分子 CT：一种统一分子不同尺度的几何和表示学习的新模型

1.研究背景

分子动力学 MD（molecular dynamics）模式是化学、生物、物理、材料等领域重要的研究工具，而随着近些年人工智能的高速发展，以深度学习（deep learning）为代表的机器学习（machine learning）技术也被广泛地应用于 MD 模拟的研究中[1]。近年来人工智能的蓬勃发展，特别是基于人工神经网络（artificial neural networks）的深度学习技术在科学计算领域也得到了广泛的应用。人工神经网络是由简单的类神经元处理单元组成的参数化模型，通常可分为输入层，隐藏层和输出层，例如多层感知机 MLP（multi-layer perceptron）就是一种经典的人工神经网络。随着近年来深度学习的重新崛起，通过增加隐藏层的数量可以产生更 “深”的模型，即“深度神经网络”[2, 3]。根据“万能近似定理”（universal approximation theorem）[4]，人工神经网络是一种“万能近似器”，也就是说理论上神经网络可以拟合任何形式的有限维数学函数，从而实现许多非常复杂的功能。目前深度学习在计算机视觉 CV（computer vision）和自然语言处理 NLP（natural language processing）等领域取得了空前的成功，在分子模拟领域也有很多研究人员利用深度学习进行如势能函数拟合[5, 6]、粗粒化（coarse grained）力场[7, 8]、增强抽样（enhanced sampling）[9, 10]及动力学建模[11, 12]等工作。

由于原子及分子体系与图像或文本的归纳偏置（inductive biases）不同，因此适合原子或分子体系的神经网络模型跟 CV 和 NLP 中常用的模型非常不一样。由于能量和力等性质是原子种类和空间坐标的函数，这些函数具有参考系不变性（或称为旋转平移不变性）等多粒子体系的对称性，所以一个可以良好描述原子或分子体系的神经网络需要能够保证这些归纳偏置。这类神经网络一般是用来拟合原子体系的从头算（*ab initio*）势能面（potential energy surface），也就是将原子类型和位置坐标映射为能量的函数。势能面原则上可以通过求解薛定谔方程（Schrödinger equation）来实现，然而薛定谔方程的求解非常耗时，所以通常会

使用一种替代函数来估算其近似解，而神经网络则恰恰是一种可以实现这一功能的工具。

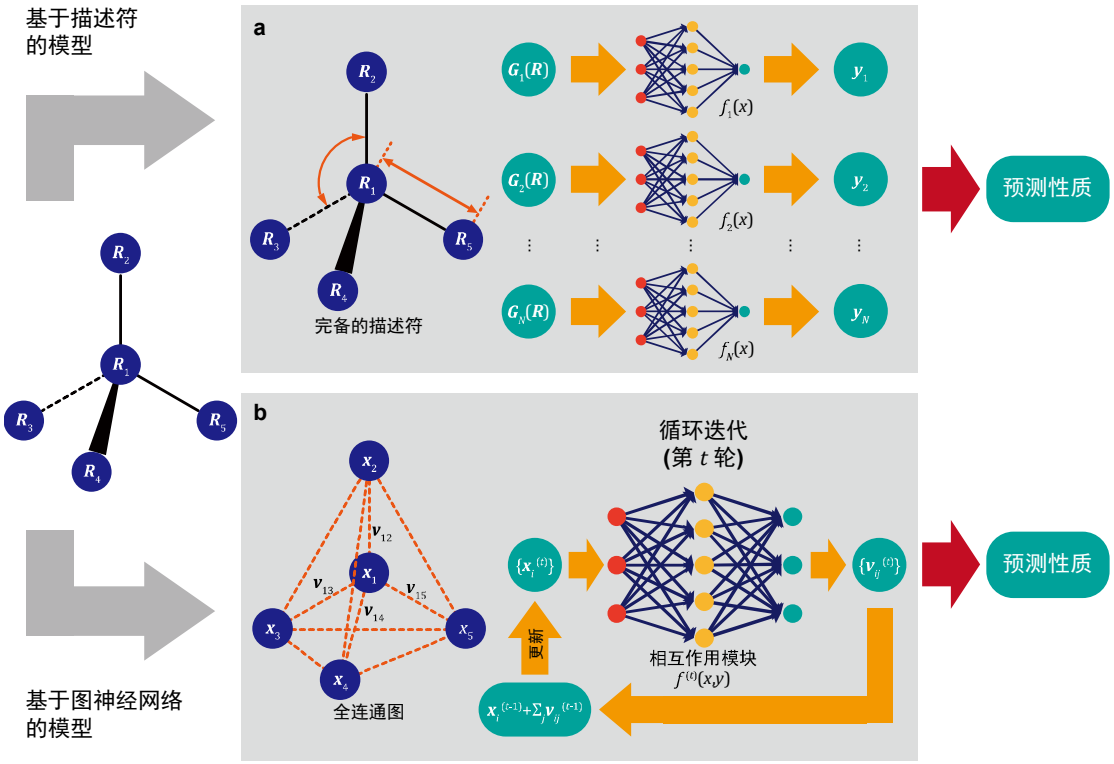

图1 两类深度分子模型：基于描述符的模型（上）和基于图神经网络的模型（下）

早在 2007 年，由苏黎世联邦理工学院 Jörg Behler 博士和 Michele Parrinello 教授开发的 BPNN（Behler-Parrinello neural network）模型[13]就通过将原子的化学环境转变为具有特定对称性和旋转平移旋转不变性的“描述符”（descriptor），从而将其输入神经网络并用于估算各种材料体系的势能面。BPNN 模型的诞生开辟了“基于描述符的分子模型”这一类型（图 1 上），后来诞生的如 ANI[14]、TensorMol[15]及 DPMD[6]等模型都是这类模型。这种模型的性能很大程度上取决于描述符的选择，即“特征工程”（feature engineering）的过程，一旦特征集固定下来，分子模型的表现力就会受到限制，即使改变网络结构也通常不会有较大改进。例如在 BPNN 模型中，当体系的原子类型增加时，所需描述符的数量也将迅速增加。也就是说，这类模型是不通用的，其在规模或组分不同的体系间进行迁移学习的难度很大。当然对于特定体系来说，只要给予适当的描述符，无论是神经网络模型和还是如高斯过程之类的非参数模型都可以表现的很好[5, 16]。

另一方面，信息传递神经网络 MPNN（massage-passing neural network）[17]的

出现则开始通过使用一种相对正交方式来处理原子/分子体系的机器学习问题。MPNN 与基于描述符的模型不同，原子的位置坐标被直接输入到神经网络中，从而用于提取原子化学环境的“表示”（representation），也就是说在预测体系性质之前先学习每个原子的“表示”。这类模型大多基于图神经网络 GNN（graph neural networks），因为 GNN 可以自然地捕捉到分子体系的先验性质，如原子的置换不变性。这类模型包括 DTNN[18]、SchNet[19]、PhysNet 和 DimNet[20]等。这种从基于描述符的模型到基于 GNN 的分子模型（图 1 下）的发展非常类似于卷积神经网络 CNN（convolutional neural network）发明后 CV 研究领域的范式转变，因为与基于描述符的模型相比，基于 GNN 的模型能够根据不同的化学环境进行调整，因此在很多情况下有着更好的性能和可迁移性。

尽管有越来越多深度分子模型被开发出来，但目前这些模型的大多应用于针对原子体系性质的学习，也就是说，元素相同的两个原子间的位置交换对于模型来说是等价的。但对于包括分子模拟在内的很多情况来说，人们更关心特定分子性质的关系，如蛋白质结构序列、DNA 及 RNA 碱基序列、环状聚合物的连接信息等。这种由粒子位置和特定分子构型表示的分子体系通常称为“分子力学”。分子力学模型打破了传统的基于 GNN 模型的元素交换不变性，而对基于描述符的模型来说同样也存在这个问题。因此，发展能同时根据元素和原子类型进行学习的模型是十分必要的。

本文将介绍我们课题组开发的一种可以根据有无粒子间的连接信息从而对体系进行“双重”编码的深度分子模型框架，其既可以根据体系的几何结构，即原子空间位置进行编码；也可以根据原子间的连接信息，如原子间的键连或氨基酸残基序列进行编码。这种具有“双重表达”的深度分子模型的特点在于：

1）该框架统一了针对原子体系和分子体系的图神经网络分子模型，从而可以使用神经网络对多粒子体系进行多尺度的建模。

2）该模型的多体算子借鉴了目前机器学习领域最先进的 Transformer 架构[21]用于学习分子体系的几何结构，可以实现原子或分子体系的高效表示学习。

3) 该模型含有一个图灵完整的多体粒子计算模块，称之为神经交互单元 NIU（neural interaction unit），它能通过一组可训练参数在任意的原子或分子体系

间进行迁移学习。

鉴于该模型框架与 Transformer 的相似性，我们将其命名为分子构型变形金刚 MolCT（molecular configuration transformer），中文简称“分子 CT”模型。该模型将使我们对各种不同的多粒子体系以统一的方式进行建模。

2.团队介绍

作者：张骏[1,2]，陈迪青[1]，张辉耀[3]，周亚强[3]，雷耀坤[1,4]，杨奕[1,*]，高毅勤[1,4,*]。

单位：1. 深圳湾实验室；2. 昌平实验室；3. 华为技术有限公司；4. 北京大学

*通讯作者：

杨奕博士，深圳湾实验室副研究员。2010 年本科毕业于山东大学，获理学（材料物理）学士学位。2015 年研究生毕业于北京大学，获理学（物理化学）博士学位，导师为高毅勤教授。2016 年 1 月至 2019 年 2 月于苏黎世联邦理工学院从事博士后研究工作，合作导师为 Michele Parrinello 教授。2019 年 4 月入职深圳湾实验室，成为系统与物理生物学研究所副研究员。2020 年获得华为首批“HUAWEI Ascend Expert (HAE)”荣誉称号，目前为 MindSpore 专家委员成员，MindSpore 资深布道师，MindSpore 社区分子模拟工作组（WG-MM）负责人。Email: yangyi@szbl.ac.cn

高毅勤教授，北京大学化学与分子工程学院教授，深圳湾实验室系统与物理生物学研究所资深研究员。1993 年本科毕业于四川大学化学系，1996 年获得中科院化学所硕士学位，2001 年获得加州理工学院博士学位，导师为 Rudolph A. Marcus 教授。2001 至 2002 年间于加州理工学院从事博士后研究工作，合作导师为 Rudolph A. Marcus 教授。2002 至 2004 年间于哈佛大学从事博士后研究工作，合作导师为 Martin Karplus 教授。2005 至 2009 年间任得克萨斯农工大学化学系助理教授。2010 年起任北京大学化学与分子工程学院教授，2013 年起任北京大学生物医学前沿创新中心研究员，2019 年起任深圳湾实验室系统与物理生物学研究所资深研究员。Email: gaoyq@pku.edu.cn

3.论文主要内容简介

近些年来，深度学习在逐渐改变计算化学领域的研究模式，特别是在分子模拟领域显示出巨大的应用潜力，这就需要一种可以对分子体系进行编码的神经网络架构。本文将介绍我们课题组开发的一种全新的深度神经网络架构——“分子 CT”。分子 CT 模型由一个关联感知编码器模块和一个可计算的通用几何学习单元组成，其能支持不同的粒子数，并通过粒子间的连接信息实现分子的旋转平移不变性。分子 CT 模型具有很高的计算效率，适用于针对各种体系的学习场景，且具备面向不同体系的迁移学习能力。我们在不同规模的分子体系上的训练结果表明，分子 CT 模型相比基准模型可以使用更少参数实现更好的学习效果。

4.代码链接

论文链接：https://arxiv.org/abs/2012.11816

代码链接：

5.算法框架技术要点

1) 多粒子体系的统一表达

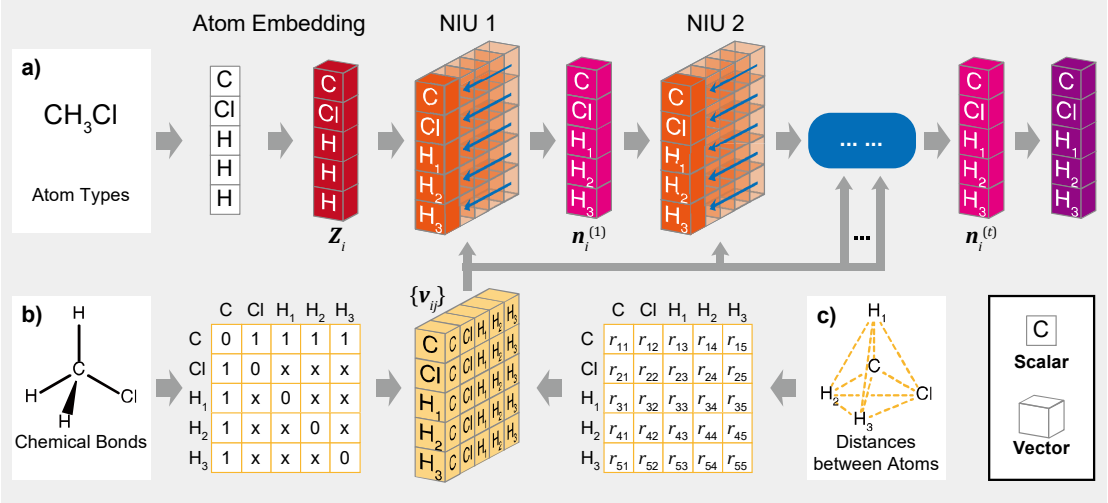

图2 MolCT 模型的基本架构与双重表示

大多数多粒子体系可以根据其中的两种信息进行描述，一是粒子间的连接信息（图 2 b），二是粒子的坐标（图 2 c）。因此分子体系可以视为一个由节点（node，或称为“点 vertex”）和边（edge）组成的图（graph），其中粒子的基本信息如原子类型被编码到“节点”中，而不同节点之间的“边”则可由粒子间的距离及可能连接信息来表示。而在不考虑粒子间连接信息的情况下，该模型则可以简化为针对原子体系的模型。这种分子图根据“边”的不同既可以是有向图也可以是无向图，从而使我们能够以一种统一的方式表示原子体系、分子体系、蛋白质模型甚至是粗粒化模型。

其中初始的节点向量（node vector）$\boldsymbol{z}_i$用来表示对粒子i的编码以区分不同的粒子类型（如元素或原子类型），类似于 NLP 中对单词进行的“嵌入（embedding）”操作[22]。初始节点向量$\boldsymbol{z}_i$是一个通用的、可迁移的嵌入表示，可在特定的多粒子体系中进行共享。例如在针对量子化学计算的学习中，初始节点向量可以是一个根据原子序数独热码（one-hot）进行嵌入的向量；而在粗粒化模型中，粗粒化粒子的节点向量则可以根据代表不同的残基类型的编号进行嵌入。

而两个节点之间的“边”可以用“关系边向量（relational edge vector）”$\boldsymbol{v}_{ij}$或“位置边向量（positional edge vector）”$\boldsymbol{e}_{ij}$表示。其中关系边向量$\boldsymbol{v}_{ij}$表示粒子i和粒子j之间的连接信息，例如在有机分子体系中，$\boldsymbol{v}_{ij}$用于表示原子之间的键连（图 1A），而对于蛋白质或 DNA/RNA 来说，$\boldsymbol{v}_{ij}$则用于表示氨基酸或碱基序列中残基的相对位置和方向（图 1B）。关系边向量可以通过将不同类型的连接信息映射到可优化的向量上来实现[17]，即$\boldsymbol{v}_{ij} \in \mathbb{R}^{1\times d}$。而位置边向量$\boldsymbol{e}_{ij}$则用来编码分子的空间几何信息，一般使用粒子间距离$r_{ij} = \left|\mathbf{R}_i - \mathbf{R}_j\right|$来表示粒子间的相对位置从而保证模型的旋转平移不变性，其中$\boldsymbol{R}_i$表示粒子i的笛卡尔坐标。这里我们选择使用$\log r_{ij}$对“边”进行特征化，这样对$\log r_{ij}$进行线性变换（如神经网络中的矩阵乘法）的学习就相当于学习r_{ij}的幂函数，而许多物理上的相互作用也经常是距离的幂函数，例如库仑势中的r_{ij}^{-1}。这一变换还确保了泡利不相容原理中的渐进奇异性，即原子不重叠（$r_{ij} \neq 0$）。虽然原则上可以直接将$\log r_{ij}$广播为位置向量$\boldsymbol{e}_{ij}$，但直接用一个向量来表示一个标量是非常低效的。这里我们借鉴了 Transformer

模型中的位置嵌入（positional embedding），通过径向基函数 RBF（radical basis functions）将r_{ij}扩展为一个d维的行向量:

$$\boldsymbol{e}(r_{ij}) = [e_1(r_{ij}), e_2(r_{ij}), \dots, e_1(r_{ij}),] \tag{1}$$

$$e_k(r_{ij}) = \exp\left[-\frac{(\log r_{ij} - \mu_k)^2}{2\sigma^2}\right] \tag{2}$$

其中μ_k为在$[\log r_{\min}, \log r_{\text{cut}}]$间的均匀分布，$\sigma$是决定模型空间分辨率的超参数。如图 3 所示，该 RBF 将从距离下限$r_{\min}$到截断距离（cut-off distance）r_{cut}之间的距离信息连续地编码到$\boldsymbol{e}_{ij}$中，且越接近$r_{\min}$的距离编码越稠密，从而契合粒子状态通常由其近距离的局部环境决定的物理直觉。其他模型如 SchNet 和 PhysNet 等也是用 RBF 对距离进行展开，不过它们所使用的基函数多为对r_{ij}的直接展开。

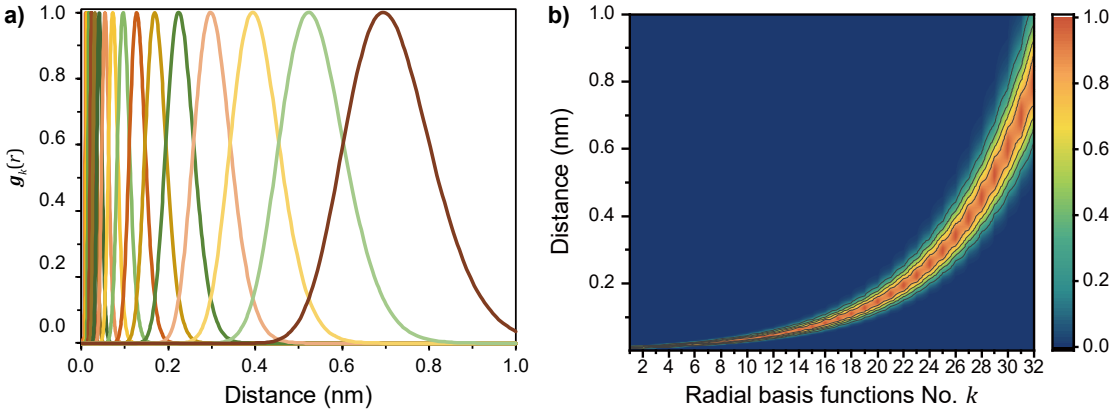

图3 MolCT 模型使用的径向基函数 RBF。

2) MolCT 模型的交互算子

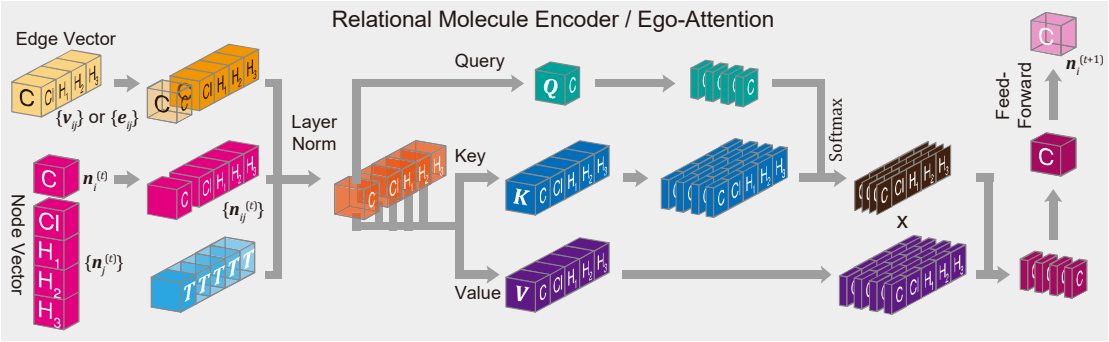

图4 分子 CT 模型交互层的基本算子:关系分子编码器/自我注意力机制示意图。

“交互层（interaction layer）”是 GNN 分子模型的核心，其作用是将每个节点向量与其周围的节点向量和边向量进行交互，从而转变为新的节点向量。这里我们使用作为 Transformer 模型[21]基础的多头注意力 MHA（multi-head attention 机制作为分子 CT 模型交互层的基本算子。对于注意力机制来说，当给出一个“查询向量（query vector）”$\boldsymbol{Q} \in \mathbb{R}^{1\times D}$、$N$个“键向量（key vector）”$\boldsymbol{K} \in \mathbb{R}^{N\times D}$和$N$个“值向量（value vector）”$\boldsymbol{V} \in \mathbb{R}^{N\times D}$（$\boldsymbol{K}$和$\boldsymbol{V}$两个向量间行与行的对应关系是固定的）后，便可以根据下列的非线性操作进行“单头”注意力操作：

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \boldsymbol{a}\boldsymbol{V} \tag{3}$$

$$\boldsymbol{a} = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D}}\right) \tag{4}$$

这里 Softmax 操作根据$\boldsymbol{Q}$和$\boldsymbol{K}$生成一组归一化的注意力系数（attention coefficients）$\boldsymbol{\alpha} \in \mathbb{R}^{1\times N}$用于对$\boldsymbol{V}$的数值进行加权求和，从而生成与$\boldsymbol{Q}$形状相同的新向量。此外每个节点往往还需要一个邻居掩码（neighbour mask），用来将周围没有连接的粒子的注意力系数α变为零。

而具有k个“头”（heads）的多头注意力（MHA）机制则为：

$$\text{MHA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = [\text{head}_1, \text{head}_2, \ldots, \text{head}_k,]W^{(O)} \tag{5}$$

$$\text{head}_i = \text{Attention}\left[\boldsymbol{Q}\boldsymbol{W}_i^{(Q)}, \boldsymbol{K}\boldsymbol{W}_i^{(K)}, \boldsymbol{V}\boldsymbol{W}_i^{(V)}\right] \tag{6}$$

其中的$\boldsymbol{W}_i^{(Q)} \in \mathbb{R}^{D\times D/k}$、$\boldsymbol{W}_i^{(K)} \in \mathbb{R}^{D\times D/k}$、$\boldsymbol{W}_i^{(V)} \in \mathbb{R}^{D\times D/k}$及$\boldsymbol{W}^{(V)} \in \mathbb{R}^{D\times D}$是用于仿射变换（affine projection）的可优化矩阵。

初始节点向量$\{\boldsymbol{z}_i\}_{i=1,\ldots,N}$只含有该粒子本身信息的信息，但将其与周围的关系边向量$\{\boldsymbol{v}_{ij}\}_{i,j=1,\ldots,N}$进行交互，便可将其嵌入为包含周围连接信息的节点向量$\{\mathbf{n}_i\}_{i=1,\ldots,N}$。对于每个初始节点向量$\boldsymbol{z}_i$，可根据下列公式计算查询向量$\boldsymbol{Q}$、键向量$\boldsymbol{K}$和值向量$\boldsymbol{V}$：

$$\boldsymbol{Q}_i = \boldsymbol{z}_i \tag{7}$$

$$\boldsymbol{k}_{j|i} = \boldsymbol{z}_j + \boldsymbol{v}_{ij}\boldsymbol{W}^{(K_2)} \tag{8}$$

$$\boldsymbol{v}_{j|i} = \boldsymbol{z}_j + \boldsymbol{v}_{ij}\boldsymbol{W}^{(V_2)} \tag{9}$$

$$\boldsymbol{K}_i = \left(\boldsymbol{k}_{1|i}^T, \boldsymbol{k}_{2|i}^T, \ldots, \boldsymbol{k}_{N|i}^T, \right)^T \tag{10}$$

$$\boldsymbol{V}_i = \left(\boldsymbol{v}_{1|i}^T, \boldsymbol{v}_{2|i}^T, \ldots, \boldsymbol{v}_{N|i}^T, \right)^T \tag{11}$$

其中$\boldsymbol{W}^{(K_2)}$和$\boldsymbol{W}^{(V_2)}$是作用于边的矩阵。与 Transformer 模型一样，在多层注意力机制之后一般还需要经过一个前馈网络 FFN（feed-forward network）并进行层标准化（layer normalization）[23]，从而更新节点嵌入：

$$\boldsymbol{n}_i = \mathrm{FFN}[\boldsymbol{z}_i + \mathrm{MHA}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i)^T] \tag{12}$$

我们将上述操作称之为关系分子编码器 RME（relational molecular encoder）。多头注意力机制和前馈网络构成了 RME 的基本组件，而在模型中可以由多个这样的组件进行组合。RME 中涉及到包括 MHA 和层标准化操作在内的所有操作都与粒子数N和粒子的输入顺序无关，这意味着同一个网络可以在粒子数不同的体系上运行，并保证相同粒子的置换不变性。

初始节点向量通常只能通过节点类型（如元素类型或原子类型等）来区分不同的粒子，但 RME 可以将初始节点向量转化为“关系感知（relation-aware）”的节点向量从而进一步区分相同类型不同环境的粒子。例如一个 sp^2 碳原子和一个 sp^3 碳原子具有相同的初始节点向量，但由于这两个碳原子具有不同的键级（与其他原子的键连数目不同），在经过 RME 之后这两个碳原子的关系感知向量就会不同。因此 RME 可对空间立体几何结构未知的体系进行嵌入表示，所以这种是一种可以很好地预测分子构型性质等下游学习任务的算子。

但仅通过 RME 仍然无法描述跟体系空间立体结构有关的信息，如分子构象相关的性质。由于初始节点向量通常并不会提供额外的多体相互作用信息，所以

关系感知节点嵌入向量$\boldsymbol{n}_i$一般也不包含构象信息，其只有再经过与周围位置边向量$\{\mathbf{e}_{ij}\}_{i,j=1,\ldots,N}$的交互才能完善该粒子的“表示”。然而，位置边向量$\boldsymbol{e}_{ij}$一般也只包含两个粒子的相对位置，但很多基于分子结构的性质（如键角和二面角等）是关于多个粒子位置的函数，这就需要一个多体算子去传递和整合多体例子间的几何信息从而学习粒子的表达。实际上绝大多数基于 GNN 的分子模型都具有这类算子，例如由 MPNN 模型提出的消息传递（messages passing）机制[17]，以及 SchNet 模型提出的连续过滤卷积 CFC（continuous-filter convolution）[19]操作等。

这里我们通过类似 Transformer 模型中的“位置嵌入（positional embedding）”操作将空间信息导入到注意力机制中，从而实现了一种具有旋转平移不变性和交换不变性的多体算子。由于在物理上单个粒子的绝对位置并没有意义，所以节点的位置嵌入将取决于参考节点，一旦一个节点$\boldsymbol{n}_i$被选作参考，所有其它$\{\mathbf{n}_j\}_{j=1,\ldots,N}$的位置嵌入向量$\{\mathbf{P}_{j|i}\}_{j=1,\ldots,N}$与$\boldsymbol{n}_i$的关系就可以通过线性变换计算出来：

$$\boldsymbol{P}_{j|i} = \boldsymbol{e}_{ij}\boldsymbol{W}^{(P)} + \boldsymbol{b}^{(P)} \tag{13}$$

$\mathbf{W}^{(P)} \in \mathbb{R}^{d\times D}$和$\mathbf{b}^{(P)} \in \mathbb{R}^{1\times D}$是可优化参数。位置边向量$\boldsymbol{e}_{ij}$在这里是一个常数向量，以节点$i$为中心的位置嵌入向量$\{\tilde{\mathbf{n}}_{j|i}\}_{j=1,\ldots,N}$可以参照如下公式:

$$\tilde{\boldsymbol{n}}_{j|i} = \boldsymbol{n}_j \otimes \boldsymbol{P}_{j|i} \tag{14}$$

这里$\otimes$表示元素乘。向量$\tilde{\mathbf{n}}_{i|i} \in \mathbb{R}^{1\times D}$可作为查询向量$\boldsymbol{Q}$，而$\{\tilde{\mathbf{n}}_{j|i}\}_{j=1,\ldots,N}$可用于构建键向量$\boldsymbol{K}$和值向量$\boldsymbol{V}$，并通过层前标准化（pre-layer normalization）[24]使向量的数值保持在适当的大小：

$$\boldsymbol{Q}_i = \mathrm{LayerNorm}\left(\tilde{\boldsymbol{n}}_{j|i}\right) \tag{15}$$

$$\boldsymbol{K}_i = \boldsymbol{V}_i = \mathrm{LayerNorm}\left(\left[\tilde{\boldsymbol{n}}_{1|i}^T, \tilde{\boldsymbol{n}}_{2|i}^T, \ldots, \tilde{\boldsymbol{n}}_{N|i}^T,\right]^T\right) \tag{16}$$

因此节点向量$\boldsymbol{n}_i$可通过下面的公式进行更新：

$$\boldsymbol{n}_i = \boldsymbol{n}_i + \mathrm{MHA}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i)^T \tag{17}$$

嵌入后的节点向量也可以像 RME 那样进一步经过前馈网络 FFN 的处理：

$$\boldsymbol{n}_i = \boldsymbol{n}_i + \mathrm{FFN}[\mathrm{LayerNorm}(\boldsymbol{n}_i)] \tag{18}$$

其他基于 GNN 的分子模型一般也会有类似这一前馈网络的操作，然而我们的测试发现，公式(17)本身的表达能力已经足够强，所以完全可以舍弃这一步操作从而减少参数规模和计算量。

此外尽管自我注意力机制允许粒子“注意”到体系中其他的所有粒子，但一般情况下会设置一个截断距离来限制粒子只“注意”到周围的粒子。这里可以像 BPNN 模型那样通过一个衰减函数将大于截断距离的粒子间的注意力系数平滑地减小零。这种有限视野假设（limited horizon assumption）广泛地应用于针对原子体系的学习，这不但符合物理直觉，也有助于防止过拟合和降低计算成本。

上述的一系列操作可以使节点i通过以其自身为中心的相对位置嵌入与其他节点进行相互作用，因此我们称这种方法为自我注意力 EA（ego-attention）机制，该机制允许图中的每个节点都使用同一套参数。由于含有 Softmax 操作，自我注意力机制天然就是一个多体算子，因此该机制即使只进行一步操作也足以处理复杂的多体相互作用，这是 MPNN 的信息传递操作或者 SchNet 的连续过滤卷积操作所难以实现的。自我注意力机制还允许我们对模型的工作机制进行分析，例如可以根据学到的注意力系数来通过观察一个粒子是如何“注意”到其他粒子的，从而研究粒子间互动的模式。另外该机制的计算过程主要为张量乘法，因此可以相对容易地实现 GNN 模型通常难以实现的并行计算。

3) 神经交互单元

有了节点嵌入向量$\mathbf{n}_i^{(t)}$及其相关的边向量$\{\mathbf{e}_{ij}\}_{j=1,\dots,N}$，便可以通过自我注意力机制实现一套从$\mathbf{n}_i^{(t)}$到$\mathbf{n}_i^{(t+1)}$进行演化的动力学规则：

$$\mathbf{n}_i^{(0)} \xrightarrow{\mathrm{EA}} \mathbf{n}_i^{(1)} \xrightarrow{\mathrm{EA}} \cdots \xrightarrow{\mathrm{EA}} \mathbf{n}_i^{(T)} \tag{19}$$

在基于 GNN 的分子模型中，通常需要迭代或重复若干次的多体算子（即交互层）的操作。其中迭代或者重复的次数，也就是即公式(19)中将$\mathbf{n}_i^{(0)}$演化为$\mathbf{n}_i^{(T)}$的次数T通常是一个对模型的最终表现起到决定性作用的超参数。而如何选择合适次数T并不是一件很容易的事，因为迭代次数的多少决定了节点嵌入的演化程度，原则上只有足够大的T才能以使每个嵌入向量$\mathbf{n}_i^{(T)}$的数值最终都到达不动点。而粒子表征学习必要的计算复杂度，即柯氏复杂度（Kolmogorov complexity）取决于粒子的种类和环境，因此不同粒子体系合适的迭代次数T是不一样的。最极端的情况是一个体系中的大部分粒子都可以在一个很小的次数T_1内实现收敛，但极少数粒子却必须经过一个远大于T_1的次数T_2才能实现收敛，这种情况下就需要让所有粒子经过T_2次的计算才能结束计算，从而浪费大量的计算资源。而且从计算科学的角度看，这种由固定的T个交互层组成的模型在计算上是不通用的。换句话说，这样的模型不是图灵完备的（Turing completeness），从而限制了模型在不同分子体系间进行迁移学习的能力。

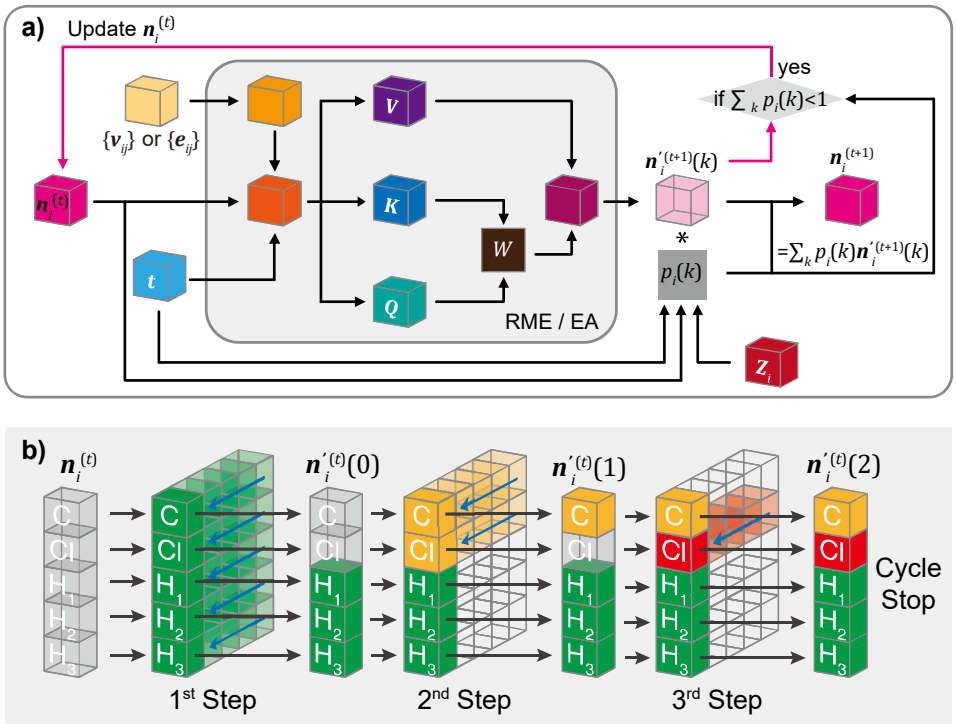

图5 自适应计算时间工作原理示意图。

为了解决这一问题，我们在自我注意力机制的基础上设计了一种计算上通用的交互层，称之为神经交互单元 NIU（neural interaction unit）。节点在每个 NIU

中会共享相同的参数进行多次迭代计算，而每个节点的迭代计算步骤T在是学习过程中动态确定的。这种自适应的迭代步数是通过自适应计算时间（adaptive computational time）[25]算法实现的，该算法最初为递归神经网络 RNN（recursive neural network）设计，后来也被 Universal Transformer 模型[26]所采用的。具体来说，NIU 中设置了一个思考网络（pondering network），它根据每个节点在当前步骤计算得到的嵌入向量$\mathbf{n}_i^{(t)}$来判断是否应该停止该节点的嵌入计算。我们使用一个以 Sigmoid 为激活函数的全连接层作为思考网络，输入嵌入时间的节点向量从而计算中止概率P_{halt}：

$$P_{\text{halt}}\left[\mathbf{n}_i^{(t)}\right] = \text{Sigmoid}\left\{\text{FFN}\left[\mathbf{n}_i^{(t)} \oplus \boldsymbol{T}^{(t)}\right]\right\} \tag{20}$$

其中⊕表示元素加，$\boldsymbol{T}^{(t)}$为时间嵌入向量（time-embedding vector），其形式为：

$$\boldsymbol{T}^{(t)} = \begin{cases} \sin(\omega_k t) & \text{if } i = 2k \\ \cos(\omega_k t) & \text{if } i = 2k+1 \end{cases} \tag{21}$$

$$\omega_k = \frac{1}{10000^{2k/D}} \tag{22}$$

一旦思考网络决定节点i在第t次迭代时中止计算，则 NIU 将在后续的计算中直接地把第t次计算获得的向量$\boldsymbol{n}_i^{(t)}$复制过来。这一过程会持续到所有节点的计算都结束为止，该过程中那些容易学习的粒子会提前中止迭代。

由于参数的通用性和时间嵌入的可扩展性，NIU 在训练完成之后其迭代次数仍然可以进行调整。因此可以在训练过程中对迭代次数T设置一个上限（相当于对模型的柯氏复杂度进行正则化），但在推理过程中取消这一限制。鉴于 NIU 具有充分的“记忆”，因此其在计算上是通用的，也就是说 NIU 像著名的神经图灵机（neural Turing machine）[27]和可微神经计算机（differentiable neural computer）[28]一样是图灵完备的。

完整的分子 CT 模型将由 RME 和 NIU 组成，从而同时处理粒子体系的连接信息（如分子构型）和空间几何信息（如分子构象）。实际操作中还可以同时将几个不同的 NIU 串联起来使用，从而实现更大的模型规模并加快训练过程。

4) 多层次的表征学习

作为一种基于图神经网络的深度分子模型，分子 CT 能够在不同层次上对多粒子体系进行表征学习。首先，对粒子表示的学习可以用来预测单个粒子的性质。例如用于拟合从头算势能面时，总势能可以被分配到体系的各个原子上，因此可以为每个原子学习相应的表达$\boldsymbol{n}_i^{(T)}$，再使用神经网络或核方法构建一个节点读出函数（readout function）$f_\theta^{(\mathcal{N})}$用于预测每个粒子的能量$E_{i,\theta}$和力$\boldsymbol{F}_{i,\theta}$：

$$E_{i,\theta} = f_\theta^{(\mathcal{N})}\left[\boldsymbol{n}_i^{(T)}\right] \tag{23}$$

$$\boldsymbol{F}_{i,\theta} = -\nabla_{\boldsymbol{R}} f_\theta^{(\mathcal{N})}\left[\boldsymbol{n}_i^{(T)}\right] \tag{24}$$

而训练用的损失函数（loss function）为：

$$\text{loss} = (1-\lambda)\left(E_0 - \sum_{i=1,N} E_{i,\theta}\right)^2 + \lambda \sum_{i=1,N} \left\|\boldsymbol{F}_{i,0} - \boldsymbol{F}_{i,\theta}\right\|^2 \tag{25}$$

其中E_0是所有体系总能量的标签，$F_{i,0}$是各个原子受力的标签，λ是一个调节损失函数中能量和力的比例的超参数。通过最小化这一损失函数，模型将与读出函数$f_\theta^{(N)}$一起被优化，从而实现预测目标。

分子 CT 模型也可以对“边”的表示进行学习，即利用边读出函数$f_\theta^{(E)}$预测任意两粒子间的性质。边读出函数通过读取一对粒子的嵌入向量$\left\{\boldsymbol{n}_i^{(T)}, \boldsymbol{n}_j^{(T)}\right\}$及相关的边向量$\left(\boldsymbol{v}_{ij}, \boldsymbol{e}_{ij}\right)$来进行预测：

$$\text{Relation}\left(\boldsymbol{n}_i, \boldsymbol{n}_j\right) = f_\theta^{(\mathcal{E})}\left(\left\{\boldsymbol{n}_i^{(T)}, \boldsymbol{n}_j^{(T)}\right\}, \boldsymbol{v}_{ij}, \boldsymbol{e}_{ij}\right) \tag{26}$$

上述公式的结果与两个节点向量的输入顺序无关，因此这属于一种关系神经网络（relational neural networks）。“边”级别的表征学习可用于针对粒子间关系的推理，例如预测蛋白质的两个氨基酸残基间的接触概率、以及预测粒子间的长程相互作用（如库仑相互作用）等。

最后，分子 CT 模型还可以学习“图”的表示，通过引入一个图读出函数或池化（pooling）函数$f_\theta^{(\mathcal{G})}$将所有的节点向量以交换不变的方式合并到为一个全局向量$\boldsymbol{g}$中，从而实现对于整个体系整体性质的预测。

$$\boldsymbol{g} = f_\theta^{(\mathcal{G})}\left(\left\{\boldsymbol{n}_1^{(T)}, \boldsymbol{n}_2^{(T)}, \ldots, \boldsymbol{n}_N^{(T)}\right\}\right) \tag{27}$$

图的表示对于分子强度量的学习特别有用，其在分子构象的聚类、降维以及各种核方法等很多领域都有着巨大的应用潜力。

6.实验结果

1) 自我注意力机制测试

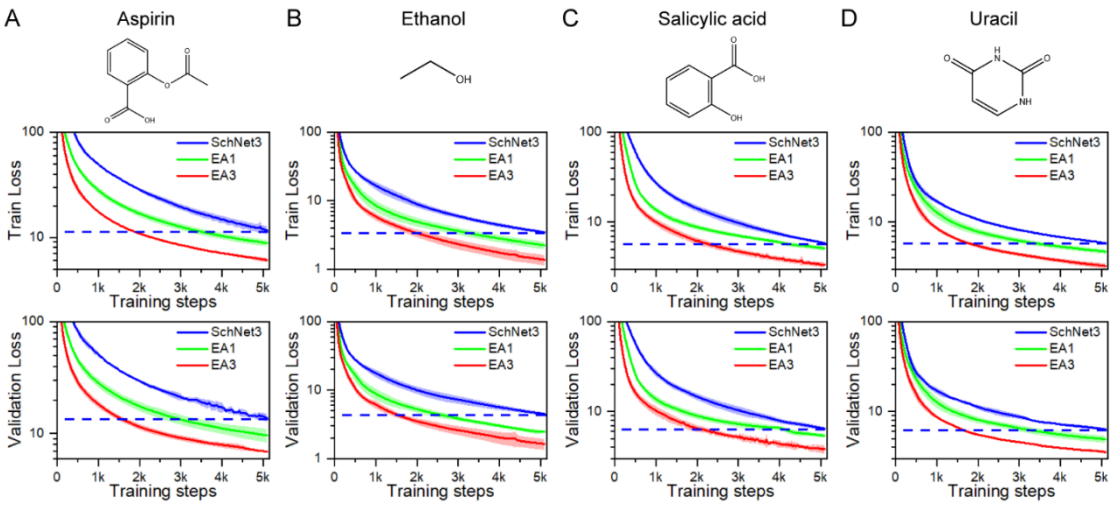

图6 分子 CT 模型的自我注意力(EA)机制与 SchNet 模型的连续过滤卷积(CFC)的效果对比。

我们首先将分子 CT 模型的 EA（自我注意力）机制与 SchNet 模型[19]所采用的 CFC（连续过滤卷积）在 MD17 数据集[29]上进行了对比测试。使用 EA 的模型使用了 8 个“头”进行 MHA（公式(5)），这里我们分别建立了单层（以下简称 EA1）和三层（以下简称 EA3）的 EA 模型，并与三层 SchNet 模型（以下简称 SchNet3）进比较。其中 EA 和 SchNet 模型的每个交互层都使用相同规模的参数量（即使用相同大小的向量维度），从而保证公平的比较。不过需要注意的是，

这里 EA 模型没有使用公式(18)中的 FFN，而这对于 SchNet 模型来说却是必不可少的。因此单层 EA 算子的参数规模比单层 CFC 算子少了近一半，也就是说即便是 EA3 模型的参数规模也只有 SchNet3 模型的一半左右。

如图 6 所示，EA1 模型在所有体系中的测试结果都要优于作为基准的 SchNet3 模型，在训练过程中无论是训练损失还是验证损失都要比 SchNet3 小 20%左右。这表明自我注意力机制是一种高效的多体算子，单层 EA 算子的效果甚至高于三层的 CFC 算子。通过叠加多层 EA 算子可以进一步提高整个模型的效果，且提升的效果元大于 CFC 算子的叠加，因此 EA 算子很适合用于组建更“深”的大模型。

2) 神经交互单元测试

我们又测试了开启了自适应计算时间的完整 NIU（神经交互单元）的效果。NIU 在 EA 算子的基础之上增加了一个思考网络，从而不需要为算子提前设置迭代次数T，而是可以在训练中自动地学习每个节点需要运行 EA 算子多少次。

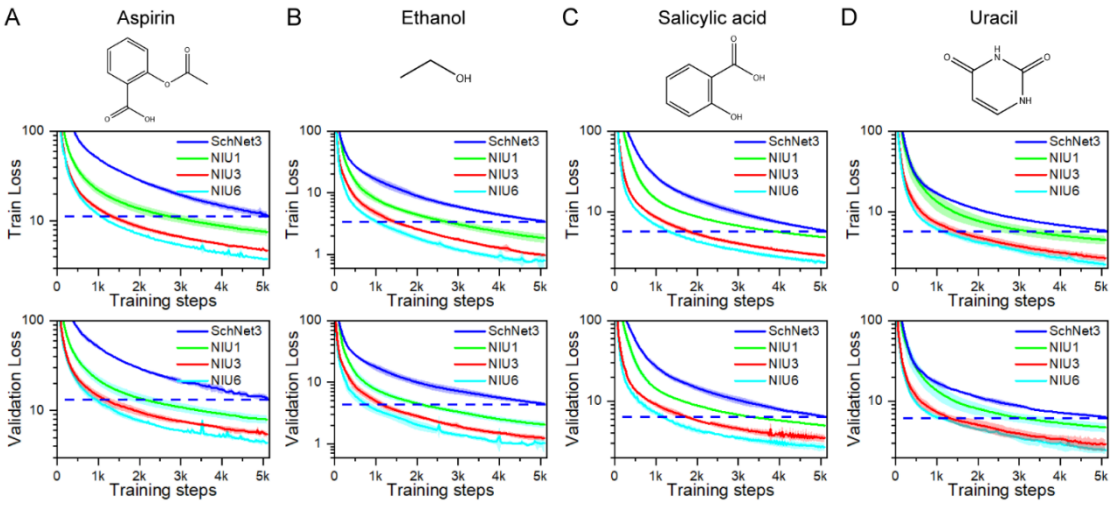

图7 使用了神经交互单元 NIU（开启自适应计算时间）MolCT 模型与 SchNet 模型的训练效果对比。

图 7 显示，单层模型 NIU1 可以在小得多的模型参数规模的情况下，其训练效果仍然可以远优于作为基准 SchNet3 模型。而通过增加 NIU 层数可以实现更好的优化效果，这里我们最多测试了六层 NIU，可以极大地提升训练的精度。此外，与单纯的 EA 模型相比，NIU 只额外多了非常少的参数量，却可以很大程度

上提升自我注意力机制的表现。特别是增加 NIU 层数的效果比单纯增加 EA 层数的效果更加明显，而参数规模却并没有增加特别多。即使是六层 NIU 模型，其参数规模仍然小于三层 SchNet 模型，因此 NIU 也非常适合搭建深层模型。

3) 带有连接信息的分子体系测试

前面的两个测试验证了分子 CT 模型对于原子体系，即在没有粒子间连接信息情况下的学习能力。然而许多分子模拟体系都包含连接信息，这里我们使用丙氨酸二肽（Alanine dipeptide）体系测试了分子 CT 模型对于带有连接信息的分子体系的效果。我们所使用的训练数据集与 MD17 等来自量子化学计算的数据集不同，该数据集中的数据标签来自使用 AMBER-FF03 力场[30]建立的隐式溶剂[31]中的丙氨酸二肽的分子模型（ACE-ALA-NME）。分子力场与量子化学不同，它需要根据原子间的化学键连接来进行计算，因此使用深度分子模型对基于分子力场的生成的数据进行训练需要确保模型可以处理分子体系的连接信息。

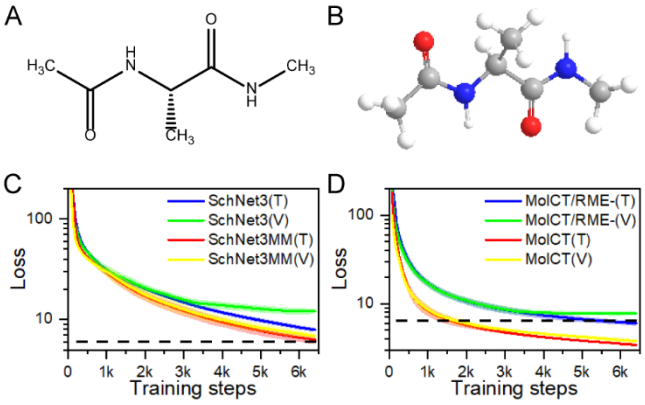

图8 针对丙氨酸二肽体系的带有连接信息的训练测试。

这里我们首先测试了 SchNet。我们一开始仍然根据原子的元素类型（即原子核电荷数）来区分 SchNet 模型不同的节点，图 8 C 显示，训练过程中 SchNet3 模型的验证损失和训练损失之间很快就出现了较大差异，说明出现了过拟合。该训练任务的失败是意料之中的，因为 SchNet 模型没有考虑分子中的键连，所以无法区分具有不同原子类型（即键连关系）但元素相同的节点。

然后我们根据分子力场中的原子类型为丙氨酸二肽分子的原子类型进行重新编号，从而让 SchNet 模型可以区分不同的原子类型（这里称之为 SchNet3MM）。图 9C 显示，这种人为干预的原子类型重新编号可以提升 SchNet 模型在分子力

场数据集上的表现，并没有过早地出现过拟合。这一结果表明连接信息可以打破体系内在的对称性或不变性，因此如果要让类似 SchNet 这样的模型捕捉这些性质就必须人为去整合更多的额外信息。

而分子 CT 模型可以凭借 RME 读取分子体系的连接信息。图 9D 显示分子 CT 模型对于丙氨酸二肽分子力场数据集的学习效果非常好，与在 MD17 等原子体系数据集上的训练一样有效。跟 SchNet3MM 模型只能根据原子类型人为修改初始节点嵌入不同，分子 CT 只需要读入原子真实的元素类型作为初始节点嵌入，再通过 RME 的学习便能自动将其转化为连接信息，不需要进行人工预处理。作为对照，如果从分子 CT 模型中移除 RME，该模型（图 9D 中的 MolCT/RME-）也会像 SchNet3 那样过早地出现过拟合，表明 RME 在处理含有连接信息的多粒子体系中起到关键作用。

7.MindSpore 代码实现

Cybertron 是一款基于华为全场景人工智能框架 MindSpore 的图神经网络分子模型通用架构，该程序内置了 MolCT、SchNet 和 PhysNet 三种分子模型，并允许开发者使用该框架实现自己的分子模型。

Cybertron 程序代码中的 examples 目录内置了九个教程脚本，内容分别为：

Tutorial_00.py：数据预处理

Tutorial_01.py：基础教程（一）

Tutorial_02.py：基础教程（二）

Tutorial_03.py：归一化数据集与验证数据集的使用

Tutorial_04.py：模型参数与超参数的读取（一）

Tutorial_05.py：多任务训练（一）

Tutorial_06.py：多任务训练（二）

Tutorial_07.py：带有力的数据集的拟合

Tutorial_08.py：模型参数与超参数的读取（二）

Cybertron 的使用可参照以上教程。

8.总结与展望

本文介绍了我们课题组开发的一种可以实现多粒子系统通用学习的全新深度神经网络架构“分子 CT”模型。“分子 CT”模型具有双重表示的能力，因此可以保证多粒子体系的旋转平移不变性，并引入了那些可以影响体系性质的粒子间的连接信息。分子 CT 模型基于可以反应体系物理化学性质的图神经网络，其中粒子间距在取对数后被映射为位置边向量，并使用类似 Transformer 模型的进行嵌入，而粒子间的连接信息也可用类似的方式嵌入网络。因此分子 CT 模型的优势在于使用了一种统一的架构中对原子体系、分子体系甚至是蛋白质残基等粗粒化粒子体系进行建模。

我们还为分子 CT 模型开发了一种对体系的空间几何性质进行学习的高效多体算子“自我注意力”机制，使得模型在学习体系的多体相互作用方面有着很好的表现。与其他类似的模型相比，分子 CT 模拟能够以更少的参数规模实现相当甚至更好的效果，从而方便用户的训练和使用。而我们设计的“神经交互单元”，可以让分子 CT 模型实现大多数 GNN 分子模型所无法实现的交互层自适应性迭代。因此分子 CT 模型是一种在计算上通用的图灵完备的分子模型，使用其可以对不同类型或不同环境中的体系实现自适应的表示学习，这一点对于针对不同体系的迁移学习或者元学习（meta-learning）至关重要。最后，通过对自我注意力系数和自适应计算时间的分析可以获得粒子间互动的线索，从而使神经网络模型不再是一个“黑箱”。最后，由于自我注意力机制的计算效率较高且能够实现自动并行，因此分子 CT 模型可以发展为具有超大参数规模的预训练大模型，这对于标签昂贵的下游任务的迁移学习来说是非常有价值的。

目前越来越多的研究开始使用深度学习来解决那些具有挑战性的分子模拟问题，如求解薛定谔方程、预测蛋白质结构、粗粒化生物分子力场、分子结构的增强抽样以及化学反应机理的研究等等。因此能同时兼容原子体系和分子体系，且具有强大通用性和可扩展性的分子 CT 模型对于分子模型来说有着非常重要的意义，我们有理由相信其能成为分子模拟领域中那些前沿研究领域的重要工具。

参考文献

[1] Zhang, J.; Lei, Y.-K.; Zhang, Z.; Chang, J.; Li, M.; Han, X.; Yang, L.; Yang, Y. I.; Gao, Y. Q. A Perspective on Deep Learning for Molecular Modeling and Simulations [J]. J Phys Chem A, 2020, 124(34): 6745-6763.

[2] Lecun, Y.; Bengio, Y.; Hinton, G. Deep Learning [J]. Nature, 2015, 521(7553): 436-444.

[3] Schmidhuber, J. Deep Learning in Neural Networks: An Overview [J]. Neural Netw, 2015, 61: 85-117.

[4] Hornik, K.; Stinchcombe, M.; White, H. Multilayer Feedforward Networks Are Universal Approximators [J]. Neural Netw, 1989, 2(5): 359-366.

[5] Mueller, T.; Hernandez, A.; Wang, C. Machine Learning for Interatomic Potential Models [J]. J Chem Phys, 2020, 152(5): 050902.

[6] Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics [J]. Phys Rev Lett 2018, 120(14): 143001.

[7] Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; De Fabritiis, G.; Noé, F.; Clementi, C. Machine Learning of Coarse-Grained Molecular Dynamics Force Fields [J]. ACS Cent Sci, 2019, 5(5): 755-767.

[8] Zhang, J.; Lei, Y.-K.; Yang, Y. I.; Gao, Y. Q. Deep Learning for Variational Multiscale Molecular Modeling [J]. J Chem Phys, 2020, 153(17): 174115.

[9] Zhang, J.; Yang, Y. I.; Noé, F. Targeted Adversarial Learning Optimized Sampling [J]. J Phys Chem Lett, 2019, 10(19): 5791-5797.

[10] Bonati, L.; Zhang, Y.-Y.; Parrinello, M. Neural Networks-Based Variationally Enhanced Sampling [J]. Proc Nat Acad Sci USA, 2019, 116(36): 17641-17647.

[11] Mardt, A.; Pasquali, L.; Wu, H.; Noé, F. Vampnets for Deep Learning of Molecular Kinetics [J]. Nat Commun, 2018, 9(1): 5.

[12] Zhang, J.; Lei, Y.-K.; Che, X.; Zhang, Z.; Yang, Y. I.; Gao, Y. Q. Deep Representation Learning for Complex Free-Energy Landscapes [J]. J Phys Chem Lett, 2019, 10(18): 5571-5576.

[13] Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces [J]. Phys Rev Lett 2007, 98(14): 146401.

[14] Smith, J. S.; Isayev, O.; Roitberg, A. E. Ani-1: An Extensible Neural Network Potential with Dft Accuracy at Force Field Computational Cost [J]. Chem Sci, 2017, 8(4): 3192-3203.

[15] Yao, K.; Herr, J. E.; Toth, D. W.; Mckintyre, R.; Parkhill, J. The Tensormol-0.1 Model Chemistry: A Neural Network Augmented with Long-Range Physics [J]. Chem Sci, 2018, 9(8): 2261-2269.

[16] Bartók, A. P.; De, S.; Poelking, C.; Bernstein, N.; Kermode, J. R.; Csányi, G.; Ceriotti, M. Machine Learning Unifies the Modeling of Materials and Molecules

[J]. Science Advances, 2017, 3(12): e1701816.

[17] Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry [C] //Proceedings of Machine Learning Research, Sydney, 2017. ML Research Press: 1263-1272.

[18] Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-Chemical Insights from Deep Tensor Neural Networks [J]. Nat Commun, 2017, 8(1): 13890.

[19] Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. Schnet – a Deep Learning Architecture for Molecules and Materials [J]. J Chem Phys, 2018, 148(24): 241722.

[20] Gasteiger, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs [J/OL]. arXiv preprint, 2020: arXiv:2003.03123 [2020-03-06]. https://arxiv.org/abs/2003.03123.

[21] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need [C], Long Beach, 2017. New York: Curran Associates, Inc.: 5998-6008.

[22] Wang, S.; Zhou, W.; Jiang, C. A Survey of Word Embeddings Based on Deep Learning [J]. Computing, 2020, 102(3): 717-740.

[23] Lei Ba, J.; Kiros, J. R.; Hinton, G. E. Layer Normalization [J/OL]. 2016: arXiv:1607.06450 [2016-07-01]. https://arxiv.org/abs/1607.06450.

[24] Xiong, R.; Yang, Y.; He, D.; Zheng, K.; Zheng, S.; Xing, C.; Zhang, H.; Lan, Y., *et al.* On Layer Normalization in the Transformer Architecture [J/OL]. 2020: arXiv:2002.04745 [2020-02-12]. https://arxiv.org/abs/2002.04745.

[25] Graves, A. Adaptive Computation Time for Recurrent Neural Networks [J/OL]. 2016: arXiv:1603.08983 [2016-03-29]. https://arxiv.org/abs/1603.08983.

[26] Dehghani, M.; Gouws, S.; Vinyals, O.; Uszkoreit, J.; Kaiser, Ł. Universal Transformers [J/OL]. 2018: arXiv:1807.03819 [2018-07-10]. https://arxiv.org/abs/1807.03819.

[27] Graves, A.; Wayne, G.; Danihelka, I. Neural Turing Machines [J/OL]. 2014: arXiv:1410.5401 [2014-10-20]. https://arxiv.org/abs/1410.5401.

[28] Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S. G.; Grefenstette, E., *et al.* Hybrid Computing Using a Neural Network with Dynamic External Memory [J]. Nature, 2016, 538(7626): 471-476.

[29] Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine Learning of Accurate Energy-Conserving Molecular Force Fields [J]. Science Advances, 2017, 3(5): e1603015.

[30] Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters [J]. Proteins, 2006, 65(3): 712-725.

[31] Onufriev, A.; Bashford, D.; Case, D. A. Exploring Protein Native States and Large-

Scale Conformational Changes with a Modified Generalized Born Model [J]. Proteins, 2004, 55(2): 383-394.