Ours starts from the lower bound of MI, which is symmetric in terms of the denoising directions. We believe that such symmetry are treating the two views equally, and can better reveal the mutual concept, making the pre-trained representation more robust to the position augmentations. **(5) Empirical baseline.** PTSSL lacks the comparisons with other pre-training methods, while we compare with 7 SOTA pre-training methods, especially those driven by maximizing the MI with the same augmentations. Without such comparisons, it is hard to tell the effectiveness of the pseudo-force matching for geometric data. **(6) Score network.** Last but not least, the score network designed in PTSSL does not satisfy the SE(3) equivariant property.

B Preliminaries

Molecular Geometry Graph. Molecules can be naturally featured in a geometric formulation, *i.e.*, all the atoms are spatially located in 3D Euclidean space. Note that the covalent bonds are added heuristically by expert rules, so they are only applicable in 2D topology graph not 3D geometry graph. Besides, atoms are not static, but in a continual motion along a potential energy surface [2]. The 3D structures at the local minima on this surface are named *conformer*, as shown in Figure 1. Conformers at such equilibrium state possess nice properties and we would like to model them during pretraining.

Geometric Neural Network. We denote each conformer as $\boldsymbol{g} = (X, R)$. Here $X \in \mathbb{R}^{n\times d}$ is the atom attribute matrix and $R \in \mathbb{R}^{n\times 3}$ is the atom 3D-coordinate matrix, where n is the number of atoms and d is the feature dimension. The representations for the i-th node and whole molecule are:

$$h_i = \text{GNN-3D}(T(\boldsymbol{g}))_i = \text{GNN-3D}(T(X, R))_i, \qquad h = \text{READOUT}\big(h_0, \ldots, h_{n-1}\big), \tag{7}$$

where T is the transformation function like atom masking, and READOUT is the readout function. In this work, we take the mean over all the node representations as the readout function.

Energy-Based Model and Denoising Score Matching. Energy-based model (EBM) is a flexible tool for modeling the underlying data distribution in the form of Gibbs distribution as $p_\theta(\boldsymbol{x}) = \exp(-E(\boldsymbol{x}))/A$, where $p_\theta(\boldsymbol{x})$ is the model distribution, A is the normalization constant and it is intractable due to the high cardinality of the data space. Recently, there has been various progress in solving this intractable function, including contrastive divergence [11], noise contrastive estimation [22], and score matching (SM) [27, 51, 52]. Specifically, SM solves this by introducing a concept called score: it is the gradient of the log-likelihood with respect to the data. SM then matches the model score and data score using Fisher divergence. Further along this research line, denoising score matching (DSM) [59] combines SM with denoising auto-encoding. The main advantage of DSM is that its solution is equivalent to SM yet with a computationally feasible and efficient solver. In this work, we will explore how DSM can be applied for molecule geometry representation learning by utilizing the distance information, one of the most fundamental factors in the geometric data.

Problem Setup. Our goal here is to apply a self-supervised pretraining algorithm on a large molecular geometric dataset, and adapt the pretrained representation for fine-tuning on geometric downstream tasks. For both the pretraining and downstream tasks, only the 3D geometric information is available, and our solution is agnostic in terms of the backbone geometric neural network.

C An Example On The Importance of Atom Coordinates

First it has been widely acknowledged [13] that the atom positions or molecule shapes are important factors to the quantum properties. Here we carry out an evidence example to empirically verify this. The goal here is to make predictions on 12 quantum properties in QM9.

The molecule geometric data includes two main components as input features: the atom types and atom coordinates. Other key information can be inferred accordingly, including the pairwise distances and torsion angles. We consider corruption on each of the component to empirically test their importance accordingly.

- Atom type corruption. There are in total 118 types of atom types, and the standard embedding option is to apply the one-hot encoding. In the corruption case, we replace all the atom types with a hold-out index, *i.e.*, index 119.

- Atom coordinate corruption. Originally QM9 includes atom coordinates that are in the stable state, and now we replace them with the coordinates generated with MMFF [23] from RDKit [30].

Table 4: An evidence example on molecular data. The goal is to predict 12 quantum properties (regression tasks) of 3D molecules (with 3D coordinates on each atom). The evaluation metric is MAE.

Model	Mode	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
SchNet	Stable Geometry	0.070	50.59	32.53	26.33	0.029	0.032	14.68	14.85	0.122	14.70	14.44	1.698
	Type Corruption	0.074	52.07	33.64	26.75	0.032	0.032	21.68	22.93	0.231	23.01	22.99	1.677
	Coordinate Corruption	0.265	110.59	79.92	78.59	0.422	0.113	57.07	58.92	18.649	60.71	59.32	5.151
PaiNN	Stable Geometry	0.048	44.50	26.00	21.11	0.016	0.025	8.31	7.67	0.132	7.77	7.89	1.322
	Type Corruption	0.057	45.61	27.22	22.16	0.016	0.025	11.48	11.60	0.181	11.15	10.89	1.339
	Coordinate Corruption	0.223	108.31	73.43	72.35	0.391	0.095	48.40	51.82	16.828	51.43	48.95	4.395

We take SchNet and PaiNN as the backbone 3D GNN models, and the results are in Table 4. We can observe that (1) Both corruption examples lead to performance decrease. (2) The atom coordinate corruption may lead to more severe performance decrease than the atom type corruption. To put this into another way is that, when we corrupt the atom types with the same hold-out type, it is equivalently to removing the atom type information. Thus, this can be viewed as using the equilibrium atom coordinates alone, and the property prediction is comparatively robust. This observation can also be supported from the domain perspective. According to the valence bond theory, the atom type information can be implicitly and roughly inferred from the atom coordinates.

Therefore, by combining all the above observations and analysis, one can draw the conclusion that, *for molecule geometry data, the atom coordinates reveal more fundamental information for representation learning*.

D Mutual Information Maximization with Energy-Based Model

In this section, we will give a detailed discussion on the mutual information (MI) maximization with energy-based model (EBM).

First, let us recall the definition of MI. MI measures the non-linear dependency between two variables, defined as:

$$I(\boldsymbol{g}_1; \boldsymbol{g}_2) = \mathbb{E}_{p(\boldsymbol{g}_1, \boldsymbol{g}_2)}\big[\log \frac{p(\boldsymbol{g}_1, \boldsymbol{g}_2)}{p(\boldsymbol{g}_1)p(\boldsymbol{g}_2)}\big]. \tag{8}$$

Notice that to keep consistent with the notations above, we will be using $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ as the two variables. Then we can obtain a lower bound to MI:

$$I(\boldsymbol{g}_1; \boldsymbol{g}_2) = \mathbb{E}_{p(\boldsymbol{g}_1, \boldsymbol{g}_2)}\Big[\log \frac{p(\boldsymbol{g}_1, \boldsymbol{g}_2)}{p(\boldsymbol{g}_1)p(\boldsymbol{g}_2)}\Big] \geq \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1, \boldsymbol{g}_2)}\Big[\log p(\boldsymbol{g}_1|\boldsymbol{g}_2) + \log p(\boldsymbol{g}_2|\boldsymbol{g}_1)\Big] \triangleq \mathcal{L}_{\text{MI}}. \tag{9}$$

Thus, we transform the MI maximization problem into maximizing the summation of two conditional log-likelihoods. Such objective function opens a wider venue for estimating MI, *e.g.*, using the EBM to estimate Equation (9).

Adaptation to Geometric Data The 3D geometric information, or the atomic coordinates are critical to molecular properties. Then based on this, we propose a geometry perturbation, which adds small noises to the atom coordinates. This geometry perturbation possess certain motivations from both domain and machine learning perspectives. (1) From the practical experiment perspective, the statistical and systematic errors [10] on conformation estimation are unavoidable. Coordinate perturbation is a natural way to enable learning representations robust to such noises. (2) From the domain aspect, molecules are not static but in a continuous motion in the 3D Euclidean space, and we can obtain a potential energy surface accordingly. We are interested in modeling the conformer, *i.e.*, the 3D coordinates with the lowest energy. However, even the conformer at the lowest energy point can have vibrations, and coordinate perturbation can better capture such movement yet with the same order of magnitude on energies. (3) As will be illustrated later, our proposed method can be simplified as denoising atomic distance matching. (4) Leveraging coordinate perturbation for model regularization has also been empirically verified its effectiveness for supervised molecule geometric representation learning [19]. Such characteristics of the molecular geometry motivate us to apply the coordinate perturbation. If we take each of the two views as adding noise to the coordinates from the other view, then the objective in Equation (9) essentially states that we want to conduct coordinate denoising, as shown in Figure 3. Yet, this is not a trivial task due to the complicated geometric space (*e.g.*, 3D coordinates) reconstruction.

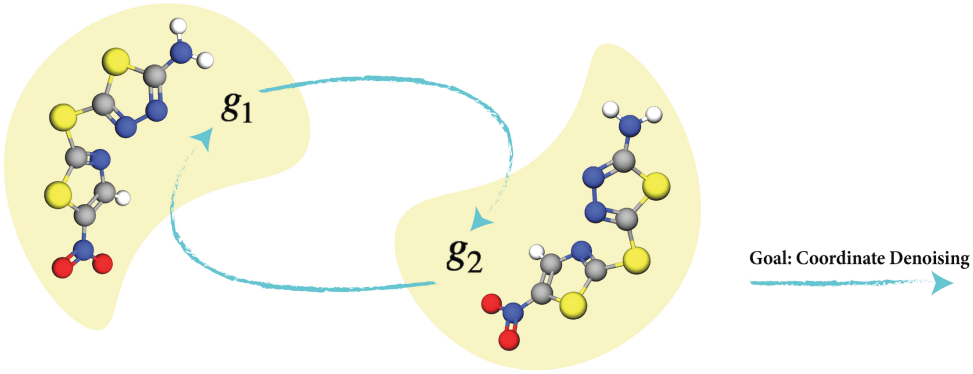

Figure 3: Pipeline for denoising coordinate matching.

D.1 An EBM framework for MI estimation

The lower bound in Equation (9) is composed of two conditional log-likelihood terms, and then we model the conditional likelihood with EBM. This gives us:

$$\mathcal{L}_{\text{EBM}} = -\frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log\frac{\exp(f_{\boldsymbol{g}_1}(\boldsymbol{g}_1,\boldsymbol{g}_2))}{A_{\boldsymbol{g}_1|\boldsymbol{g}_2}} + \log\frac{\exp(f_{\boldsymbol{g}_2}(\boldsymbol{g}_2,\boldsymbol{g}_1))}{A_{\boldsymbol{g}_2|\boldsymbol{g}_1}}\Big], \tag{10}$$

where $f_{\boldsymbol{g}_1}(\boldsymbol{g}_1,\boldsymbol{g}_2) = -E(\boldsymbol{g}_1|\boldsymbol{g}_2)$ and $f_{\boldsymbol{g}_2}(\boldsymbol{g}_2,\boldsymbol{g}_1) = -E(\boldsymbol{g}_2|\boldsymbol{g}_1)$ are the negative energy functions, and $A_{\boldsymbol{g}_1|\boldsymbol{g}_2}$ and $A_{\boldsymbol{g}_2|\boldsymbol{g}_1}$ are the corresponding partition functions. The energy functions can be flexibly defined, thus the bottleneck here is the intractable partition function due to the high cardinality. To solve this, existing methods include noise-contrastive estimation (NCE) [22] and score matching (SM) [51, 52], and we will describe how to apply them for MI maximization.

D.2 EBM-NCE for MI estimation

Under the EBM framework, if we solve Equation (10) with Noise-Contrastive Estimation (NCE) [22], the final objective is termed EBM-NCE, as:

$$\begin{aligned}\mathcal{L}_{\text{EBM-NCE}} = &-\frac{1}{2}\mathbb{E}_{p_{\text{data}}(y)}\Big[\mathbb{E}_{p_n(\boldsymbol{g}_1|\boldsymbol{g}_2)}[\log\left(1-\sigma(f_{\boldsymbol{g}_1}(\boldsymbol{g}_1,\boldsymbol{g}_2))\right)] + \mathbb{E}_{p_{\text{data}}(\boldsymbol{g}_1|\boldsymbol{g}_2)}[\log\sigma(f_{\boldsymbol{g}_1}(\boldsymbol{g}_1,\boldsymbol{g}_2))]\Big] \\ &-\frac{1}{2}\mathbb{E}_{p_{\text{data}}(x)}\Big[\mathbb{E}_{p_n(\boldsymbol{g}_2|\boldsymbol{g}_1)}[\log\left(1-\sigma(f_{\boldsymbol{g}_2}(\boldsymbol{g}_2,\boldsymbol{g}_1))\right)] + \mathbb{E}_{p_{\text{data}}(\boldsymbol{g}_2|\boldsymbol{g}_1)}[\log\sigma(f_{\boldsymbol{g}_2}(\boldsymbol{g}_2,\boldsymbol{g}_1))]\Big].\end{aligned} \tag{11}$$

All the detailed derivations can be found in [22]. Specifically, EBM-NCE is equivalent to the Jensen-Shannon estimation for MI, while the mathematical intuitions and derivation processes are different. Besides, it also belongs to the contrastive SSL venue. That is, it aims at aligning the positive pairs and contrasting the negative pairs.

D.3 EBM-SM for MI estimation: GeoSSL

In this subsection, we will be focusing on the geometric data like molecular geometry. Recall that we have two views: $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, and the goal is to maximize the lower bound of the mutual information in Equation (9). Because the two views share the same atomic features, it can be reduced to:

$$\begin{aligned}\mathcal{L}_{\text{MI}} &= \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log p(\boldsymbol{g}_1|\boldsymbol{g}_2)\Big] + \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log p(\boldsymbol{g}_2|\boldsymbol{g}_1)\Big] \\ &= \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log p(\langle X_1,R_1\rangle|\langle X_2,R_2\rangle)\Big] + \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log p(\langle X_2,R_2\rangle|\langle X_1,R_1\rangle)\Big] \\ &= \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log p(R_1|\boldsymbol{g}_2)\Big] + \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log p(R_2|\boldsymbol{g}_1)\Big] \\ &= \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_1,\boldsymbol{g}_2)}\Big[\log\frac{\exp(f(R_1,\boldsymbol{g}_2))}{A_{R_1|\boldsymbol{g}_2}}\Big] + \frac{1}{2}\mathbb{E}_{p(\boldsymbol{g}_2,\boldsymbol{g}_1)}\Big[\log\frac{\exp(f(R_2,\boldsymbol{g}_1))}{A_{R_2|\boldsymbol{g}_1}}\Big],\end{aligned} \tag{12}$$

where the $f(\cdot)$ are the negative of energy functions, and $A_{R_1|\boldsymbol{g}_2}$ and $A_{R_2|\boldsymbol{g}_1}$ are the intractable partition functions. The first equation in Equation (12) results from that the two views share the same atom types. This equation can be treated as denoising the atom coordinates of one view from the geometry of the other view. In the following, we will explore how to use the score matching for solving EBM, and further transform the coordinate-aware mutual information maximization to the denoising distance matching (GeoSSL) as the final objective.

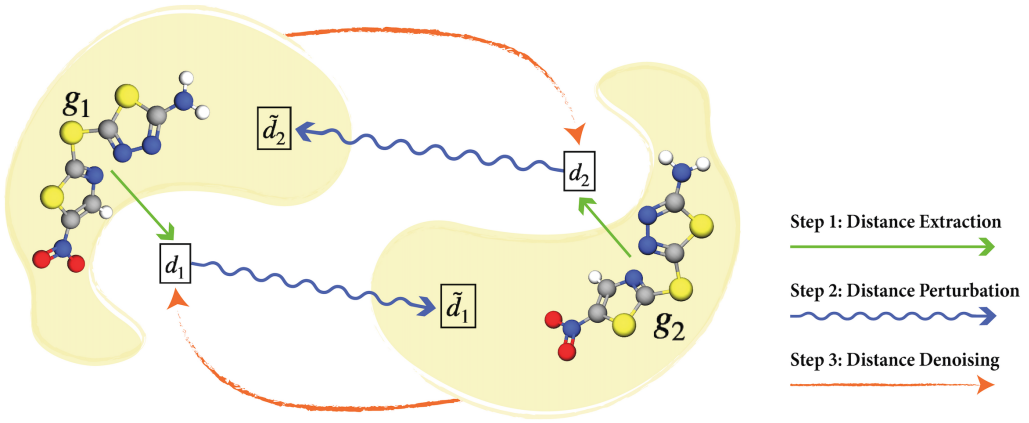

Figure 4: Pipeline for GeoSSL (GeoSSL). The $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are around the same local minima, yet with coordinate noises perturbation. Originally we want to do coordinate denoising between these two views. Then as proposed in GeoSSL, we transform it to an equivalent problem, *i.e.*, distance denoising. This figure shows the three key steps: extract the distances from the two geometric views, then perform distance perturbation, and finally denoise the perturbed distances.

Score Definition. The two terms in Equation (3) are in the mirroring direction. Thus in what follows, we may as well adopt a proxy task that these two directions can calculated separately, and take one direction for illustration, *e.g.*, $\log \frac{\exp(f(R_1,\boldsymbol{g}_2))}{A_{R_1|\boldsymbol{g}_2}}$. The score is defined as the gradient of the log-likelihood w.r.t. the data, *i.e.*, the atom coordinates in our case. Because the normalization function is a constant w.r.t. the data, it will disappear during the score calculation. To adapt it into our setting, the score is obtained as the gradient of the negative energy function w.r.t. the atom coordinates, as:

$$s(R_1, \boldsymbol{g}_2) = \nabla_{R_1} \log p(R_1|\boldsymbol{g}_2) = \nabla_{R_1} f(R_1, \boldsymbol{g}_2). \tag{13}$$

If we assume that the learned optimal energy function, *i.e.*, $f(\cdot)$, possesses certain physical or chemical information, then the score in Equation (13) can be viewed as a special form of the pseudo-force. This may require more domain-specific knowledge, and we leave this for future exploration.

Score Decomposition: From Coordinates To Distances. Through back-propagation [46], the score on atom coordinates can be further decomposed into the scores attached to pairwise distances:

$$\begin{aligned} s(R_1, \boldsymbol{g}_2)_i &= \frac{\partial f(R_1, \boldsymbol{g}_2)}{\partial r_{1,i}} \\ &= \sum_{j \in \mathcal{N}(i)} \frac{\partial f(R_1, \boldsymbol{g}_2)}{\partial d_{1,ij}} \cdot \frac{\partial d_{1,ij}}{\partial r_{1,i}} \\ &= \sum_{j \in \mathcal{N}(i)} \frac{1}{d_{1,ij}} \cdot \frac{\partial f(R_1, \boldsymbol{g}_2)}{\partial d_{1,ij}} \cdot (r_{1,i} - r_{1,j}) \\ &= \sum_{j \in \mathcal{N}(i)} \frac{1}{d_{1,ij}} \cdot s(\boldsymbol{d}_1, \boldsymbol{g}_2)_{ij} \cdot (r_{1,i} - r_{1,j}), \end{aligned} \tag{14}$$

where $s(\boldsymbol{d}_1, \boldsymbol{g}_2)_{ij} \triangleq \frac{\partial f(R_1, \boldsymbol{g}_2)}{\partial d_{1,ij}}$. Such decomposition has a nice underlying intuition from the pseudo-force perspective: the pseudo-force on each atom can be further decomposed as the summation of pseudo-forces on the pairwise distances starting from this atom. Note that here the pairwise atoms are connected in the 3D Euclidean space, not by the covalent-bonding.

Denoising Distance Matching (DDM). Then we adopt the denoising score matching (DSM) [59] to our task. To be more concrete, we take the Gaussian kernel as the perturbed noise distribution on each pairwise distance, *i.e.*, $q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{g}_2) = \mathbb{E}_{p_{\text{data}}(\boldsymbol{d}_1|\boldsymbol{g}_2)}[q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1)]$, where σ is the deviation in Gaussian perturbation. One main advantage of using the Gaussian kernel is that the following gradient of conditional log-likelihood has a closed-form formulation: $\nabla_{\tilde{\boldsymbol{d}}_1} \log q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1, \boldsymbol{g}_2) = (\boldsymbol{d}_1 - \tilde{\boldsymbol{d}}_1)/\sigma^2$, and the goal of DSM is to train a score network to match it. This trick was first introduced in [59], and has been widely utilized in the score matching applications [49, 50].

To adapt this into our setting, this is essentially saying that we want to train a "distance network", *i.e.*, $s_\theta(\tilde{\boldsymbol{d}}_1|\boldsymbol{g}_2)$, to match the distance perturbation, or we can say it aims at matching the pseudo-force with the pairwise distances from another aspect. By taking the Fisher divergence as the discrepancy metric and the trick mentioned above,

the estimation $s_\theta(\tilde{\boldsymbol{d}}_1, \boldsymbol{g}_2) \approx \nabla_{\tilde{\boldsymbol{d}}_1} \log q(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1, \boldsymbol{g}_2)$ can be simplified to the following:

$$D_F(q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{g}_2)||p_\theta(\tilde{\boldsymbol{d}}_1|\boldsymbol{g}_2)) = \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\boldsymbol{d}_1|\boldsymbol{g}_2)}\mathbb{E}_{q_\sigma(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1,\boldsymbol{g}_2)}\big[\|s_\theta(\tilde{\boldsymbol{d}}_1, \boldsymbol{g}_2) - \frac{\boldsymbol{d}_1 - \tilde{\boldsymbol{d}}_1}{\sigma^2} + \|^2\big] + C. \tag{15}$$

Final objective. We adopt the following four model training tricks from [33, 49, 50] to stabilize the score matching training process. (1) We carry out the distance denoising at L-level of noises. (2) We add a weighting coefficient $\lambda(\sigma) = \sigma^\beta$ for each noise level, where β is the annealing factor. (3) We scale the score network by a factor of $1/\sigma$. (4) We sample the exactly same atoms from the two geometry views with masking ratio r. Finally, by considering the two directions and all the above tricks, the objective function becomes the follows:

$$\begin{aligned}\mathcal{L}_{\text{GeoSSL}} =& \frac{1}{2L}\sum_{l=1}^{L}\sigma_l^\beta \mathbb{E}_{p_{\text{data}}(\boldsymbol{d}_1|\boldsymbol{g}_2)}\mathbb{E}_{q(\tilde{\boldsymbol{d}}_1|\boldsymbol{d}_1,\boldsymbol{g}_2)}\Big[\Big\|\frac{s_\theta(\tilde{\boldsymbol{d}}_1, \boldsymbol{g}_2)}{\sigma_l} - \frac{\boldsymbol{d}_1 - \tilde{\boldsymbol{d}}_1}{\sigma_l^2}\Big\|_2^2\Big] \\ &+ \frac{1}{2L}\sum_{l=1}^{L}\sigma_l^\beta \mathbb{E}_{p_{\text{data}}(\boldsymbol{d}_2|\boldsymbol{g}_1)}\mathbb{E}_{q(\tilde{\boldsymbol{d}}_2|\boldsymbol{d}_2,\boldsymbol{g}_1)}\Big[\Big\|\frac{s_\theta(\tilde{\boldsymbol{d}}_2, \boldsymbol{g}_1)}{\sigma_l} - \frac{\boldsymbol{d}_2 - \tilde{\boldsymbol{d}}_2}{\sigma_l^2}\Big\|_2^2\Big].\end{aligned} \tag{16}$$

D.4 Discussions

Using the energy-based model (EBM) to solve MI maximization can open a novel venue, especially for high-structured data like molecular geometry. To solve EBM, existing methods include noise-contrastive estimation (NCE) [22], score matching (SM) [52], etc. To put this under the MI maximization setting, EBM-NCE is essentially a contrastive learning method, where the goal is to align the positive pairs and contrast the negative pairs simultaneously. While EBM-SM or GeoSSL is a generative self-supervised learning (SSL) on distance denoising, and it is especially appealing in the field for geometric data representation learning.

Score matching can be smoothly adopted to 3D geometric setting. Because scores are defined as gradients of the energy function with respect to the atom positions, it can be thought of a form of pseudo-forces. Following this, GeoSSL can be viewed as a pseudo-force matching, which is more natural to the molecular structures. However, further understanding of this requires more domain knowledge in understanding or designing of the energy function. This is beyond the score of this paper, and we would like to leave it for future exploration.

Recently, there have been a certain works [33] proving that 3D geometric information is useful for 2D topology. Here we want to conjecture that the reverse direction is also meaningful: 2D topology can be also useful for 3D representation. This may not seem reasonable from the domain perspective, since 2D topology can be heuristically obtained from the 3D geometry, *i.e.*, all the 2D information is redundant to 3D geometry. However, from the machine learning theory perspective [4, 16], this is still helpful in reducing the sample complexity. From a higher level perspective, we want to explicitly point out that such gap between machine learning and scientific domain has been widely existed, and it would be an interesting direction for further exploration.

E Experiments

In this section, we would like to discuss the experiment details of our work. The main structure is as follows:

- In Appendix E.1, we introduce the computation resources.
- In Appendix E.2, we introduce the pretraining dataset.
- In Appendix E.3, we introduce the pretraining baselines.
- In Appendices E.4 to E.6, we introduce the downstream datasets.
  - Notice that because the performance of QM9 and MD17 is quite stable after fixing the seed (*e.g.*, 42), we we will not run cross-validation. This also follows the main literature [35, 43, 44].
  - Yet, for LBA & LEP, these two datasets are quite small and are very sensitive to the data splitting, so we pick up 5 seeds (12, 22, 32, 42, and 52) and run cross validation on them.
- In Appendix E.7, we list the key hyperparameters for all the pretraining baselines and GeoSSL.
- In Appendix E.8, we show the empirical results using SchNet as the backbone.
- In Appendix E.9, we show the empirical results using PaiNN as the backbone.

E.1 Computational Resources

We have 20 V100 GPU cards for computation at an internal cluster. Each job can be finished within 3-24 hours.

E.2 Pretraining Dataset: Molecule3D

The PubChemQC database is a large-scale database with around 4M molecules with 3D geometries, and it calculates both the ground-state and excited-state 3D geometries using DFT (density functional theory). Due to

the high computational cost, only several thousand molecules can be processed every day, and this dataset takes years of efforts in total. Following this, Molecule3D [63] takes the ground-state geometries and transforms the data formats into a deep learning-friendly way. It also parses essential quantum properties for each molecule, including energies of the highest occupied molecular orbital (HOMO) and the lowest occupied molecular orbital (LUMO), the energy gap between HOMO-LUMO, and the total energy. For our molecular geometry pretraining, we take a subset of 1M molecules with 3D geometries from Molecule3D.

E.3 Pretraining Baselines

Self-Supervised Learning Pretraining Baselines We first consider the four coordinate-MI-unaware SSL methods: (1) *Type Prediction* is to predict the atom type of masked atoms; (2) *Distance Prediction* aims to predict the pairwise distances among atoms; (3) *Angle Prediction* is to predict the angle among triplet atoms, *i.e.*, the bond angle prediction; (4) *3D InfoGraph* adopts the contrastive learning paradigm by taking the node-graph pair from the same molecule geometry as positive and negative otherwise. Next, following the coordinate-aware MI maximization framework introduced in Equation (2), we include two contrastive and one generative SSL baselines. (5) InfoNCE [58] and (6) EBM-NCE [33] are the two widely-used contrastive learning loss functions, where the goal is to simultaneously align the positive views and contrast the negative views. (7) Representation reconstruction (RR) [33] is a generative SSL that is proxy to maximize the MI. It is a more general form of non-contrastive SSL methods like BOYL [20] and SimSiam [8], and the goal is to reconstruct each view from its counterpart in the representation space. Following this, our proposed GeoSSL can be classified as generative SSL, yet it aims at denoising the pairwise distances instead.

Supervised Pretraining Baseline We also compare our method with a supervised pretraining baseline. As aforementioned, the large-scale pretraining dataset uses the DFT to calculate the energy, and extracts the most stable conformers with the lowest energies, which reveal the most fundamental properties of molecules in the 3D Euclidean space. Thus, such energies can be naturally adopted as the supervised signals, and we take this as a supervised pretraining baseline.

E.4 Dataset: QM9

QM9 [40] is a dataset of 134K molecules consisting of 9 heavy atoms. It includes 12 tasks that are related to the quantum properties. For example, U0 and U298 are the internal energies at 0K at 0K and 298.15K respectively, and U298 and G298 are the other two energies that can be transferred from H298 respectively. The other 8 tasks are quantum mechanics related to the DFT process. We follow [43] in preprocessing the dataset (including unit transformation for each task).

Current work is using different data split (in terms of the splitting size). Originally there are 133,885 molecules in QM9, where 3,054 are filtered out, leading to 130,831 molecules. During the benchmark, we find that the performance on QM9 is very robust to either using (1) 110K for training, 10K for val, 10,831 for test or using (2) 100K for training, 13,083 for val and 17,748 for test. In this paper, we are using option (1).

E.5 Dataset: MD17

MD17 [9] is a dataset on molecular dynamics simulation. It includes eight tasks, corresponding to eight organic molecules, and each task includes the molecule positions along the potential energy surface (PES), as shown in Figure 1. The goal is to predict the energy-conserving interatomic forces for each atom at each molecule position. We list some basic statistics in Table 5. We follow [35, 44] in preprocessing the dataset (including unit transformation for each task).

Table 5: Some basic statistics on MD17.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
Train	1K	1K	1K	1K	1K	1K	1K	1K
Validation	1K	1K	1K	1K	1K	1K	1K	1K
Test	209,762	47,863	553,092	991,237	324,250	318,231	440,790	131,770

E.6 Dataset: LBA & LEP

Atom3D [56] is a newly published dataset. It gathers several core tasks for 3D molecules, including binding affinity. The binding affinity prediction is to measure the strength of binding interaction between a small molecule to the target protein. Here we will model both the small molecule and large molecule (protein) with their 3D atom coordinates provided.

During the binding process, there is a cavity in a protein that can potentially possess suitable properties for binding a small molecule (ligand), and it is termed a pocking [53]. Because of the large volume of protein, we

Table 6: Some basic statistics on LBA & LEP. For LBA, we use split-by-sequence-identity-30: we split protein-ligand complexes such that no protein in the test dataset has more than 30% sequence identity with any protein in the training dataset. For LEP, we split the complex pairs by protein target.

Pretraining	LBA	LEP
Train	3,507	304
Validation	466	110
Test	490	104
Split	split-by-identity-30	split-by-target

follow [56] by only taking the binding pocket, where there are no more than 600 atoms for each molecule and protein pair. To be more concrete, we consider two binding affinity tasks. (1) The first task is ligand binding affinity (LBA). It is gathered from [60] and the task is to predict the binding affinity strength between a small molecule and a protein pocket. (2) The second task is ligand efficacy prediction (LEP). The input is a ligand and both the active and inactive conformers of a protein, and the goal is to predict whether or not the ligand can activate the protein's function. We list some basic statistics in Table 6.

E.7 Hyperparameter Specification

We list all the detailed hyperparameters in this subsection. For all the methods, we use the same optimization strategy, *i.e.*, with learning rate as 5e-4 and cosine annealing learning rate schedule [37]. The other hyperparameters for each pretraining method are listed in Table 7. For the other hyperparameters, we are using the default hyperparameters, as attached in the codes.

Table 7: Hyperparameter specifications.

Pretraining	Hyperparameter	Value
Supervised	task	{total energy}
Type Prediction	masking ratio	{0.15, 0.3}
Distance Prediction	prediction rate	{1}
Angle Prediction	prediction rate	{1e-3, 1e-4}
RR	perturbed noise μ	{0}
	perturbed noise σ	{0.3}
	masking ratio r	{0, 0.3}
InfoNCE	perturbed noise μ	{0}
	perturbed noise σ	{0.3, 1}
	masking ratio r	{0, 0.3}
EBM-NCE	perturbed noise μ	{0}
	perturbed noise σ	{0.3, 1}
	masking ratio r	{0, 0.3}
GeoSSL	perturbed noise μ	{0}
	perturbed noise σ	{0.3}
	masking ratio r	{0, 0.3}
	L	{30, 50}
	σ_1	{0.01}
	σ_L	{10}
	annealing factor β	{0.05, 0.2, 2, 5, 10}

E.8 SchNet as Backbone Model

We want to highlight that some backbone models (*e.g.*, DimeNet++ and SphereNet) may perform better or on par with the PaiNN, as shown in Table 3. Yet they will be out of GPU memory. Thus, considering all (including the model performance, computation efficiency, and memory cost) together, we adopt PaiNN as the backbone model in the main paper.

In this section, we carry out experiments using SchNet as the backbone model. We follow the same process as in Section 3, *i.e.*, we compare our method with one randomly-initialized and seven pretraining baselines. The results on QM9, MD17, LBA and LEP are in Tables 8 to 10 accordingly. From these three tables, we can observe that in general, GeoSSL can reach the most optimal results, yielding 21 best performance in 22 downstream tasks, and can reach comparative performance on the remaining task (within top 2 model). This can largely support the effectiveness of our proposed method, GeoSSL. In addition, we also want to mention that a lot of pretraining tasks show the negative transfer issue. Comparing to the results in Section 3, we conjecture that this is related to the task (both pretraining and downstream tasks) and the backbone model. Yet, this is beyond the scope of our work, and we would like to leave this as a future direction.

Table 8: Downstream results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
–	0.070	50.59	32.53	26.33	0.029	0.032	14.68	14.85	0.122	14.70	14.44	1.698
Supervised	0.070	51.34	32.62	27.61	0.030	0.032	14.08	14.09	0.141	14.13	13.25	1.727
Type Prediction	0.084	56.07	34.55	30.65	0.040	0.034	18.79	19.39	0.201	19.29	18.86	2.001
Distance Prediction	0.068	49.34	31.18	25.52	0.029	0.032	13.93	13.59	0.122	13.64	13.18	1.676
Angle Prediction	0.084	57.01	37.51	30.92	0.037	0.034	15.81	15.89	0.149	16.41	15.76	1.850
3D InfoGraph	0.076	53.33	33.92	28.55	0.030	0.032	15.97	16.28	0.117	16.17	15.96	1.666
RR	0.073	52.57	34.44	28.41	0.033	0.038	15.74	16.11	0.194	15.58	14.76	1.804
InfoNCE	0.075	53.00	34.29	27.03	0.029	0.033	15.67	15.53	0.125	15.79	14.94	1.675
EBM-NCE	0.073	52.86	33.74	28.07	0.031	0.032	14.02	13.65	0.121	13.70	13.45	1.677
GeoSSL (ours)	**0.066**	**48.59**	**30.83**	**25.27**	**0.028**	**0.031**	**13.06**	**12.33**	**0.117**	**12.48**	**12.06**	**1.631**

Table 9: Downstream results on 8 force prediction tasks from MD17. We take 1K for training, 1K for validation, and the number of molecules for test are varied among different tasks, ranging from 48K to 991K. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
–	1.196	0.404	0.542	0.879	0.534	0.786	0.562	0.730
Supervised	1.863	0.413	0.512	1.254	0.846	1.005	**0.529**	0.899
Type Prediction	1.293	0.787	0.547	0.879	1.030	1.076	0.614	0.738
Distance Prediction	1.414	0.453	0.845	1.371	0.591	0.819	0.588	0.993
Angle Prediction	3.030	0.450	0.485	0.845	1.112	1.214	0.791	1.016
3D InfoGraph	1.545	0.448	0.640	1.080	0.827	1.096	0.735	0.760
RR	1.878	0.450	0.690	2.255	0.960	1.382	0.784	1.188
InfoNCE	1.286	0.396	0.512	1.007	0.778	1.060	0.667	0.933
EBM-NCE	1.271	0.400	0.570	0.972	0.605	0.862	0.576	0.790
GeoSSL (ours)	**1.176**	**0.368**	**0.434**	**0.779**	**0.460**	**0.700**	0.561	**0.679**

Table 10: Downstream results on 2 binding affinity tasks. We select three evaluation metrics for LBA: the root mean squared error (RMSD), the Pearson correlation (R_p) and the Spearman correlation (R_S). LEP is a binary classification task, and we use the area under the curve for receiver operating characteristics (ROC) and precision-recall (PR) for evaluation. We run cross validation with 5 seeds, and the best results are in **bold**.

Pretraining	LBA			LEP	
	RMSD ↓	R_P ↑	R_C ↑	ROC ↑	PR ↑
–	1.489 ± 0.02	0.522 ± 0.01	0.501 ± 0.01	0.436 ± 0.03	0.369 ± 0.02
Supervised	1.477 ± 0.04	0.528 ± 0.02	0.503 ± 0.03	0.462 ± 0.05	0.392 ± 0.03
Type Prediction	1.483 ± 0.04	0.498 ± 0.03	0.481 ± 0.03	0.570 ± 0.04	0.509 ± 0.07
Distance Prediction	1.461 ± 0.06	0.535 ± 0.04	0.512 ± 0.04	0.502 ± 0.06	0.415 ± 0.05
Angle Prediction	1.499 ± 0.01	0.475 ± 0.01	0.462 ± 0.02	0.532 ± 0.06	0.449 ± 0.03
3D InfoGraph	1.467 ± 0.06	0.526 ± 0.03	0.500 ± 0.03	0.515 ± 0.05	0.412 ± 0.04
RR	–	–	–	0.439 ± 0.04	0.365 ± 0.02
InfoNCE	1.528 ± 0.05	0.483 ± 0.02	0.464 ± 0.02	0.588 ± 0.06	0.523 ± 0.05
EBM-NCE	1.499 ± 0.03	0.509 ± 0.02	0.498 ± 0.02	0.493 ± 0.07	0.429 ± 0.06
GeoSSL (ours)	**1.432 ± 0.02**	**0.550 ± 0.02**	**0.529 ± 0.02**	**0.633 ± 0.03**	**0.541 ± 0.03**

E.9 PaiNN as Backbone Model

Due to the page limit, we illustrate the results on of ligand binding affinity (LBA) and ligand efficacy prediction (LEP) on PaiNN here Table 11.

Table 11: Downstream results on 2 binding affinity tasks. We select three evaluation metrics for LBA: the root mean squared error (RMSD), the Pearson correlation (R_p) and the Spearman correlation (R_S). LEP is a binary classification task, and we use the area under the curve for receiver operating characteristics (ROC) and precision-recall (PR) for evaluation. We run cross validation with 5 seeds, and the best results are in **bold**.

Pretraining	LBA			LEP	
	RMSD ↓	R_P ↑	R_C ↑	ROC ↑	PR ↑
–	1.463 ± 0.06	0.572 ± 0.02	0.568 ± 0.02	0.675 ± 0.04	0.549 ± 0.05
Supervised	1.551 ± 0.08	0.539 ± 0.03	0.533 ± 0.03	0.696 ± 0.03	0.554 ± 0.03
Charge Prediction	2.316 ± 0.80	0.387 ± 0.11	0.400 ± 0.11	0.630 ± 0.05	0.557 ± 0.07
Distance Prediction	1.542 ± 0.08	0.545 ± 0.03	0.540 ± 0.03	0.521 ± 0.07	0.479 ± 0.07
Angle Prediction	–	–	–	0.545 ± 0.07	0.504 ± 0.07
3D InfoGraph	–	–	–	0.540 ± 0.03	0.469 ± 0.03
RR	1.515 ± 0.07	0.545 ± 0.03	0.539 ± 0.03	0.654 ± 0.05	0.518 ± 0.06
InfoNCE	1.564 ± 0.05	0.508 ± 0.03	0.497 ± 0.05	0.693 ± 0.06	0.571 ± 0.08
EBM-NCE	1.499 ± 0.06	0.547 ± 0.03	0.534 ± 0.03	0.691 ± 0.05	0.603 ± 0.07
GeoSSL (ours)	**1.451 ± 0.03**	**0.577 ± 0.02**	**0.572 ± 0.01**	**0.776 ± 0.03**	**0.694 ± 0.06**