Deep Molecular Representation Learning via Fusing Physical and Chemical Information

Shuwen Yang	extsuperscript{1,}, Ziyao Li	extsuperscript{2,} Guojie Song	extsuperscript{1,†} Lingsheng Cai	extsuperscript{1}

	extsuperscript{1}Key Laboratory of Machine Perception and Intelligence (MOE), Peking University, Beijing, China
	extsuperscript{2}Center for Data Science, Peking University, Beijing, China
{swyang,leeezy,gjsong,cailingsheng}@pku.edu.cn

Abstract

Molecular representation learning is the first yet vital step in combining deep learning and molecular science. To push the boundaries of molecular representation learning, we present PhysChem, a novel neural architecture that learns molecular representations via fusing physical and chemical information of molecules. PhysChem is composed of a physicist network (PhysNet) and a chemist network (ChemNet). PhysNet is a neural physical engine that learns molecular conformations through simulating molecular dynamics with parameterized forces; ChemNet implements geometry-aware deep message-passing to learn chemical / biomedical properties of molecules. Two networks specialize in their own tasks and cooperate by providing expertise to each other. By fusing physical and chemical information, PhysChem achieved state-of-the-art performances on MoleculeNet, a standard molecular machine learning benchmark. The effectiveness of PhysChem was further corroborated on cutting-edge datasets of SARS-CoV-2.

1 Introduction

The intersection between deep learning and molecular science has recently caught the eye of researchers in both areas. Remarkable progress was made in applications including molecular property prediction [37, 16], molecular graph generation [38, 12, 29], and virtual screening for drug discovery [8, 32], yet learning representations for molecules remains the first yet vital step. Molecular representation learning, or learning molecular fingerprints, aims to encode input notations of molecules into numerical vectors, which later serve as features for downstream tasks. Earlier deep molecular representation methods generally used off-the-shelf network architectures including message-passing neural networks [9], graph attention networks [37] and Transformers [11, 25]. These methods took either line notations (e.g. SMILES) or graph notations (i.e. structural formulas) of molecules as inputs, whereas physical and chemical essence of molecules was largely neglected. Notably, a trend of integrating 3D conformations (i.e. the 3D Cartesian coordinates of atoms) into molecular representations recently emerged [4, 15, 30], while most of these methods assume the availability of labeled conformations of target molecules.

In order to push the boundaries of molecular representation learning, we revisited molecules from both physical and chemical perspectives. Modern physicists generally regard molecules as particle systems that continuously move following the laws of (quantum) mechanics. The dominant conformations of molecules reflect the equilibriums of these micro mechanical systems, and are thus of wide interest.

	extsuperscript{*}Equal Contribution.

	extsuperscript{†}Corresponding Author.

SMILES (Simplified Molecular Input Line Entry Specification [35]) is a widely used protocol that specifies (non-unique) line notations for molecules, CCO for ethanol, for example.

which merged the states of all nodes into graph representations. In this work, ChemNet modifies

\[ \mathbf{h}_i \leftarrow U(h_i, m_i), \quad i \in V, \]

where \( i, j \in V \) were graph nodes, \( h_i \)s and \( e_{i,j} \)s were states of nodes and edges, and \( M(\cdot), U(\cdot) \) were the message and update functions. For graph-level tasks, MPNNs further defined a readout function which merged the states of all nodes into graph representations. In this work, ChemNet modifies the message-passing scheme of MPNNs: messages are extended to incorporate geometries of local structures, and states of both atoms and bonds are updated. See Section 3.4 for more details.

Neural Physical Engines Recent studies showed that neural networks are capable of learning annotated (or pseudo) potentials and forces in particle systems, which made fast molecular simulations and protein-folding tasks possible. Notably, it was further shown in [16, 28] that neural networks alone can simulate molecular dynamics for conformation prediction. As an instance,
HamNet [16] proposed a neural physical engine that operated on a generalized space, where positions and momentums of atoms were defined as high-dimensional vectors. In the engine, atoms moved following Hamiltonian Equations with parameterized kinetic, potential and dissipation functions. PhysNet in our work is a similar engine. Nevertheless, instead of learning parameterized energies and calculating their negative derivatives as forces, we directly parameterize the forces between each pair of atoms. In addition, HamNet considered gravitations and repulsions of molecules based on implicit positions, while it ignored the effects of chemical interactions: for example, the types of chemical bonds were ignored in the energy functions. PhysChem fixes this issue via the cooperation mechanism between two specialist networks. Specifically, PhysNet takes chemical expertise (the bond states) from ChemNet and introduces torsion forces, i.e. forces that origin from torsions in chemical bonds, into the dynamics. See Section 3.3 for more details.

**Multi-Task Learning and Model Fusion** The cooperation mechanism in PhysChem shares similar motivations with multi-task learning and model fusion. Multi-task learning [40, 6] is now almost universally used in deep learning models. Representations are shared among a collection of related tasks in order to learn the common ideas. Model fusion, on the other hand, merges different models on identical tasks to improve performances [24, 13]. Notably, these techniques have been previously applied to molecular tasks [31, 13, 32]. In PhysChem, conformation learning and property prediction tasks are jointly trained, with two specialist networks fused to achieve better performances. Nevertheless, the cooperation mechanism in PhysChem roots from observations in physics and chemistry, and enjoys better interpretability than straight-forward ensemble strategies. The advantages of the cooperation mechanism over multi-task strategies are also empirically shown in Section 4.

### 3 Proposed Method: PhysChem

#### 3.1 Preliminaries

**Notations** In the statements and equations below, we use italic letters for scalars and indices, bold lower-case letters for (column) vectors, bold upper-case letters for matrices, calligraphic letters for sets, and normal letters for annotations. Common neural network layers are directly referred to: FC ($x$) denotes a fully-connected layer with input $x$; GCN ($A, X$) denotes a (vanilla) graph convolutional network (GCN)[14] with adjacency matrix $A$ and feature matrix $X$; GRU cell ($s, x$) denotes a cell of the Gated Recurrent Units (GRU) [5] with recurrent state $s$ and input signal $x$; LSTM ($\{x_t\}$) denotes a Long Short-Term Memory network [10] with input signals $\{x_t\}$; MaxPool ($\{x_t\}$) denotes a max-pooling layer with inputs $\{x_t\}$. Exact formulas are available in the appendix. $\oplus$ is the operator of concatenations. $\| \|$ denotes the l_2 norm of the input vector.

**Problem Definition** PhysChem considers molecular representation learning as a supervised learning task. It takes notations of molecules as inputs, conformations and chemical / biomedical properties as supervisions. PhysChem assumes a pre-conducted featurization process, after which a molecule can be denoted as an attributed graph $\mathcal{M} = (\mathcal{V}, \mathcal{E}, n, m, X^v, X^e)$. Here, $\mathcal{V}$ is the set of $n$ atoms, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of $m$ chemical bonds, $X^v \in \mathbb{R}^{n \times d_v} = (x^v_1, \cdots, x^v_n)^\top$ is the matrix of atomic
features, and \( X^c \in \mathbb{R}^{m \times d_e} = (x^c_1, \cdots, x^c_m)^\top \) that of bond features. Based on above inputs, PhysNet outputs the dominant conformations of molecules, and ChemNet outputs the representations as well as the predicted chemical / biomedical properties of molecules.

**Overview** Figure 1 is a sketch of PhysChem. An *initializer* first encodes the inputs into initial atom and bond states (\( v^{(0)} \) and \( e^{(0)} \)) for ChemNet, along with the initial atomic positions and momenta (\( q^{(0)} \) and \( p^{(0)} \)) in PhysNet. Subsequently, \( L \) PhysNet blocks simulate neural molecular dynamics in the generalized space; \( L \) ChemNet blocks conduct geometry-aware message-passing for atoms and bonds. Between each couple of blocks, implicit conformations (qs) and states of chemical bonds (es) are shared as expertise. At the top of PhysNet, a *conformation decoder* transforms implicit atomic positions into the 3D Euclidean space; of ChemNet, a sequence of *readout layers* aggregate atom states into molecular representations, based on which molecular properties are predicted with task-specific layers. Notably, two specialist networks are jointly optimized in PhysChem.

### 3.2 Initializer

Given a molecular graph \( \mathcal{M} = (\mathcal{V}, \mathcal{E}, n, m, X^v, X^c) \), the initializer generates the initial values for both PhysNet and ChemNet variables. In the initializer, we first encode the input features into initial atom and bond states with fully connected layers, i.e.

\[
v^{(0)} = \text{FC} \left( x^v_i \right), \quad e^{(0)} = \text{FC} \left( x^c_{i,j} \right), \quad (i, j) \in \mathcal{E}.
\]

We then adopt the initialization method in [16] to generate initial positions (\( q^{(0)} \)) and momenta (\( p^{(0)} \)) for atoms: a bond-strength adjacency matrix \( A \in \mathbb{R}^{n \times n} \) is estimated with sigmoid-activated FC layers on bond features, according to which a GCN captures the chemical environments of atoms (as \( \bar{v} \)); an LSTM then determines unique positions for atoms, especially for those with identical chemical environments (carbons in benzene, for example). Denoted in formula, the initialization follows

\[
A(i, j) = \begin{cases} 
0, & (i, j) \notin \mathcal{E} \\
\text{FC}_{\text{sigmoid}} \left( x^c_{i,j} \right), & (i, j) \in \mathcal{E},
\end{cases}
\]

\[
\tilde{\mathcal{V}} = \text{GCN} \left( A, V^{(0)} \right),
\]

\[
\left\{ \left( q^{(0)}_i \oplus p^{(0)}_i \right) \right\} = \text{LSTM} \left( \{ \bar{v}_i \} \right), \quad i \in \mathcal{V}
\]

where \( V^{(0)} = (v^{(0)}_1, \cdots, v^{(0)}_n)^\top \) and \( \tilde{\mathcal{V}} = (\tilde{v}_1, \cdots, \tilde{v}_n)^\top \) are the atom states. The order of atoms in the LSTM is specified by the canonical SMILES of the molecule.

### 3.3 PhysNet

Overall, PhysNet simulates the dynamics of atoms in a generalized, \( d_f \)-dimensional space (\( d_f \geq 3 \)). In a molecule as a classic dynamical system, atoms move following Newton’s Second Law:

\[
dq/dt = p/m, \quad dp/dt = f,
\]

where \( q, p \) and \( m \) are the position, momentum and mass of an atom, and \( f \) is the force that the atom experiences. With uniform temporal discretization, the above equations may be approximated with

\[
q_{s+1} = q_s + p_s \tau/m, \quad p_{s+1} = p_s + f_s \tau, \quad s = 0, 1, 2, \ldots
\]

where \( s \) is the index of timestamps and \( \tau \) is the temporal interval. The calculations in PhysNet simulate such a process. Parameters of PhysNet blocks learn to derive the forces \( f \) from intermediate molecular conformations and states of chemical bonds. Correspondingly, two types of forces are modeled in PhysNet, namely the *positional forces* and the *torsion forces*.

**Positional Forces** The positional forces in PhysNet model the gravitations and repulsions between pairs of atoms. In conventional molecular force fields, these forces are generally modeled with (negative) derivatives of *Lennard-Jones potentials*. We therefore propose a similar form for the positional forces taking atomic distances as determinants:

\[
f_{j,i}^{\text{pos}} = (r_{j,i}^{-2} - r_{j,i}^{-1}) m_j m_i d_{j,i}, \quad r_{j,i} = \| \text{FC} \left( q_i - q_j \right) \|, \quad i, j \in \mathcal{V}, \quad i \neq j,
\]

where \( d_{j,i} = \frac{q_i - q_j}{\| q_i - q_j \|} \) is the unitary directional vector from atom \( j \) to \( i \). Instead of using \( l_2 \) distances in the generalized space, we use parameterized *interaction distances* \( r_{j,i} \) to estimate the forces, which
increase the capability of the network. Here, $r^{-2} - r^{-1}$ approximates the landscape of derivatives of Lennard-Jones potentials\textsuperscript{5} with lower-degree polynomials. The approximation is made to avoid numerical issues of using high-degree polynomials, most typically, the gradient explosion.

**Torsion Forces** The torsion forces model the mechanical effects of torsions in local structures. The torsion forces are defined between pairs of atoms that are directly connected by chemical bonds. Estimating local torsions only with positions and momenta of atoms is somehow suboptimal, as the network lacks prior knowledge of chemical characteristics of the bonds (default bond lengths, for example). Alternatively, we leverage the bond states $(\mathbf{e}_{i,j})$ in ChemNet as chemical expertise to derive the torsion forces. Specifically, the torsion forces $f_{\text{tor}}$ are calculated as

$$f_{\text{tor}}^{(i,j)} = w_{j,i} \mathbf{d}_{j,i}, \quad w_{j,i} = \mathbf{C} (\mathbf{e}_{j,i}), \quad (i,j) \in \mathcal{E}. \tag{8}$$

As the torsion forces model chemical interactions, we do not explicitly incorporate atomic mass into the calculation. Notably, atomic information is implicitly considered in the torsion forces, as the bond states integrate atom states at both ends in ChemNet (see the next subsection).

**Dynamics** After estimating the positional and torsion forces, PhysNet simulates the dynamics of atoms following Equation (6). In the $l$-th block of PhysNet, $S$ steps of dynamics are simulated:

$$q_i^{(l,s+1)} = q_i^{(l,s)} + \frac{\tau}{m_i} p_i^{(l,s)}, \quad p_i^{(l,s+1)} = p_i^{(l,s)} + \tau f_i^{(l,s)}, \quad s = 0, 1, \ldots, S - 1. \tag{9}$$

Here, $q_i^{(l,0)} = q_i^{(l)}$, $q_i^{(l+1)} = q_i^{(l,0)}$ ($p$ similarly). $f_i$ is the resultant force on atom $i$, i.e.

$$f_i = \sum_{j \in \mathcal{V}} f_{j,i}^{\text{pos}} + \sum_{(i,j) \in \mathcal{E}} f_{i,j}^{\text{tor}}. \tag{10}$$

### Conformation Decoder and Loss

We use a simple linear transformation to decode implicit atomic positions (and momenta) from the generalized space into the real 3D space:

$$\hat{\mathbf{R}} = \mathbf{Q} \mathbf{W}_{\text{dec}}, \quad \mathbf{W}_{\text{dec}} \in \mathbb{R}^{d_j \times 3}, \tag{11}$$

where $\mathbf{Q} = (q_1, \ldots, q_n)^T \in \mathbb{R}^{n \times d_j}$ is the position matrix in the generalized space, and $\hat{\mathbf{R}} = (\hat{r}_1, \ldots, \hat{r}_n)^T \in \mathbb{R}^{n \times 3}$ that of the predicted 3D conformation. We further propose Conn-$k$ ($k$-hop connectivity loss), a spatially invariant loss that supervises PhysNet based on local distance errors: if $\mathcal{C}^{(k)} \in \{0, 1\}^{n \times n}$ is the $k$-hop connectivity matrix\textsuperscript{6} and $\mathbf{D}$ is the distance matrix\textsuperscript{7}, then the $k$-hop connectivity loss is defined as

$$L_{\text{Conn-}k} (\hat{\mathbf{R}}, \mathbf{R}) = \frac{1}{n} \left\| \hat{\mathbf{C}}^{(k)} \odot \left( \mathbf{D} - \hat{\mathbf{D}} \right) \odot \left( \hat{\mathbf{D}} - \mathbf{D} \right) \right\|_{F}, \tag{12}$$

where $\odot$ is the element-wise product, $\| \cdot \|_{F}$ is the Frobenius norm, $(\mathbf{D}, \hat{\mathbf{D}})$ are distance matrices of the real and predicted conformations $(\mathbf{R}, \hat{\mathbf{R}})$, and $\hat{\mathbf{C}}^{(k)}$ is the normalized $k$-hop connectivity matrix\textsuperscript{7}. The total loss of PhysNet is defined as the weighted sum of Conn-3 losses on all timestamps:

$$L_{\text{phys}} = \frac{1}{Z} \sum_{l,s} \eta L_{\text{Conn-}3} (\mathbf{R}^{(l,s)}), \quad l = 1, 2, \ldots, L, \quad s = 1, 2, \ldots, S, \tag{13}$$

where $\eta < 1$ is a decay factor and $Z = 0.01$ is an empirical normalization factor.

### 3.4 ChemNet

ChemNet conducts message-passing for both atoms and chemical bonds. Besides generating messages with atom states, ChemNet also considers local geometries including bond lengths and angles to adequately characterize local chemical environments. Specifically, in each ChemNet block, triplet descriptors are established in the atomic neighborhoods. Atoms merge relevant triplet descriptors to

\textsuperscript{5}$U^{1-L} = ar^{-12} - br^{-6}$, $f^{1-L} = \frac{\partial U^{1-L}}{\partial r} = \left( 12a r^{-11} - 6br^{-5} \right) \frac{4}{a r}$
\textsuperscript{6}$\mathcal{C}_{i,j}^{(k)} = 1$ if and only if atoms $i$ and $j$ are $k$- or less-hop connected on the molecular graph.
\textsuperscript{7}$\mathbf{D}_{i,j} = \| \mathbf{r}_i - \mathbf{r}_j \|$, where $\mathbf{r} \in \mathbb{R}^3$ are 3D coordinates of atoms.
\textsuperscript{8}$\hat{\mathbf{C}}^{(k)} = \mathbf{L}^{-1/2} \mathcal{C}^{(k)} \mathbf{L}^{-1/2}$, where $\mathbf{L}$ is the diagonal degree matrix with $L_{i,i} = \sum_j \mathcal{C}_{i,j}^{(k)}$. 

\[5\]
generate messages, and then aggregate received messages to update their states. Chemical bonds also update their states by aggregating the states of atoms at both ends.

**Triplet Descriptor** The triplet descriptors \( t_{i,j,k} \) are descriptive vectors defined on all atom triplets \((i, j, k)\) with \((i, j) \in \mathcal{E} \land (i, k) \in \mathcal{E} \):

\[
l_{i,j,k}^{(l)} = \text{FC} \left( v_i^{(l)} \oplus v_j^{(l)} \oplus v_k^{(l)} \oplus l_{i,j}^{(l)} \oplus l_{i,k}^{(l)} \oplus a_{j,i,k}^{(l)} \right), \quad j, k \in \mathcal{N}(i),
\]

where \( \mathcal{N}(i) = \{ j \mid (i, j) \in \mathcal{E} \} \) denotes the atomic neighborhood of \( i \), \( v_i \) are atom states, and \( l_{i,j} = \text{FC} (l_{i,j}) \) and \( a_{j,i,k} = \text{FC} (\cos \langle z_{j,i,k} \rangle) \) are representations of bond lengths and bond angles. The motivation of constructing triplet descriptors is that these features are geometrically deterministic and invariant: i) all bond lengths and angles in an atomic neighborhood together compose a minimum set of variables that uniquely determine the geometry of the neighborhood (deterministic); ii) these features all enjoy translational and rotational invariances (invariant). When PhysNet and ChemNet are jointly optimized, \( l_{i,j}^{(l)} \) and \( z_{j,i,k}^{(l)} \) are calculated from the intermediate 3D conformation in the \( l \)-th PhysNet block; when real conformations of target molecules are available, these values can be replaced by the ground-truths (e.g. ChemNet (real conf.) in Section 4).

**Message-passing** After establishing the triplet descriptors, ChemNet generates atomic messages by merging relevant descriptors. The message from atom \( j \) to \( i \) in the \( l \)-th block is calculated as

\[
m_{i,j}^{(l)} = \text{MaxPool} \left( \left\{ t_{i,j,k}^{(l)} \mid k \in \mathcal{N}(i) \right\} \right), \quad j \in \mathcal{N}(i), \quad l = 1, 2, \cdots, L.
\]

Subsequently, ChemNet utilizes a similar architecture to [37] to conduct message passing. Centric atoms aggregate the received messages with attention scores determined by the bond states:

\[
m_i^{(l)} = \sum_{j \in \mathcal{N}(i)} \alpha_{i,j}^{(l)} m_{i,j}^{(l)}, \quad \left\{ \alpha_{i,j}^{(l)} \mid j \in \mathcal{N}(i) \right\} = \text{softmax} \left( \left\{ \text{FC} \left( e_{i,j}^{(l)} \right) \mid j \in \mathcal{N}(i) \right\} \right).
\]

Atom states are then updated with GRU cells that take previous states as recurrent states. A similar process is then conducted for all chemical bonds, where messages are generated by the states of atoms at both ends. Denoted in formula,

\[
v_i^{(l+1)} = \text{GRU}_{\text{cell}} \left( v_i^{(l)}, m_i^{(l)} \right), \quad e_{i,j}^{(l+1)} = \text{GRU}_{\text{cell}} \left( e_{i,j}^{(l)}, \text{FC} \left( v_i^{(l+1)} \oplus v_j^{(l+1)} \right) \right).
\]

**Representation Readout** The molecular representation is finally read out with \( T \) global attentive layers [37], where a virtual *meta-atom* continuously collects messages from all atoms in the molecule and updates its state. We initialize and update the state of the meta-atom \( v_{\text{meta}}^{(t)} \) following:

\[
m_{\text{meta}}^{(t)} = \sum_{i \in \mathcal{V}} \alpha_{i}^{(t)} v_i^{(L)}, \quad \left\{ \alpha_{i}^{(t)} \mid i \in \mathcal{V} \right\} = \text{softmax} \left( \left\{ \text{FC} \left( v_{\text{meta}}^{(t)} \oplus v_i^{(L)} \right) \mid i \in \mathcal{V} \right\} \right),
\]

\[
v_{\text{meta}}^{(t)} = \text{FC} \left( \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} v_i^{(L)} \right), \quad v_{\text{meta}}^{(t+1)} = \text{GRU}_{\text{cell}} \left( v_{\text{meta}}^{(t)}, m_{\text{meta}}^{(t)} \right), \quad t = 1, 2, \cdots, T.
\]

The final state of the meta-atom is used as the molecular representation. Labeled chemical and/or biomedical properties are used to supervise the representations, and the loss of ChemNet, \( L_{\text{chem}} \), is determined task-specifically. The total loss of PhysChem is the weighted sum of losses in the two networks controlled by a hyperparameter \( \lambda \), i.e.

\[
L_{\text{total}} = \lambda L_{\text{phys}} + L_{\text{chem}}.
\]

4 **Experiments**

4.1 **Experimental Setup**

**Datasets and Featurization** We evaluated PhysChem on quantum mechanics (QM7, QM8 and QM9) and physical chemistry (IPOP, FREESOLV and ESOL) datasets, as well as drug effectiveness datasets of the notorious SARS-CoV-2. Statistics of the datasets are in Table 1. QM7, QM8 and QM9 contain stable organic molecules with up to 7, 8 or 9 heavy atoms. 3D atomic coordinates as
Table 1: Statistics of the datasets used in this paper.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>QM7</th>
<th>QM8</th>
<th>QM9</th>
<th>LIPOP</th>
<th>FREESolV</th>
<th>ESOL</th>
<th>COVID19</th>
</tr>
</thead>
<tbody>
<tr>
<td># molecules</td>
<td>7,160</td>
<td>21,786</td>
<td>133,885</td>
<td>4,200</td>
<td>642</td>
<td>1,128</td>
<td>14,332</td>
</tr>
<tr>
<td># targets</td>
<td>1</td>
<td>16</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task type</th>
<th>Category</th>
<th>With conformation?</th>
<th>Regression</th>
<th>Property</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quantum Mechanics</td>
<td>Yes</td>
<td></td>
<td>Physical Chemistry</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Physical Chemistry</td>
<td>No</td>
<td></td>
<td></td>
<td>No</td>
</tr>
</tbody>
</table>

Baselines For conformation learning tasks, we compared PhysChem with i) a Distance Geometry (DG) method tuned with the Universal Force Fields (UFF) [22], which was implemented in the RDKit package [2] and thus referred to as RDKit; ii) CVGAE and CVGAE+UFF [18], which learned to generate low-energy molecular conformations with deep generative graph neural networks (either with or without UFF tuning); iii) HamEng [16], which learned stable conformations via simulating Hamiltonian mechanics with neural physical engines. For property prediction tasks, we compared PhysChem with i) MoleculeNet, for which we reported the best performances achieved by methods collected in [36] (before 2017); ii) 3DGCN [4], which augmented conventional GCNs with input bond geometries; iii) DimeNet [15], which conducted directional message-passing by representing pairs of atoms; iv) Attentive FP [37], which used local and global attentive layers to derive molecular representations; and v) CMPNN [30], which used communicative kernels to conduct deep message-passing. We conducted experiments with official implementations of HamEng, CVGAE, Attentive FP and CMPNN; for other baselines, as identical evaluation schemes were adopted, we referred to the reported performances in corresponding citations and left unreported entries blank.

PhysChem Variants We also compared PhysChem with several variants, including i) PhysNet (s.a.), a stand-alone PhysNet that ignored all chemical expertise by setting \( e_{j,i} = 0 \); ii) ChemNet (s.a.), a stand-alone ChemNet (ChemNet (s.a.)) that ignored all physical expertise by equally setting all bond lengths and angles \( l_{i,j} = l, a_{j,i,k} = a \); iii) ChemNet (real conf.) and ChemNet (rdkit conf.), two ChemNet variants that leveraged \( l_{i,j} \) and \( a_{j,i,k} \) in real conformations (real conf.) or RDKit-generated conformations (rdkit conf.); and iv) ChemNet (m.t.), a multi-task ChemNet variant that used a straight-forward multi-task strategy for conformation learning and property prediction tasks: we applied the conformation decoder and loss on atom states \( u \) and optimized ChemNet with the weighted sum of losses of both tasks (i.e. \( L_{total} \) in Equation (19)).

Implementation and Training Details Unless otherwise specified, we used \( L = 2 \) pairs of blocks for PhysChem. In the initializer, we used a 2-layer GCN and a 2-layer LSTM. In each PhysNet block, we set \( d_j = 3, S = 4 \) and \( \tau = 0.25 \). In each ChemNet block, we set the dimensions of atom and bond states as 128 and 64, correspondingly. In the representation readout block, we used \( T = 1 \) global attentive layers with 256-dimensional meta-atom states. For property prediction tasks with multiple targets, the targets were first standardized and then fitted simultaneously. We use the mean-square-error (MSE) loss for all regression tasks and the cross-entropy loss for all classification tasks to train the models. Other implementation details, such as hyperparameters of baselines, are provided in the appendix.

---

9The datasets are publicly available at [http://moleculenet.ai/datasets-1](http://moleculenet.ai/datasets-1).
10The datasets are available at [https://opendata.ncats.nih.gov/covid19/](https://opendata.ncats.nih.gov/covid19/) (CC BY 4.0 license) and are continuously extended. The data used in our experiments were downloaded on February 16th, 2021.
11We use the 2020.03.1.0 version of the RDKit package at [http://www.rdkit.org/](http://www.rdkit.org/).
### 4.2 Results

#### Quantum Mechanical Datasets

As real conformations are available in the QM datasets, we evaluated PhysChem on both conformation learning and property prediction tasks. On conformation learning tasks, a Distance MSE metric is reported, which sums the squared errors of all pair-wise distances in each molecule, normalizes it by the number of atoms, and then takes averages across molecules. Note that this metric is equivalent to the Conn-$\infty$ loss with $k = \infty$ in Equation (12). On property prediction tasks, the mean-absolute-errors (MAE) for regression targets are reported. When multiple targets are requested (on QM8 and QM9), we report the Multi-MAE metric in [16] which sums the MAEs of all targets (standardized for QM9). Table 2 and 3 show the results. On conformation learning tasks, PhysNet (s.a.) displays significant advantages on learning conformations of small molecules. Specifically, the comparison between PhysNet (s.a.) and HamEng indicates that directly learning forces in neural physical engines may be superior to learning energies and their derivatives. With massive data samples (QM9), the specialist, PhysNet (s.a.), obtains better results than PhysChem; while on datasets with limited samples (QM7), chemical expertise demonstrates its effectiveness. On property prediction tasks, PhysChem obtains state-of-the-art performances on QM datasets. The comparison between PhysChem, ChemNet (s.a.) and ChemNet (m.t.) shows that the elaborated cooperation mechanism in PhysChem is necessary, as the straight-forward multi-task strategy leads to severe negative transfer. In addition, the results of ChemNet (real conf.) and ChemNet (rdkit conf.) show that leveraging real (or generated, geometrically correct) conformations of test molecules indeed helps on some datasets, while the elevations are somehow insignificant.

#### Physical Chemical & Biomedical Datasets

On LIPOP, FREE SOLV, ESOL and COVID19 with no labeled conformations, we used RDKit-generated conformations to satisfy models that requested conformations of training and / or test molecules. Although these generated conformations are less accurate, distance geometries in local structures are generally correctly displayed. We report the rooted-mean-squared-errors (RMSE) for regression tasks, and the multi-class ROC-AUC (Multi-AUC) metric on COVID19. Table 4 shows the results. PhysChem again displays state-of-the-art performances. Notably, as the numbers of samples in physical chemical datasets (LIPOP, FREE SOLV...)}
Table 4: Performances of property prediction on Lipophilicity, FreeSolv, ESOL, and COVID19.

<table>
<thead>
<tr>
<th>Dataset Metric</th>
<th>Lipophilicity</th>
<th>FreeSolv</th>
<th>ESOL</th>
<th>SARS-CoV-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lipophilicity</td>
<td>FreeSolv</td>
<td>ESOL</td>
<td>SARS-CoV-2</td>
</tr>
<tr>
<td>w/o conf.</td>
<td>0.655</td>
<td>1.150</td>
<td>0.580</td>
<td>–</td>
</tr>
<tr>
<td>MoleculeNet</td>
<td>0.589±0.036</td>
<td>0.962±0.197</td>
<td>0.612±0.027</td>
<td>0.695±0.009</td>
</tr>
<tr>
<td>Attentive FP</td>
<td>–</td>
<td>0.808±0.129</td>
<td>0.547±0.011</td>
<td>0.692±0.007</td>
</tr>
<tr>
<td>CMPNN</td>
<td>0.577±0.016</td>
<td>0.730±0.227</td>
<td>0.561±0.035</td>
<td>0.757±0.012</td>
</tr>
<tr>
<td>PhysChem</td>
<td>0.568±0.014</td>
<td>0.692±0.112</td>
<td>0.499±0.025</td>
<td>0.792±0.013</td>
</tr>
<tr>
<td>w/ conf.</td>
<td>–</td>
<td>0.824±0.014</td>
<td>0.580±0.069</td>
<td>–</td>
</tr>
<tr>
<td>3DGCN</td>
<td>0.557±0.014</td>
<td>0.731±0.024</td>
<td>0.504±0.016</td>
<td>–</td>
</tr>
<tr>
<td>HamNet</td>
<td>0.532±0.011</td>
<td>0.663±0.029</td>
<td>0.504±0.023</td>
<td>0.795±0.010</td>
</tr>
<tr>
<td>ChemNet (rdkit conf.)</td>
<td>0.557±0.014</td>
<td>0.731±0.024</td>
<td>0.504±0.016</td>
<td>–</td>
</tr>
</tbody>
</table>

Figure 2: Further analyses of PhysChem.

and ESOL) are sparse, the conformation learning task for PhysNet is comparatively tough. This leads to a larger gap of performances between PhysChem and ChemNet (rdkit conf.), yet the former is still better than most baselines. The largest elevation of performances is observed on COVID19, which indicates that PhysChem is further competent on more complicated, structure-dependent tasks.

**Further Analyses** Figure 2(a) shows the effect of the loss hyperparameter $\lambda$ on QM9. When $\lambda \leq 0.1$, increasing $\lambda$ benefits both tasks. This indicates that aiding the learning of conformations also helps to predict chemical properties. With larger $\lambda$, the property prediction task is then compromised. Figure 2(b) visualizes the predicted conformations of baselines and PhysNet blocks. Local structures such as bond lengths, angles and planarity of aromatic groups are better preserved by PhysChem.

# 5 Conclusion and Future Work

In this paper, we propose a novel neural architecture, PhysChem, that learns molecular representations via fusing physical and chemical information. Conformation learning and property prediction tasks are jointly trained in this architecture. Beyond straight-forward multi-task strategies, PhysChem adopts an elaborated cooperation mechanism between two specialist networks. State-of-the-art performances were achieved on MoleculeNet and SARS-CoV-2 datasets. Nevertheless, there is still much space for advancement of PhysChem. For future work, a straight-forward improvement is to enlarge the capability of the model by further simplifying as well as deepening the architecture of PhysChem. In addition, proposing strategies to train PhysChem with massive unlabeled molecules is yet another promising direction.

**Broader Impact** For the machine learning community, our work proposes a more interpretable architecture on molecular machine learning tasks and demonstrates its effectiveness. We hope that the specialist networks and domain-related cooperation mechanism in PhysChem will inspire researchers in a wider area of deep learning to develop novel architectures under the same motivation. For the drug discovery community, PhysChem leads to direct applications on ligand-related tasks including conformation and property prediction, protein-ligand binding affinity prediction, etc. With limited possibility, the abuse of PhysChem in drug discovery may violate some ethics of life science.

**Acknowledgments** This work was supported by the National Natural Science Foundation of China (Grant No. 61876006).
References


A Neural Network Layers

We hereby specify the detailed formulas of the neural network layers used in this paper.

- Fully-Connected Layers

\[ FC(x) = \sigma(Wx) \]  

(20)

- Graph Convolutional Networks (with two layers)

\[ GCN(A, X) = \sigma(A\sigma(AXW_1)W_2) \]  

(21)

- Cells of Gated Recurrent Units

\[ r = \text{sigmoid}(W_r(s \oplus x)) \]  

(22)

\[ z = \text{sigmoid}(W_z(s \oplus x)) \]  

(23)

\[ h = \tanh(W_h((r \odot s) \oplus x)) \]  

(24)

\[ \text{GRU}_{\text{cell}}(s, x) = (1 - z) \odot s + z \odot h \]  

(25)
• Long Short-Term Memory (LSTM) (with one layer)

\[ h_0 \text{ and } c_0 \text{ randomly initialized; } \]

\[ z = \text{tanh}(Wz(h_{t-1} \oplus x_t)) \quad (26) \]
\[ z^i = \text{sigmoid}(W_i(h_{t-1} \oplus x_t)) \quad (27) \]
\[ z^o = \text{sigmoid}(W_o(h_{t-1} \oplus x_t)) \quad (28) \]
\[ z^f = \text{sigmoid}(W_f(h_{t-1} \oplus x_t)) \quad (29) \]
\[ c_t = z^f \odot c_{t-1} + z^i \odot z \quad (30) \]
\[ h_t = z^o \odot \text{tanh}(c_t) \quad (31) \]
\[ \text{LSTM} \left( \{x_t\} \right) = \{y_t \mid y_t = \text{FC}(h_t)\} \quad (32) \]

\[ \text{Max Pooling with } x_t \in \mathbb{R}^D \]

\[ \text{MaxPool} \left( \{x_t\} \right)[i] = \max_{x_s \in \{x_t\}} x_s[i], \quad d = 1, 2, \ldots, D \quad (33) \]

B Detailed Algorithms

We hereby formalize the algorithms of the initializer, PhysNet blocks and ChemNet blocks.

**Algorithm 1 Initializer**

**Input:** Molecular Graph \( \mathcal{M} = (\mathcal{V}, \mathcal{E}, n, m, X^v, X^e) \)

**Output:** \( v^{(0)}, e^{(0)}, q^{(0)}, p^{(0)} \)

1: Encode atom and bond states

\[ v_i^{(0)} = \text{FC}(x_i^v), \quad i \in \mathcal{V} \quad e_{i,j}^{(0)} = \text{FC}(x_{i,j}^e), \quad (i, j) \in \mathcal{E} \]

2: Encode atomic chemical environments with GCNs

\[ A(i, j) = \begin{cases} 0, & (i, j) \notin \mathcal{E} \\ \text{FC}_{\text{sigmoid}}(x_{i,j}^e), & (i, j) \in \mathcal{E} \end{cases}, \quad \tilde{V} = \text{GCN} \left( A, V^{(0)} \right), \]

3: Derive initial atomic positions / momenta

\[ \left\{ \left( q_i^{(0)} \oplus p_i^{(0)} \right) \right\} = \text{LSTM} \left( \{\tilde{v}_i\} \right), \quad i \in \mathcal{V} \]
Algorithm 2 PhysNet block

**Input:** atomic positions & momenta \((q, p)\), atomic mass \((m)\), bond states \((e)\)

**Output:** new atomic positions & momenta \((q, p)\)

1: for \(i \in V\) do
2: for \(j \in V \land j \neq i\) do
3: \(r_{j,i} \leftarrow \|FC(q_i - q_j)\|
4: \(d_{j,i} \leftarrow (q_i - q_j)/\|q_i - q_j\|
5: f_{j,i}^{\text{pos}} \leftarrow (r_{j,i} - 2d_{j,i} - r_{j,i}^{1})m_{j}m_{i}d_{j,i}
6: end for
7: for \(j \in N(i)\) do
8: \(w_{j,i} \leftarrow FC(e_{j,i})
9: f_{j,i}^{\text{for}} \leftarrow w_{j,i}d_{j,i}
10: end for
11: \(f_{i} \leftarrow \sum_{j \in V \land j \neq i} f_{j,i}^{\text{pos}} + \sum_{j \in N(i)} f_{j,i}^{\text{for}}
12: end for
13: for \(i \in V\) do
14: \(q_{i} \leftarrow q_{i} + \tau p_{i}/m_{i}
15: p_{i} \leftarrow p_{i} + \tau f_{i}
16: end for

Algorithm 3 ChemNet block (in combination with PhysNet)

**Input:** atom & bond states \((v, e)\), atomic positions \(q\)

**Output:** new atom & bond states \((v, e)\)

1: for \(i \in V\) do
2: for \(j \in N(i)\) do
3: \(r_{j,i} \leftarrow q_{i} - q_{j}, \ l_{i,j} \leftarrow \|r_{j,i}\|, \ l_{i,j} \leftarrow FC(l_{i,j})
4: end for
5: for \(j, k \in N(i) \land j \neq k\) do
6: \(\cos(a_{j,i,k}) \leftarrow r_{j,i} \cdot r_{k,i}
7: a_{j,i,k} \leftarrow FC(\cos(a_{j,i,k}))
8: t_{i,j,k} \leftarrow FC \left( v_{i}^{(l)} \oplus v_{j}^{(l)} \oplus v_{k}^{(l)} \oplus i_{i,j}^{(l)} \oplus i_{j,k}^{(l)} \oplus a_{j,i,k} \right)
9: end for
10: for \(j \in N(i)\) do
11: \(m_{i,j} \leftarrow \text{MaxPool}(\{t_{i,j,k} \mid k \in N(i) \land k \neq j\})
12: end for
13: \( \{e_{i,j} \mid j \in N(i)\} \leftarrow \text{softmax}(\{FC(e_{i,j}) \mid j \in N(i)\})
14: m_{i} \leftarrow \sum_{j \in N(i)} \alpha_{i,j}m_{i,j}
15: v_{i} \leftarrow \text{GRU}_{\text{cell}}(v_{i}, m_{i})
16: end for
17: for \((i, j) \in E\) do
18: \(m_{i,j}^{\epsilon} \leftarrow FC(v_{i} \oplus v_{j})
19: e_{i,j} \leftarrow \text{GRU}_{\text{cell}}(e_{i,j}, m_{i,j}^{\epsilon})
20: end for

C Data Licenses

- MoleculeNet datasets (QM7, QM8, QM9, LIPOP, FREE SOLV, ESOL) are licensed under an MIT license. See [http://moleculenet.ai/datasets-1](http://moleculenet.ai/datasets-1).

- SARS-CoV-2 datasets (COVID19) are licensed under a CC BY 4.0 license, 2020. See [https://opendata.ncats.nih.gov/covid19/](https://opendata.ncats.nih.gov/covid19/)
D Preprocessing COVID19

We downloaded data in COVID19 datasets on February 16th, 2021. 13 assays of 14,332 drugs were used as targets. Note that not all assays on all drugs were available, so we applied masks on unavailable entries in both training and evaluation stages. The 13 assays we analysed were

- (a) 3CL enzymatic activity;
- (b) ACE2 enzymatic activity;
- (c) HEK293 cell line toxicity;
- (d) Human fibroblast toxicity;
- (e) MERS Pseudotyped particle entry;
- (f) MERS Pseudotyped particle entry (Huh7 tox counterscreen);
- (g) SARS-CoV Pseudotyped particle entry;
- (h) SARS-CoV Pseudotyped particle entry (VeroE6 tox counterscreen);
- (i) SARS-CoV-2 cytopathic effect (CPE);
- (j) SARS-CoV-2 cytopathic effect (host tox counterscreen);
- (k) Spike-ACE2 protein-protein interaction (AlphaLISA);
- (l) Spike-ACE2 protein-protein interaction (TruHit Counterscreen);
- (m) TMPRSS2 enzymatic activity.

We preprocessed regression targets into classification ones: we partitioned the potency indices into bins as \{inactive, \[10, \infty\), \[1, 10\), \(0, 1\)\}, i.e. \{inactive, low potency, medium potency, high potency\} (the smaller index indicates stronger potency), and predicted which bin the indices were in. Correspondingly, predictions on all assays were transformed into 4-class classification tasks. Distribution on the classes are shown in Figure S3.

![Figure S3: Distributions of classes of 13 assays on COVID19.](image-url)
E  Featurization

We used the identical featurization process to [37]. Details are included in Table S5.

Table S5: Featurization details (Table from [37].)

<table>
<thead>
<tr>
<th>Atom Features</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>atom symbol</td>
<td>16</td>
<td>[B, C, N, O, F, Si, P, S, Cl, As, Se, Br, Te, I, At, metal] (one-hot)</td>
</tr>
<tr>
<td>degree</td>
<td>6</td>
<td>number of covalent bonds [0,1,2,3,4,5] (one-hot)</td>
</tr>
<tr>
<td>formal charge</td>
<td>1</td>
<td>electrical charge (integer)</td>
</tr>
<tr>
<td>radical electrons</td>
<td>1</td>
<td>number of radical electrons (integer)</td>
</tr>
<tr>
<td>hybridization</td>
<td>6</td>
<td>[sp, sp2, sp3, sp3d, sp3d2, other] (one-hot)</td>
</tr>
<tr>
<td>aromaticity</td>
<td>1</td>
<td>whether the atom is part of an aromatic system [0/1] (one-hot)</td>
</tr>
<tr>
<td>hydrogens</td>
<td>5</td>
<td>number of connected hydrogens [0,1,2,3,4] (one-hot)</td>
</tr>
<tr>
<td>chirality</td>
<td>1</td>
<td>whether the atom is chiral center [0/1] (one-hot)</td>
</tr>
<tr>
<td>chirality type</td>
<td>2</td>
<td>[R, S] (one-hot)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bond Features</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond type</td>
<td>4</td>
<td>[single, double, triple, aromatic] (one-hot)</td>
</tr>
<tr>
<td>conjugation</td>
<td>1</td>
<td>whether the bond is conjugated [0/1] (one-hot)</td>
</tr>
<tr>
<td>ring</td>
<td>1</td>
<td>whether the bond is in ring [0/1] (one-hot)</td>
</tr>
<tr>
<td>stereo</td>
<td>4</td>
<td>[StereoNone, StereoAny, StereoZ, StereoE] (one-hot)</td>
</tr>
</tbody>
</table>

F  Baselines

For property prediction tasks, italic results of baselines (3DGCN [4], CMPNN [30], DimeNet [15], and HamNet [16]) in the tables in Section 4 were directly took from corresponding references. As MoleculeNet [36] suggested standard training and evaluation schemes for all collected datasets (including QM7, QM8, QM9, LiPOP, FREEsolv, ESOL), to which all baselines conformed, the performances were comparable. For these baselines, we left untested entries in the tables blank. For Attentive FP [37], as the paper did not report the standard errors of its performances, we conduct experiments on its official implementation [12] with the exact hyperparameters included in its appendix. For COVID-19 datasets, we tested the official implementations of Attentive FP and CMPNN [13]. Suggested parameters were used for these baselines. We also reported conformation learning results of RDKit, HamEng [16] and CVGAE [18]. We referred to the official implementations of RDKit [14] and HamEng [15]. As there were some errors in the official code of CVGAE, we reimplemented it strictly following details in the paper. We used the default parameters suggested in the papers.

G  Hyperparameters

Table S6 shows the hyperparameters of PhysChem on different datasets. As similar architectures were used, most parameters of ChemNet and training in PhysChem are borrowed from corresponding hyperparameters in Attentive FP [37] (on corresponding datasets): for example, the dimensionalities of atom states and the learning rates. Despite the modification of structures, we found that these setups worked so we did not tune these hyperparameters.

Notably, we used $d_f = 3$ in our experiment. Empirically, setting $d_f = 3$ already leads to promising results. Using $d_f \gg 3$ may slightly improve the performances of conformation learning at the compromise of model efficiency. We show the effect of $d_f$ in Section K.

12 https://github.com//OpenDrugAI/AttentiveFP
13 https://github.com//SY575/CMPNN
14 https://github.com//PKUterran/HamNet
15 Version 2020.03.1.0 at http://www.rdkit.org/
16 https://github.com//PKUterran/HamNet
17 We are truly sorry to cause confusions by stating in the main paper that $d_f \gg 3$ and that we used $d_f = 128$ by default. We will correct this issue in the future versions of our paper.
### Table S6: Hyperparameters of PhysChem on all datasets.

<table>
<thead>
<tr>
<th>Module</th>
<th>Hyperparameter</th>
<th>QM7</th>
<th>QM8</th>
<th>QM9</th>
<th>Lipop</th>
<th>F.S.</th>
<th>ESOL</th>
<th>COVID19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>dim of atom features (a^)</td>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of bond features (a^e)</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Int.</td>
<td>#layer of the GCN</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>#layer of the LSTM</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of GCN hidden layers</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of LSTM hidden layers</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PhysNet</td>
<td>#blocks (L)</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of (q, p (d_r))</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of interaction dist.</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>#steps (S)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>interval (\tau)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0.25</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mass (m)</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>loss decay factor (\eta)</td>
<td>0.625</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>normalization factor (Z)</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChemNet</td>
<td>#blocks (L)</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of atom states (v)</td>
<td>200</td>
<td>200</td>
<td>128</td>
<td>256</td>
<td>160</td>
<td>120</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>dim of bond states (e)</td>
<td>100</td>
<td>100</td>
<td>64</td>
<td>256</td>
<td>160</td>
<td>120</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>dim of bond length repr.(I)</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of bond angle repr. (a)</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of atom messages (v_m)</td>
<td>200</td>
<td>200</td>
<td>128</td>
<td>256</td>
<td>160</td>
<td>120</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>dim of bond messages (e_m)</td>
<td>100</td>
<td>100</td>
<td>64</td>
<td>256</td>
<td>160</td>
<td>120</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>#layers of readout fn. (T)</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim of meta-atom states (v_{meta})</td>
<td>300</td>
<td>300</td>
<td>256</td>
<td>256</td>
<td>160</td>
<td>120</td>
<td>200</td>
</tr>
<tr>
<td>Training</td>
<td>loss param. (\lambda)</td>
<td>0.01</td>
<td>0.1</td>
<td>1</td>
<td>0.1</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>learning rate</td>
<td>1e-4</td>
<td>1e-5</td>
<td>2e-6</td>
<td>1e-4</td>
<td>3e-3</td>
<td>3e-3</td>
<td>1e-4</td>
</tr>
<tr>
<td></td>
<td>exponential decay</td>
<td>0.95</td>
<td>0.95</td>
<td>0.995</td>
<td>0.995</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>batch size</td>
<td>100</td>
<td>81</td>
<td>20</td>
<td>32</td>
<td>128</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>epoch</td>
<td>100</td>
<td>100</td>
<td>300</td>
<td>400</td>
<td>800</td>
<td>800</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>dropout rate</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>weight decay</td>
<td>1e-6</td>
<td>1e-6</td>
<td>1e-5</td>
<td>1e-5</td>
<td>1e-5</td>
<td>1e-5</td>
<td>1e-5</td>
</tr>
<tr>
<td></td>
<td>batch normalization</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### H Additional Metric on Conformation Learning

Compared to distance-based metric, traditional RMSD is subjected to a optimal alignment between predicted and reference conformations, so it can’t smoothly measure their discrepancy when there are poorly predicted regions in the conformation [19]. However, to further justify the results of our model, we still evaluate RMSD of several baselines and results are list in Table H.

Table S7: RMSD and distance-based metric for baselines on QM9 conformation learning, with the same settings as Table 2

<table>
<thead>
<tr>
<th>Baseline</th>
<th>RMSD</th>
<th>Distance MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Guess</td>
<td>1.972</td>
<td>2.799</td>
</tr>
<tr>
<td>RDKit</td>
<td>0.805</td>
<td>1.920</td>
</tr>
<tr>
<td>CVGAE</td>
<td>1.364</td>
<td>1.052</td>
</tr>
<tr>
<td>HamEng</td>
<td>0.826</td>
<td>0.418</td>
</tr>
<tr>
<td>PhysChem</td>
<td>0.732</td>
<td>0.255</td>
</tr>
<tr>
<td>PhysNet (s.a.)</td>
<td>0.709</td>
<td>0.197</td>
</tr>
<tr>
<td>PhysNet (s.a.) (Mixed Loss)</td>
<td><strong>0.483</strong></td>
<td>0.218</td>
</tr>
</tbody>
</table>

Here, Mixed Loss indicates that we used a combination of RMSD loss and Conn-3 loss following [16].

### I Individual MAEs for the separate tasks in QM9

We evaluate PhysChem, ChemNet (s.a.) and ChemNet (real conf.) on 12 separate tasks on QM9, and results are listed in Table I.

### J Computational Resources & Efficiency

The training and inferences of PhysChem and the baselines were conducted on a total of 8 NVIDIA Tesla P100 GPUs. We recorded the running time of PhysChem and baselines including CVGAE [13].
Table S8: Individual MAEs for the separate tasks in QM9.

<table>
<thead>
<tr>
<th>Target</th>
<th>Unit</th>
<th>std. dv.</th>
<th>ChemNet (s.a.)</th>
<th>PhysChem</th>
<th>ChemNet (real conf.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mu)</td>
<td>(D)</td>
<td>1.532</td>
<td>0.524</td>
<td>0.537</td>
<td>0.461</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>(b^3)</td>
<td>8.196</td>
<td>0.617</td>
<td>0.511</td>
<td>0.438</td>
</tr>
<tr>
<td>(\epsilon_{\text{HOMO}})</td>
<td>hartree</td>
<td>0.022</td>
<td>0.00425</td>
<td>0.00401</td>
<td>0.00370</td>
</tr>
<tr>
<td>(\epsilon_{\text{LUMO}})</td>
<td>hartree</td>
<td>0.047</td>
<td>0.00468</td>
<td>0.00434</td>
<td>0.00395</td>
</tr>
<tr>
<td>(\Delta_{\epsilon})</td>
<td>hartree</td>
<td>0.048</td>
<td>0.00599</td>
<td>0.00564</td>
<td>0.00512</td>
</tr>
<tr>
<td>(\langle R^2\rangle)</td>
<td>(b^2)</td>
<td>279.942</td>
<td>33.930</td>
<td>29.877</td>
<td>25.390</td>
</tr>
<tr>
<td>ZPVE</td>
<td>hartree</td>
<td>0.033</td>
<td>0.00132</td>
<td>0.00120</td>
<td>0.00109</td>
</tr>
<tr>
<td>(U_0)</td>
<td>hartree</td>
<td>40.102</td>
<td>1.413</td>
<td>0.912</td>
<td>0.570</td>
</tr>
<tr>
<td>(U)</td>
<td>hartree</td>
<td>40.101</td>
<td>1.415</td>
<td>0.913</td>
<td>0.576</td>
</tr>
<tr>
<td>(H)</td>
<td>hartree</td>
<td>40.101</td>
<td>1.413</td>
<td>0.914</td>
<td>0.571</td>
</tr>
<tr>
<td>(G)</td>
<td>hartree</td>
<td>40.102</td>
<td>1.412</td>
<td>0.918</td>
<td>0.571</td>
</tr>
<tr>
<td>(c_v)</td>
<td>(\text{cal mol}^{-1})</td>
<td>4.064</td>
<td>0.294</td>
<td>0.228</td>
<td>0.186</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>1.209</td>
<td>1.096</td>
<td>1.040</td>
</tr>
</tbody>
</table>

Attentive FP [37] and HamNet [16] of the conformation learning and property prediction tasks on QM9. The results are shown in Table S9. For HamNet, we separately reported the running time of its two modules, namely the Hamiltonian Engine (Ham. Eng.) and the Fingerprint Generator (FP Gen.). Our model displayed comparable efficiency to the introduced baselines on both tasks.

Table S9: Training and Inference time of PhysChem and baselines on QM9 (unit: ms/mol).

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
<th>Trn. Time</th>
<th>Inf. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conf. Learn.</td>
<td>HamNet (Ham. Eng.)</td>
<td>20.31</td>
<td>6.21</td>
</tr>
<tr>
<td></td>
<td>CVGAE</td>
<td>7.03</td>
<td>2.04</td>
</tr>
<tr>
<td></td>
<td>PhysNet (s.a.)</td>
<td>7.54</td>
<td>2.21</td>
</tr>
<tr>
<td>Prop. Pred.</td>
<td>HamNet (FP Gen.)</td>
<td>2.25</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>Attentive FP</td>
<td>5.85</td>
<td>1.51</td>
</tr>
<tr>
<td></td>
<td>ChemNet (real conf.)</td>
<td>3.13</td>
<td>1.03</td>
</tr>
<tr>
<td>Both</td>
<td>HamNet</td>
<td>22.56</td>
<td>7.08</td>
</tr>
<tr>
<td></td>
<td>PhysChem</td>
<td>9.05</td>
<td>3.01</td>
</tr>
</tbody>
</table>

K  The Effect of Dimensionality of the Generalized Space

Figure S4 shows the effect of the hyperparameter \(d_f\) in PhysChem. PhysChem is robust to the selection of \(d_f\). Empirically, using larger \(d_f\) leads to equivalent performances on property prediction, and slightly improved ones on conformation learning. Notably, using \(d_f = 3\) already leads to promising results. This is different to that in HamNet [16].
Figure S4: The effect of $d_f$ on QM7.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 4.1.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 2, 3 and 4.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details are in the appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See Section 4.1.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Section 4.1.
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes] All publicly available.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

19
(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]