# Physical Consistency Bridges Heterogeneous Data in Molecular Multi-Task Learning

**Yuxuan Ren**[*†], **Dihan Zheng**[*†], **Chang Liu**[‡], **Peiran Jin, Yu Shi, Lin Huang, Jiyan He**[†], **Shengjie Luo**[†], **Tao Qin, Tie-Yan Liu**

Microsoft Research AI for Science

## Abstract

In recent years, machine learning has demonstrated impressive capability in handling molecular science tasks. To support various molecular properties at scale, machine learning models are trained in the multi-task learning paradigm. Nevertheless, data of different molecular properties are often not aligned: some quantities, *e.g.* equilibrium structure, demand more cost to compute than others, *e.g.* energy, so their data are often generated by cheaper computational methods at the cost of lower accuracy, which cannot be directly overcome through multi-task learning. Moreover, it is not straightforward to leverage abundant data of other tasks to benefit a particular task. To handle such data heterogeneity challenges, we exploit the specialty of molecular tasks that there are physical laws connecting them, and design consistency training approaches that allow different tasks to exchange information directly so as to improve one another. Particularly, we demonstrate that the more accurate energy data can improve the accuracy of structure prediction. We also find that consistency training can directly leverage force and off-equilibrium structure data to improve structure prediction, demonstrating a broad capability for integrating heterogeneous data.

## 1 Introduction

The field of machine learning has witnessed a blossom of progress in solving molecular science tasks in recent years, including molecular property prediction [1, 2, 3], machine-learning force field (energy/force prediction) [4, 5, 6, 7, 8], electronic structure [9, 10, 11, 12, 13], molecular structure generation [14, 15, 16, 17] and design [18, 19, 20]. In molecular research, these tasks are often required jointly: for example, energy prediction is required for molecular stability and dynamics, and equilibrium structure (conformation) prediction offers the most probable and characteristic structure for understanding molecule interaction and functions. Multi-task learning is hence adopted, where a model is trained to predict multiple properties using the same number of decoders (output heads) built on a shared encoder (backbone model) (Fig. 1) [21]. This paradigm is also used for pre-training a model by leveraging as much data as possible that are scattered over various tasks and domains [22, 23, 24].

Nevertheless, science tasks have some unique challenges beyond conventional machine learning tasks, which cannot be adequately addressed by multi-task learning alone. It is more costly to curate a dataset for molecular science tasks since it calls for running physics-theoretic computation algorithms, which come with a stringent accuracy-efficiency trade-off. Molecular-science datasets are hence generated each with a specific algorithmic portfolio that is economic for the particular purpose.

[*]Equal contribution.

[†]Work done during an internship at Microsoft Research AI for Science.

[‡]Correspondence to: <changliu@microsoft.com>

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



https://doi.org/10.52202/079017-2330

This incurs two challenges regarding data heterogeneity. **(1)** Different properties in a dataset may come from algorithms in different levels of theory, meaning different levels of accuracy. This limits the prediction accuracy on some tasks. For example, labeling the energy of a molecular structure is yet affordable for common molecules using algorithms in the density functional theory (DFT) level, but producing the equilibrium structure of a molecule requires repeated energy evaluations hence is tens to hundreds of times more costly. As a result, DFT-level equilibrium structure data are available only in a limited scale [25, 26, 27], while larger-scale datasets [28, 29] have to resort to lower levels of theory to generate equilibrium structures at scale, which come with a lower level of accuracy. Using such data, multi-task learning alone cannot predict structures in an accuracy higher than the data-generation method. **(2)** Different datasets focus on different tasks, so combining these datasets to enhance the performance on a particular task is not straightforward. For example, there are datasets [30, 31, 32] that are concerned with force prediction and off-equilibrium structures, which do not provide direct supervision to equilibrium structure prediction. Although including these additional tasks in multi-task learning could help learn a better encoder (or, representation), this is only based on an empirical observation from a general machine learning perspective and does not directly exploit relevant physical information in the additional tasks.

In this work, we highlight that science tasks also provide fortunate specialties that come to the rescue. In contrast to conventional machine learning tasks which are primarily defined by data, science tasks originate in fundamental physical laws, and data are rather the demonstration of such laws. These laws impose explicit constraints between tasks, hence define the "physical consistency" between model predictions on these tasks. By enforcing such consistency, model predictions for different tasks are connected and can explicitly share the information in the data of one task to the prediction for other tasks, hence bridging data heterogeneity. From another perspective, while sharing a common encoder connects various decoders at the input end, physical consistency closes the loop by connecting the decoders at the output end (Fig. 1). With the additional information from the physical consistency, capabilities beyond conventional multi-task learning are enabled: **(1)** data from a higher-level of theory of one task can improve the accuracy of a physically related task, and **(2)** the abundant data of a physically related task can directly improve the performance of the concerned task.

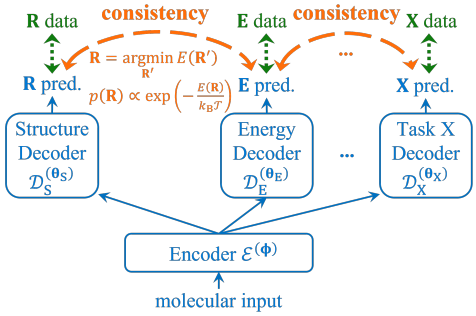


Figure 1: Illustration of the idea of physical consistency. To support multiple tasks ("Task X" represents a general task), the model (blue solid lines) builds multiple decoders on a shared encoder, which are trained by multi-task learning with data of respective tasks (green dotted double arrows). Physical consistency losses enforce physical laws between tasks (orange dashed double arrows), hence bridge data heterogeneity and directly improve one task from others.

Concretely, we demonstrate the practical value of the physical consistency between energy prediction and equilibrium structure prediction, two central tasks in molecular science. The consistency can be constructed from two perspectives: the equilibrium structure of a molecule is the structure that attains the minimal energy of the molecule (Sec. 3.2), and the equilibrium structure is a sample of the thermodynamic equilibrium distribution at a low temperature (Sec. 3.3). When adopting the denoising diffusion formulation [33, 34, 35] for structure prediction, we show that the two physical laws can be translated into consistency loss functions which connect the energy-prediction model with the structure-prediction model at large and small diffusion steps, respectively. We apply the consistency losses in the multi-task learning on the PubChemQC B3LYP/6-31G*//PM6 dataset [29], which is perhaps the largest public dataset with DFT-level (B3LYP/6-31G*) energy labels thus highly relevant to model pre-training. But its equilibrium structures are generated in the semi-empirical level (PM6), which is in a lower level of accuracy. We use the consistency losses to transmit the information of the higher-level theory in the energy model to the structure model by only optimizing parameters of the structure decoder, which achieves the followings. **(1)** Consistency losses improve structure prediction accuracy beyond multi-task learning alone, when evaluated using DFT-level structures from the PCQM4Mv2 [27] and QM9 [25] datasets. The advantage persists even after both have undergone finetuning. **(2)** Datasets providing DFT-level force labels on off-equilibrium structures are better leveraged to improve structure prediction accuracy beyond only including force prediction in multi-task learning. Since force is the gradient of the energy, the additional data allow

https://doi.org/10.52202/079017-2330

the energy model to learn a better energy landscape, which in turn leads to more accurate structure prediction through the consistency losses. This is a more direct pathway for data on an additional task to benefit structure prediction beyond learning a better representation in multi-task learning. We remark that the improvement is not from any more accurate structure data, but from bridging data heterogeneity using the "free lunch" redeemed from physical laws.

### 1.1 Related Work

**Structure prediction.** In recent years, deep generative models have been used as a powerful tool to generate molecular structures. Due to the subtlety that a structure being rotated as a whole is essentially the same structure, early methods opt to generate intermediate geometric variables, such as inter-atomic distances [36, 37] or torsional angles [38, 39]. Directly generating Cartesian coordinates of atoms has also been explored where the rotational equivalence is handled by alignment [40] or leveraging gradient of distances [41, 42]. More recently, diffusion models have been used to generate torsional angles [43], or atom coordinates [16, 44] using equivariant models. Given the generality and superior performance, we adopt an equivariant diffusion model for structure prediction. While these works may generate multiple meta-stable structures, we aim at the single equilibrium structure for each molecule, and investigate the benefit from an energy model.

**Leveraging physical laws between tasks.** A noticeable example is leveraging the connection between energy and thermodynamic equilibrium distribution to compensate for potentially biased data to better learn the distribution. The equilibrium distribution in a canonical ensemble is the Boltzmann distribution, which is directly determined by the energy function. From the machine learning perspective, the energy function provides an unnormalized density function of the target distribution. Using a flow-based model [45], Boltzmann generators [14] inject the energy supervision via the evidence lower bound objective. Bootstrapped $\alpha$-divergence objective is thereafter introduced to mitigate mode collapse [46]. Zheng et al. [17] use a diffusion model where the energy supervises the score model at the start diffusion step and is propagated to intermediate steps by enforcing a PDE. Similar techniques are also explored for conventional machine learning tasks [47, 48]. More recently, Bose et al. [49] directly connect the energy to intermediate-step scores. In the opposite direction, Arts et al. [50] leverage Boltzmann-distribution samples to learn a coarse-grained energy model. In parallel with these works, the current work is devoted to the investigation of leveraging the connection between energy and structure prediction, and is not using an oracle energy function considering the cost of DFT calculation.

## 2 Technical Background

Before delving into details of the consistency between energy and structure prediction, we first introduce the problem formulation, and diffusion-based generative formulation for structure prediction.

Chemically, a molecule is specified by the types (*i.e.*, chemical elements) of atoms and bonds, jointly forming a molecular graph $\mathcal{G}$. A molecule in physical reality can take different structures (conformations) $\mathbf{R} \in \mathbb{R}^{A\times 3}$, *i.e.*, the collection of 3-dimensional coordinates of its $A$ atoms. Many properties of molecule $\mathcal{G}$ depend on its specific structure $\mathbf{R}$, *e.g.*, the (inter-atomic potential) energy, so energy prediction is in the form $E_{\mathcal{G}}^{(\theta)}(\mathbf{R})$ with model parameters $\theta$.

Among possible structures, the equilibrium structure is the one that attains the minimal energy, and is the most representative structure for the molecule. As mentioned, the diffusion formulation has been a preferred choice for equilibrium structure prediction, which samples from a distribution $p_{\mathcal{G}}(\mathbf{R})$ concentrated at the equilibrium structure. For this, a primitive structure is sampled from a simple distribution, *e.g.*, the standard Gaussian, which undergoes a diffusion process that transforms the simple distribution to the desired distribution. This is done by reversing the process in the opposite direction, which is easier to construct. For example, the process can be taken as the Langevin diffusion that converges to standard Gaussian [35]: $\mathrm{d}\mathbf{R}_t = \beta_t \nabla \log \mathcal{N}(\mathbf{R}_t; \mathbf{0}, \mathbf{I})\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\mathbf{W}_t = -\frac{\beta_t}{2}\mathbf{R}_t\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\mathbf{W}_t$, where $\beta_t$ is a time dilation factor [51], and $\mathbf{W}_t$ denotes the Wiener process. The process starts from the desired distribution $p_{\mathcal{G},0} = p_{\mathcal{G}}$ at $t = 0$ and ends after a sufficiently long period $T$ when the distribution converges, $p_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse process is known to

 https://doi.org/10.52202/079017-2330

follow [52]:

$$\mathrm{d}\mathbf{R}_{\bar{t}} = \frac{\beta_{T-\bar{t}}}{2}\mathbf{R}_{\bar{t}}\,\mathrm{d}\bar{t} + \beta_{T-\bar{t}}\nabla \log p_{\mathcal{G},T-\bar{t}}(\mathbf{R}_{\bar{t}})\,\mathrm{d}\bar{t} + \sqrt{\beta_{T-\bar{t}}}\,\mathrm{d}\mathbf{W}_{\bar{t}}, \tag{1}$$

where $\bar{t} := T - t$, which transforms $\mathcal{N}(\mathbf{0}, \mathbf{I})$ at $\bar{t} = 0$ to the desired distribution at $\bar{t} = T$, hence the generation process is constructed. To simulate the process, the only unknown is the (Fisher's) score function $\nabla \log p_{\mathcal{G},t}(\mathbf{R})$ at each diffusion instant, for which a machine-learning model $\mathbf{s}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\mathbf{R})$ is introduced. To learn to fit $\nabla \log p_{\mathcal{G},t}(\mathbf{R})$, a practical approach is by optimizing the denoising score matching loss [53]: $\mathbb{E}_{p_{\mathcal{G},0}(\mathbf{R}_0)}\mathbb{E}_{p(\mathbf{R}_t|\mathbf{R}_0)}\|\mathbf{s}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\mathbf{R}_t) - \nabla_{\mathbf{R}_t}\log p(\mathbf{R}_t|\mathbf{R}_0)\|^2$, which is convenient since from the Langevin diffusion, we can derive $p(\mathbf{R}_t|\mathbf{R}_0) = \mathcal{N}(\mathbf{R}_t; \sqrt{\bar{\alpha}_t}\mathbf{R}_t, (1-\bar{\alpha}_t)\mathbf{I})$, where $\bar{\alpha}_t := \exp(-\int_0^t \beta_s\,\mathrm{d}s)$, which is a known distribution. Leveraging the reparameterization trick, the loss is reformed as [35]:

$$\mathbb{E}_{\mathrm{Unif}(t;0,T)}(1-\bar{\alpha}_t)\mathbb{E}_{p_{\mathcal{G},0}(\mathbf{R}_0)}\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_t;\mathbf{0},\mathbf{I})}\left\|\mathbf{s}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\sqrt{\bar{\alpha}_t}\mathbf{R}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t) + \frac{\boldsymbol{\epsilon}_t}{\sqrt{1-\bar{\alpha}_t}}\right\|_2^2, \tag{2}$$

where the expectation w.r.t $\mathbb{E}_{p_{\mathcal{G},0}(\mathbf{R}_0)}$ can be estimated by averaging over data. Once the score model is trained, it can generate structures by replacing $\nabla \log p_{\mathcal{G},t}(\mathbf{R})$ and simulating Eq. (1). If only $p_{\mathcal{G},0}(\mathbf{R}_0)$ is desired (instead of $p_{\mathcal{G}}(\mathbf{R}_{0:T})$), then an equivalent simulation approach can be adopted known as probability-flow ODE [35]:

$$\mathrm{d}\mathbf{R}_{\bar{t}} = \frac{\beta_{T-\bar{t}}}{2}\big(\mathbf{R}_{\bar{t}} + \nabla \log p_{\mathcal{G},T-\bar{t}}(\mathbf{R}_{\bar{t}})\big)\,\mathrm{d}\bar{t}, \tag{3}$$

which is equivalent to the deterministic process in denoising diffusion implicit model (DDIM) [54].

An alternative to the score-prediction formulation is the denoising formulation [55, 56]. By defining a "denoising model" $\mathbf{S}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\mathbf{R}_t)$ which formulates the score model following:

$$\mathbf{s}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\mathbf{R}_t) = \frac{\sqrt{\bar{\alpha}_t}\mathbf{S}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\mathbf{R}_t) - \mathbf{R}_t}{1-\bar{\alpha}_t}, \tag{4}$$

the training loss function Eq. (2) in terms of $\mathbf{S}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\mathbf{R}_t)$ becomes:

$$\mathbb{E}_t\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\mathbb{E}_{p_{\mathcal{G},0}(\mathbf{R}_0)}\mathbb{E}_{\boldsymbol{\epsilon}_t}\left\|\mathbf{S}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\sqrt{\bar{\alpha}_t}\mathbf{R}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t) - \mathbf{R}_0\right\|_2^2, \tag{5}$$

which follows the intuition to denoise a perturbed structure by predicting the original structure. This formulation better aligns with the notion of "structure prediction", hence can be benefited from successful model architectures [18, 19], and matches structure pre-training strategies [57, 24, 22, 58].

## 3 Method

We begin with the basic formulation of multi-task learning. We then present the two consistency training approaches between energy and structure prediction, based on two physical laws between the two tasks. The approach to directly leveraging physically-related datasets is described at last.

### 3.1 Multi-Task Learning for Energy and Structure Prediction

Both energy and structure prediction tasks require a comprehensive understanding of the input molecular graph $\mathcal{G}$ and structure $\mathbf{R}$, so a shared encoder $\mathcal{E}^{(\boldsymbol{\Phi})}_{\mathcal{G},t}(\mathbf{R})$ with parameters $\boldsymbol{\Phi}$ is employed. The time step $t$ is required by the diffusion formulation, which is taken as 0 for energy prediction indicating the input structure is unperturbed and real. For energy and structure prediction, the corresponding decoders $\mathcal{D}^{(\boldsymbol{\theta}_{\mathrm{E}})}_{\mathrm{E}}$ and $\mathcal{D}^{(\boldsymbol{\theta}_{\mathrm{S}})}_{\mathrm{S}}$ are introduced. For structure prediction, the denoising formulation is adopted (end of Sec. 2). Under this formulation, the two tasks are handled by:

$$E^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{E}})}_{\mathcal{G}}(\mathbf{R}) = \mathcal{D}^{(\boldsymbol{\theta}_{\mathrm{E}})}_{\mathrm{E}}\big(\mathcal{E}^{(\boldsymbol{\Phi})}_{\mathcal{G},t=0}(\mathbf{R})\big), \quad \mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\mathcal{G},t}(\mathbf{R}) = \mathcal{D}^{(\boldsymbol{\theta}_{\mathrm{S}})}_{\mathrm{S}}\big(\mathcal{E}^{(\boldsymbol{\Phi})}_{\mathcal{G},t}(\mathbf{R}), \mathbf{R}\big). \tag{6}$$

On one datapoint $(\mathcal{G}, \mathbf{R}, E)$, the multi-task loss is (*c.f.* Eq. (5)): $L_{\text{multi-task}}(\boldsymbol{\Phi}, \boldsymbol{\theta}_{\mathrm{E}}, \boldsymbol{\theta}_{\mathrm{S}}|\mathcal{G}, \mathbf{R}, E)$

$$= \lambda_{\mathrm{E}}\left|E^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{E}})}_{\mathcal{G}}(\mathbf{R}) - E\right| + \mathbb{E}_t\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\mathbb{E}_{\boldsymbol{\epsilon}_t}\left\|\mathbf{S}^{(\boldsymbol{\theta})}_{\mathcal{G},t}(\sqrt{\bar{\alpha}_t}\mathbf{R} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t) - \mathbf{R}\right\|_2^2. \tag{7}$$

A subtlety with the models is geometric invariance and equivariance. Indeed, if a structure $\mathbf{R}$ is translated and rotated as a whole, the resulting atom coordinates represent essentially the same structure. The energy and the probability density should keep invariant after the transformation. For energy prediction, this can be guaranteed by using an invariant encoder $\mathcal{E}^{(\boldsymbol{\Phi})}_{\mathcal{G},t}(\mathbf{R})$ (no requirement on $\mathcal{D}^{(\boldsymbol{\theta}_{\mathrm{E}})}_{\mathrm{E}}$). For the probability density, it is known [59] that rotational invariance can be achieved by the invariance of $p_{\mathcal{G},T}$, which is satisfied by $\mathcal{N}(\mathbf{0},\mathbf{I})$, and the equivariance of the denoising model $\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\mathcal{G},t}(\mathbf{R})$. For this reason, the input structure $\mathbf{R}$ re-enters the structure decoder $\mathcal{D}^{(\boldsymbol{\theta}_{\mathrm{S}})}_{\mathrm{S}}$, which is implemented with an equivariant architecture. Translational invariance of density can be achieved by centering the structures; see ref. [60] for reasoning.

Nevertheless, multi-task learning is restricted by the level of accuracy of training data. As mentioned, structure data are often generated in a lower level of accuracy than energy data due to the more demanding nature. To alleviate this limitation, we exploit physical laws between molecular energy and structure, and propose consistency training losses accordingly to bridge the energy and structure models, thereby enhancing the accuracy of structure prediction from the more accurate energy model.

### 3.2 Optimality Consistency

One direct relationship between energy and equilibrium structure is that the equilibrium structure $\mathbf{R}_{\text{eq}}$ minimizes the energy, *i.e.*, $\mathbf{R}_{\text{eq}} = \operatorname{argmin}_{\mathbf{R}} E_{\mathcal{G}}(\mathbf{R})$. To enforce this optimality condition, we propose an optimality consistency loss $L_{\text{optim-cons}}$, in the form of "increase after perturbation" loss. It is based on the idea that the energy of the equilibrium structure should be lower than that of its perturbed version. Denoting $\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\text{pred}}$ as the model-predicted equilibrium structure, the loss on a molecule $\mathcal{G}$ can be written as:

$$L_{\text{optim-cons}}(\boldsymbol{\theta}_{\mathrm{S}} \mid \boldsymbol{\Phi}, \boldsymbol{\theta}_{\mathrm{E}}, \mathcal{G}) = \mathbb{E}_{\boldsymbol{\eta}} \max\left\{0,\ E^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{E}})}_{\mathcal{G}}(\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\text{pred}}) - E^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{E}})}_{\mathcal{G}}(\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\text{pred}} + \boldsymbol{\eta})\right\}, \quad (8)$$

where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathrm{Diag}(\boldsymbol{\sigma}^2))$ is a small perturbation, and each element of $\boldsymbol{\sigma}^2$ is independently sampled from $\mathrm{Unif}(0, \sigma^2_{\text{max}}]$. Since the purpose is to improve structure prediction accuracy by leveraging the energy model which has seen more accurate labels, so the consistency loss only optimizes the parameters exclusively for the structure prediction utility, *i.e.*, structure decoder parameters $\boldsymbol{\theta}_{\mathrm{S}}$. Other parameters $\boldsymbol{\Phi}$ and $\boldsymbol{\theta}_{\mathrm{E}}$ do not optimize this loss. In this way, the consistency loss would not contaminate the energy prediction model with the less accurate structure prediction model.

The standard way to produce $\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\text{pred}}$ requires simulating the diffusion process (Eq. (1)) or the equivalent ODE (DDIM) (Eq. (3)), which calls the denoising model recursively. So optimizing $L_{\text{optim-cons}}$ would involve backpropagation through the simulation process, which can be impractically costly and numerically unstable. Fortunately, we can exploit the intuition in the denoising formulation and find a much cheaper way to generate structure. The intuition of the denoising model $\mathbf{S}_{\mathcal{G},t}(\mathbf{R}_t)$ is to recover the original structure $\mathbf{R}_0$ from the perturbed structure $\mathbf{R}_t$. Although this is informationally impossible at the instance level, the denoising model still has a definite learning target at the distributional level, which is $\mathbb{E}[\mathbf{R}_0|\mathbf{R}_t]$ [56]. Particularly, when $t = T$, the correlation between $\mathbf{R}_0$ and $\mathbf{R}_T$ diminishes, so $\mathbf{S}_{\mathcal{G},T}(\mathbf{R}_T)$ learns to output $\mathbb{E}[\mathbf{R}_0]$, the expectation of the target distribution, which is the equilibrium structure since the distribution concentrates at that structure. Under this perspective, the model-predicted structure can be generated by $\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\text{pred}} = \mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\mathcal{G},T}(\mathbf{R}_T)$, where $\mathbf{R}_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. This only requires one evaluation of the denoising model.

Nevertheless, the rotational invariance of the structure distribution introduces more subtleties.

**Proposition 1.** *Let $\mathbf{S}^{(\boldsymbol{\theta})} : \mathbb{R}^{A\times 3} \to \mathbb{R}^{A\times 3}$ be a rotationally equivariant function; that is, for any rotation matrix $\mathbf{Q} \in \mathrm{SO}(3)$ and structure $\mathbf{R} \in \mathbb{R}^{A\times 3}$, we have $\mathbf{S}^{(\boldsymbol{\theta})}(\mathbf{R}\mathbf{Q}) = \mathbf{S}^{(\boldsymbol{\theta})}(\mathbf{R})\mathbf{Q}$. Then, for any target structure $\mathbf{R}^{\star} \in \mathbb{R}^{A\times 3}$, the minimizer of the denoising loss function $L(\boldsymbol{\theta}) = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\|\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon}) - \mathbf{R}^{\star}\|_2^2$ is the zero map; that is, $\mathbf{S}^{(\boldsymbol{\theta})}(\mathbf{R}) = \mathbf{0}$, for any $\mathbf{R}$.*

See Appendix A for proof. This conclusion reveals that the learning target of the denoising model at $T$ is trivially all-zero, which cannot serve to generate the equilibrium structure. To circumvent this, a simple choice is to denoise from a time step $\tau$ that is close but smaller than $T$:

$$\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\text{pred}} = \mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\mathrm{S}})}_{\mathcal{G},\tau}(\boldsymbol{\epsilon}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I}). \quad \text{(for large } \tau) \quad (9)$$


https://doi.org/10.52202/079017-2330

The target of the denoising model at $\tau$ is $\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},\tau}(\boldsymbol{\epsilon}) = \mathbb{E}[\mathbf{R}_0|\mathbf{R}_\tau = \boldsymbol{\epsilon}]$, which is equivariant w.r.t $\mathbf{R}_\tau$. So the input $\mathbf{R}_\tau$ provides an orientation information hence breaking the rotational symmetry of the corresponding distribution $p(\mathbf{R}_0|\mathbf{R}_\tau)$. The resulting expectation then would not average a structure over orientations evenly, hence not zero. More explicitly, since $p(\mathbf{R}_0|\mathbf{R}_\tau) \propto p(\mathbf{R}_0)p(\mathbf{R}_\tau|\mathbf{R}_0) \propto p(\mathbf{R}_0)\exp\left\{-\frac{\|\mathbf{R}_0-\mathbf{R}_\tau/\sqrt{\bar{\alpha}_\tau}\|_2^2}{2(1/\bar{\alpha}_\tau-1)}\right\}$, we have $p(\mathbf{R}_0|\mathbf{R}_\tau) \propto p(\mathbf{R}_0)\mathcal{N}(\mathbf{R}_0;\mathbf{R}_\tau/\sqrt{\bar{\alpha}_\tau},(1/\bar{\alpha}_\tau-1)\mathbf{I})$, where the Gaussian factor assigns larger probability along the direction of $\mathbf{R}_\tau$, hence breaks the rotational symmetry from $p(\mathbf{R}_0)$. Under this choice, the optimality consistency loss in Eq. (8) is specified as: $L_{\text{optim-cons}}(\boldsymbol{\theta}_\mathrm{S} \mid \boldsymbol{\Phi}, \boldsymbol{\theta}_\mathrm{E}, \mathcal{G})$

$$= \mathbb{E}_{\boldsymbol{\eta}}\mathbb{E}_{\boldsymbol{\epsilon}} \max\left\{0,\ E^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{E})}_{\mathcal{G}}\big(\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},\tau}(\boldsymbol{\epsilon})\big) - E^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{E})}_{\mathcal{G}}\big(\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},\tau}(\boldsymbol{\epsilon})+\boldsymbol{\eta}\big)\right\}. \quad \text{(for large } \tau\text{)} \tag{10}$$

### 3.3 Score Consistency

The optimality consistency loss only supervises the denoising model at large time steps. For small time steps, an alternative perspective on the physical law between equilibrium structure and energy can help. Physically, a molecule $\mathcal{G}$ in a real system can take different structures with different probabilities. When the system is in thermodynamic equilibrium, the probability distribution of the structures can be determined from the energy function of the molecule. Particularly, in a system with fixed volume and temperature $\mathcal{T}$, the structure distribution is the well-known Boltzmann distribution, $p_{\mathrm{B};\mathcal{G},\mathcal{T}}(\mathbf{R}) \propto \exp\left(-E_{\mathcal{G}}(\mathbf{R})/(k_\mathrm{B}\mathcal{T})\right)$, where $k_\mathrm{B}$ is the Boltzmann constant. When temperature approaches zero, the distribution becomes concentrated on the equilibrium structure. This aligns with the learning target of the diffusion model for equilibrium structure prediction. Through the expression of the Boltzmann distribution, the structure model can thus be connected to the energy model.

To enforce this connection, ideally, the density function $p^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G}}(\mathbf{R})$ modeled by the denoising model should match that defined by the energy model. Since the latter only provides an unnormalized density, we enforce their scores to match: $\mathbb{E}_{q(\mathbf{R})}\left\|\nabla \log p^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G}}(\mathbf{R}) + \nabla E^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{E})}_{\mathcal{G}}(\mathbf{R})/(k_\mathrm{B}\mathcal{T})\right\|_2^2$, where $q(\mathbf{R})$ is a reference distribution. Nevertheless, it is computationally costly to evaluate the density function from the diffusion model: $\log p^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G}}(\mathbf{R}) = \log p_T(\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_T) + \int_0^T \frac{\beta_t}{2}\frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\left(3A\sqrt{\bar{\alpha}_t} - \nabla\cdot\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},t}(\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_t)\right)\mathrm{d}t$, where $\mathbf{R}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{t\in[0,T]}$ is the solution to the ODE $\frac{\mathrm{d}\mathbf{R}_t}{\mathrm{d}t} = \frac{\beta_t}{2}\frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\left(\sqrt{\bar{\alpha}_t}\mathbf{R}_t - \mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},t}(\mathbf{R}_t)\right)$ with initial condition $\mathbf{R}_0 = \mathbf{R}$ [35]. Significant computational cost would be incurred from invoking and backpropagating through an ODE solver to evaluate and optimize the density.

We hence turn to another way to leverage this connection. Note from Eq. (4), the denoising model can be used to recover the score model, which targets the score function of the marginal distribution at the corresponding diffusion time instant. Particularly, the score model at $t = 0$ should approximate the score of the desired distribution, which is $p_{\mathrm{B};\mathcal{G},\mathcal{T}}$ for small $\mathcal{T}$. The energy model can hence provide supervision to the score model by enforcing this connection: $L_{\text{score-cons}}(\boldsymbol{\theta}_\mathrm{S} \mid \boldsymbol{\Phi}, \boldsymbol{\theta}_\mathrm{E}, \mathcal{G})$

$$= \mathbb{E}_{p_\tau(\mathbf{R}_\tau)}\left\|\frac{\sqrt{\bar{\alpha}_\tau}\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},\tau}(\mathbf{R}_\tau) - \mathbf{R}_\tau}{1-\bar{\alpha}_\tau} + \frac{\nabla_{\mathbf{R}_\tau}\mathbf{E}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{E})}_{\mathcal{G}}(\mathbf{R}_\tau)}{k_\mathrm{B}\mathcal{T}}\right\|_2^2. \quad \text{(for small } \tau\text{)} \tag{11}$$

In this score consistency loss, we have avoided taking the time step $\tau$ exactly zero for numerical stability consideration. Indeed, when $\tau \to 0$, the denoising model $\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},\tau}$ approaches the identity map, and $\bar{\alpha}_\tau$ approaches 1, making the first term in Eq. (11) an indeterminate form of type 0/0, which may render numerical stability issues. For the reference distribution to generate data to evaluate the loss, one can choose either the data distribution $p_{\mathcal{G},0}$ for which the energy model gives more confident results, or the perturbed distribution $p_{\mathcal{G},\tau}$ (can be sampled by adding noise to a data sample $\mathbf{R}_0$ following $p(\mathbf{R}_t|\mathbf{R}_0)$) for relevance to how the denoising model $\mathbf{S}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_\mathrm{S})}_{\mathcal{G},\tau}$ is invoked. Through some trials, we found the latter gives slightly better results.

Finally, as is the case for the optimality consistency loss, the score consistency loss also only optimizes the parameters $\boldsymbol{\theta}_\mathrm{S}$ of the structure decoder, to ensure the energy model would not be misled by the less accurate structure data. To implement this unconventional optimization requirement, we list detailed algorithms for the two consistency losses in Appendix B.1 in terms of the actual model components $\mathcal{E}^{(\boldsymbol{\Phi})}_{\mathcal{G},t}$, $\mathcal{D}^{(\boldsymbol{\theta}_\mathrm{E})}_{\mathrm{E}}$, and $\mathcal{D}^{(\boldsymbol{\theta}_\mathrm{S})}_{\mathrm{S}}$.

### 3.4 Leveraging Physically-Related Data

Besides the difference in the level of theory to generate data, the heterogeneity of molecular-science datasets also lies in the difference of concerned quantities. Compared to the enormous chemical space, available datasets for equilibrium structure are still not abundant, while there is a vast amount of data generated for physically related but different tasks, for example, labels of atomic forces, and data on off-equilibrium structures. We highlight that the consistency losses can leverage such datasets in an explicit way to further improve equilibrium structure prediction. Note that the consistency losses Eqs. (10, 11) works by offering the information at a higher level of theory in the energy model, in the form of *energy landscape* on the structure space; *i.e.*, ranking different structures in optimality consistency, and providing energy gradient in score consistency. For better learning the landscape, the force labels, which are negative gradients of the energy, provides first-order information of the landscape, and energy and force labels on multiple off-equilibrium structures enable better exploration on the structure space. This approach provides a more direct and concrete information path to equilibrium structure prediction than helping learn a better representation in multi-task learning. For learning a better energy landscape, the force labels are used to directly supervise the gradient of the energy model. The loss term for a datapoint $(\mathcal{G}, \mathbf{R}, \mathbf{F})$ is:

$$L_{\text{force}}(\boldsymbol{\phi}, \theta_{\text{E}} \mid \mathbf{R}, \mathbf{F}) = \left\| \nabla_{\mathbf{R}} E_{\mathcal{G}}^{(\boldsymbol{\phi}, \theta_{\text{E}})}(\mathbf{R}) + \mathbf{F} \right\|_2^2,$$

where $\mathbf{F}$ is the force label. Note that there may be multiple $(\mathbf{R}, \mathbf{F})$ data pairs for one molecule $\mathcal{G}$, which provide even richer information on the energy landscape.

## 4 Experiments

In this section, we demonstrate the advantages of incorporating the proposed consistency losses into multi-task learning. Implementation details are provided in Appendix B.

### 4.1 Setup

**Datasets.** We consider multi-task learning of energy and structure prediction on the PubChemQC B3LYP/6-31G*//PM6 dataset [29] (abbreviated as PM6), which is seemingly the largest ($\sim$86M molecules) public available dataset with DFT-level property labels, hence a preferred setting for pre-training a molecular model. The energy labels are in the DFT (B3LYP/6-31G*) level, while the equilibrium structures are produced at the semi-empirical PM6 [61] level, which is less accurate than DFT. Consistency training is hence considered to improve structure prediction accuracy using the more accurate energy data. To evaluate the effect of improved structure prediction accuracy beyond the PM6 level, the accuracy is evaluated against structures generated at the DFT level, which are available in the PCQM4Mv2 dataset [27] (abbreviated as PCQ) and the QM9 dataset [25].

**Evaluation.** Each of the evaluation datasets of PCQ and QM9 is spilt into three disjoint sets for training, validation, and test. The training and validation sets are for optional fine-tuning (see Sec. 4.4). Following existing convention [41, 44, 40], each test set is prepared by uniformly randomly selecting 200 distinct molecules from PCQ or QM9 that do not appear in the training dataset (PM6), which already makes the test molecules sufficiently dissimilar from training molecules (Appendix C.6).

On each test molecule, we sample 200 structures using the model, calculate their rooted mean square deviations (RMSDs) against the equilibrium structure in the test set, and evaluate the mean and the minimum over these RMSDs. Due to the geometric invariance of the structure distribution, the RMSD is evaluated after translational and rotational alignment of two structures using the Kabsch algorithm [62]. We consider both the denoising (Eq. (9)) and the DDIM (Eq. (3)) approaches for structure sampling. We also provide coverage evaluation results in Appendix C.2.

In each setting, we independently repeat the evaluation process for five times using different random seeds, and report the mean of the repeats in the following tables. The standard deviations and t-test p-values are collectively provided in Appendix C.5. In settings using consistency training, both the optimality (Eq. (10)) and score consistency losses (Eq. (11)) are added to the multi-task training loss (Eq. (7)). Validation results for training in terms of both energy prediction and structure generation are provided in Appendix C.4.

 https://doi.org/10.52202/079017-2330

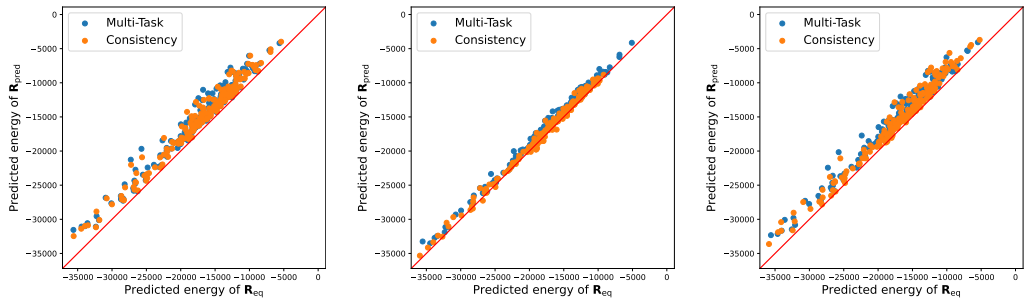


Figure 2: Comparison of energy (eV) on the model-generated structure $\mathbf{R}_{\text{pred}}$ using the denoising method and the equilibrium structure $\mathbf{R}_{\text{eq}}$ in the PCQ dataset. Each point represents the model-predicted energy values on the two structures for one test molecule. Models are trained on (**left**) the PM6 dataset, (**middle**) the PM6 dataset and SPICE force dataset, and (**right**) the PM6 dataset with a subset of force labels. The closer a point lies to the diagonal line, the closer the energy of the predicted structure is to the minimum energy, indicating a closer prediction of equilibrium structure.

### 4.2 Structure Prediction Results

We first evaluate the effect of consistency training over multi-task training following the above settings. The results are shown in Table 1. We see that consistency training enhances the accuracy of structure prediction beyond multi-task learning consistently, without using any more accurate structure data. The improvement is significant when compared to the standard deviations provided in Appendix Table C.8, which are as low as around 0.003 Å (Appendix Table C.10 verifies t-test significance). We note that this improvement is not at the cost of a lower energy prediction accuracy, as indicated by the energy prediction results in Appendix Table C.6(left).

To further examine that the improvement is from the effect of consistency training, we evaluate the energy of the predicted structure and the true DFT-level equilibrium structure in the PCQ dataset for each test molecule using the model, which is shown in the scatter plot of Fig. 2(left). We observe that when using consistency training, the overall energy is reduced, indicating that the predicted structures indeed have lower energy hence closer to the true DFT-level equilibrium structure. To quantify this improvement, in Appendix C.3 we define an "energy gap" metric, and the results shown in Table C.5 consolidate the observation.

### 4.3 Results using Physically-Related Data

We next investigate the effect of consistency training for leveraging physically related data, as explained in Sec. 3.4. For this, we consider the force data in the SPICE dataset (PubChem subset) [32]. The force data are also available on multiple off-equilibrium structures for each molecule. We note the subtlety that the setting of DFT in SPICE, $\omega$B97M-D3(BJ)/def2-TZVPPD, is different from that for generating energy labels in the PM6 dataset. (Up to our knowledge, there does not seem to exist a public force dataset that matches the DFT setting as the PM6 dataset.) On one hand, calculation on near-equilibrium structures of small molecules is not very sensitive to DFT settings, especially for force calculation (even less affected than energy), while the force labels on multiple structures could be more valuable to learning the energy landscape despite the mismatch. So we still consider it as a

Table 1: Test RMSD (Å; lower is better) of structure prediction by multi-task learning and consistency learning on PM6 dataset.

| Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|
| Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| Struct. Stat. | Mean | Min | Mean | Min | Mean | Min | Mean | Min |
| Multi-Task | 1.189 | 0.655 | 1.041 | 0.361 | 0.928 | 0.545 | 0.669 | 0.197 |
| Consistency | **1.158** | **0.645** | **1.007** | **0.346** | **0.848** | **0.490** | **0.650** | **0.194** |

https://doi.org/10.52202/079017-2330

Table 2: Test RMSD (Å; lower is better) of structure prediction by multi-task learning and consistency learning on the PM6 dataset *with additional SPICE force dataset or PM6 subset force data*.

| Additional Training Data | Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| | Struct. Stat. | Mean | Min | Mean | Min | Mean | Min | Mean | Min |
| SPICE force | Multi-Task | 1.161 | 0.631 | 1.047 | 0.373 | 0.876 | 0.486 | 0.670 | 0.207 |
| | Consistency | **1.147** | **0.590** | **1.013** | **0.345** | **0.842** | **0.485** | **0.644** | **0.194** |
| PM6 subset force | Multi-Task | 1.199 | 0.672 | 1.027 | 0.365 | 0.914 | 0.545 | 0.648 | 0.193 |
| | Consistency | **1.113** | **0.629** | **1.019** | **0.351** | **0.836** | **0.488** | **0.646** | **0.192** |

relevant investigation setting. Note energy labels in SPICE are not used, and additional structures are not used for training the structure decoder. On the other hand, to reduce the gap, we also generated in-house force labels on a subset of PM6 structures using the same DFT setting (B3LYP/6-31G*) as for the PM6 dataset energy labels. The systematic error is controlled, although for each molecule there is only one labeled structure.

The results after incorporating SPICE force data and PM6 subset force data in training are shown in Table 2. We first observe that consistency training still outperforms multi-task training in structure prediction in all cases (see Appendix Tables C.8 and C.10 for standard derivations and t-test p-values). We note that this improvement is not at the cost of a lower energy prediction accuracy, as indicated by Appendix Table C.6(middle) indicates energy prediction is also not compromised in consistency training in this case. When compared to Table 1, we see that the inclusion of force data does not uniformly enhance multi-task learning performance, since the mechanism to learn a better representation is still implicit and indirect and may require extensive tuning. In contrast, using consistency losses improves structure prediction more consistently when physically related data are available in training. These observations indicate that consistency loss training can potentially assist the model in more effectively utilizing data from different sources or modalities.

Energy analysis is presented in Fig. 2(middle) utilizing SPICE force data, and in Fig. 2(right) with force labels on a subset of PM6 molecules. The data indicate that training with consistency loss results in lower predicted energies than multi-task training. Furthermore, we observe that the structures predicted by models trained with force datasets (Fig. 2, middle and right) have lower predicted energies compared to those trained exclusively on the PM6 dataset (Fig. 2, left), illustrating the advantage of incorporating force labels.

### 4.4 Fine-Tuning Results

In addition to the zero-shot prediction evaluations, we further fine-tune the pre-trained models investigated in Sec. 4.2 using DFT-level structures in PCQ or QM9 training datasets. The results presented in Table 3 indicate that the inclusion of consistency loss in pre-training still enhances the accuracy of structure prediction even after fine-tuning. See Appendix Tables C.9 and C.10 for statistical significance. This could be attributed to that using consistency losses in pre-training can already inform the model of more accurate structure information, leading to a state that is more relevant to the underlying physics, which is a favored starting point for further improvements through

Table 3: Test RMSD (Å; lower is better) *after finetuning* for structure prediction pre-trained by multi-task learning and consistency learning on the PM6 dataset.

| Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|
| Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| Struct. Stat. | Mean | Min | Mean | Min | Mean | Min | Mean | Min |
| Multi-Task | 1.158 | 0.614 | 0.921 | 0.220 | 0.889 | 0.467 | 0.501 | 0.090 |
| Consistency | **1.152** | **0.610** | **0.918** | **0.218** | **0.835** | **0.420** | **0.493** | **0.076** |


https://doi.org/10.52202/079017-2330

fine-tuning. Results of fine-tuning models that are pre-trained with SPICE force and PM6 subset force are shown in Appendix C.1, which further demonstrates the advantages of the consistency loss.

## 5 Conclusion and Discussion

This work leverages physical laws between molecular tasks to bridge data heterogeneity in multi-task learning. Consistency losses are designed to enforce physical laws between inter-atomic potential energy prediction and equilibrium structure prediction. They have shown to improve structure prediction beyond the typical accuracy level of structure data by leveraging abundant energy data in a higher level of accuracy, and can directly leverage force and off-equilibrium structure data to further improve the accuracy. The advantage still holds after finetuning. We would like to highlight that the improvement comes "for free" as no additional data (*e.g.*, more accurate structure data) are required, demonstrating the value of physical laws in learning molecular tasks. The idea bears broader generality as data heterogeneity is ubiquitous in the science domain, and data for a specific task are often limited in either abundancy or accuracy.

The current work is limited to the consistency between energy and structure prediction, while more consistency laws can be considered in molecular science. Apart from mentioned works in connecting energy and thermodynamic distribution, more possibilities include electronic structure and molecular properties, and fine-grained and coarse-grained structures and macroscopic statistics. The significance of improvement in this work is still limited by the abundance of the data involved. Further improvement can be expected with more abundant/diverse but possibly less accurate structure data from, *e.g.*, RDKit [63] or experimental measurements, or energy/force datasets in matching level of theory that more extensively explore the structural space.

## References


[1] Oliver T Unke and Markus Meuwly. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15(6): 3678–3693, 2019.

[2] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do Transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

[3] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021.

[4] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and E Weinan. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 120(14):143001, 2018.

[5] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. GemNet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34: 6790–6802, 2021.

[6] Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

[7] Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, 2022.

[8] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14(1):579, 2023.

[9] He Li, Zun Wang, Nianlong Zou, Meng Ye, Runzhang Xu, Xiaoxun Gong, Wenhui Duan, and Yong Xu. Deep-learning density functional theory Hamiltonian for efficient ab initio electronic-structure calculation. *Nature Computational Science*, 2(6):367–377, 2022.

[10] He Zhang, Chang Liu, Zun Wang, Xinran Wei, Siyuan Liu, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Self-consistency training for density-functional-theory Hamiltonian prediction. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Vw4Yar2fmW.

[11] James Kirkpatrick, Brendan McMorrow, David HP Turban, Alexander L Gaunt, James S Spencer, Alexander GDG Matthews, Annette Obika, Louis Thiry, Meire Fortunato, David Pfau, et al. Pushing the frontiers of density functionals by solving the fractional electron problem. *Science*, 374(6573):1385–1389, 2021.

[12] R. Remme, T. Kaczun, M. Scheurer, A. Dreuw, and F. A. Hamprecht. KineticNet: Deep learning a transferable kinetic energy functional for orbital-free density functional theory. *The Journal of Chemical Physics*, 159(14):144113, 10 2023. ISSN 0021-9606. doi: 10.1063/5.0158275.

[13] He Zhang, Siyuan Liu, Jiacheng You, Chang Liu, Shuxin Zheng, Ziheng Lu, Tong Wang, Nanning Zheng, and Bin Shao. Overcoming the barrier of orbital-free density functional theory for molecular systems using deep learning. *Nature Computational Science*, Mar 2024. ISSN 2662-8457. doi: 10.1038/s43588-024-00605-8.

[14] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[15] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[16] Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hoogeboom22a.html.

[17] Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, He Zhang, Shidi Tang, Hongxia Hao, Peiran Jin, Chi Chen, Frank Noé, Haiguang Liu, and Tie-Yan Liu. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00837-3.

[18] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[19] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, pages 1–9, 2023.

[20] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[21] Rich Caruanaf. Multitask learning. *Machine learning*, 28:41–75, 1997.

[22] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-Mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6K2RM6wVqKu.

[23] Duo Zhang, Xinzijian Liu, Xiangyu Zhang, Chengqian Zhang, Chun Cai, Hangrui Bi, Yiming Du, Xuejian Qin, Jiameng Huang, Bowen Li, Yifan Shan, Jinzhe Zeng, Yuzhi Zhang, Siyuan Liu, Yifan Li, Junhan Chang, Xinyan Wang, Shuo Zhou, Jianchuan Liu, Xiaoshan Luo, Zhenyu Wang,

 https://doi.org/10.52202/079017-2330

Wanrun Jiang, Jing Wu, Yudi Yang, Jiyuan Yang, Manyi Yang, Fu-Qiang Gong, Linshuang Zhang, Mengchao Shi, Fu-Zhi Dai, Darrin M. York, Shi Liu, Tong Zhu, Zhicheng Zhong, Jian Lv, Jun Cheng, Weile Jia, Mohan Chen, Guolin Ke, Weinan E, Linfeng Zhang, and Han Wang. DPA-2: Towards a universal large atomic model for molecular and material simulation. *arXiv preprint arXiv:2312.15492*, 2023.

[24] Shengjie Luo, Tianlang Chen, Yixian Xu, Shuxin Zheng, Tie-Yan Liu, Liwei Wang, and Di He. One transformer can understand both 2d & 3d molecular data. In *The Eleventh International Conference on Learning Representations*, 2023.

[25] 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.

[26] Maho Nakata and Tomomi Shimazaki. PubChemQC project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017.

[27] Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. OGB-LSC: A large-scale challenge for machine learning on graphs. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[28] Maho Nakata, Tomomi Shimazaki, Masatomo Hashimoto, and Toshiyuki Maeda. PubChemQC PM6: Data sets of 221 million molecules with optimized molecular geometries and electronic properties. *Journal of Chemical Information and Modeling*, 60(12):5891–5899, 2020.

[29] Maho Nakata and Toshiyuki Maeda. PubChemQC B3LYP/6-31G*//PM6 dataset: the electronic structures of 86 million molecules using B3LYP/6-31G* calculations. *arXiv preprint arXiv:2305.18454*, 2023.

[30] 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.

[31] Stefan Chmiela, Valentin Vassilev-Galindo, Oliver T Unke, Adil Kabylda, Huziel E Sauceda, Alexandre Tkatchenko, and Klaus-Robert Müller. Accurate global machine learning force fields for molecules with hundreds of atoms. *Science Advances*, 9(2):eadf0873, 2023.

[32] Peter Eastman, Pavan Kumar Behara, David L Dotson, Raimondas Galvelis, John E Herr, Josh T Horton, Yuezhi Mao, John D Chodera, Benjamin P Pritchard, Yuanqing Wang, et al. SPICE, a dataset of drug-like molecules and peptides for training machine learning potentials. *Scientific Data*, 10(1):11, 2023.

[33] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[34] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[36] Gregor Simm and Jose Miguel Hernandez-Lobato. A generative model for molecular distance geometry. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8949–8958. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/simm20a.html`.

[37] Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=pAbm1qfheGk`.

[38] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

[39] Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3D conformer ensembles. *Advances in Neural Information Processing Systems*, 34:13757–13769, 2021.

[40] Jinhua Zhu, Yingce Xia, Chang Liu, Lijun Wu, Shufang Xie, Yusong Wang, Tong Wang, Tao Qin, Wengang Zhou, Houqiang Li, Haiguang Liu, and Tie-Yan Liu. Direct molecular conformation generation. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[41] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9558–9568. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/shi21b.html.

[42] Shitong Luo, Chence Shi, Minkai Xu, and Jian Tang. Predicting molecular conformation via dynamic graph score matching. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19784–19795. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a45a1d12ee0fb7f1f872ab91da18f899-Paper.pdf.

[43] Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 24240–24253. Curran Associates, Inc., 2022.

[44] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. GeoDiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

[45] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[46] Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XCTVFJwS9LJ.

[47] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[48] Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8pvnfTAbu1f.

[49] Avishek Joey Bose, Tara Akhound-Sadegh, Jarrid Rector-Brooks, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from Boltzmann densities. In *International Conference on Machine Learning*, 2024.

[50] Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zügner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

[51] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

 https://doi.org/10.52202/079017-2330

[52] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

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

[54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=St1giarCHLP`.

[55] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pages 21696–21707, 2021.

[56] Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[57] Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.

[58] Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16284–16294, 2023.

[59] Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pages 5361–5370. PMLR, 2020.

[60] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, pages 40001–40039, 2023.

[61] James JP Stewart. Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements. *Journal of Molecular modeling*, 13: 1173–1213, 2007.

[62] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32 (5):922–923, 1976.

[63] Greg Landrum et al. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8(31.10):5281, 2013.

[64] Yu Shi, Shuxin Zheng, Guolin Ke, Yifei Shen, Jiacheng You, Jiyan He, Shengjie Luo, Chang Liu, Di He, and Tie-Yan Liu. Benchmarking graphormer on large-scale molecular modeling datasets. *arXiv preprint arXiv:2203.04810*, 2022.

[65] Tianlang Chen, Shengjie Luo, Di He, Shuxin Zheng, Tie-Yan Liu, and Liwei Wang. GeoMFormer: A general architecture for geometric molecular representation learning. In *International Conference on Machine Learning*, 2024.

[66] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, San Diega, CA, USA, 2015.

[67] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 2015.

# Appendix

## A Proof of Proposition 1

We first prove the following two preliminary Lemmas.

**Lemma A.1.** *Let* $\boldsymbol{\epsilon} = \left[\boldsymbol{\epsilon}_1^\top, \ldots, \boldsymbol{\epsilon}_A^\top\right]^\top$*, and* $\boldsymbol{\epsilon}_i$ *are independent, identically distributed random vectors with* $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3)$*.* $\mathbf{Q}$ *is a random variable uniformly distributed over* $\mathrm{SO}(3)$*. Then the random matrix* $\boldsymbol{\gamma} = \boldsymbol{\epsilon}\mathbf{Q}$ *has the same distribution as* $\boldsymbol{\epsilon}$*.*

***Proof.*** Consider the probability density function of $\boldsymbol{\gamma}$:

$$p_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}) = \int p_{\boldsymbol{\gamma},\mathbf{Q}}(\boldsymbol{\gamma}, \mathbf{Q})\mathrm{d}\mathbf{Q} = \int p_{\mathbf{Q}}(\mathbf{Q}) p_{\boldsymbol{\gamma}|\mathbf{Q}}(\boldsymbol{\gamma} \mid \mathbf{Q})\mathrm{d}\mathbf{Q}.$$

Since $\boldsymbol{\gamma} = \boldsymbol{\epsilon}\mathbf{Q}$, we have:

$$p_{\boldsymbol{\gamma}|\mathbf{Q}}(\boldsymbol{\gamma} \mid \mathbf{Q}) = \prod_{i=1}^{A} \frac{1}{(2\pi)^{3/2}} \exp\left\{-\frac{\|\mathbf{Q}^\top \boldsymbol{\gamma}_i\|_2^2}{2}\right\} = \frac{1}{(2\pi)^{3/2}} \prod_{i=1}^{A} \exp\left\{-\frac{\|\boldsymbol{\gamma}_i\|_2^2}{2}\right\} = p_{\boldsymbol{\epsilon}}(\boldsymbol{\gamma}).$$

Thus, it follows that:

$$p_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}) = \int p_{\mathbf{Q}}(\mathbf{Q}) p_{\boldsymbol{\gamma}|\mathbf{Q}}(\boldsymbol{\gamma} \mid \mathbf{Q})\mathrm{d}\mathbf{Q} = \int p_{\mathbf{Q}}(\mathbf{Q}) p_{\boldsymbol{\epsilon}}(\boldsymbol{\gamma})\mathrm{d}\mathbf{q} = p_{\boldsymbol{\epsilon}}(\boldsymbol{\gamma}),$$

which demonstrates that $\boldsymbol{\gamma}$ is identically distributed as $\boldsymbol{\epsilon}$. □

**Lemma A.2.** *Assume* $\mathbf{Q}$ *is a random variable uniformly distributed over* $\mathrm{SO}(3)$*. Then* $\mathbb{E}_{\mathbf{Q}}\mathbf{Q} = \mathbf{0}$*.*

***Proof.*** For any fixed $\mathbf{Q}_0 \in \mathrm{SO}(3)$, the distribution of $\mathbf{Q}\mathbf{Q}_0$ is identical to that of $\mathbf{Q}$ due to the uniformity of the distribution over the group $\mathrm{SO}(3)$. Consequently, we have:

$$\mathbb{E}_{\mathbf{Q}}\mathbf{Q} = \mathbb{E}_{\mathbf{Q}}\mathbf{Q}\mathbf{Q}_0 = \left[\mathbb{E}_{\mathbf{Q}}\mathbf{Q}\right]\mathbf{Q}_0$$

Suppose $\mathbb{E}_{\mathbf{Q}}\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]$, where $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ are the columns of $\mathbb{E}_{\mathbf{Q}}\mathbf{Q}$. Then, it follows that:

$$[\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3] = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]\mathbf{Q}_0.$$

Selecting $\mathbf{Q}_0 = \mathrm{Diag}\{-1, -1, 1\}$ yields $\mathbf{q}_1 = -\mathbf{q}_1$ and $\mathbf{q}_2 = -\mathbf{q}_2$, , which implies that $\mathbf{q}_1 = \mathbf{q}_2 = \mathbf{0}$. Similarly, choosing $\mathbf{Q}_0 = \mathrm{Diag}\{1, -1, -1\}$ results in $\mathbf{q}_3 = -\mathbf{q}_3$, hence $\mathbf{q}_3 = \mathbf{0}$. Therefore, we conclude that $\mathbb{E}_{\mathbf{Q}}\mathbf{Q} = \mathbf{0}$. □

We now proceed to prove Proposition 1. Let $\mathbf{Q}$ be a random variable uniformly distributed over $\mathrm{SO}(3)$. Then according to Lemma A.1, the random matrix $\boldsymbol{\gamma} = \boldsymbol{\epsilon}\mathbf{Q}$ is identically distributed as $\boldsymbol{\epsilon}$. Then we have:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{\epsilon}}\|\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon}) - \mathbf{R}^\star\|_2^2 = \mathbb{E}_{\boldsymbol{\gamma}}\|\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\gamma}) - \mathbf{R}^\star\|_2^2 = \mathbb{E}_{\boldsymbol{\epsilon},\mathbf{Q}}\|\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon}\mathbf{Q}) - \mathbf{R}^\star\|_2^2 \\ &= \mathbb{E}_{\boldsymbol{\epsilon},\mathbf{Q}}\|\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon})\mathbf{Q} - \mathbf{R}^\star\|_2^2 = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\mathbb{E}_{\mathbf{Q}}\|\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon}) - \mathbf{R}^\star\mathbf{Q}^\top\|_2^2\right]. \end{aligned}$$

For any $\boldsymbol{\epsilon} \in \mathbb{R}^{A\times 3}$, the objective function $\mathbb{E}_{\mathbf{Q}}\left\|\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon}) - \mathbf{R}^\star\mathbf{Q}^\top\right\|_2^2$ achieves its minimum when $\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon}) = \mathbb{E}_{\mathbf{Q}}[\mathbf{R}^\star\mathbf{Q}^\top] = \mathbf{R}^\star[\mathbb{E}_{\mathbf{Q}}\mathbf{Q}^\top]$. Since $\mathbb{E}_{\mathbf{Q}}\mathbf{Q}^\top = \mathbb{E}_{\mathbf{Q}}\mathbf{Q}$ and using Lemma A.2, we have $\mathbb{E}_{\mathbf{Q}}\mathbf{Q}^\top = \mathbf{0}$, which implies $\mathbf{S}^{(\boldsymbol{\theta})}(\boldsymbol{\epsilon}) = \mathbf{0}$. Therefore, $\mathbf{S}^{(\boldsymbol{\theta})}$ is a zero map.

## B Implementation Details

### B.1 Implementation Details for Consistency Losses

The proposed consistency loss $L_{\text{optim-cons}}$ in Eq. (10) and $L_{\text{score-cons}}$ in Eq. (11) are designed to update the parameters $\theta_{\mathrm{S}}$ within the structure prediction model $\mathbf{S}_{\mathcal{G},\tau}^{(\boldsymbol{\Phi},\theta_{\mathrm{S}})}$. To achieve this, we employ the stop gradient operation $\mathrm{SG}(\cdot)$ to prevent unnecessary gradient computation for the parameters $\boldsymbol{\Phi}$ in the encoder model $\mathcal{E}_{\mathcal{G},t}^{(\boldsymbol{\Phi})}$ and $\theta_{\mathrm{E}}$ in the energy decoder $\mathcal{D}_{\mathrm{E}}^{((\theta_{\mathrm{E}})}$. The detailed implementations of optimality consistency and score consistency are presented in Alg. B.1 Alg. B.2, respectively.


https://doi.org/10.52202/079017-2330

**Algorithm B.1** Implementation of Optimality Consistency Loss

**Require:** Encoder: $\mathcal{E}_{\mathcal{G},t}^{(\mathbf{\Phi})}(\mathbf{R})$, structure decoder $\mathcal{D}_{\mathrm{E}}^{(\theta_{\mathrm{E}})}$, energy decoder $\mathcal{D}_{\mathrm{S}}^{(\theta_{\mathrm{S}})}$, Diffusion time step $\tau$.
**Ensure:** $\nabla_{\theta_{\mathrm{S}}} L_{\text{optim-cons}}$

1: Sample $\boldsymbol{\epsilon}$ and $\boldsymbol{\eta}$.
2: Extract the molecular feature $\mathbf{h} \leftarrow \mathcal{E}_{\mathcal{G},t=\tau}^{(\mathbf{\Phi})}(\boldsymbol{\epsilon})$ using the encoder.
3: Apply the stop gradient operation to the molecular feature and obtain $\bar{\mathbf{h}} \leftarrow \mathrm{SG}(\mathbf{h})$ through Pytorch's `.detach()` method.
4: Compute the denoised structure $\hat{\mathbf{R}}_0 \leftarrow \mathcal{D}_{\mathrm{S}}^{(\theta_{\mathrm{S}})}(\bar{\mathbf{h}}, \boldsymbol{\epsilon})$ using the structure decoder.
5: Set `requires_grad = False` for the parameters in the energy model $E_{\mathcal{G}}^{(\mathbf{\Phi},\theta_{\mathrm{E}})}(\mathbf{R}) = \mathcal{D}_{\mathrm{E}}^{(\theta_{\mathrm{E}})}\left(\mathcal{E}_{\mathcal{G},t=0}^{(\mathbf{\Phi})}(\mathbf{R})\right)$
6: Evaluate the loss $L_{\text{optim-cons}} = \max\left\{0,\ E_{\mathcal{G}}^{(\mathbf{\Phi},\theta_{\mathrm{E}})}(\hat{\mathbf{R}}_0) - E_{\mathcal{G}}^{(\mathbf{\Phi},\theta_{\mathrm{E}})}(\hat{\mathbf{R}}_0 + \boldsymbol{\eta})\right\}$.
7: Determine the gradient $\nabla_{\theta_{\mathrm{S}}} L_{\text{optim-cons}}$ through automatic differentiation.

**Algorithm B.2** Implementation of Score Consistency Loss

**Require:** Encoder: $\mathcal{E}_{\mathcal{G},t}^{(\mathbf{\Phi})}(\mathbf{R})$, structure decoder $\mathcal{D}_{\mathrm{E}}^{(\theta_{\mathrm{E}})}$, energy decoder $\mathcal{D}_{\mathrm{S}}^{(\theta_{\mathrm{S}})}$, Diffusion time step $\tau$.
**Ensure:** $\nabla_{\theta_{\mathrm{S}}} L_{\text{score-cons}}$

1: Sample $\mathbf{R}_\tau$.
2: Extract the molecular feature $\mathbf{h}_0 \leftarrow \mathcal{E}_{\mathcal{G},t=0}^{(\mathbf{\Phi})}(\mathbf{R}_\tau)$ using the encoder.
3: Compute the free energy $\mathbf{E} \leftarrow \mathcal{D}_{\mathrm{E}}^{(\theta_{\mathrm{E}})}\left(\mathbf{h}_0\right)$ with the energy decoder.
4: Compute the energy gradient $\mathbf{F} \leftarrow \nabla_{\mathbf{R}_\tau}\mathbf{E}$ using PyTorch's `torch.autograd`.
5: Apply the stop gradient operation (`.detach()` method in Pytorch) to the energy gradient: $\bar{\mathbf{F}} \leftarrow \mathrm{SG}\left(\mathbf{F}\right)$.
6: Extract the molecular feature $\mathbf{h}_\tau \leftarrow \mathcal{E}_{\mathcal{G},t=\tau}^{(\mathbf{\Phi})}(\mathbf{R}_\tau)$ using the encoder.
7: Obtain $\bar{\mathbf{h}}_\tau \leftarrow \mathrm{SG}(\mathbf{h}_\tau)$ using `.detach()`.
8: Compute the denoised structure $\hat{\mathbf{R}}_0 \leftarrow \mathcal{D}_{\mathrm{S}}^{(\theta_{\mathrm{S}})}\left(\mathrm{SG}(\bar{\mathbf{h}}_\tau), \mathbf{R}_\tau\right)$ with structure decoder.
9: Evaluate the loss $L_{\text{score-cons}} = \left\| \frac{\sqrt{\bar{\alpha}_\tau}\hat{\mathbf{R}}_0 - \mathbf{R}_\tau}{1-\bar{\alpha}_\tau} + \frac{\bar{\mathbf{F}}}{k_{\mathrm{B}}\mathcal{T}} \right\|_2^2$.
10: Determine the gradient $\nabla_{\theta_{\mathrm{S}}} L_{\text{score-cons}}$ through automatic differentiation.

### B.2 Model Architecture

The model is composed of an encoder, and two decoders for structure and energy prediction.

The encoder $\mathcal{E}_{\mathcal{G},t}^{(\mathbf{\Phi})}(\mathbf{R})$ is a simple modification to the Graphormer model [2, 64], which additionally adopts diffusion time $t$ embedding into both node features and pairwise-distance attention bias. The encoder consists of 24 layers of Graphormer, with the dimension of both hidden and feed-forward layers set to 768. It utilizes a multi-head attention mechanism with 32 heads and employs 128 Gaussian Basis kernels for enhancing the positional encoding.

For the time embedding, we implement a `SinusoidalPositionEmbeddings` module and a `TimeStepEncoder` module. The former generates time-dependent sinusoidal embeddings, and the latter refines these embeddings using a feed-forward network with a GELU activation function. The resulting time embeddings are then integrated into the node features to inform the model of temporal information.

Additionally, we incorporate time embeddings into the attention mechanism by computing a structure-based attention bias. This is achieved by calculating the outer product of the time embeddings and using the result as an additive bias in the self-attention layers. This integration allows the model to adapt its attention based on the temporal relationships between nodes in the graph.

On top of the encoder, the energy decoder $\mathcal{D}_{\mathrm{E}}^{(\theta_{\mathrm{E}})}(\mathbf{h})$ is a simple MLP layer concatenated to the invariant node features $\mathbf{h}$.

Table C.1: Index of main result tables in the paper.

| Training settings | Test RMSD | Standard deviation | Paired t-test | Coverage | Validation (structure) | Validation (energy) | EGap |
|---|---|---|---|---|---|---|---|
| PM6 | Table 1 | Table C.8 | Table C.10 | Table C.3 | Table C.7 | Table C.6 | Table C.5 |
| PM6 w/ force | Table 2 | Table C.8 | Table C.10 | Table C.3 | Table C.7 | Table C.6 | Table C.5 |
| PM6 + finetuning | Table 3 | Table C.9 | Table C.10 | Table C.4 | N/A | N/A | N/A |
| PM6 w/ force + finetuning | Table C.2 | Table C.9 | - | Table C.4 | N/A | N/A | N/A |

The structure decoder $\mathcal{D}_{\text{S}}^{(\theta_{\text{S}})}(\mathbf{h}, \mathbf{R})$ adopts the GeoMFormer architecture [65], which takes the invariant node features $\mathbf{h}$ from the output of the encoder, and the atom coordinates $\mathbf{R}$. The output is denoised atom coordinates which are equivariant w.r.t the input coordinates $\mathbf{R}$. These modules are combined to form the energy and structure prediction models following Eq. (6).

### B.3 Training Details

Our pre-training procedure is executed in two discrete stages. Initially, the model is subjected to training exclusively utilizing the multi-task loss function, $L_{\text{multi-task}}$, for a total of 300,000 iterations. Subsequently, in the second stage, we integrate the proposed consistency loss and the force loss, $L_{\text{force}}$, into the training regimen, which then proceeds for an additional 200,000 iterations. All the models are trained with the Adam optimizer [66] with batch size 256. The learning rate is set to $2 \times 10^{-4}$ with a linear warm-up phase in the initial 10,000 steps, which followed by a linear decay schedule thereafter.

The weights of the energy loss and the diffusion denoising loss, *i.e.*, the first and the second terms in Eq. (7), are set to $1.0$ and $0.01$, respectively. The weights of the optimality consistency loss Eq. (10) and the score consistency loss Eq. (11) are set to $0.1$ and $1.0$, respectively.

We employ a sigmoid schedule across 1,000 diffusion time steps for $\beta_t$, with $\beta_0 = 1 \times 10^{-4}$ and $\beta_T = 2 \times 10^{-2}$. For the optimality consistency loss, the diffusion time step $\tau$ is sampled uniformly from $[400, 700]$. For the score consistency loss, the diffusion time step $\tau$ is sampled uniformly from $[5, 300]$, and $k_{\text{B}}\mathcal{T}$ is set to $0.1\,\text{eV}$.

The multi-task model is trained on an $8\times$ Nvidia V100 GPU server for approximately one week. The model with the consistency loss is trained on one $16\times$ Nvidia V100 GPU server.

## C Additional Results

In this section, we provide additional results under various combinations of settings, and additional metrics and supporting evidence to complement the main results. For clarity, we summarize main result tables in Table C.1 for easier indexing.

### C.1 Additional Fine-Tuning Results

This section demonstrate more fine-tuning results listed in the main text. Besides the fine-tuned PM6 dataset pre-trained mode, we performed fine-tuning experiments on the model pre-trained with the SPICE force and PM6 subset force datasets, as well. The results are presented in Table C.2. Similar to the case in Table 3, we can again observe that while finetuning improves structure prediction accuracy in all settings (compared to Table 2), pre-training the model with consistency loss still enhances the accuracy universally.

### C.2 Coverage Evaluation for Structure Generation

Following the common practice in structure-generation literature, we also test the coverage of the ground-truth structure over model-generated structures. This metric is defined as:

$$\text{COV}(\mathbb{S}_{\text{gen}}, \mathbf{R}_{\text{eq}}) = \frac{1}{|\mathbb{S}_{\text{gen}}|}\Big|\{\hat{\mathbf{R}} \in \mathbb{S}_{\text{gen}} \mid \text{RMSD}(\mathbf{R}_{\text{eq}}, \hat{\mathbf{R}}) < \delta\}\Big|, \tag{C.1}$$


https://doi.org/10.52202/079017-2330

Table C.2: Test RMSD (Å; lower is better) *after finetuning* for structure prediction pre-trained by multi-task learning and consistency learning on the PM6 dataset *with additional SPICE force data or PM6 subset force data*.

| (Pre-)Training Set | Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| | Struct. Stat. | Mean | Min | Mean | Min | Mean | Min | Mean | Min |
| PM6 with SPICE force | Multi-Task | 1.161 | 0.618 | 0.930 | 0.219 | 0.855 | 0.444 | 0.505 | 0.081 |
| | Consistency | **1.132** | **0.581** | **0.916** | **0.215** | **0.832** | **0.418** | **0.492** | **0.073** |
| PM6 with subset force | Multi-Task | 1.143 | 0.603 | 0.927 | 0.224 | 0.855 | 0.441 | 0.497 | 0.080 |
| | Consistency | **1.099** | **0.542** | **0.914** | **0.215** | **0.822** | **0.419** | **0.490** | **0.076** |

where $\mathbb{S}_{\text{gen}}$ denotes the set of generated structures, $\mathbf{R}_{\text{eq}}$ denotes the ground-truth equilibrium structure provided from the evaluation dataset, $\delta$ is a threshold parameter, and $|\cdot|$ takes the cardinality of a set. Note that since for the task of equilibrium structure prediction, there is only one ground-truth structure, we only evaluate the so-called precision coverage, since the recall coverage (by switching the roles of $\mathbb{S}_{\text{gen}}$ and $\mathbf{R}_{\text{eq}}$ in the definition Eq. (C.1)) evaluates to 1 in all cases. Here, the RMSD is evaluated after alignment of the two structures by Kabsch algorithm [62]. We choose $\delta$ as 0.9 and 1.25 for QM9 and PCQ dataset. Same as the RMSD test, we use the same test molecules and sample 200 structure for each molecule. The coverage (COV) of the PM6 dataset model prediction and including force dataset model prediction are shown in Table C.3. The results present consistent results as the RMSD. After adding the consistency loss, all have improvement over the multi-task setting. Besides, incorporating the force data can also improve the accuracy.

Table C.3: Test coverage (higher is better) of structure prediction by multi-task learning and consistency learning on the PM6 dataset, and together with additional SPICE force data or PM6 subset force data.

| Training Set | Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| | Struct. Stat. | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| PM6 | Multi-Task | 0.597 | 0.670 | 0.649 | 0.675 | 0.468 | 0.478 | 0.717 | 0.780 |
| | Consistency | **0.626** | **0.695** | **0.671** | **0.730** | **0.604** | **0.660** | **0.732** | **0.825** |
| PM6 with SPICE force | Multi-Task | 0.614 | 0.660 | 0.645 | 0.685 | 0.550 | 0.580 | 0.715 | 0.765 |
| | Consistency | **0.644** | **0.710** | **0.675** | **0.725** | **0.617** | **0.705** | **0.740** | **0.835** |
| PM6 with subset force | Multi-Task | 0.590 | 0.650 | 0.662 | 0.705 | 0.493 | 0.520 | 0.736 | 0.835 |
| | Consistency | **0.677** | **0.775** | **0.671** | **0.720** | **0.643** | **0.690** | **0.737** | **0.845** |

We also evaluated the coverage on the fine-tuned models. The results are shown in Table C.4, where we again observe that pre-training with consistency loss improves structure prediction accuracy even after fine-tuning.

### C.3 Energy Gap Analysis

Our goal is to predict the equilibrium structure $\mathbf{R}_{\text{eq}} = \operatorname{argmin}_{\mathbf{R}} E_{\mathcal{G}}(\mathbf{R})$. It is desirable for the predicted structure to approximate this state of minimal energy. To quantify the proximity of the predicted structure to the equilibrium state, we introduce the energy gap metric, which is defined as:

$$\text{EGap} = \frac{E_{\mathcal{G}}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\text{E}})}(\mathbf{R}_{\text{pred}}) - E_{\mathcal{G}}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\text{E}})}(\mathbf{R}_{\text{eq}})}{|E_{\mathcal{G}}^{(\boldsymbol{\Phi},\boldsymbol{\theta}_{\text{E}})}(\mathbf{R}_{\text{eq}})|}.$$

The $\text{EGap}$ metric serves to evaluate the energy difference between the predicted and equilibrium structures, with a smaller energy gap signifying a more accurate prediction.

Table C.4: Test coverage (higher is better) *after finetuning* for structure prediction pre-trained by multi-task learning and consistency learning on the PM6 dataset, and together with additional SPICE force data or PM6 subset force data.

| (Pre-)Training Set | Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| | Struct. Stat. | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| PM6 | Multi-Task | 0.629 | 0.675 | 0.721 | **0.800** | 0.543 | 0.595 | **0.790** | 0.960 |
| | Consistency | **0.632** | **0.700** | **0.724** | 0.795 | **0.622** | **0.690** | 0.788 | **0.990** |
| PM6 with SPICE force | Multi-Task | 0.632 | 0.705 | 0.714 | 0.760 | 0.597 | 0.660 | 0.784 | 0.980 |
| | Consistency | **0.649** | **0.725** | **0.721** | **0.795** | **0.631** | **0.710** | **0.791** | **0.990** |
| PM6 with subset force | Multi-Task | 0.643 | 0.705 | 0.719 | **0.795** | 0.600 | 0.650 | 0.786 | 0.990 |
| | Consistency | **0.680** | **0.780** | **0.720** | 0.794 | **0.643** | **0.715** | **0.789** | **0.990** |

To compute this metric, we randomly select 200 molecules from the intersection of PM6 and PCQ dataset, using the structure from the PCQ dataset as $\mathbf{R}_{\text{eq}}$. The predicted structure $\mathbf{R}_{\text{pred}}$ is generated using the denoising method. The results are presented in Table C.5. We observe that the incorporation of the consistency loss reduces the EGap metric, particularly in cases with additional force labels. These results demonstrate that the consistency loss effectively transfers information from the energy model to the structure model.

Table C.5: Comparison of averaged EGap between structure prediction by multi-task learning and consistency learning. Lower EGap values suggest that the energy of the predicted structure is closer to the theoretical minimum energy.

| Training Set | PM6 | PM6 with SPICE force | PM6 with PM6 subset force |
|---|---|---|---|
| Multi-Task | 0.1278 | 0.0546 | 0.1163 |
| Consistency | **0.1172** | **0.0306** | **0.1013** |

### C.4 Validation Results

To make sure that the conclusions drawn from the above results are solid, we further provide validation results for energy and structure.

For the energy, we randomly selected 200 molecules from the intersection of the PM6 dataset (training set) and the PCQ dataset (test set). This choice allows evaluating energy on both PM6 structure and PCQ structure for each molecule, where the former reflects training quality, and the latter reflects the utility for consistency learning of structure prediction. Although the original PCQ dataset [27] does not provide energy labels, we noticed that it is curated based on the PubChemQC Project dataset [26], which provides energy labels on the same PCQ structures under the same DFT settings (B3LYP/6-31G*) as the PM6 energy labels.

The results are listed in Table C.6 and boxplots of the distributions of energy prediction MAE (eV) evaluated on the PM6 structures are shown in Fig. C.1. We can observe that energy prediction on PM6 structures are reasonably accurate, and it gets even better when consistency training is activated. This indicates that consistency training does not sacrifice energy prediction. The energy prediction error is larger on PCQ structures, which is not surprising since the model does not see data with PCQ structures in training. But as we mentioned in the main paper, better generalization can be expected if force data are available, which enriches the information on the energy landscape hence improving the generalization on unseen structures. From the table, we can observe that the energy prediction accuracy on PCQ structures is indeed improved. We also notice that including force data in training even improves energy prediction accuracy on PM6 structures, for which a possible explanation is that introducing additional relevant prediction tasks could help the model learn a more comprehensive representation of the input molecular structure, which in turn helps other tasks.


https://doi.org/10.52202/079017-2330

Table C.6: Validation MAE (eV) of energy prediction trained by multi-task learning and consistency learning. Validation molecules are randomly selected from the intersection of PM6 and PCQ, and results on both PM6 structures and PCQ structures of the molecules are shown.

| Validation structures from | Training Set | PM6 | PM6 with SPICE force | PM6 with PM6 subset force |
|---|---|---|---|---|
| PM6 structures | Multi-Task | 96.08 | 83.15 | 79.00 |
| | Consistency | **88.58** | **78.56** | **72.61** |
| PCQ structures | Multi-Task | 136.66 | 135.93 | 117.22 |
| | Consistency | **127.06** | **110.80** | **108.71** |

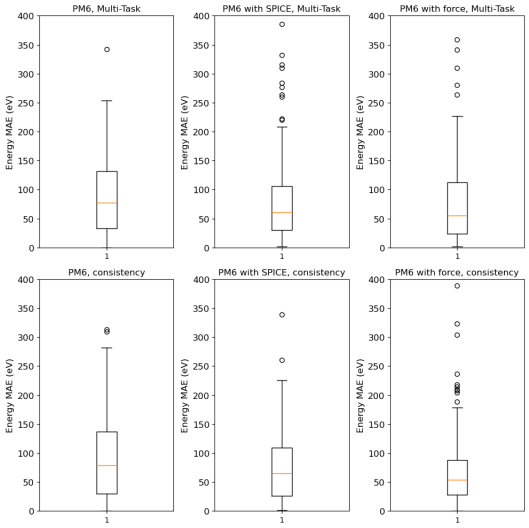


Figure C.1: Box plots for the distributions of energy prediction MAE (eV) evaluated on the PM6 structures of randomly selected 200 molecules from the intersection of PM6 and PCQ datasets.

For structure prediction, we validate the trained model by evaluating the RMSD of generated structures on 200 randomly selected PM6 molecules. From the results shown in Table C.7, we can see that the models trained by both multi-task learning and consistency learning achieve reasonable predictions.

Table C.7: Validation RMSD (Å) evaluated on the PM6 validation set of structure prediction trained by multi-task learning and consistency learning on the PM6 dataset, and together with additional SPICE force data or PM6 subset force data. Predicted structures are generated by the DDIM method.

| Training Set | PM6 | | PM6 with SPICE force | | PM6 with PM6 subset force | |
|---|---|---|---|---|---|---|
| Struct. Stat. | Mean | Min | Mean | Min | Mean | Min |
| Multi-Task | 0.970 | 0.287 | 0.983 | 0.314 | 0.953 | 0.267 |
| Consistency | 0.931 | 0.280 | 0.944 | 0.286 | 0.958 | 0.290 |

### C.5 Standard Deviations and Statistical Significance

To evaluate the significance of our comparison results, for each experimental setup, we repeated five independent runs using different random seeds (666666, 666667, 666668, 666669, 666670). The reported results in the main paper are the means of the five repeats, while the standard deviations are listed in Table C.8 and Table C.9 here, which cover the results for the pre-trained models and for fine-tuned models, respectively. We can see that all the standard deviations are around 0.003 Å, which

is clearly lower than the improvements by consistency training, hence indicating the significance of the effectiveness of the proposed methods.

To more seriously assess the statistical significance, we also conducted a paired t-test for each comparison. The p-values are listed in Table C.10, where each number represents the probability of the observed results under the null hypothesis that the means by multi-task learning and consistency learning are the same, hence the lower value the more significant that consistency learning outperforms multi-task learning. Nearly all p-values are well below the 0.05 significance threshold, indicating that the improvement by consistency learning is significant. Exceptional cases almost all correspond to the setting where the result in each repeat is the "Min"imum RMSD over 200 generated samples, in which case multi-task learning may also has a chance to hit the target structure.

Table C.8: Standard deviations for the test RMSD (Å) of structure prediction by multi-task learning and consistency learning on the PM6 dataset (corresponding to Table 1), and together with additional SPICE force data or PM6 subset force data (corresponding to Table 2).

| Training Set | Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| | Struct. Stat. | Mean | Min | Mean | Min | Mean | Min | Mean | Min |
| PM6 | Multi-Task | 0.0013 | 0.0045 | 0.0019 | 0.0059 | 0.0004 | 0.0026 | 0.0017 | 0.0015 |
| | Consistency | 0.0021 | 0.0021 | 0.0016 | 0.0053 | 0.0018 | 0.0022 | 0.0080 | 0.0016 |
| PM6 with SPICE force | Multi-Task | 0.0021 | 0.0042 | 0.0020 | 0.0067 | 0.0013 | 0.0040 | 0.0084 | 0.0024 |
| | Consistency | 0.0026 | 0.0024 | 0.0017 | 0.0030 | 0.0019 | 0.0034 | 0.0025 | 0.0030 |
| PM6 with subset force | Multi-Task | 0.0113 | 0.0048 | 0.0044 | 0.0127 | 0.0015 | 0.0134 | 0.0020 | 0.0025 |
| | Consistency | 0.0027 | 0.0040 | 0.0023 | 0.0043 | 0.0011 | 0.0030 | 0.0033 | 0.0030 |

Table C.9: Standard deviations for the test RMSD (Å) *after finetuning* for structure prediction pre-trained by multi-task learning and consistency learning on the PM6 dataset (corresponding to Table 3), and together with additional SPICE force data or PM6 subset force data (corresponding to Table C.2).

| (Pre-)Training Set | Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| | Struct. Stat. | Mean | Min | Mean | Min | Mean | Min | Mean | Min |
| PM6 | Multi-Task | 0.0021 | 0.0059 | 0.0025 | 0.0036 | 0.0008 | 0.0026 | 0.0011 | 0.0044 |
| | Consistency | 0.0024 | 0.0064 | 0.0020 | 0.0033 | 0.0011 | 0.0028 | 0.0022 | 0.0023 |
| PM6 with SPICE force | Multi-Task | 0.0023 | 0.0053 | 0.0012 | 0.0051 | 0.0016 | 0.0038 | 0.0019 | 0.0033 |
| | Consistency | 0.0023 | 0.0047 | 0.0030 | 0.0013 | 0.0018 | 0.0030 | 0.0027 | 0.0015 |
| PM6 with subset force | Multi-Task | 0.0022 | 0.0042 | 0.0016 | 0.0053 | 0.0013 | 0.0036 | 0.0019 | 0.0014 |
| | Consistency | 0.0019 | 0.0038 | 0.0019 | 0.0072 | 0.0016 | 0.0068 | 0.0016 | 0.0023 |


https://doi.org/10.52202/079017-2330

Table C.10: Paired t-test p-values on structure prediction RMSD means over 5 repeats (standard deviations are shown in Tables C.8 and C.9) corresponding to the results in Table 1 (row 1, pre-training on PM6), Table 2 (rows 2 and 3, pre-training on PM6 together with force labels), and Table 3 (row 4, pre-training on PM6 then finetuning). Values lower than the 0.05 significance threshold are shown in bold.

| Test Set | PCQ | | | | QM9 | | | |
|---|---|---|---|---|---|---|---|---|
| Generated by | Denoising | | DDIM | | Denoising | | DDIM | |
| Struct. Stat. | Mean | Min | Mean | Min | Mean | Min | Mean | Min |
| PM6 | $\mathbf{7.8_{\times 10^{-4}}}$ | $\mathbf{5.6_{\times 10^{-3}}}$ | $\mathbf{3.8_{\times 10^{-7}}}$ | $\mathbf{8.9_{\times 10^{-4}}}$ | $\mathbf{4.5_{\times 10^{-7}}}$ | $\mathbf{8.2_{\times 10^{-7}}}$ | $\mathbf{1.1_{\times 10^{-3}}}$ | $\mathbf{7.4_{\times 10^{-4}}}$ |
| PM6 w/ SPICE force | $\mathbf{2.4_{\times 10^{-5}}}$ | $\mathbf{1.0_{\times 10^{-5}}}$ | $\mathbf{6.0_{\times 10^{-4}}}$ | $\mathbf{3.1_{\times 10^{-5}}}$ | $\mathbf{9.9_{\times 10^{-8}}}$ | $4.5_{\times 10^{-1}}$ | $\mathbf{2.3_{\times 10^{-2}}}$ | $\mathbf{1.5_{\times 10^{-4}}}$ |
| PM6 w/ subset force | $\mathbf{4.5_{\times 10^{-3}}}$ | $\mathbf{2.4_{\times 10^{-5}}}$ | $\mathbf{2.9_{\times 10^{-3}}}$ | $\mathbf{2.4_{\times 10^{-2}}}$ | $\mathbf{6.5_{\times 10^{-7}}}$ | $\mathbf{1.2_{\times 10^{-4}}}$ | $5.1_{\times 10^{-2}}$ | $7.5_{\times 10^{-1}}$ |
| PM6 + finetuning | $\mathbf{4.2_{\times 10^{-3}}}$ | $2.4_{\times 10^{-1}}$ | $\mathbf{2.4_{\times 10^{-3}}}$ | $1.7_{\times 10^{-1}}$ | $\mathbf{1.2_{\times 10^{-7}}}$ | $\mathbf{2.5_{\times 10^{-6}}}$ | $\mathbf{1.0_{\times 10^{-4}}}$ | $\mathbf{8.5_{\times 10^{-3}}}$ |

### C.6 Evaluation on Dissimilar Molecules

To further consolidate the effectiveness of our method, we give a closer look into the influence of the similarity of the test dataset to the training dataset. Recall that we have excluded identical molecules appearing in the training dataset from the test dataset, but there remains the possibility that the test dataset may still contain similar molecules as those in the training dataset. For this, we first investigate the Tanimoto similarity [67] between the training and test datasets. This similarity can be computed from the molecular graphs of two molecules, while considering structural similarity between the two molecules. We plot in Fig. C.2 where each column visualizes the count of PCQ test molecules that have a certain portion of similar (Tanimoto similarity > 0.7) molecules in the PM6 training dataset. We can observe that most (almost all) of the test molecules only has less than $1 \times 10^{-7}$ ($2.5 \times 10^{-7}$) similar molecules in PM6. This indicates that excluding identical molecules appearing in the training dataset from the test dataset already makes the test dataset sufficiently dissimilar to the training dataset, hence the presented results are valid prediction evaluations that are not close to "memorizing the training molecules".

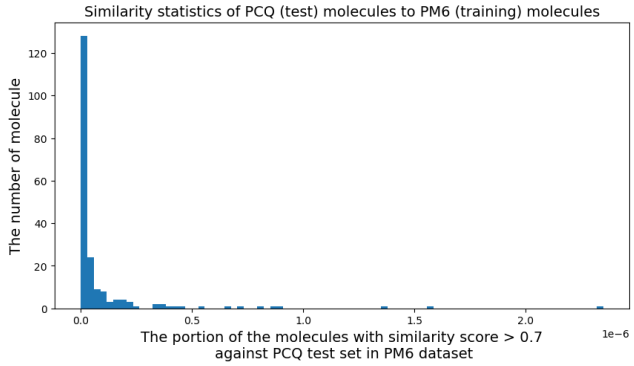


Figure C.2: Histogram showing the distribution of the portion of similar (Tanimoto similarity > 0.7) molecules in PM6 (the training dataset) over the 200 PCQ test molecules. Note that the x-axis is scaled by $1 \times 10^{-6}$.

For a completely sanitized evaluation on dissimilar molecules, we provide the results evaluated on PCQ test molecules that do not have any similar (Tanimoto similarity > 0.7) molecules in the PM6 (training) dataset. Such molecules make up 49% of the total 200 PCQ test molecules. Table C.11 shows the results in both RMSD and Coverage using the denoising generation method, corresponding to the settings in Tables 1 and C.3 (row block 1) (pre-training on PM6), and Tables 2 and C.3 (row blocks 2 and 3) (pre-training on PM6 together with force labels). The results confirmed the advantage

of consistency learning on dissimilar molecules that cannot rely on memorizing the training dataset, which further consolidates the conclusion.

Table C.11: Test RMSD (Å; lower is better) and coverage (higher is better) *on dissimilar molecules* for structure prediction pre-trained by multi-task learning and consistency learning on the PM6 dataset, and together with additional SPICE force data or PM6 subset force data.

| Training Set | Method | RMSD (Å) | | Cov | |
|---|---|---|---|---|---|
| | | Mean | Min | Mean | Median |
| PM6 (*c.f.* Tables 1 or C.3 (row block 1)) | Multi-Task | 1.175 | 0.642 | 0.613 | 0.675 |
| | Consistency | **1.135** | **0.625** | **0.644** | **0.745** |
| PM6 w/ SPICE force (*c.f.* Tables 2 (row block 1) or C.3 (row block 2)) | Multi-Task | 1.136 | 0.609 | 0.639 | 0.735 |
| | Consistency | **1.121** | **0.579** | **0.672** | **0.790** |
| PM6 w/ subset force (*c.f.* Tables 2 (row block 2) or C.3 (row block 3)) | Multi-Task | 1.174 | 0.653 | 0.612 | 0.660 |
| | Consistency | **1.099** | **0.616** | **0.697** | **0.830** |


https://doi.org/10.52202/079017-2330

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have accurately reflect the paper's contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in the Sec. 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have provided complete proof for the assumptions in Sec. 3 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have disclosed all the needed information for reproducibility in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**


https://doi.org/10.52202/079017-2330

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Releasing data and code requires an internal asset releasing review process in our organization. We have started the process, but cannot guarantee availability during the review period.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer:[Yes]

Justification: We have included all details of the model architecture, data processing, and hyperparameter settings in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have report the standard deviations and statistical significance of the main results in Appendix C.5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The required computational resources are provided in Appendix B.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work develops techniques for better solving molecular science tasks using machine learning. The techniques could potentially foster the development of related industries, *e.g.*, drug discovery and material design. Possible caveats include inaccurate generated structures or predicted properties which may lead to failure in downstream research.


https://doi.org/10.52202/079017-2330

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not release data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our paper has properly credited and cited the code and data used in this paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The new asset in this paper is the in-house generated force labels on a subset of PM6 molecules. Releasing the new asset requires internal releasing review process in our organization. We cannot guarantee releasing the asset during the review period.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing experiments nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.


https://doi.org/10.52202/079017-2330

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.