基于多模态表示和提示语微调的分子性质预测

重庆大学硕士学位论文

（专业学位）

学生姓名：赵卓然

指导教师：周　庆 教　授

专业学位类别：工　程

领　　域：计算机技术

研究方向：人工智能

答辩委员会主席：尚明生 教授

授位时间：2024 年 6 月

Molecular Property Prediction Based on Multimodal Representations and Prompt-based Fine-tuning

A thesis submitted to Chongqing University

in partial fulfillment of the requirement

for the professional degree of Master of

Engineering

by

Zhuoran Zhao

(Computer Technology)

Supervisor: Prof. Qing Zhou

June, 2024

摘 要

分子性质预测在药物发现中扮演着关键角色，它有助于识别具有目标性质的候选分子。传统的药物发现方法需要对药物分子性质进行严格的实验测定，而人工智能（Artificial Intelligence，AI）技术的快速发展使其能够通过学习分析现有药物分子性质数据，提前预测潜在的药物副作用，从而在药物设计的早期阶段规避潜在风险，加速了候选药物的筛选和优化过程，降低了研发消耗。

然而，传统的基于神经网络的分子性质预测方法面临着数据表示不一致、模型泛化能力不足等问题。为应对这些挑战，本文在分子性质预测方法中引入多模态对比学习技术和大语言模型提示学习技术。这需要解决三个技术问题，一是如何对齐分子不同形态的空间表示，二是如何引入分子多粒度表示并结合化学领域知识，三是如何减少多个下游任务带来的模型训练和存储成本。本文针对这些问题开展研究，主要工作和贡献如下：

（1）提出了一种基于对比学习的多模态多粒度分子性质预测模型。为了确保分子图和简化分子输入线性表达式（Simplified Molecular Input Line Entry System，SMILES）有效融合，采用对比学习方法对齐不同模态的分子表示。同时，提出一个包含交叉注意力机制的多模态融合编码器，进一步优化不同模态数据的融合，提升模型的跨模态学习性能。在模型预训练阶段，引入三种粒度的预训练任务，分别关注原子级别、官能团级别和分子级别的特征，有效结合了药物化学领域的专业知识。多粒度的预训练方法使模型能够充分利用分子结构中的丰富信息，提取出对预测任务重要的特征。实验表明，在多个分子性质预测数据集中，本模型的预测准确率均优于基线算法。

（2）提出了一种基于提示语微调的分子性质预测模型。该模型采用少样本学习，不需要为每个下游任务训练单独的模型，仅需将构建的提示语输入大模型，即可对不同分子性质进行泛化学习。提示语包含两个部分，第一部分为分子描述，通过检索算法从外部数据库中获得；第二部分为待预测分子的嵌入表示，通过在待预测分子与正负样本分子之间构建关系图，采用图神经网络（Graph Neural Network，GNN）更新节点表示获得。最后，使用已训练好的大语言模型对药物分子进行性质预测，在利用大语言模型丰富的知识的同时，减少了模型训练成本和参数存储成本。通过实验验证，表明了模型的有效性，其中 ROC-AUC 指标值平均可以达到 86.3%，均优于基线算法。

关键词：分子性质预测；多模态对齐；图神经网络；提示语微调；关系图谱

Abstract

Molecular property prediction plays a critical role in drug discovery, aiding in the identification of candidate molecules with desired properties. Traditional drug discovery methods require rigorous experimental determination of drug molecular properties. However, the rapid advancement of Artificial Intelligence (AI) technologies enables the prediction of potential drug side effects by learning from existing drug molecular property data. This can circumvent potential risks in the early stages of drug design, accelerating the screening and optimization process for candidate drugs and reducing research and development costs.

Nevertheless, traditional neural network-based methods for molecular property prediction face challenges such as inconsistent data representation and insufficient model generalization capabilities. To address these challenges, this thesis introduces multimodal contrastive learning techniques and large language model prompt learning techniques into the molecular property prediction methodology. This involves addressing three technical issues: aligning the spatial representations of molecules in different forms, incorporating multimodal granular representations with domain knowledge in chemistry, and reducing the training and storage costs associated with multiple downstream tasks. The research carried out in this thesis is focused on these issues, with the main work and contributions as follows:

(1) A multimodal and multigranular molecular property prediction model based on contrastive learning is proposed. To ensure the effective integration of molecular graphs and Simplified Molecular Input Line Entry System (SMILES) representations, a contrastive learning approach is used to align molecular representations across different modalities. A multimodal fusion encoder with a cross-attention mechanism is introduced to further optimize the fusion of different modal data, enhancing the model's cross-modal learning performance. During the model's pre-training phase, three granular pre-training tasks are introduced, focusing on atomic, functional group, and molecular level features, effectively integrating professional knowledge from the field of pharmaceutical chemistry. The multigranular pre-training approach allows the model to fully utilize the rich information in the molecular structure, extracting features important for prediction tasks. Experiments show that the model's prediction accuracy is superior to baseline algorithms on multiple molecular property prediction datasets.

(2) A molecular property prediction model based on prompt-tuning is proposed. This model employs few-shot learning and does not require separate models to be trained for each downstream task. Instead, constructed prompts are input into the large model, enabling generalized learning across different molecular properties. The prompt consists of two parts: the first part is the molecular description, obtained from external databases through a retrieval algorithm; the second part is the embedding representation of the molecule to be predicted, obtained by constructing a relational graph between the target molecule and positive and negative sample molecules and updating the node representation using a Graph Neural Network (GNN). Finally, the well-trained large language model is used for drug molecule property prediction, reducing model training costs and parameter storage costs while leveraging the rich knowledge of the large language model. Experimental validation has demonstrated the effectiveness of the model, with an average ROC-AUC metric reaching 86.3%, outperforming baseline algorithms.

Keywords: Molecular Property Prediction; Multimodal Alignment; Graph Neural Network; Prompt Tuning; Relational Graphs

目 录

1 绪 论

1.1 研究工作的背景与意义

越来越多的行业都已经将深度学习运用在了各自领域，制药行业[1]也不例外。分子性质预测[2-4]一直是制药领域的一大挑战。传统的通过湿实验[5]（Wet-lab Experimentation）开发新药是一个昂贵且耗时耗力的过程，只有通过所有化学药物性质的预测，例如毒性，生物活性，代谢稳定性等，才有可能成为候选的新药。如此复杂的过程严重阻碍了药物开发的效率，如果可以通过深度学习的方法[6]对分子进行初步的性质预测和虚拟筛选，将会大大提高药物研发的效率，降低制药成本。

如果使用深度学习的方式预测药物分子性质，至关重要的一步是对分子进行抽象表示，目前主流的方法有两种，一种是基于 SMILES [7]（Simplified Molecular Input Line Entry System），另一种是基于分子图表示。SMILES 是一种用于描述化学物质结构的序列信息，通常用于表示分子，将分子视为序列结构数据，如此便可以采用很多自然语言处理的技术来对分子进行表征，例如 GPT [8]、BERT [9]等。这些技术在分子性质预测领域也有着不错的表现。另一种是将分子描述为具有节点（原子）和边（化学键）的分子图，而不是固定长度的序列嵌入。因此，许多处理图数据的深度学习方法也可以被应用在分子性质预测上，例如 GNN [10-11]，图 Transformer [12-13]等。但是不同模态分子的表示所包含的信息并不完全相同，SMILES 中所标注的手性结构、同位素标记、特殊化合价原子是分子图难以表现的特征。而分子图也有其独特的结构特征，他可以描述分子的拓扑结构，尤其是在药物分子中经常出现的环状结构和多种多样的异构体。因此，两种表示都有其不可替代性。另外，与其他工程领域不同，想要获得准确的药物分子性质需要大量的湿实验，这导致了具有标签的公共数据集非常少，在这些小量数据集上进行训练很容易使模型过拟合，泛化能力较差。在自然语言处理[14-15]领域，为了提高模型泛化能力，研究人员提出了自监督学习的技术，从大规模的无标签数据中预训练模型，同为序列结构的 SMILES 也作为预训练数据集对 BERT 模型进行预训练。随着 Transformer [15]在各领域大放异彩，拥有图结构的药物分子也逐渐加入预训练大军。然而，这些方法都未能将药物分子多模态性入预训练模型。

科学家为每个分子构建的分子描述信息往往被研究人员所忽略，自然语言的描述有时候并不像标准化序列数据[16]以及图数据[2]那样易于处理。但随着大语言模型[17-18]（Large Language Model，LLM）的发展，将这些分子描述作为特征嵌入变为可能。利用更丰富的分子特征可以使模型的预测工作更加的准确。针对分子性质

预测中的这些问题，本文对基于多模态分子表示以及分子描述在大模型中的应用进行了研究。

1.2 国内外研究现状

分子性质预测的早期研究主要依赖于统计学和物理化学的方法。定量结构-活性关系[19]（QSAR）模型是一种基于统计学的方法，使用一组预定义的分子描述符来预测分子活性。这种方法的主要挑战在于选择合适的描述符和统计模型。尽管早期的 QSAR 模型通常使用线性回归[20]或逻辑回归[21]，但现代的 QSAR 模型已开始使用更复杂的机器学习方法，如支持向量机[22]和随机森林[23]。分子动力学模拟[24]是一种基于物理的方法，使用牛顿运动定律来模拟分子的运动，这种方法可以提供分子的详细动态信息，例如分子的结构、能量和动态性质[25]。然而，由于其高计算成本，这种方法通常只适用于小分子或短时间尺度的模拟[26-27]。量子力学计算是另一种基于物理的方法，使用薛定谔方程来计算分子的性质[28-29]，这种方法可以提供分子的精确信息，例如分子的能级、电子分布和反应能量。然而，这种方法的计算成本也非常高，通常只适用于小分子的计算。还有一些集中在开发手工特征，如扩展连通性指纹[30]，库仑矩阵[31]和对称函数[32]。由于这些方法泛化能力和可扩展性较差，当面对复杂多样的问题时，这些方法也有它们的局限性。目前，分子的图结构和文本表示在深度学习中得到了广泛的应用。在以下的部分，本小节将详细阐述这些工作。

在分子图表示学习领域，近年来的工作主要集中在图神经网络（Graph Neural Networks）、自监督学习预训练等方向。图神经网络是一种强大的深度学习模型，可以直接在图结构数据上进行学习，这种模型能自主学习图的复杂特征，例如节点的局部环境和拓扑结构等。图卷积最初被用来做监督学习，提取图结构数据特征并学习图表示，如 GraphConv[33]和 Weave[4]。近年来，许多基于 GNNs 的模型被提出，包括, 图注意力网络[34]，图同构网络[35]等。这些模型在结构和性能上都有所不同，为分子图表示学习提供了更多样的选择。自监督学习是一种无监督学习的方法，通过设计预测任务来学习数据的分布。在分子图表示学习中，自监督学习可以通过预测分子的一部分，例如一个原子或一个化学键，来学习分子表示。最近，一些研究者提出学习分子图的无监督表示。相关研究采用了对比学习的方法对分子图数据进行了预训练[36]，引入了三种用于分子图的数据增强方法[37]。还有一些研究提出了一些预训练策略，如属性掩蔽[38]和上下文预测[39]等。此外，还有一些基于自监督学习的方法被提出，例如 InfoGraph[40]和 GraphCL [38]。这些方法充分利用了无标签数据的信息，提高了分子图表示的质量。Transformer 是一种利用自注意力的预训练模型，在分子图表示学习中，Transformer 可以通过在大规模

无标签数据上进行预训练，然后在小规模标签数据上对模型进行微调。近年来，许多基于 Transformer 的方法被提出，例如 Molecule Transformer[41]，ChemBERTa [42]和 ChemBERTa-2[43]。这些方法通过预训练微调的策略，有效地利用了大规模无标签数据和小规模标签数据，提高了分子图表示学习能力，以及分子性质预测的准确率。

在分子文本表示学习领域，SMILES 是一种用于描述分子结构的文本表示方式。它使用特定的符号和规则将分子的原子和化学键编码成一串字符。这些研究将 SMILES 视为文本并使用最新的自然语言处理技术将它们转化为分子表示。然而，由于 SMILES 是一种无结构的文本数据，直接在 SMILES 上应用传统的机器学习方法，比如支持向量机和随机森林等，通常效果不佳。因此，需要一种能够理解 SMILES 序列结构的模型，深度学习为处理 SMILES 提供了一种有效的方法。最初，循环神经网络[44]（Recurrent Neural Network, RNN）被用于处理 SMILES。RNN 可以处理任意长度的序列数据，并能捕捉序列中的长距离依赖关系。例如，Duvenaud 等人[45]提出了一种基于 RNN 的模型，用于预测分子的各种性质。最近，MolGPT[46]模型采用了生成式方法对分子序列进行预训练并预测分子性质。还有一些基于自监督学习的方法，例如掩码语言模型（Masked Language Model）它通过预测序列中被掩盖的部分来学习词的表示，最终通过池化得到总体句子的表示。SMILES-BERT [47]和 ChemBERTa[42]使用掩蔽语言模型来提取分子特征，然后在小规模标签数据上进行微调。

在多模态分子表示学习领域，最近的研究引入了用于分子表示的多模态预训练模型[48-50]，这些模型的都提出从分子图和 SMILES 两种模态中融合信息。虽然 SMILES 和分子图这两者是根据黄金法则转换的，本质是相同的物质，但处理不同类型数据的模型能捕捉到的分子特征信息是不一样的。SMILES 中的一些非原子符号信息，例如类似手性结构的表示使用的”\”和”/”，或表示立体异构的”@”，人为赋予它的意义令语言模型在整个 SMILES 中很难根据上下文理解其含义，但在分子图中表示的就非常明确，分子图所具备的拓扑信息可以很容易的表示这些关系。因此这些多模态方法可以在多维信息中捕获更广泛的分子特征。然而，这些研究的主要问题在于不同模态的表征嵌入到同一子空间时，它们之间往往是有距离的，这种距离就是本文提到的对齐问题，这导致了模型很难将多模态表示融合。除了多模态，分子在不同的粒度上也包含丰富的化学信息，比如官能团结构普遍意义上决定着化学分子的性质。MolCLR [36]提出在分子级别表示分子，MPG [51]关注子图级别的匹配任务，而 K-bert [52]则主要强调原子级别的属性预测。然而，这些模型在他们的预训练策略中缺乏明确的粒度，并且缺乏相关的领域知识，阻碍了模型利用分子本身结构的特征信息，这也是导致模型泛化能力差的原因。

大语言模型在最近一段时间有了突破性的进展，通过在大规模语料库上预训练 Transformer 模型，提出了预训练语言模型（Pre-trained Language Models，PLMs）[53]，在解决各种自然语言处理（Natural Language Processing，NLP）任务方面显示出强大的能力。大语言模型可以根据模型结构分为三类，第一类是基于编码器的模型架构，训练方式采用掩码语言模型，比较常见的预训练任务是预测掩码单词，代表模型有 BERT，RoBERTa [54]。第二类是基于解码器的模型架构，这种模型主要用于生成式任务，采用自回归语言模型的训练方式，预测下一个出现的单词，代表模型有熟知的 GPT[8]系列，LLaMa[55]等。第三类是基于编解码器的模型架构，它与传统 transformer 相似，通常具有更强的序列学习和生成能力，尤其擅长实现输入序列到输出序列的结构映射，所以在机器翻译、聊天机器人等任务上有更好的应用前景，常见的编解码器大语言模型有 T5 [56]，GLM [57]。在分子化学领域，也有很多将分子表示与大语言模型结合的研究，例如有对分子进行翻译的 MolT5 模型，它可以给大模型输入 SMILES，大模型可以对该分子做出一些描述和解释，也可以根据用户输入的分子描述生成相对应的分子。还有可以预测化学反应生成物和反应物的模型[58]。以及像 Drugchat[59]的模型，在药物分子图上实现类似于 ChatGPT[60]的药物问答系统，使用者输入药物分子图，并向该系统提出关于这个药物分子的相关问题，DrugChat 将会多轮交互的方式解答用户所提出的这些问题。

1.3 主要工作内容

本文主要针对多模态药物分子表示以及性质预测问题展开研究，药物分子多模态包括一维药物分子 SMILES 文本数据和二维药物分子图数据，本文的主要工作内容包含以下两点：

（1）基于对比学习的多模态分子性质预测模型。现有的分子表示方法主要分为文本 SMILES 表示和分子图表示两类，以往的分子表示模型没有充分考虑多模态信息，单独使用一种会丢失分子特征，不利于分子预测的准确性。因此本文首先提出基于对比学习的多模态分子性质预测模型，利用多模态融合表示与预训练相结合提升分子嵌入的表达能力。为了保证多模态表征在特征空间中对齐，本模型通过对比学习的方法来缩小不同模态分子嵌入的距离，达到多模态表示对齐的目标。另外为了让不同特征能够有效融合，本模型在多模态编码器中加入了交叉注意力，使得模型能够将不同模态之间的表示相互注意，并充分融合。在预训练阶段，本文设计了三种不同粒度的预训练策略，分别为基于原子的属性预测，基于子结构的官能团预测以及图文对匹配任务，通过多级预训练任务最大限度的利用已知分子的结构特征，捕获重要信息。本文提出的模型受益于在大量数据上的

预训练以及从多模态分子表示中提取的丰富信息，在分类和回归两种类型的下游任务中均表现出良好性能。

（2）提出了一种基于提示语微调的分子性质预测模型。目前对于预训练微调模型存在着模型训练成本和参数存储成本高的问题。因此本文首先引入少样本学习，不再需要为每个下游任务训练单独的模型，仅需将构建的提示语输入大语言模型，即可对分子性质进行泛化学习。对于提示语，主要包含两个部分，第一部分为分子描述，通过检索算法从外部数据库中即可获得。第二部分为待预测分子嵌入，通过 KNN 算法在待预测的分子与正例和负例之间构建出一个关系图，采用 GNN 更新节点表示得到最终嵌入表示。最终，使用已训练好的大语言模型对所有分子进行对应的性质预测，在利用大语言模型丰富的知识的同时，减少了模型的训练成本和参数存储成本。最后，在真实数据集中的实验结果表明，本文提出的融合领域知识的医疗文本分类算法相比 4 种基线算法效果更优异。

1.4 本文组织结构

本文共分为五章，每章节的主要内容如下：

第一章：绪论，本章节首先介绍了药物分子性质预测的背景与意义，然后归纳了药物分子性质预测领域的国内外现状，分析了该领域现存在的一些问题，最后介绍了本文的主要内容和组织结构。

第二章：相关理论和技术。本章节主要介绍了本文涉及到的相关技术和理论，其中包括基于 Transformer 的语言模型、图表示学习和提示语学习。此外还介绍了本文实验中用于评估模型性能的评价指标。

第三章：提出了基于对比学习的多模态分子性质预测模型。本章节首先进行了模型概述，指出了现有使用单模态分子表示方法导致的特征缺失和多模态表示不能对齐的问题。然后从文本编码，图编码和融合编码三个方面对该模型进行了详细的描述和解释，介绍了对比学习策略和多粒度预训练策略。最后在分类和回归两种下游任务数据集上对该模型进行了微调和实验验证。

第四章：提出了基于提示语微调的分子性质预测模型。本章节首先进行了概述，指出了现有的丰富分子描述特征未被充分利用，以及现有的预训练微调模型泛化能力不强的问题。之后对模型的不同模块进行了详细的阐述，最终在多个下游分类任务数据集上对该模型实验验证。

第五章：总结和展望。本章节对本文的主要工作进行了总结，并反思了不足之处，最后对未来的研究工作做出了展望。

2 相关理论和技术

本章主要对基于 Transformer 模型架构的自然语言处理模型、图表示学习，提示语学习相关算法等相关理论和技术进行相应的介绍，为后续的研究提供理论支持。

2.1 Transformer

Transformer 是一种在自然语言处理领域广泛应用的深度学习模型架构。它最初在 2017 年中被提出，由 Google 的研究人员设计。这种模型架构的主要优点是能够处理序列数据，而且不需要像循环神经网络那样依赖于序列中的时间步骤。与传统的循环神经网络和长短期记忆网络相比，Transformer 完全摒弃了循环结构，使得模型可以更好地利用现代计算机硬件进行并行计算处理。这种设计显著减少了训练时间。

Transformer 的核心是自注意力（Self-Attention）或者多头注意力（Multi-Head Attention）的机制。Transformer 通过自注意力机制能够捕捉序列中任意两个元素之间的关系，无论它们的位置有多远。这使得 Transformer 在处理长序列时能够有效地理解和保持长距离依赖关系。

Transformer 架构可以很容易地扩展至大规模数据集和模型，例如 BERT、GPT 系列等。这些模型在多种 NLP 任务上都取得了突破性的进展而且 Transformer 架构可以通过调整模型大小，Transformer 编码器层数，解码器层数，隐藏层嵌入维度等，来适应不同的计算资源和性能需求，从而使其在不同的应用场景中都能够发挥作用。

Transformer 模型由编码器和解码器两部分组成。编码器负责将输入序列转换成一种连续的表示，这种表示捕捉到了输入序列中的各种语义信息。解码器则负责根据这种连续的表示通过交叉注意力的方式生成输出序列。在训练阶段，编码器和解码器是同时进行训练的，使得模型能够在生成输出序列时考虑到输入序列的全局信息，解码器的输出通过最后一个线性层和一个 Softmax 层，将每个位置的向量转换为预测的下一个单词的概率分布。在推理过程中，模型会逐步生成输出，每次生成一个单词，并将其作为下一步的输入，直到生成结束符号或达到最大长度。需要注意的是，解码器在预测时，存在一个 mask 矩阵，该矩阵会对后续答案进行掩蔽，防止模型在预测输出时观测到正确答案，导致推理过程作弊。Mask 矩阵往往是一个上三角矩阵，上三角的值均为负无穷大，这样做的原因是为了让计

算结果通过 softmax 函数计算后输出接近于零，从而在计算加权和时不会对结果产生影响。如图 2.1 是 Transformer 的整体架构图。

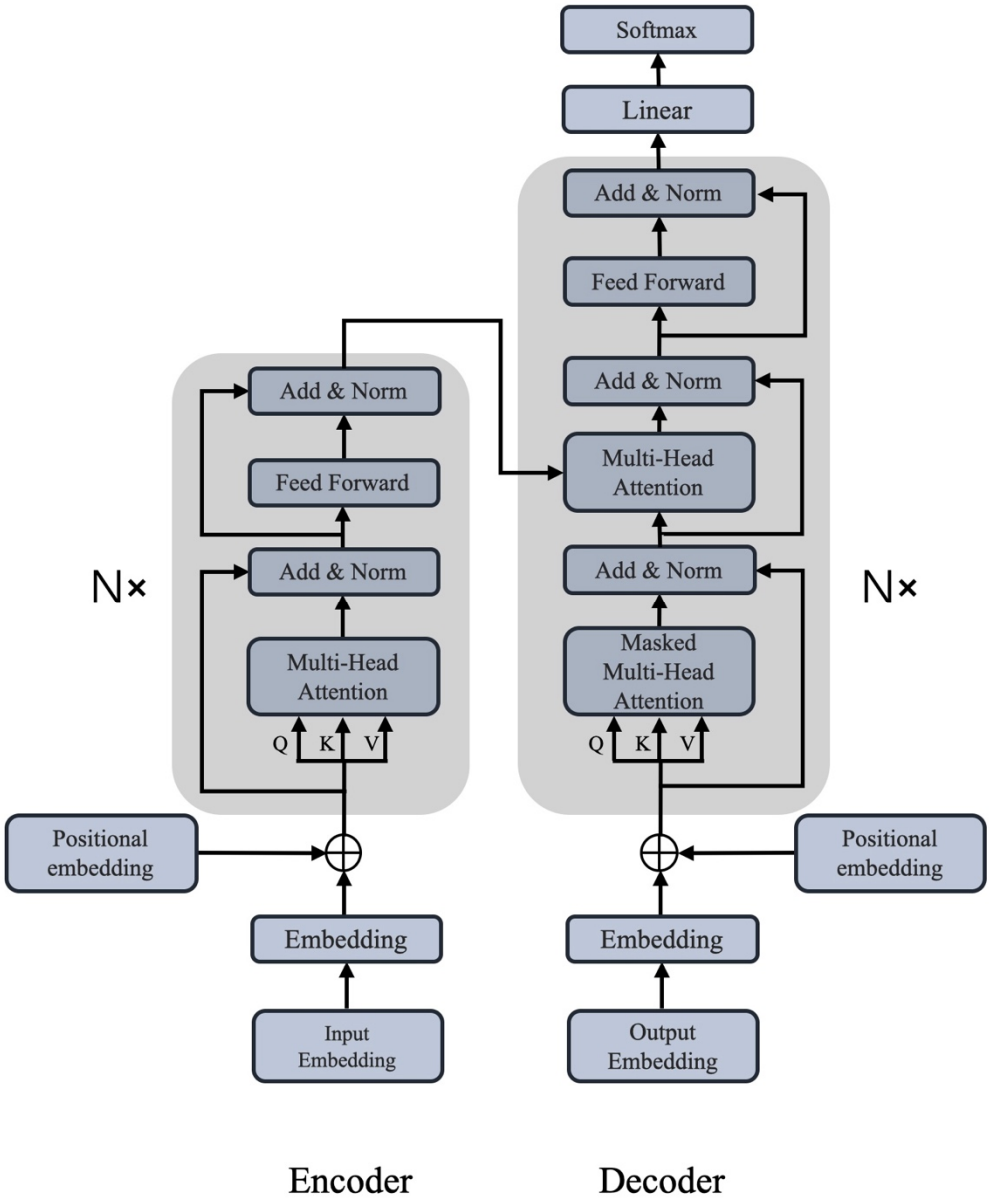


图 2.1 Transformer 架构

Fig 2.1 Transformer

2.1.1 编码器

在 Transformer 模型的编码器部分，每个编码器层主要包含两个子模块：自注意力机制（Self-Attention）模块和前馈神经网络模块（Feed-Forward Neural Network）。

自注意力机制是 Transformer 的核心组成部分，它允许模型在编码一个单词时，考虑到输入序列中所有单词的信息。这是通过计算输入序列中每个单词与其他所有单词的注意力分数来实现的。具体来说，对于输入序列中的每个单词，首先计算其“查询”（Query）、“键”（Key）和“值”（Value）向量。然后，通过计算查询向量与所有键向量的点积，得到注意力分数。最后，对注意力分数进行 Softmax 归一化，并用它们对所有值向量进行加权求和，得到该单词的输出向量。

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathbf{V} \tag{2.1}$$

其中，$\mathbf{Q}$，$\mathbf{K}$ 和 $\mathbf{V}$分别代表查询、键和值矩阵，$\sqrt{\mathrm{d_k}}$是键向量的维度。

前馈神经网络是一个简单的两层全连接网络，它对自注意力的输出进行进一步的处理。这个网络对每个位置的输入是独立处理的，也就是说，它不会改变输入的顺序。公式如下：

$$\text{FFN}(x) = \max(0, x\mathbf{W}_1 + b_1)\,\mathbf{W}_2 + b_2 \tag{2.2}$$

其中$\text{FFN}(x)$表示对输入x执行的前馈神经网络函数。$x\mathbf{W}_1 + b_1$表示输入x与权重$\mathbf{W}_1$的矩阵乘法，然后加上偏置b_1。$\max(0, x\mathbf{W}_1 + b_1)$表示 ReLU 激活函数，它将所有负值置为 0。然后，结果与第二层权重W_2进行矩阵乘法，并添加第二层偏置b_2，得到最终的输出。

由于 Transformer 模型本身并不具有捕获序列中的位置信息的能力，也就是说无论如何调整输入序列的顺序，自注意力机制的输出都是一样的。因此需要通过位置编码来提供这些信息。在原始的 Transformer 模型中，位置编码使用了一种基于正弦和余弦函数的方法。对于输入序列中的每一个位置pos，其位置编码是一个d维的向量其中d是模型的嵌入维度。这个向量的第$2i$元素和第$2i+1$元素分别由以下两个公式计算得出：

$$PE_{(p,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right) \tag{2.3}$$

$$PE_{(p,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{2.4}$$

其中i表示向量的维度，i从 0 开始，对于每一个d维的向量，i的取值范围是$[0, \mathrm{d}/2]$。

这种位置编码的方式也是可以处理任意长度的序列的，正弦和余弦函数是周期函数，所以对于任意大的pos，都能得到合理的位置编码。最后，这个位置编码会被加到每个词的嵌入向量上，这样模型在处理每个词的时候，就能同时考虑到这个词的内容和位置信息。

2.1.2 解码器

在 Transformer 模型中，解码器（Decoder）主要包含三个模块：自注意力模块、编码器-解码器注意力模块和前馈神经网络模块。自注意力的计算与编码器完全相

同。编码器-解码器注意力模块，这个模块的主要作用是让解码器能够使用到编码器的输出信息。具体来说，它会计算解码器的每一个位置的元素与编码器输出序列中所有位置元素的关联性。这样可以让模型在生成新的元素时，能够考虑到输入序列和编码器输出序列的全局信息。其计算公式与自注意力机制相同，只不过**K**和**V**的输入是编码器的输出。

在 Transformer 模型中，为了防止解码器在预测下一个位置的输出时看到未来的信息，该模型需要使用一个掩码的机制。这个掩码是一个矩阵，与注意力机制中的查询和键的点积得到的注意力矩阵大小相同，用于掩盖未来位置的信息。具体来说，掩码矩阵中的每个元素对应于注意力矩阵中的一个元素。如果希望屏蔽某个位置的信息，就在掩码矩阵中的对应位置填入一个非常大的负数，该负数在经过 Softmax 函数后，会变成接近于 0 的值，从而在最终的加权和中，这个位置的信息就被有效地忽略掉。

2.1.3 基于 Transformer 的语言模型

在 Transformer 的基础上，根据语言模型的类型可以分为编码器语言模型，解码器语言模型，编解码器语言模型，这些都是基于 Transformer 模型的变体。

编码器语言模型的代表是 BERT，BERT 是由 Google 在 2018 年提出的一种预训练语言表示的方法，它使用了 Transformer 的编码器。BERT 的主要创新之处在于使用了双向的 Transformer 编码器和新的预训练目标。对于每个输入 Token，BERT 模型都会输出一个高维向量，这个向量是通过对输入 Token 和它的上下文进行编码得到的。BERT 模型的预训练任务分为两个：掩码语言模型(Masked Language Model，MLM) 和句子预测 (Next Sentence Prediction，NSP)。在掩码语言模型任务中,模型的输入序列的一部分 Token 会被随机地替换为特殊的[MASK] Token 嵌入，然后模型的任务是预测这些[MASK] Token 嵌入的原始值，该任务的损失函数为公式 2.5。这使得 BERT 模型能够学习到输入序列中每个 Token 的上下文表示。在句子预测任务中，模型的输入是两个句子，模型的任务是预测第二个句子是否是第一个句子的下一个句子。这使得 BERT 模型能够学习到句子之间的关系。

$$\mathrm{L_{MLM}} = -\frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}} \log \mathrm{P}\left(\mathbf{W_i}|\mathbf{W_{context}}\right) \tag{2.5}$$

其中，$\mathbf{W_i}$是被 mask 的词，$\mathbf{W_{context}}$是上下文词，N是被 mask 的词的数量。这个公式计算了模型预测被 mask 的词的负对数似然损失。

$$L_{\mathrm{NSP}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(y_i') + (1-y_i) log(1-y_i')] \tag{2.6}$$

其中，N是样本数量，y_i是第 i 个样本的真实标签（0 或 1），y_i'是模型预测的概率。

BERT 是第一个无监督的、双向训练的预训练模型。这使得 BERT 能够理解句子中每个词的上下文，其次 BERT 的预训练-微调模式在大量未标记的文本上进行预训练，然后在具体任务的数据上进行微调。这使得 BERT 能够在文本分类、实体识别、问答、情感分析等自然语言处理任务上取得良好的效果。

解码器语言模型的代表是 GPT（Generative Pretrained Transformer），是一个基于 Transformer 的自回归语言模型。GPT 基于 Transformer 解码器部分，由多层 Transformer 解码器堆叠而成自注意力机制使其捕获序列中的长距离依赖关系，并且计算复杂度较低。预训练阶段，模型在大规模的未标注文本数据上进行训练，学习语言的统计规律。具体来说，它使用自回归方法进行训练，基于前面所有已经生成的词，在每个时间步预测下一个词，损失函数为公式 2.7。这使得 GPT 能够生成流畅、连贯的文本。微调阶段，模型在特定任务的标注数据上进行训练，学习任务相关的知识。GPT 能够适应各种不同的 NLP 任务，包括文本分类、问答、文本生成等。GPT 的输入是一个词序列，表示为词嵌入向量。每个词嵌入向量都通过一个位置编码向量进行修改，以表示词在序列中的位置。位置编码向量可以是固定的绝对位置，也可以是可学习的相对位置。GPT 的输出是一个词的概率分布，表示下一个词的预测。这个概率分布是通过在最后一层的输出上应用一个线性变换和一个 Softmax 函数得到的。

$$\mathrm{L}(\theta) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \log \mathrm{P}\left(\mathbf{W}_{\mathrm{i}} | \mathbf{W}_{1:\mathrm{i}-1}, \theta\right) \tag{2.7}$$

其中，$\mathbf{W}_{1:\mathrm{i}-1}$是前$i-1$个词，$\mathbf{W}_{\mathrm{i}}$是第$i$个词，$\theta$是模型的参数，$N$是语料库中的总词数，模型会基于前面的词和模型的参数去预测下一个词。

GPT 作为一个生成模型，可以生成连贯、流畅的文本，这使得它在文本生成、对话系统、创作辅助等任务上表现出色。它也能够理解前后文的关系，这使得它能够处理长距离依赖、理解复杂句子结构，虽然深度学习模型通常被认为是黑箱模型，但 GPT 模型的注意力机制可以提供一定程度的可解释性，帮助理解模型的决策过程。

编解码器语言模型的代表是 T5（Text-to-Text Transfer Transformer），是 Google 在 2019 年提出的一种新型预训练模型。T5 模型的主要思想是将所有的 NLP 任务都统一为文本生成任务。无论是文本分类、问答、摘要，还是翻译等任务，都可以通过一种统一的方式来处理。这种方式被称为前缀自回归，即给模型一个提示（如“翻译：”或“问答：”），然后让模型生成一个文本作为输出。

T5 模型并未在 Transformer 架构上直接做出改变，而是改革了预训练模型的应用方式。传统预训练模型在预训练阶段通常需要预先知道未来要解决的具体任务，而 T5 在预训练阶段并不设定具体任务，而是学习大量无标签文本的语言模型。在

微调阶段，它会根据特定任务的有标签数据进行训练，以适应任务需求，从而使预训练模型的适用范围更广，灵活性更高。

在预训练阶段，T5 采用了因果语言模型，而非传统的 Masked Language Model。这意味着在生成单词时，模型只能看到该单词之前的单词，而不能看到之后的单词，这种方式更符合人类阅读文本的习惯，能够使模型生成的文本更接近自然语言。此外，T5 模型使用了相对位置编码，相比于传统 Transformer 模型的绝对位置编码，这种方式能更好地处理长序列，并且使模型对输入序列的长度更具有适应性。

2.2 图神经网络

在图数据结构中，节点通常代表实体，边则代表实体之间的关系。图可以有效地模拟现实世界中的许多系统，例如社交网络、蛋白质相互作用网络、学术引文网络和知识图谱等。而在本文中更多使用的是药物分子图的表示，一个图结构通常可以表示为$G = (V, E)$，其中V表示节点集合，E表示边集合。根据边的有向性可以将图分成有向图和无向图。在无向图中，边没有方向，表示的是双向关系。例如，在社交网络中，如果 A 和 B 是朋友，那么这种关系是双向的。而在有向图中，边有方向，表示的是单向关系。例如，在学术引文网络中，如果论文 A 引用了论文 B，那么这种关系是单向的，从 A 到 B 有引用关系，但从 B 到 A 没有引用关系。本文主要研究的药物分子图中，节点表示原子，边表示不同类型的化学键，如图 2.2 是 SMILES 和分子图的对应关系

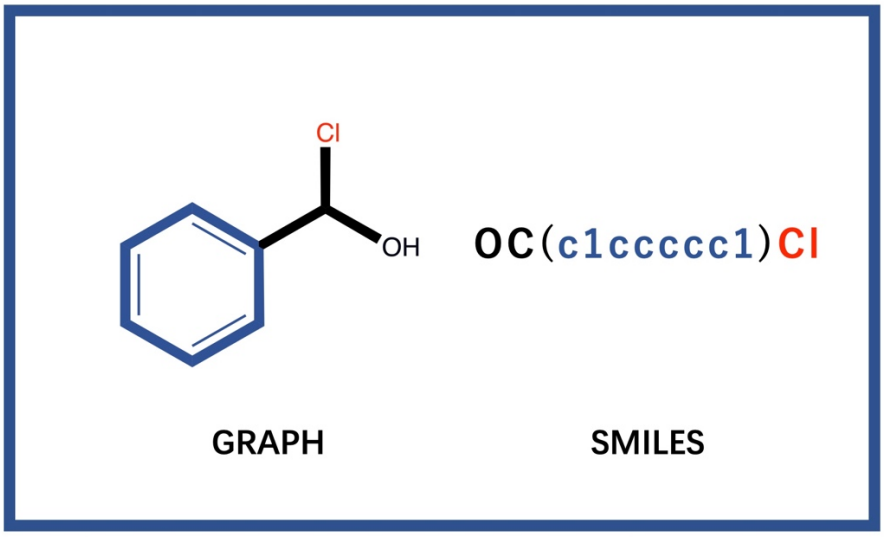


图 2.2 分子图和 SMILES

Fig 2.2 Molecular graph and SMILES

图神经网络（Graph Neural Networks，GNN）是一种深度学习模型，专门用于处理图数据。与传统的深度学习模型不同，GNN 能够直接在图形数据上进行操作，捕捉节点间的复杂关系。由于图的大小和结构可以非常不同，如何处理不同大小和结构的图也是一个挑战。为了解决这些问题，研究人员提出了各种各样的 GNN 变体和扩展，如图注意力网络（Graph Attention Networks，GAT）、图同构网络（Graph Isomorphism Networks，GIN）等。本文主要介绍，图卷积网络和图注意力网络。

2.2.1 图卷积网络

图卷积网络（Graph Convolutional Networks，GCNs）扩展了卷积运算，使其不仅适用于传统的网格数据（如图像），还适用于图数据。GCNs 的基本思想是学习一个函数映射$f(\cdot)$，通过这个映射$f(\cdot)$，图中的每个节点可以聚合自身的特征x_i和其邻居的特征x_j（$j \in N(v_i)$），生成新的节点表示v_i。具体来说，每个节点通过这个函数映射，能够整合其自身特征和周围节点的特征，从而得到一个新的、更全面的特征表示。这种新的特征表示能够捕捉到更多的上下文信息和节点之间的关系。图卷积网络是许多复杂图神经网络模型的基础，这些模型包括基于自动编码器的模型、生成模型以及时空网络等。这些模型都利用了图卷积网络的这种特性，以学习复杂的图结构和节点间的复杂关系。图 2.3 直观地展示了图神经网络学习节点表示的步骤。通过这种方法，GCN 模型可以有效地学习和理解图结构中的复杂模式和关系。

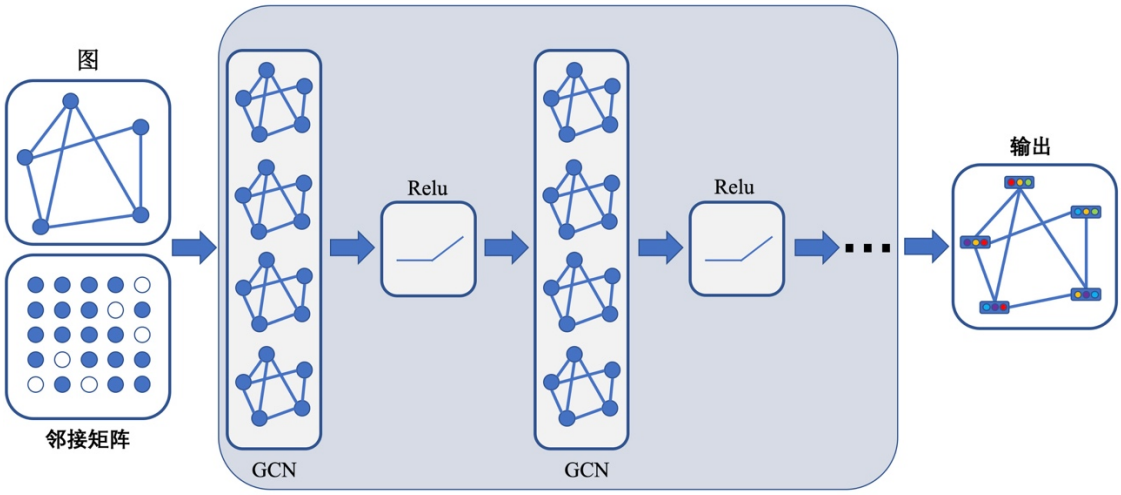


图 2.3 GCN 层

Fig 2.3 GCN layer

图卷积神经网络主要有两种类型：谱域方法和空域方法。基于谱的方法从图信号处理的视角定义图卷积，它采用滤波器来执行图卷积操作，这种操作在本质

上是从图信号中滤除噪声。另一方面，基于空间的方法将图卷积视为从邻近节点中汇集特征信息的过程。在图卷积网络的框架下，当算法在节点层面进行操作时，图池化（Graph Pooling）模块可以和图卷积层交替使用，将原始图粗化为更高级别的子结构。种架构设计可以用于提取图的多层次表示，并执行图分类任务。这种架构设计不仅可以提取出图的各级别表示，还可以执行图分类任务，这是因为它能够捕捉到图的全局结构信息和局部特征信息，从而生成更全面的图表示。

（1）谱域方法

谱图卷积网络是一种基于图的傅里叶变换的图神经网络。它的核心思想是在图的谱域，即图拉普拉斯矩阵的特征向量组成的空间上进行卷积操作。谱图神经网络中的图均视为无向图，无向图的一种常见并且鲁棒的数学表示方式是使用正则化图拉普拉斯矩阵，正则化图拉普拉斯矩阵是图拉普拉斯矩阵的一种变体，它的定义为：

$$L_{\text{norm}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \tag{2.8}$$

其中$\mathbf{D}$度矩阵且$\mathbf{D}$为对角矩阵$\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$，$\mathbf{L}$是拉普拉斯矩阵，$\mathbf{A}$是邻接矩阵

正则化图拉普拉斯矩阵确实具有实对称半正定的性质。这意味着它可以通过特征值分解或者奇异值分解进行分解。对于一个实对称矩阵L_{norm}，可以将其分解为：

$$L_{\text{norm}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{2.9}$$

其中，$\mathbf{\Lambda}$是一个对角矩阵，其对角线上的元素是L_{norm}的特征值，$\mathbf{U}$是一个正交矩阵，其列是L_{norm}的归一化特征向量。

在图信号处理中，一个图的信号可以被看作是定义在图的节点上的函数。如果把这个函数记作x，那么$x(i)$就表示第i个节点的信号值。对于图的傅里叶变换，如果把$L = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$作为图拉普拉斯矩阵的特征分解，那么图信号$x$的傅里叶变换可以被定义为：

$$\hat{x} = \mathbf{U}^T x \tag{2.10}$$

其中$\hat{x}$是x的傅里叶变换。

图卷积是一种特殊的卷积，它在图的结构上进行。对于输入信号x，假设g_θ是一个参数为θ的滤波器，那么滤波器在图上的卷积操作定义为：

$$g_\theta * x = \mathbf{U}g_\theta\mathbf{U}^T x \tag{2.11}$$

其中，$\mathbf{U}$是正则化图拉普拉斯的特征向量矩阵，$\mathbf{U}^T x$是图傅里叶变换。

首先在傅里叶空间中对输入信号进行滤波，然后再通过逆傅里叶变换将滤波后的信号转换回原始空间。基于谱的图卷积神经网络方法的一个主要缺点就是需要将整个图加载到内存中以执行图卷积。这是因为，基于谱的方法需要计算图的拉普拉斯矩阵的特征向量，这个过程需要知道整个图的信息。在处理大型图时，

这种方法将非常消耗内存和计算资源，因此并不高效。为了解决这个问题，研究者们提出了一些基于空间的图卷积神经网络方法。

（2）空域方法

基于空间的图卷积神经网络中比较经典的方法是由 Jure Leskovec 和他的团队在 2017 年提出的 GraphSAGE[61]，其模型示意图如下。

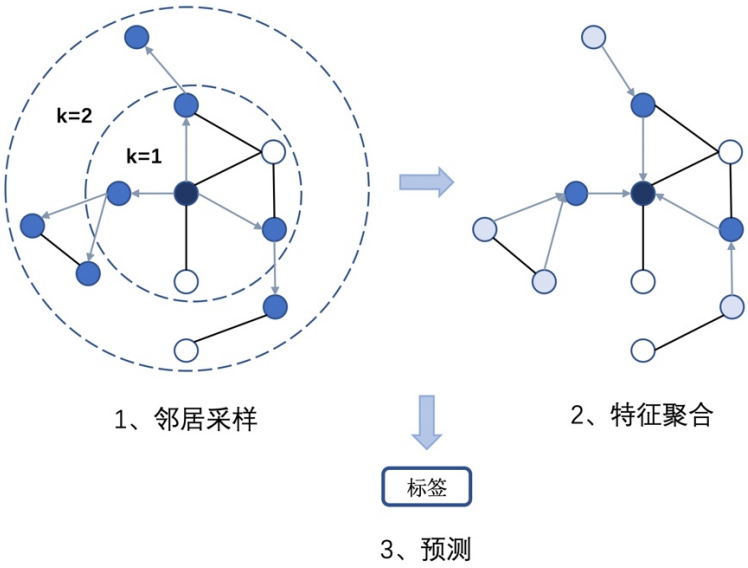


图 2.4 GraphSAGE 层

Fig 2.4 GraphSAGE layer

GraphSAGE 的主要目标是学习一个函数，该函数可以从节点的局部网络邻域生成节点的低维嵌入。该方法对于每一个需要生成嵌入的节点在其邻域中均匀地采样固定数量的邻居节点。这个过程会在多层进行，每一层都会采样邻居的邻居，这样可以得到一个固定大小的邻域，即使原图的度分布非常不均匀。在每一层，GraphSAGE 会聚合节点和其邻居节点的特征来生成新的特征。这个聚合函数可以是元素级别的平均、求和、最大值，也可以是更复杂的函数，如 LSTM[62]。这个过程可以表示为：

$$h_{N(v)}^{(k)} = \text{AGGREGATE}^{(k)}\left(\{h_u^{(k-1)}, \forall u \in N(v)\}\right) \tag{2.12}$$

其中，$h_{N(v)}^{(k)}$是节点v在第k层的邻居节点的聚合特征，$N(v)$是节点v的邻居节点，$h_u^{(k-1)}$是节点u在第$k-1$层的特征。

在每一层，GraphSAGE 会使用一个非线性变换来更新节点的特征，这个非线性变换会考虑节点自身的特征和其邻居节点的聚合特征。这个过程可以表示为：

$$h_v^{(k)} = \sigma\left(\mathbf{W}^{(k)} \cdot \text{CONCAT}\left(h_v^{(k-1)}, h_{N(v)}^{(k)}\right)\right) \tag{2.13}$$

其中$h_v^{(k)}$是节点v在第k层的特征，$\mathbf{W}^{(k)}$是第k层的权重，σ是非线性激活函数，CONCAT 是连接操作。

GraphSAGE 的主要优点是可以在大型图上进行训练，并且可以生成未见过的节点的嵌入。然而，它的缺点是需要预先定义邻居采样和特征聚合的方式，这可能会限制模型的灵活性。

2.2.2 图注意力网络

注意力机制是一种在深度学习模型中选择性地关注输入数据的重要部分的技术。这种机制的灵感来源于人类的视觉注意力机制，即人类在观察图像时会关注一些特定的、重要的部分，而忽略其它不重要的部分。在深度学习中，注意力机制的引入帮助模型更好地处理序列数据，特别是在处理长序列时，可以使模型更好地聚焦于当前任务相关的部分。

在图神经网络中，由 Velickovic 等人提出的图注意力网络（Graph Attention Network）是一种基于空间的图神经网络方法，主要目标是学习节点的嵌入，同时考虑其邻居节点的特征和其重要性。这种重要性是通过注意力机制来确定的，能够动态地调整邻居节点对目标节点特征更新的影响。图注意力网络主要将节点周围的邻居节点采样并给定一个权重矩阵对其进行训练，周围节点对中心节点的信息贡献程度由注意力系数决定，这样可以使信息最大化利用，有效邻域的权重会增大，而一般邻域节点的权重会减小，最终将不同的注意力权重与各自表示相乘并经过池化得到最终的中心节点表示。如图为 GAT 模型框架：

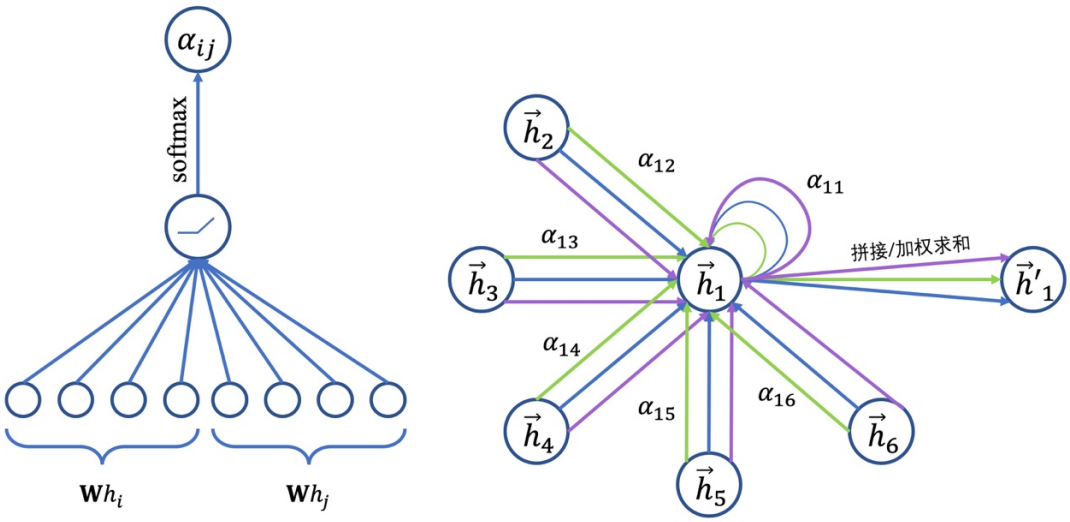


图 2.5 GAT 层

Fig 2.5 GAT layer

图注意力网络对于每个节点，首先应用一个线性变换来更新其特征，可以表示为：

$$h_i' = \mathbf{W} \cdot h_i \tag{2.14}$$

其中，h_i和h_i'分别是节点i的原始特征和更新后的特征，$\mathbf{W}$是权重矩阵。

对于每对节点i和j，计算一个注意力权重，该权重表示节点j对i的重要性。计算过程如下：

$$e_{ij} = \text{LeakyReLU}(\alpha^T[\mathbf{W}h^i||\mathbf{W}h^j]) \tag{2.15}$$

其中e_{ij}是节点j对节点i的注意力权重，α是注意力机制的参数，||表示连接操作，LeakyReLU 是非线性激活函数。

对于每个节点i，对所有邻居节点的注意力权重进行 softmax 归一化。这个过程可以表示为：

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in N(i)}\exp(e_{ik})} \tag{2.16}$$

其中，α_{ij}是归一化后的注意力权重，$N(i)$是节点i的邻居节点。

最终进行特征的更新，对于每个节点i用归一化的注意力权重对邻居节点的特征进行加权平均，然后应用一个非线性激活函数来更新节点的特征，该过程表示为：

$$h_i' = \sigma\left(\sum_{j\in N(i)} \alpha_{ij}\mathbf{W}h_j\right) \tag{2.17}$$

其中，h_i'为最终的节点特征，σ为非线性激活函数。

GAT 的主要优点是可以在大型图上进行训练，并且可以生成未见过的节点的嵌入。此外，它可以自动地确定邻居节点的重要性，而无需预先定义。然而，它的缺点是计算注意力权重需要更多的计算资源。

2.3 提示语学习

在过去，NLP 的任务主要依赖于预训练（Pretrained）和微调（Model Tuning）的模式进行解决。然而，这种模式需要对每个新的任务都训练一个新的模型，这既不能共享，也不够高效。对于预训练的大型语言模型来说，这就好像是为每个任务都进行了专门定制，效率低下。为了让已经训练好的大模型参数可以共享到每一个下游任务当中，减少计算资源的占用，并提升计算效率，研究人员提出了提示语学习，提示语学习的本质是通过提前给定大模型一些设置信息，这些信息中包括角色定义，下游任务描述以及输出指令模式。如果为包含 few-shot 样例，提示语中也要包含给定模型的提示样本，让模型参照给定的样例通过上下文学习来对特定任务进行预测输出。如图为预训练微调和提示语学习的对比。左侧是传

统的模型调整（Model Tuning）范式：对于每个不同的任务，都需要对整个预训练语言模型进行微调，每个任务都有一套独特的参数。右侧是提示语微调[64]的方法：对于各种不同的任务，只需要插入特定的提示语参数，每个任务都独立地训练这些提示语参数，而不是调整整个预训练语言模型。这种方法可以大大减少训练时间，并极大地提高模型的利用效率。

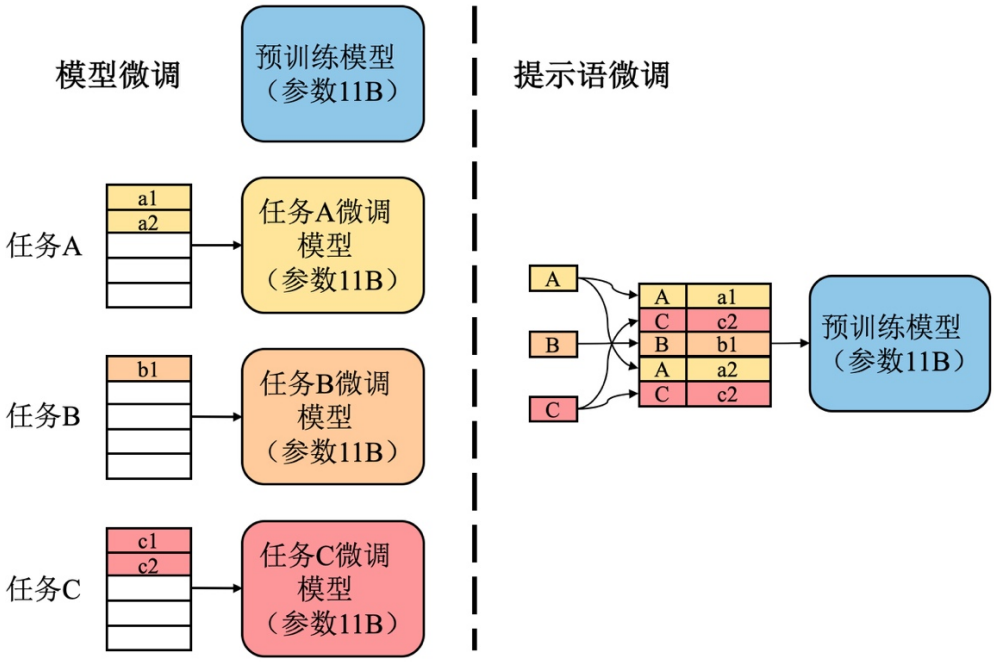


图 2.6 提示语微调和模型微调对比

Fig 2.6 Model tuning and prompt tuning

2.3.1 提示语模板设计

提示语模板的设计[65]就是为了构建一个适合特定下游任务的提示语函数$f_{prompt}(\cdot)$，这个函数的目标是将预训练的语言模型引导到特定的任务空间中，使模型能更好地解决这个任务。在提示语学习中，提示语函数通常被设计为一种语言模式，这种模式能够清晰地表述出任务的需求。例如，在问答任务中，一个可能的提示语是“问题：{问题} 答案：”，其中“{问题}”是要被替换的部分。

根据槽位的位置，可以将提示语模板的设计分为两类主要类型：完形填空式（Cloze）和前缀式（Prefix）。

在完形填空式这种类型的提示语中，空位[S]被放置在模板的中间位置。例如，对于一个问答任务，可能会设计一个像这样的提示语：“问题是：‘{问题}’，其答案是：‘[S]’。” 在这个例子中，模型的任务是找出可以填入空位[S]的正确答案。

在前缀类型的提示语中，空位[S]被放置在模板的末尾。例如，对于同样的问答任务，可能会设计一个像这样的提示语：“问题是：‘{问题}’，答案是：”在这个例子中，模型的任务是继续写下一段文本，作为问题的答案。

根据模板是否由人为指定，可以将提示语的生成方式分为两类：人为设计的模板和自动学习的模板。

在人为设计的模板中，研究人员或开发人员会根据任务的需求和模型的特性，设计出一种或多种模板来生成提示语。这种方式的优点是可以利用人的专业知识和理解来优化模板的设计，从而提高模型的性能。然而，它也有一些缺点，例如可能需要大量的人工努力，且对于一些复杂或未知的任务，可能难以设计出有效的模板。

在自动学习的模板[66]中，模型会通过学习数据自动生成提示语，包括离散的提示语和连续的的提示语，离散提示语为模型词表当中的离散字符，而连续提示语为连续的一组字符嵌入。这种方式的优点是可以自动适应各种任务，且不需要人工设计模板。但可学习提示语需要大量的训练数据，生成的提示语模板可能并不具有人类认知内的语义。

2.3.2 基于提示语的训练策略

大语言模型训练和任务适应的方式包括三种，分别是零样本（zero-shot）、少样本（Few-shot）和全数据（Full-data），在零样本下，模型在训练时并未使用到任何特定任务的数据。模型的训练通常依赖于大规模的、非任务特定的数据。然后，为了完成某个特定任务，模型会使用预设的提示来理解和执行任务。这种方式被称为″零样本″，因为对于特定的任务，并没有使用任何训练数据。少样本情况下，模型在训练时使用了少量的特定任务的数据。这些数据可以帮助模型更好地理解和完成任务。在全样本情况下，模型利用所有的训练数据来学习任务的各个方面，有更多的信息可以学习。然而，这种方法的一个潜在问题是过拟合，即模型可能会过度适应训练数据，而在新的、未见过的数据上表现不佳。

2.4 模型评价指标

2.4.1 分类任务评价指标

本文中的药物性质分类实验包含二分类，多分类以及回归任务，在介绍多分类评价指标之前先介绍二分类评价指标。

二分类混淆矩阵（Confusion Matrix）展示了模型预测与实际标签之间的关系，通常包含四个不同的组成部分：真正例（TP）、假正例（FP）、真负例（TN）和假负例（FN），具体形式参见表 2.1。

表 2.1 混淆矩阵

Table 2.1 Confusion matrix

	预测为正例	预测为负例
实际为正例	真正例（TP）	假负例（FN）
实际为负例	假正例（FP）	真负例（TN）

下面将介绍基于混淆矩阵计算的评价标准：

（1）准确率（Accuracy）：准确率是正确预测的数量除以总预测的数量。对于不平衡数据集，准确率可能会产生误导。例如，如果 99%的样本都属于同一类，那么一个将所有样本都预测为这一类的模型的准确率就会达到 99%，但这并不代表该模型具有良好的预测能力，计算公式如下：

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.18}$$

（2）精确率（Precision）：精确率是正确预测的正例（真正例）在所有预测为正例的样本中的比例。关注于预测为正例的样本的准确性，特别适用于对假正例非常敏感的情况。可能会忽略假负例的影响，即实际为正例但预测为负例的情况。

计算公式如下：

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.19}$$

（3）召回率（Recall）：召回率是正确预测的正例（真正例）在所有实际为正例的样本中的比例。关注于实际为正例的样本是否被准确预测，特别适用于对假负例非常敏感的情况。可能会忽略假正例的影响，即实际为负例但预测为正例的情况。

计算公式如下：

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.20}$$

（4）F1 分数（F1 Score）：F1 分数是精确率和召回率的调和平均值，可以同时考虑精确率和召回率。同时考虑了精确率和召回率，是这两者的调和平均值，因此在某种程度上克服了只看准确率、精确率或召回率的缺点。特别适合于数据不平衡的情况。F1 分数可能会掩盖精确率和召回率之间的差异。例如，如果精确率很高而召回率很低，或者反过来，F1 分数可能仍然看起来很好。因此，有时候也需要单独查看精确率和召回率。其计算公式如下：

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.21}$$

（5）ROC-AUC：ROC 曲线是通过将分类阈值从高到低移动，计算出一系列的假正例率（FPR）和真正例率（TPR），并以 FPR 为横轴，TPR 为纪轴画出的曲线。AUC 是 ROC 曲线下的面积，数值介于 0.5（随机分类器）和 1（完美分类器）之间。ROC-AUC 对正负样本的分布和阈值选择不敏感，因此在正负样本不平衡的情况下，通常比准确率更有用。ROC-AUC[67]越大，说明模型在区分正负样本方面的能力越强。

本文多分类相关任务采用采用宏平均（Macro Average）和微平均（Micro Average）：

（1）宏平均（Macro Average）：对每个类别单独计算评价指标，然后取这些指标的平均值。这种方法对所有类别赋予了相同的权重，无论类别中的样本数多少。因此，它对小类别的性能更敏感。具体计算如下：

$$\text{Macro-Precision} = \frac{1}{N}\sum_{i=1}^{N}\text{Precision}_i \tag{2.22}$$

$$\text{Macro-Recall} = \frac{1}{N}\sum_{i=1}^{N}\text{Recall}_i \tag{2.23}$$

其中，N是类别数，Precision_i和Recall_i是第i个类别的精确率和召回率。

（2）微平均（Micro Average）：它将所有类别的样本看作一个整体来计算评价指标。在计算过程中，它将所有类别的真正例、假正例和假负例累加起来，然后用于计算精确度和召回率。因此，微平均对样本数量较大的类别的性能更为敏感。其计算如下：

$$\text{Micro-Precision} = \frac{\sum_{i=1}^{N}\text{TP}_i}{\sum_{i=1}^{N}(\text{TP}_i + \text{FP}_i)} \tag{2.24}$$

$$\text{Micro-Recall} = \frac{\sum_{i=1}^{N}\text{TP}_i}{\sum_{i=1}^{N}(\text{TP}_i + \text{FN}_i)} \tag{2.25}$$

其中，TP_i，FP_i，FN_i是第i个类别的真正例、假正例和假负例的数量。

2.4.2 回归任务评价指标

回归任务的评价指标主要用于衡量预测值与实际值之间的差距或相似度。以下是一些常用的回归任务评价指标：

（1）均方误差（Mean Squared Error，MSE），预测值与实际值差值的平方的平均值。MSE 的公式为：

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.26}$$

其中，y_i是第i个观测值，$\hat{y}_i$是对应的预测值，n 是样本数量。

（2）均方根误差（Root Mean Squared Error, RMSE）：它是 MSE 的平方根，用于缩小 MSE 中较大误差的放大效应。RMSE 的公式为：

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{2.27}$$

（3）平均绝对误差（Mean Absolute Error, MAE）：它是预测值与实际值差值的绝对值的平均值。MAE 的公式为：

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \tag{2.28}$$

（4）R^2分数[68]，也被称为决定系数，衡量了模型预测的变量与实际结果的变量之间的相关程度。计算公式如下：

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.29}$$

其中，$\bar{y}$是目标变量的均值。

2.5 本章小结

本章对本文所涉及到的相关技术进行了介绍，包括 Transformer 架构的语言模型、图神经网络和提示语学习。此外，本章还对本文在药物分子性质分类和回归实验中使用的评价指标进行了介绍。

3 基于对比学习的多模态多粒度分子性质预测模型

3.1 概述

越来越多的行业将深度学习运用在了各自领域。传统的制药行业为了确保药物候选物在进入临床试验阶段之前具有合适的安全性和效能，要求科学家通过湿实验（Wet-lab Experimentation）来评估和优化这些药物分子性质，其中包括性、生物活性、代谢稳定性等。这是一件昂贵且耗时耗力的过程，如果利用深度学习进行初步的分子性质预测和虚拟筛选，将会加速药物的发现和开发过程，并降低制药成本。

想要使用深度学习的方法完成分子性质预测任务，至关重要的一步就是对分子进行表征。目前主流的表示方法包括基于 SMILES 的方法和基于分子图的方法。SMILES 是一种用于描述化学物质结构的序列信息，通常用于表示分子，将分子视为序列结构数据。该类型数据与自然语言形式相似，可以采用很多自然语言处理（Natural Language Process，NLP）的技术来对 SMILES 类型数据进行表征。NLP 技术不仅在自然语言领域有着强大的性能，在分子性质预测领域也有着不错的表现。基于分子图的方法将分子表示为具有节点（原子）和边（键）的图（Graph），而不是固定长度的特征向量。因此，许多处理图数据的深度学习方法也可以被应用在分子图表征上。然而，这两种方法输出的分子表示包含的信息并不完全相同。SMILES 中所标注的手性结构、同位素标记、特殊化合价原子等特征，是分子图难以表现的。但是分子图也有其独特的结构特征，它可以描述分子的拓扑结构，尤其是在药物分子中经常出现的单键或芳香键。因此，两种表示都有其不可替代性，单独使用一种会丢失一些分子信息，使得嵌入空间内分子的表征不够准确，不利于提高分子预测的准确性。

有研究人员采用了多模态思想，例如 GraSeq。该模型将不同模态的药物分子数据通过对应的编码模型得到多模态表示，并将其加和作为药物分子的最终嵌入。但是，不同空间的低维嵌入并不是对齐的，直接加和会丢失语义信息。为了解决这样的问题，研究人员提出了利用分子图中边的表示作为 SMILES Transformer 中的自注意力偏差，以达到更好融合多模态表示的目的，这种方法虽然将拓扑信息加入到了 Transformer 中，但仍然很难做到多模态表示是对齐的，并且，不同编码模型之间的参数并不能复用。如图 3.1 所示，将分子图不同的子结构或原子表示与 SMILES 表示中一一对应，蓝色代表苯环对应的 SMILES 中的表示为 C1CCCCC1。在原子级别上嵌入表示，对应于 SMILES 文本是将其中的字符进行嵌入表示。对于分子图就是对图节点进行嵌入表示。但对于嵌入到子空间的多模态向量来说，

并不是对齐的，也就是两个向量之间是有距离的。由于这两者的表示在本质上是同一个分子，因此当前采用这种多模态表示的方法大多没有考虑到对齐问题，只是进行简单的加和或者求平均等方法，并不能使分子表示的融合达到很好的效果。

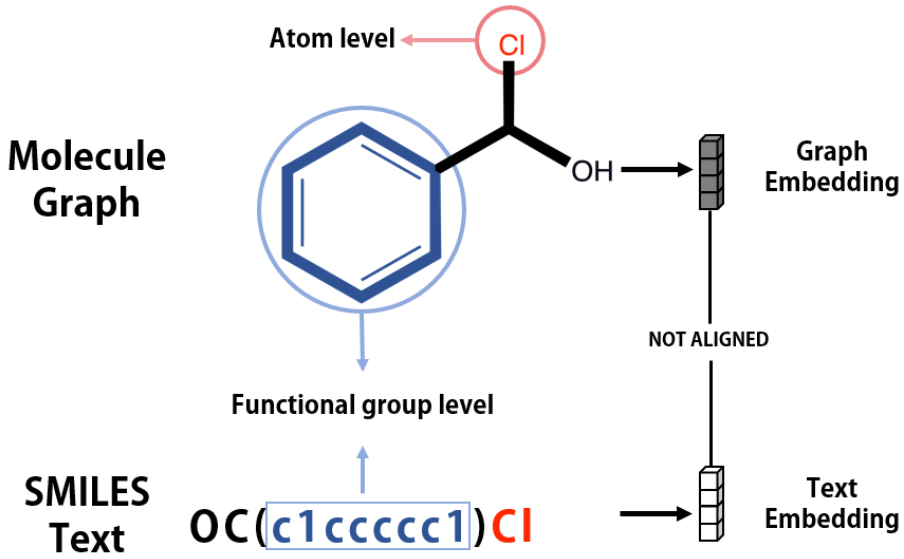


图 3.1 分子多模态对齐问题

Fig 3.1 Multimodal alignment problem

另外，与其他领域不同，想要获得药物分子性质需要大量的湿实验，从而导致具有标签的公共数据集非常少，这使得模型在这些数据集上进行训练很容易过拟合，模型的泛化能力较差。为了提高模型泛化能力，研究人员提出了自监督学习的技术，从大规模的无标签数据中预训练模型例如 motif-tree[63]，该模型的作者将药物分子图简化成树型结构，利用深度优先算法重构 motif 结构达到自监督的目的。还有 Hu 等人提出的节点级别和子图级别的预训练任务，由于该方法的子图是由中心原子 k 跳邻域采样而来，采样子图不具有化学意义，缺乏领域知识。目前来看，大部分研究的预训练策略粒度划分不够明显，实验效果欠佳。

在这项工作中，本章提出了一个基于多模态预训练的分子性质预测模型，解决了上述三个关键的问题。通过利用图和序列信息来学习药物分子的融合表示，以此来提取药物分子中更丰富的信息。其次，为了使图特征和文本特征在嵌入空间中对齐并最大化其互信息，提出了对比损失。本模型还加入了一个带有交叉注意力的多模态融合编码器，它可以帮助模型更好的融合不同模态的特征，让多模态编码器更容易执行跨模态学习。在预训练中本模型选择了三种不同粒度的预训练策略，并用到了领域知识，通过多级预训练任务最大限度的利用已知分子的结构特征，捕获重要信息，由此也解决了标签数据过少导致泛化能力较弱的问题。

为了减小资源的占用，提出了难区分负样本的采样策略，让模型从最有价值的数据中学习到更多的信息。本章模型在分类任务和回归任务中都有着不错的性能和准确的预测结果，与其他基线模型相比，本章模型的效果均超过了当前最优秀的基线模型。

3.2 模型描述

3.2.1 符号说明及问题描述

表 3.1 符号定义表

Table 3.1 Symbol definition

符号	含义
Q,K,V	查询矩阵，键值矩阵，值矩阵
$head_i$	多头注意力中的第 i 个头的输出
$N(v)$	节点v的邻居节点集合
$m_v^{(l,k)}$	是在第k次迭代中，节点v的第l层的隐藏状态
e_{uv}	节点u和v之间的边属性
L_{APP}	原子属性预测任务损失函数
L_{FGP}	官能团匹配任务损失函数
L_{GTM}	图文配对任务损失函数
L_{CON}	对比损失函数
$p_m^{g2t}(G)$	分子图 G 被映射到文本样例上的概率
$p_m^{t2g}(G)$	文本 T 被映射到分子图样例上的概率

在本章研究中面临的主要问题是如何有效地整合来自两种不同模态的分子信息，即分子的 SMILES 序列和分子图的嵌入表示。为预训练模型构建包含领域知识的有效预训练任务。并从不同粒度让模型可以学习到更加丰富的信息来构建分子表示，以预测分子的性质。

给定分子的 SMILES 序列 S 和分子图 G，首先分别对它们进行编码。采用 Transformer 编码器 f_S和 f_G来分别处理这两种模态的数据 $E_S = f_S(S)$和 $E_G = f_G(G)$，其中 E_S和 E_G分别代表 SMILES 序列和分子图的编码表示。由于 SMILES 序列和分子图的表示在特征空间还有一定的距离，本模型采用对比学习方法来最小化对比损失函数，使多模态表示在特征空间中对齐，采用的对比损失函数为$\mathcal{L}_{con} =$ ContrastiveLoss(E_S, E_G)。表示对齐后，使用交叉注意力机制来融合这两种模态的信息，从而得到一个综合的分子表示$E_M =$ CrossAttention(E_S, E_G)。交叉注意力机制允

许模型在两种模态之间相互学习，从而提取更丰富的分子特征。最后，将融合后的分子表示E_M用于分子性质预测。定义一个预测函数 p，将E_M做映射，预测值为$y_{pred} = p(E_M)$。最终的目标是最小化预测值y_{pred}和真实性质值y_{true}之间的差异$\mathcal{L}_{\text{pred}} = \text{Loss}(y_{pred}, y_{true})$。

3.2.2 模型概述

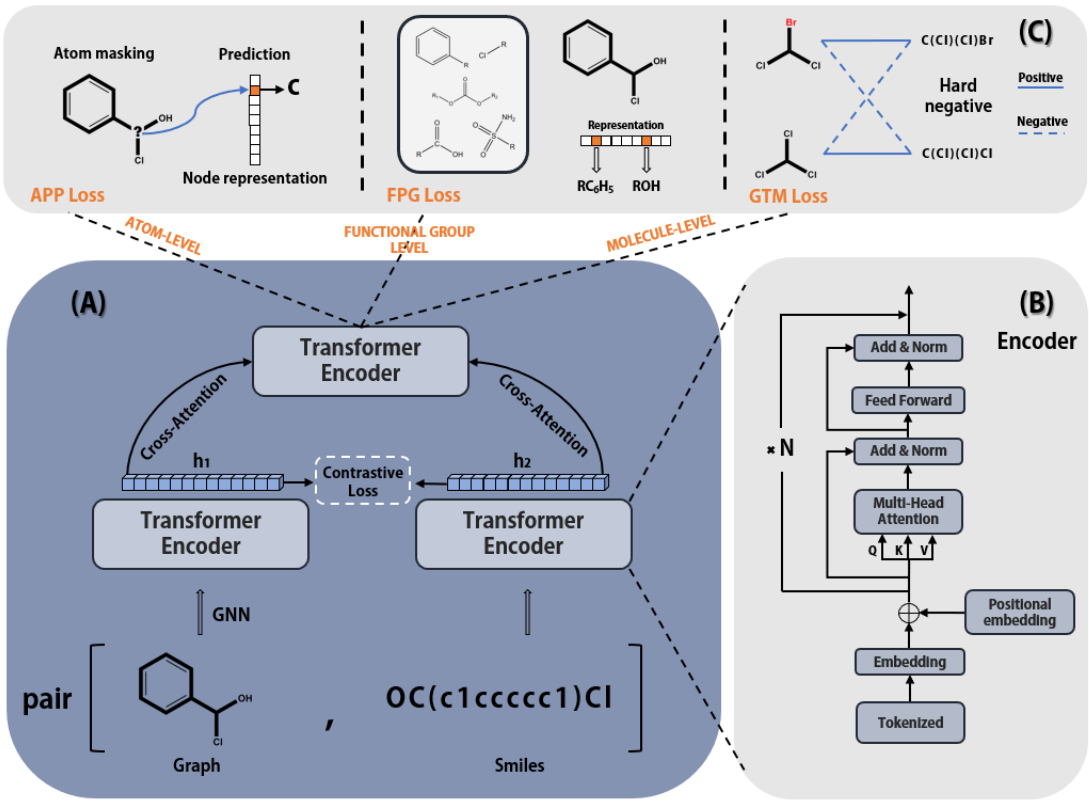


图 3.2 多模态对比学习模型

Fig 3.2 Multimodal contrastive learning model

本章模型结构如图 3.2 所示，主要分为三个部分：(1) 预训练模型将分子图和 SMILES 表示作为输入。Transformer 编码器用来对图数据和文本数据进行编码。使用对比损失对齐不同模态的嵌入向量并最大化他们的互信息。并使用带有交叉注意力的融合编码器得到分子最终的表示。 (2) Transformer 编码器的详细结构图。(3) 不同粒度的预训练任务，其中包括原子属性预测（Atom Property Prediction，APP），官能团预测（Fuctional Group Prediction，FGP），图文匹配（Graph Text Match，GTM）。

3.2.3 编码器

本模型设计了三个编码器，均使用 Transformer encoder 的基本框架，每个transformer编码器由六个transformer encoder layer组成，编码器具体模型如图3.2。其中两个编码器用于对图数据和序列数据进行编码表示，最后一个编码器加入交叉注意力模块对两个不同序列输入做融合编码。

（1）文本编码器，使用了 Transformer encoder，Transformer 是一个预训练模型，注意力机制是 Transformer 的主要组成部分，自注意力层采用一组 Queries，Keys，Values 作为输入，他计算 **Q** 和 **K** 的点积，并使用 Softmax 函数来获得值的权重，输出可以被看作矩阵：

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V} \tag{3.1}$$

其中 d_k 是 **K** 的维度，然而多头注意力允许模型联合关注来自不同位置子空间的信息，本模型采用多头注意力公式如下：

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(head_1,\dots,head_h)\mathbf{W}^O \tag{3.2}$$

$$head_i = \text{Attention}\left(\mathbf{Q}\mathbf{W}_\mathbf{i}^{\mathrm{Q}},\mathbf{K}\mathbf{W}_\mathbf{i}^{\mathrm{K}},\mathbf{V}\mathbf{W}_\mathbf{i}^{\mathrm{V}}\right) \tag{3.3}$$

其中，$\mathbf{W}_\mathbf{i}^{\mathrm{Q}} \in R^{d_{model}\times d_k}, \mathbf{W}_\mathbf{i}^{\mathrm{K}} \in R^{d_{model}\times d_k}, \mathbf{W}_\mathbf{i}^{\mathrm{V}} \in R^{d_{model}\times d_v}$，$\mathbf{W}^{\mathrm{O}} \in R^{hd_v\times d_{model}}$

不需要解码器部分来做自监督，只需要编码器模块得到SMILES的嵌入表示，SMILES 可以被看作自然语言处理中的句子，SMILES 中的每个字符可以被看作句子中的单词，将其输入 Transformer encoder 模型可以得到每一个单词的嵌入表示。

(2) 图编码器在 Transformer encoder 之前先使用了 GNN 对分子图中的节点信息进行邻域聚合，GNN 的本质是消息传递，通过聚合图节点 v 的相邻节点和边的隐藏状态来迭代更新节点 v 的隐藏状态 h_v。通常，消息传递过程涉及多次迭代，每次迭代可以进一步划分为几跳。形式上，在第 l 次迭代中，第 k 跳可以公式化为：

$$m_v^{(l,k)} = \text{AGGREGATE}^{(l)}\left(\left(h_v^{(l,k-1)}, h_u^{(l,k-1)}, e_{uv}\right)\middle| u \in N_v\right) \tag{3.4}$$

$$h_v^{(l,k)} = \sigma\left(\mathbf{W}^{(l)} m_v^{(l,k)} + b^{(l)}\right) \tag{3.5}$$

其中$m_v^{(l,k)}$是聚合的消息，$\sigma(\cdot)$为激活函数，$\text{AGGREGATE}^{(l)}(\cdot)$为聚合函数，聚合的方式[69]有很多种，可以是平均，最大池化，或者图注意力机制。

由于 Transformer 的输入是序列化的，这样做会丢失分子图中的拓扑结构，所以使用消息传递的框架聚合周围节点的特征可以保留图的拓扑信息。而且 Transformer 可以看作是构建在全连通图上的 GAT 模型变体，它能够聚合无连边节点之间的信息，弥补消息传递模型中无连边节点之间信息缺失的劣势。

（3）多模态编码器是一个关键的组件，它使用对比学习方法来对齐来自不同模态的表示。对比学习的目标是促进跨模态学习，使得在表示空间中相似的输入更接近。通过最小化不同模型嵌入之间的对比损失来实现这一点。计算图嵌入和

文本嵌入之间的相似性。使用相似性函数，将其视为余弦相似性，目标是最大化正样本的相似性得分。

$$sim(x,y)=\frac{x\cdot y}{|x||y|} \tag{3.6}$$

在这里，对于每一对图文，计算 softmax 归一化的相似性：

$$p_m^{g2t}(G)=\frac{exp\big(sim(G,T_m)\big)}{\sum_{m=1}^{M}exp\big(sim(G,T_m)\big)} \tag{3.7}$$

$$p_m^{t2g}(G)=\frac{exp\big(sim(T,G_m)\big)}{\sum_{m=1}^{M}exp\big(sim(T,G_m)\big)} \tag{3.8}$$

其中，G 和 T 分别代表图嵌入和文本嵌入, $\mathrm{y^{g2t}(G)}$ 和 $y^{t2g}(T)$ 作为真实值，其中负样本对的 label 为 0，正样本对的 label 为 1，对比损失函数被定义为 p 和 y 之间的交叉熵损失：

$$L_{CON}=-\left(\sum_{m=1}^{M}y^{g2t}(G)logp_m^{g2t}(G)+\sum_{m=1}^{M}y^{t2g}(T)logp_m^{t2g}(T)\right) \tag{3.9}$$

其中 M 是批次大小。

多模态编码器也同样基于 Transformer 编码器，引入了一个交叉注意力模块。交叉注意力的计算公式与自注意力相同。然而，自注意力是在同一序列的输入上进行操作，而这个交叉注意力以图嵌入和文本嵌入作为输入。交叉注意力使得来自不同输入序列的信息能够相互注意和融合，从而得到更全面的特征表示。要执行交叉注意力计算，只需要将输入修改为来自不同的序列即可。

3.2.4 模型预训练

对于模型预训练，采用了三种粒度的预训练策略：原子属性预测，官能团种类预测以及图-文本匹配任务，这三种预训练策略分别从原子级别，官能团级别以及分子级别对模型进行训练，不同的任务代表着不同的粒度，使模型可以学习到更丰富的信息。还采用了对比学习的方法，它将图特征和文本特征对齐，增大了正样本之间的互信息，使得多模态编码器更容易执行跨模态学习。

（1）原子属性预测，采用掩码语言模型（Mask Language Model，MLM）中的预训练策略，随机掩盖 15%的原子向量，并将他们替换成[mask]，对多模态编码器的对应位置输出进行原标签的预测，原子标签由 RDKit 计算生成，其中包括原子的度，芳香性，所含氢原子个数，是否包含手性以及手性类型等，因此原子属性预测可以被看作是多任务的分类任务，损失函数如下：

$$L_{APP}=-\sum_{i}^{N}\sum_{c=1}^{M}y_{ic}\,log(p_{ic}) \tag{3.10}$$

其中M为原子种类数量，N为原子数，c 表示原子种类，p_{ic}表示第i个原子的预测概率，y_{ic}为真实值。

（2）官能团种类预测，如图 2.3，根据领域知识，分子子结构往往决定了分子性质，对于预训练，需要从数据本身来挖掘特征和信息作为自监督任务的可靠标签，分子中非常重要的一种子结构就是官能团，并且这种结构可以利用 RDKit 检测到，对每个分子构建他的官能团标签，然后利用多模态编码器的[CLS]输出向量做官能团标签的预测，官能团预测任务是一个多任务分类，L_{FGP}仍然使用交叉熵损失函数。

$$L_{FGP} = -\sum_{i}^{N}\sum_{c=1}^{M} y_{ic}\, log(p_{ic}) \tag{3.11}$$

其中N表示分子数，c表示任务序号，p_{ic}表示第i个分子在任务c的预测概率，y_{ic}为真实值。

（3）图文本匹配，如图 2.3，预测一对图文是正样本（匹配）还是负样本（不匹配），使用多模态编码器的[CLS]词向量的输出作为一对图文的嵌入表示，将他通过一个全连接层并将其经过 SoftMax 函数去预测一个概率$\mathrm{p_{mat}}$通过交叉熵函数计算损失:

$$L_{\mathrm{GTM}} = -\sum_{i}^{N} y_{\mathrm{mat,i}} \log\left(\mathrm{p_{mat,i}}\right) \tag{3.12}$$

其中，N代表图文对的数量,$y_{\mathrm{mat,i}}$表示第i对图文对的真实值，$p_{\mathrm{mat,i}}$表示预测概率，即第i对图文对是互相匹配的（表示为正样本）。

本文提出了一个策略去采样难区分负样本，认为如果一对负样本在嵌入到低维向量后有着相似的表示，那么就认为这是一对难区分负样本，反之则为易区分负样本。难区分负样本对于模型来讲往往蕴含着更多的可利用信息，模型很难从容易区分的负样本中学到更多的东西，难区分样本应该是模型着重关注的对象，所以计算图文样本嵌入之间的相似度 sim，在一个 mini-batch 的相似矩阵中找到除对角线外的一个最大值并记录负样本对索引，让多模态编码器对该样本对进行融合编码并进行图文匹配任务。

3.2.5 模型微调

本章节所提出的模型经过大量无标签药物分子数据的预训练，已经学会了如何从 SMILES 和分子图中提取基本的分子特征，为了让模型适配不同的下游任务，并获得更好的性能，针对不同的下游任务对模型进行了微调。微调首先需要加载预训练好的模型参数，并选择冻结一部分 Transformer 的参数层。模型采用 12 层 Transformer 层，在微调阶段选择只更新每个编码器的最后一个 Transformer 编码器层，以及连接在多模态编码器模型最后的分类器，其他部分的参数不参与微调更新。模型选择多模态编码器的[CLS]向量输出作为整个分子图以及 SMILES 的融合特征表示，然后将[CLS]的表示输入多层感知器（MLP）也就是分类器来预测分子

属性，并根据预测值与标签计算交叉熵损失，回传计算梯度，并更新参数，完成在带标签的数据集上训练。最后，保存验证集效果最好的模型参数用于测试集计算准确度。

3.3 实验

3.3.1 实验环境

本章提出的基于对比学习的多模态多粒度分子性质预测模型采用 Python 语言编程，基于 Pytorch 深度学习框架实现。具体的实验环境如表 3.2 所示，预训练参数和微调参数设置在表 3.4 和表 3.5。

表 3.2 实验环境

Table 3.2 Experimental environment

名称	参数
处理器	Intel(R) Xeon(R) W-2235
主频	3.8GHz
内存	62GB
硬盘	2TB
图形处理器	NVIDIA RTX 3090 Ti
显存	24GB
操作系统	Ubuntu 20.04
开发语言	Python 3.6
深度学习框架	Pytorch 1.8.0

3.3.2 实验数据

本章实验选择了 8 个真实分子性质数据集，其中 5 个为分类预测数据集，3 个为回归预测数据集。详细的数据集信息如下：

（1）分类任务数据集：

BBBP（Blood-Brain Barrier Penetration）：血脑屏障渗透数据集源自一项关于屏障渗透性建模和预测的最新研究。作为分隔循环血液和大脑细胞外液的膜，血脑屏障阻挡了大部分药物、激素和神经递质。因此，穿透血脑屏障在针对中枢神经系统的药物开发中一直是一个长期存在的问题。这个数据集包含了超过 2000 种化合物的渗透性质，数据形式为 SMILES 串，以及对应是否可以渗透血脑屏障的标签，属于二分类数据集。

Tox21：该数据集包含了 8014 种化合物对 12 个不同目标的定性毒性测量，包括应激反应途径和核受体。

ClinTox：该数据集包含了由美国食品和药物管理局（FDA）批准的药物以及由于毒性原因未通过临床试验的药物的定性数据。

BACE：该数据库包含了一组人类 β-分泌酶 1 抑制剂的结合结果。β-分泌酶 1 是阿尔茨海默病中淀粉样蛋白质生成的关键酶。

SIDER：该数据集是一个市场上的药物和不良药物反应（ADR）的数据库，这些不良反应被分组到 27 个系统器官类别中。这个数据库提供了一种强大的资源，可以用于研究和理解药物的不良反应，以及它们如何影响人体的不同器官系统。

（2）回归任务数据集：

FreeSolv：提供了水中小分子的水合自由能数据。这个数据库中的一部分化合物也被用于 SAMPL 盲预测挑战。计算值是通过使用分子动力学模拟的炼金自由能计算得出的。

ESOL（Estimated Solubility）：该数据集是一个小型数据集，包含了 1128 种化合物的水溶性数据。水溶性是物质在水中溶解的能力，是药物设计中的一个重要参数，因为药物的水溶性影响其在体内的吸收、分布、代谢和排泄。一个药物的水溶性越高，其在体内的生物利用度通常也越高。

Lipophilicity：从 ChEMBL 数据库整理的亲脂性数据集，提供了 4200 种化合物在辛醇/水分配系数（在 pH 7.4 时的 logD）的实验结果。

下游任务数据集的分子数量及任务数量如下表，下游任务分为分类任务和回归任务，数据集分割方式采用 scaffold split 将下游数据集按照 8：1：1 的比例划分训练集，验证集和测试集。与常见的随机拆分不同，本章对于数据记的划分方式是基于分子子结构的，划分更加贴近现实，但也会让预测任务更具挑战性。

表 3.3 数据集统计信息

Table 3.3 Statistics of the real world datasets

数据集	分子数量	任务数量
BBBP	2039	1
Tox21	7831	12
ClinTox	1478	2
BACE	1513	1
SIDER	1427	27
FreeSolv	642	1
ESOL	1128	1
Lipo	4200	1

3.3.3 实验设置

本章根据分子性质预测中经常使用的技术选取了 7 个对比模型，如基于 DNN 的多任务框架，以分子指纹作为模型输入 TF_Roubus。使用了图卷积模型的 GraphConv，Weave 和 SchNet。使用了多模态特征提取架构，输入是分子图和 smiles 序列的 GraSeq。本章对比模型还挑选了一些使用预训练的模型，比如 MolCLR 利用分子图数据增广扩增数据，并做对比学习的预训练模型。还有一些受到 NLP 影响在序列表示上使用语言模型进行预训练的方法，比如 N-GRAM 和 SMILES-BERT。对比模型的详细信息如下：

TF_Roubust：是一种利用深度学习技术进行药物发现的方法。方法的核心思想是利用神经网络处理多任务学习，同时解决多个相关的预测任务。以分子指纹的形式输入用于表示分子的结构和性质，通过神经网络的学习，可以输出预测的药物性质，如生物活性、药代动力学性质、毒性等。

GraphConv：该模型提出了一种新的半监督学习方法，即图卷积网络。分子可以被看作是一个图，其中的节点代表原子，边代表化学键。通过在这个图上进行卷积操作，图卷积网络可以捕捉到分子内部的复杂结构和性质。

Weave：传统的分子指纹方法通常只编码了分子的一部分特性，而新的图卷积方法可以更全面地获取分子的信息。这种方法直接在分子的图表示上进行操作，每个节点代表一个原子，每个边代表一个化学键。这样，可以捕获分子的全局性质和局部性质，包括原子类型、化学键类型、原子的电荷状态等。它可以自动学习使得模型能够自适应地处理各种不同的分子结构，而不需要人工干预。

SchNet：提出将深度学习和量子化学相结合，这是一种专门为建模量子相互作用而设计的卷积神经网络，主要特点是其使用了连续滤波器，这使得它能够更准确地模拟分子的电子结构和量子动力学。通过在原子间的距离上应用卷积，以捕获分子的三维结构。这种方法的优点是，它可以自动地从数据中学习分子的重要特性，而无需人工设计特征。

GraSeq：在这篇文章中，作者们提出了一种名为 GraSeq 的新型深度学习方法，该方法结合了图神经网络和序列模型来预测分子性质。主要创新之处在于，它同时考虑了分子的图形结构和 SMILES。首先使用图神经网络来处理分子的图形结构，然后使用循环神经网络来处理分子的序列信息。这两种信息然后被融合在一起，用于预测分子的性质。

MolCLR: 该模型提出了一种基于图神经网络的分子对比学习方法，用于学习分子的表示。对比学习是一种无监督学习方法，它通过最大化正样本对之间的相似性并最小化负样本对之间的相似性来学习样本的表示。图神经网络被用来处理分子的图形结构，生成分子的向量表示。

N-GRAM：研究人员提出了一种名为 N-gram graph 的新型无监督图表示方法，在该模型中，一个图的 N-gram 是由 N 个连续的边组成的路径。这些 N-gram 被用来构建一个新的图，其中的节点代表原图中的 N-gram，边代表 N-gram 之间的重叠。然后，这个新图的特性被用来生成原图的向量表示。

SMILES-BERT：作者们首先将分子的 SMILES 表示转换为一系列的词，然后预训练一个编码器模型 BERT 来从这些文本中学习连续的向量表示。这个模型被训练来预测 SMILES 序列中的掩蔽单词，这是一种自监督学习任务，可以在没有标签数据的情况下进行。这些向量表示被用来预测分子性质，如溶解度、毒性等。

本章提出的预训练模型和微调模型超参数设置如下表：

表 3.4 预训练超参数

Table 3.4 The pre-trained hyperparameters

参数名称	值/范围
学习率	1e-5~5e-6
Batch 大小	64
Epoch	50
优化器	Adam
嵌入维度	768
注意力头数	8
编码器层数	6

表 3.5 微调超参数

Table 3.5 The fine-tuning hyperparameters

参数名称	值/范围
学习率	1e-5~1e-4
Batch 大小	32
Epoch	100
优化器	Adam
嵌入维度	768
注意力头数	8
编码器层数	6

3.3.4 实验结果与分析

本节将分三部分对实验结果进行分析，其中包括分类实验，回归实验以及消融实验。与当前先进的模型以及和本章模型使用了相似方法的模型在分子性质预测数据集上进行了实验效果对比，分类任务使用的评估方法是 ROC-AUC，回归任务使用的评估方法为 RMSE。

（1）分类任务实验

本模型为预训练微调模型，在实验过程中，选取了三个模型也为预训练模型，分别为 MolCLR，N-GRAM，SMILES-BERT。在上一节所提到的五个分类数据集上进行了实验，加粗字体表示最佳效果模型，括号内数字代表上下数据上下浮动的阈值，结果如下表所示：

表 3.6 无预训练模型分类数据集实验结果（ROC-AUC）

Table 3.6 Classification dataset experimental results for no pretrained（ROC-AUC）

算法	数据集				
	BBBP	Tox21	ClinTox	BACE	SIDER
TF_Roubust[70]	$0.860_{(0.087)}$	$0.698_{(0.012)}$	$0.765_{(0.085)}$	$0.824_{(0.022)}$	$0.607_{(0.033)}$
GraphConv[71]	$0.877_{(0.036)}$	$0.772_{(0.041)}$	$0.845_{(0.051)}$	$0.854_{(0.051)}$	$0.593_{(0.035)}$
weave[4]	$0.837_{(0.065)}$	$0.741_{(0.044)}$	$0.823_{(0.023)}$	$0.791_{(0.008)}$	$0.543_{(0.034)}$
SchNet[72]	$0.847_{(0.087)}$	$0.767_{(0.025)}$	$0.717_{(0.042)}$	$0.750_{(0.033)}$	$0.545_{(0.087)}$
GraSeq[73]	$0.943_{(0.000)}$	$0.819_{(0.000)}$	-	$0.838_{(0.000)}$	-
My model	$\mathbf{0.962_{(0.013)}}$	$\mathbf{0.821_{(0.012)}}$	$\mathbf{0.997_{(0.003)}}$	$\mathbf{0.881_{(0.031)}}$	$\mathbf{0.684_{(0.023)}}$

在分类实验当中首先与无预训练模型相比，本章模型在 5 个基准数据集上均表现出了最优的水准，分类任务至少提高了 2.2%准确度，同样为多模态方法的 GraSeq 只在 BBBP，Tox21 以及 BACE 上进行了实验。本章模型在相同数据集上的效果均优于 GraSeq，由于 GraSeq 采用将不同模态数据编码到同一维度并直接相加的方式，没有考虑到嵌入后不同模态表示之间的距离，这样会损失较多信息。而本章模型通过对比损失使得不同模态的嵌入对齐，令多模态编码器能够更好的融合多模态特征。并且，本章模型在 ClinTox 分类数据集上取得了近乎完全正确的预测。在分类任务中，效果较差，分类较困难的 SIDER 数据集有 27 个分类子任务，大多数模型在该数据集上的的表现仅仅只有接近平均值的水平，相较于这些模型，本章所提出的模型在多任务分类预测上也有着近 10%的提升。综上所述，本章模型同时结合了图表示学习以及自然语言处理，多模态结合的模型在性质预测方面有着较好的表现。

表 3.7 有预训练模型分类数据集实验结果（ROC-AUC）

Table 3.7 Classification dataset experimental results for pretrained（ROC-AUC）

算法	数据集				
	BBBP	Tox21	ClinTox	BACE	SIDER
MolCLR[36]	$0.736_{(0.005)}$	$0798_{(0.007)}$	$0.932_{(0.017)}$	$\mathbf{0.890_{(0.003)}}$	$0.680_{(0.011)}$
N-GRAM[74]	$0.912_{(0.030)}$	$0.769_{(0.027)}$	$0.855_{(0.037)}$	$0.876_{(0.035)}$	$0.632_{(0.005)}$
SMILES-BERT[47]	$0.959_{(0.009)}$	$0.803_{(0.010)}$	$0.985_{(0.014)}$	$0.849_{(0.021)}$	$0.568_{(0.031)}$
My model	$\mathbf{0.962_{(0.013)}}$	$\mathbf{0.821_{(0.012)}}$	$\mathbf{0.997_{(0.003)}}$	$0.881_{(0.031)}$	$\mathbf{0.684_{(0.023)}}$

与预训练模型相比，本章模型在 5 个基准数据集中的 4 个表现出最优水准，与性能较好的MolCLR和SMILES-BERT相比在BBBP，ClinTox，Tox21以及SIDER中实现了最佳性能。相比于这些单模态预训练模型，本章模型从 SMILES 和分子图中提取了更加丰富的特征信息，并且构建了多粒度预训练策略，比当前的优秀基线平均准确度高出了 7%。相比与 MolCLR，该方法的预训练数据量达到 1000 万的分子数据，而本方法只用到了 50 万个分子做预训练，在预训练时间较短并且数据量少的情况下依然保持了和 MolCLR 近似的效果。同理在 BACE 数据集中，分子的相对分子质量非常大，对于较长 SMILES 串包含过多词的问题，做了类似停用词的删减，在保证基本排列顺序不变的情况下，也达到了不错的效果。

（2）回归任务实验

本模型为预训练微调模型，在实验过程中，选取了三个模型也为预训练模型，分别为 MolCLR，N-GRAM，SMILES-BERT。在上一节所提到的三个回归数据集上进行了分类实验，加粗字体表示最佳效果模型，括号内数字代表上下数据上下浮动的阈值，结果如下表所示：

表 3.8 无预训练模型回归数据集实验结果（RMSE）

Table 3.8 Regression dataset experimental results for no pretrained（RMSE）

算法	数据集		
	FreeSolv	ESOL	Lipo
TF_Roubust	$4.122_{(0.085)}$	$1.722_{(0.038)}$	$0.909_{(0.060)}$
GraphConv	$2.900_{(0.135)}$	$1.068_{(0.050)}$	$0.712_{(0.049)}$
weave	$2.398_{(0.250)}$	$1.158_{(0.055)}$	$0.813_{(0.042)}$
SchNet	$3.215_{(0.755)}$	$1.045_{(0.064)}$	$0.909_{(0.098)}$
GraSeq	-	-	-
My model	$\mathbf{1.903_{(0.490)}}$	$\mathbf{0.789_{(0.087)}}$	$\mathbf{0.721_{(0.067)}}$

首先与无预训练模型相比，本章模型在 5 个基准数据集上均表现出了最优的水准。与分类任务相比，回归任务更具挑战性，因为后者只考虑了手动标注的离散标签。与三个使用图卷积神经网络的方法相比，本章模型不仅采用了 GCN 对分子图结构编码，并且使用 transformer 编码器对图结构数据进行了全局的注意力系数计算。这样不仅利用了采样的邻居信息，还在分子嵌入中加入了全局信息。模型训练也不会随着层数的增多而导致梯度消失。相比于 GraSeq，该模型解决了它不能处理回归数据集的问题，并且在预训练阶段采用了结合领域知识的多粒度预训练策略，模型在回归实验上取得了非常好的效果。

表 3.9 有预训练模型回归数据集实验结果（RMSE）

Table 3.9 Regression dataset experimental results for pretrained（RMSE）

算法	数据集		
	FreeSolv	ESOL	Lipo
MolCLR	$2.200_{(0.200)}$	$1.110_{(0.01)}$	$\mathbf{0.650_{(0.080)}}$
N-GRAM	$2.510_{(0.190)}$	$1.100_{(0.030)}$	$0.880_{(0.120)}$
SMILES-BERT	$2.974_{(0.510)}$	$0.841_{(0.096)}$	$0.666_{(0.029)}$
My model	$\mathbf{1.903_{(0.490)}}$	$\mathbf{0.789_{(0.087)}}$	$0.721_{(0.067)}$

与带有预训练的模型相比，同样使用对比学习的 MolCLR，它不仅使用了数据增强，而且预训练的数据量超出本模型所使用预训练数据的 20 倍，可以看出仅在 Lipo 数据集中比该模型算法效果好一些。SMILES-BERT 模型也拥有着 1800 万的预训练数据，效果也不如本章所提出的模型。而且它的预训练策略仅仅是与 Bert 模型相同的掩码预测，并没有借助到分子化学领域知识，导致它即便有大量的预训练数据，但效果依然不佳。本章模型可以在更小规模的预训练数据集中得到更优的预训练模型，也可以反映出多模态模型能够从一维和二维结构中提取到更丰富的特征信息。

（3）消融实验

为了研究预训练对于模型性能的贡献，随机初始化模型参数在五个分类数据集上进行有监督学习，并与之前的实验遵循相同的超参数设置，在表 3.10 中给出了比较结果。经过预训练的模型明显比无预训练的模型在分类任务中表现更好，经过预训练的模型平均 ROC-AUC 增长了 10%，由此也证明了预训练可以从无标签数据中学到数据的基本特征，并提高模型在下游任务的泛化能力。而且可以发现在较小的数据集上，预训练模型能够有更大的提升，这也有助于解决标签数据过少模型较难学到数据分布的问题。并且还进行对比学习有效性的验证，对齐主

要使用的是对比学习损失让分子图和分子 SMILES 序列空间表示距离缩短，具体实验结果如表 3.10，数据对比可视化操作如图 3.3。

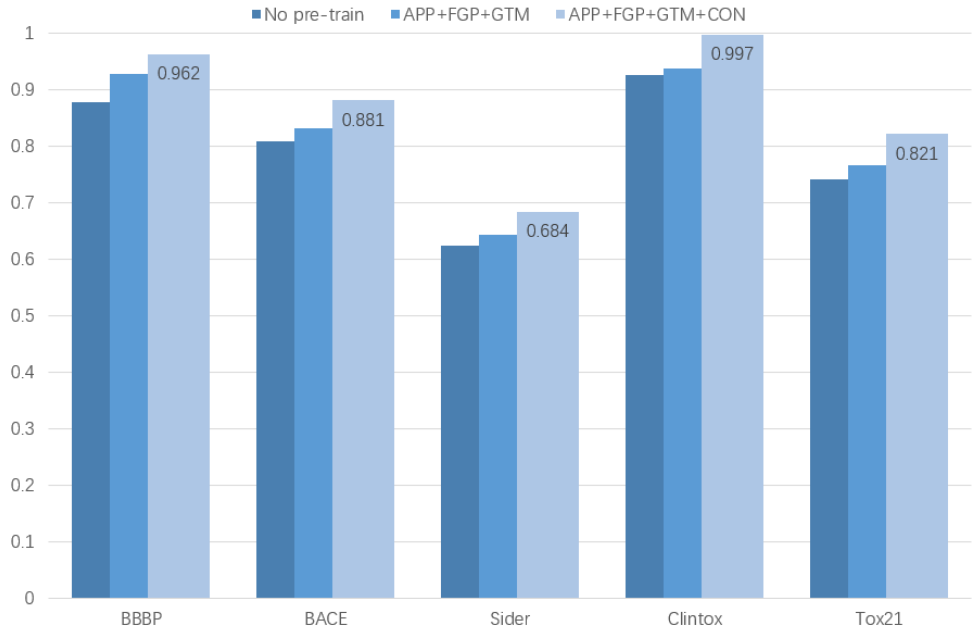


图 3.3 消融实验结果可视化

Fig 3.3 Ablation study results visualization

表 3.10 消融实验结果（ROC-AUC）

Table 3.10 The results of ablation study（ROC-AUC）

预训练规模	预训练任务	BBBP	BACE	ClinTox	Tox21	Sider
500K	无预训练	0.878	0.809	0.925	0.740	0.623
	APP+FGP+GTM	0.928	0.831	0.936	0.766	0.642
	APP+FGP+GTM +CON	0.962	0.881	0.997	0.821	0.684

本小节从两个方面来评估对比学习的作用，第一是对不同模态嵌入对齐的效果以及在下游任务中的表现。在所有下游分类数据集上测试了有无对比损失的消融实验，表 3.10 显示了所有分类数据集的实验结果指标 ROC-AUC，可以看出对比学习对于分类准确率的提升是显著的。提取了预训练过程中第 8 个 epoch 的某个分子在多模态编码器中的交叉注意力权重，如图 3.4，图 3.5 所示。可视化出 smiles 序列中被标红字符与分子图中对应原子的交叉注意力权重，以及注意力矩阵。分子图中颜色的深浅和圆形半径大小表示了权重的大小关系。可以看出，未采用对比学习的模型，很难将两个空间的向量对齐，而采用对比学习策略，可以让图嵌入和文本嵌入相互对齐，使 Transformer 更容易注意到文本和分子图之间相对应的

原子。并且预训练当中多粒度的任务模式，也是建立在图嵌入与文本嵌入对齐基础之上的。在后续的多模态融合数据当中，对齐表示的分子嵌入更易于多模态编码器对数据的融合表示学习。

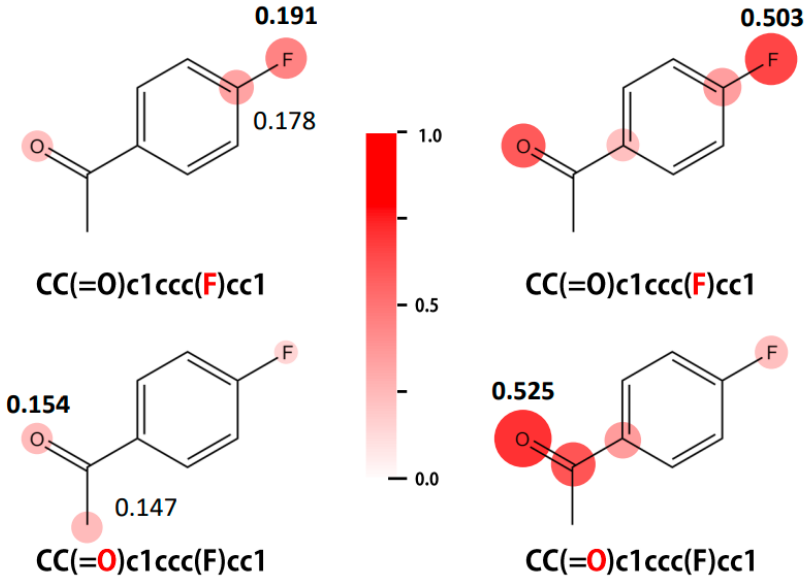


图 3.43 分子注意力可视化

Fig 3.4 Molecular attention visualization

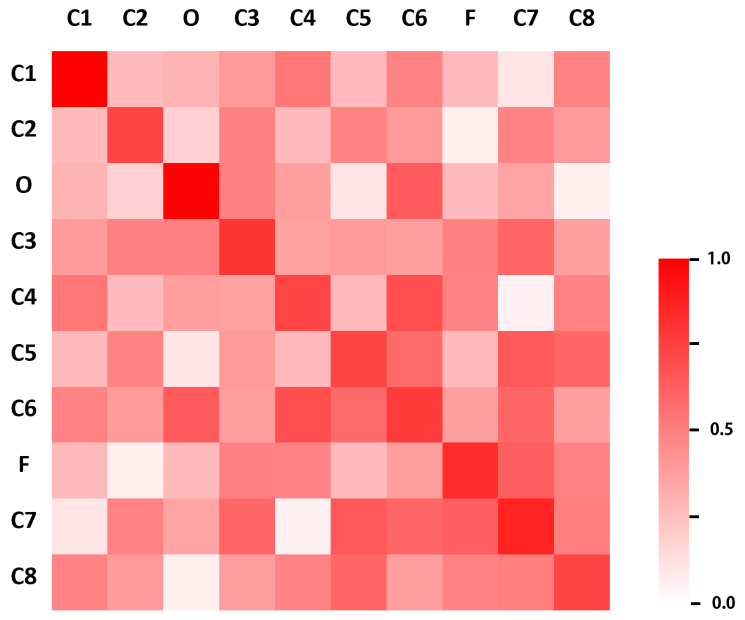


图 3.5 注意力矩阵

Fig 3.5 Attention matrix

为了探索不同粒度的预训练任务对实验结果的影响，我们使用了三种粒度的损失函数独立预训练模型，并随后在五个分类数据集上进行了微调。结果如图 3.6 所示，表明使用单一粒度的损失函数会导致准确率下降。然而，结合所有三种粒度的损失函数进行模型预训练可以获得更准确的结果。

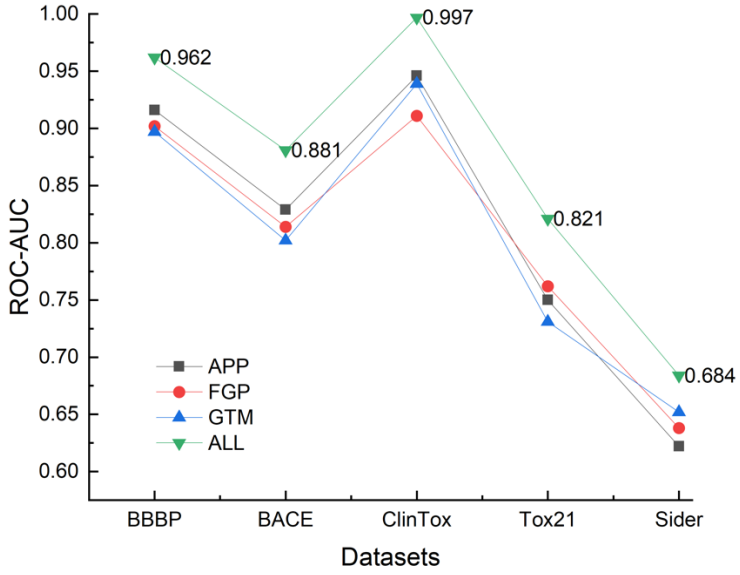


图 3.6 使用不同粒度损失函数实验结果

Fig 3.6 Results of fine-tuning the model using loss functions of different granularities

为了验证使用多粒度损失函数的模型的表达能力，我们对三种类型的损失函数进行了单独验证。我们使用一个包含 50,000 个未标记分子的玩具数据集，在相同的训练设置下，使用不同粒度的损失函数对我们的模型进行预训练。如图 3.7 所示，在训练阶段，使用组合多粒度损失函数的模型表现优于使用单一粒度损失函数的模型，这再次证明了多粒度损失函数模型的有效性。

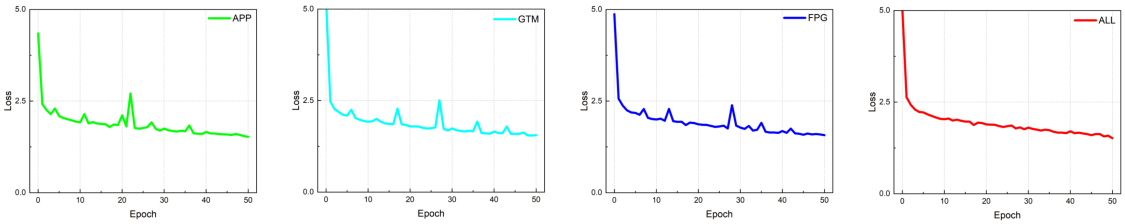


图 3.7 不同粒度损失函数的训练损失

Fig 3.7 The training losses on different granularity loss function

3.4 本章小结

目前分子表示的多模态预训练方法较少关注不同模态分子表示的对齐问题，并且预训练任务往往是自然语言处理任务的简单变化，不能有针对性地解决药物分子领域的问题。本章提出了一种基于对比学习的多模态多粒度分子性质预测模型，将分子的多种模态的表示进行对齐，并构建了三种具备领域知识的不同粒度的预训练任务。由该模型学习得到的表示可以被应用到许多不同的分子性质预测的下游任务中。在分类和回归数据集上进行了实验和对比，实验结果表明该模型能有效地学习不同模态中的丰富信息并且能够准确地预测分子性质。

4 基于 LLM 和分子描述的分子性质预测模型

4.1 概述

下游任务（如药物性质预测）一般需要对预训练模型进行调整。对于拥有丰富性质的药物分子，如果每一个下游任务都要微调预训练模型，不仅耗时耗力，而且不同的模型会占据非常大的存储空间。分子的模态不仅包括分子图以及 SMILES 表示，还包括丰富的分子描述信息。研究人员根据质谱光谱和化学实验等研究得出的药物分子性质一般以自然语言的形式所保存，并可以通过询问大语言模型（Large Language Model，LLM）生成。因此，除了从分子图和 SMILES 序列当中获得结构信息，也可以利用功能强大的大语言模型产生海量的分子描述，为预测模型提供丰富的特征信息，从而提高药物分子性质预测的准确性，为未来的药物发现和药物设计等任务提供支持。

为解决上述问题，并充分利用分子描述信息，本章对多模态对比学习模型进行优化，构建适用于不同下游任务的大模型提示语微调模型。该方法不需要为每个下游任务训练单独的模型，仅通过构建特殊的提示语（prompt）输入到大模型即可实现不同分子性质的预测。提示语由两个部分构成。第一部分即分子描述，通过检索算法从外部数据库中获取。第二部分提示语构建采用了少样本学习（few-shot learning）模式，其输入样本仅包括少数几个正例和负例。该方案利用第三章提出的模型和图神经网络编码技术，在待预测的分子与正例和负例之间构建出一个关系图，从而将正负样例与待预测分子充分联系。该关系图通过学习得到的一个适配器进一步映射为提示语的第二部分，并与分子描述文本一起提供给大语言模型。本章提出的方法仅使用已训练好的大语言模型就可以对所有分子进行对应的性质预测，在利用大语言模型丰富的知识的同时，减少了模型的训练成本和参数存储成本。该模型在第三章的分类数据集中表现优异，在增加的两个大型数据集上的准确性也有较大提升。

本章内容组织如下，首先对模型架构进行概述，其次对各模块作详细描述和说明，然后对实验所涉及的数据集和对比模型进行介绍，对实验结果进行详细分析和可视化展示，最后对本章进行小结。

4.2 模型描述

4.2.1 符号定义及问题描述

本章目标是从一组少样本分子属性预测任务 $\{T_\tau\}_{\tau=1}^{N_t}$ 中学习一个适配器，并将其作为提示语的组成部分，让大语言模型可以泛化以预测给定少数标注分子的新

属性。第τ个任务T_τ测一个分子（索引为i）在目标属性上是活跃的（$y_{\tau,i}=1$）还是不活跃的（$y_{\tau,i}=0$）每个类别提供了 K 个标注样本。然后将T_τ表述为一个 2-way K-shot 的分类任务，支持集$S_\tau=\{(x_{\tau,i},y_{\tau,i})\}_{i=1}^{2K}$包含了 2K 个标注样本，查询集$Q_\tau=\{(x_{\tau,j},y_{\tau,j})\}_{j=1}^{N_{q_\tau}}$包含了$N_{q_\tau}$个待分类的无标签样本。对于查询节点$\mathrm{v_q}$，定义其邻居集合为$\mathcal{N}(v_q)$，通过 KNN 算法选取距离最近的 k 个节点$\mathcal{N}(v_q)=\mathrm{KNN}(v_q,\mathrm{k})$，基于邻居集合$\mathcal{N}(v_q)$，构建关系图 G = (V, E)，通过消息传递机制，更新查询节点的表示h_v，$h_v^{(t+1)}=\mathrm{GNN}\left(h_v^{(t)},\left\{h_u^{(t)}:u\in\mathcal{N}(v_q)\right\}\right)$，GNN 是图神经网络。将更新后的节点表示$h_v$通过一个适配器映射函数$\phi$，映射到隐藏空间。给定一组查询节点嵌入$E=\{e_1,e_2,\ldots,e_n\}$，其中每个$e_i$表示一个查询节点的嵌入向量，以及一组分子描述$M=\{m_1,m_2,\ldots,m_n\}$，其中每个$m_i$是一个分子的文本描述，构造一个提示语$P$，需要将之前的节点嵌入$E$和分子描述嵌入$M$整合起来，将提示语用作大型语言模型 LLM 的输入。最终的目标是让已经训练好的 LLM 模型对提示语中所提出的问题进行答案预测A。将提示语P通过 LLM 的映射到答案A，$A=\mathrm{LLM}(P)$。得到的答案就是分子性质预测相关回答。

表 4.1 符号定义表

Table 4.1 Symbol definition

符号	含义
T_τ	分类任务类型
$x_{\tau,j}$	分子嵌入表示
$y_{\tau,j}$	性质标签
S_τ	特定任务支持集
Q_τ	特定任务查询集
c_τ^c	任务类原型表示
$g_{\tau,i}$	第 i 个样本的嵌入表示
$b_{\tau,i}$	属性感知分子嵌入表示
$\mathbf{A}_\tau^{(t)}$	全连接图邻接矩阵
$\widehat{\boldsymbol{A}}_\tau^{(t)}$	KNN 处理后的邻接矩阵
$x_{\tau,i}$	分子嵌入
$y_{\tau,i}$	分子性质标签
$\mathcal{N}(v_q)$	邻域节点集
Dice(A, B)	分子 A，B 摩根指纹相似度函数

4.2.2 模型概述

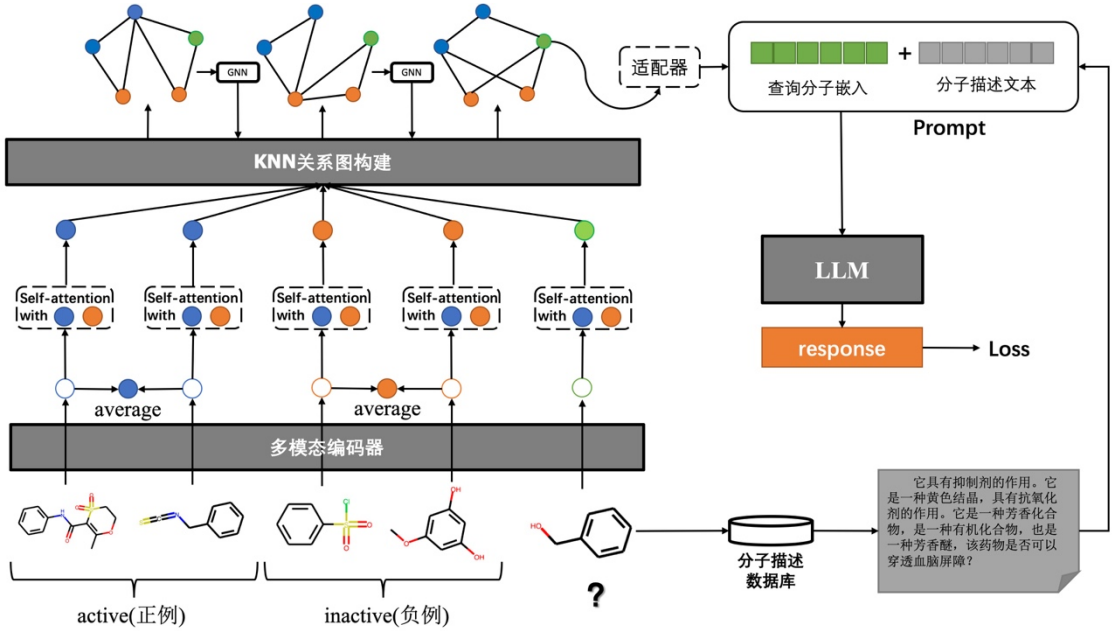


图 4.1 模型结构图

Fig 4.1 Model architecture

本章模型结构如图 4.1 所示，主要由四个模块构成：(1)分子嵌入模块。从对应的下游任务数据集中筛选出 N 个正例和 N 个负例并通过上一章所提到的多模态编码器对每种分子进行编码得到他们的嵌入表示，并将得到的嵌入表示分别与正负例的拼接表示做注意力计算得到最终表示。(2)关系图构建模块。该模块包括两个步骤，一是采用 K 最邻近算法构建正负样例及待测样本的关系图，二是利用图神经网络技术更新关系图的节点特征，两个步骤不断迭代直到产生最终的关系图。(3)分子描述模块。该模块使用基于摩根指纹的分子检索算法，从数据库中检索对应分子描述。(4)提示语构建模块。将来自模块(2)和模块(3)的输出整合为提示语并输入到大语言模型，从而获得待测分子的性质。

4.2.3 分子嵌入模块

在图 4.1 中绘制了来自特定任务数据集的 2-way，2-shot 的任务。基于 few-shot 任务设定，“2-way”意味着任务包含正例和负例两个类别，而“2-shot”则意味着每个类别只有两个用于训练的分子样本，本章提出的方法需要根据这两个类别的四个分子样本来学习，生成分子的嵌入并进行预测。

在图 4.1 中，虚线模块是可微调模块，实线模块中的参数则是冻结的，不需要训练。输入一个查询分子$x_{\tau,i}$，首先使用上一章的多模态编码器将其编码为$g_{\tau,i}$。对

于每个类别 c，选择两个分子作为样本，将它们的表示进行平均获得类原型c_τ^c，具体计算方式如公式 4.1 所示。

$$c_\tau^c = \frac{1}{|S_\tau^c|} \sum_{(x_{\tau,i}, y_{\tau,i}) \in S_\tau^c} g_{\tau,i} \tag{4.1}$$

其中，S_τ^c是所有在目标任务T_τ下，药物分子性质类别为c的样本的集合，即$S_\tau^c = \{(x_{\tau,i}, y_{\tau,i}) | (x_{\tau,i}, y_{\tau,i})\} \in S_\tau \text{ and } y_{\tau,i} = c$。$g_{\tau,i}$是第$i$个样本的嵌入表示。为了将不同属性的分子之间构建联系，将这些类原型作为T_τ的上下文信息，并进一步将它们编码为：

$$b_{\tau,i} = \left[\text{softmax}\left(\frac{C_{\tau,i} C_{\tau,i}^T}{\sqrt{d_g}}\right) C_{\tau,i}\right] \tag{4.2}$$

其中，$C_{\tau,i}^T = [g_{\tau,i}, c_{\tau,0}, c_{\tau,1}] \in R^{d_g \times 3}$，在此，$[\cdot]_j$提取出与$x_{\tau,i}$对应的第 j 行向量。这里的$b_{\tau,i}$使用了缩放点积自注意力机制来计算，以便在维度上比较每个$g_{\tau,i}$与类原型。然后，属性感知分子嵌入$p_{\tau,i}$其计算方式为：

$$p_{\tau,i} = MLP_W^p(\text{concat}[g_{\tau,i}, b_{\tau,i}]) \tag{4.3}$$

MLP_W^p表示由W^p参数化的多层感知器，它用于找到一个更低维度的空间，该空间编码的子结构与T_τ的目标属性更加相关。这个上下文化的$p_{\tau,i}$是属性感知的，可以更好地预测目标属性。

4.2.4 关系图构建模块

关系图学习模块用于捕获和利用这种分子间的属性感知关系图，使得有限的标签可以在相似的分子之间有效地传播。关系图学习模块包括两个子模块，KNN 子模块和 GNN 子模块。KNN 子模块根据分子嵌入表示利用 K-最近邻算法更新关系图的图结构，GNN 子模块则在关系图上进行消息传递，更新图节点中每个分子的嵌入表示。两个子模块交替地进行多次迭代，直到产生最终的关系图结构及每个分子的嵌入表示。

在第 t 次迭代中，令$\mathbf{G}_\tau^{(t)}$表示关系图，其中$V\tau$将$S\tau$中的 $2k$ 个分子和Q_τ中的一个查询分子作为节点。$\mathbf{A}_\tau^{(t)} \in R^{(2K+1)\times(2K+1)}$表示图$\mathbf{G}_\tau^{(t)}$的邻接矩阵。如果两个节点$x_{\tau,i}, \mathrm{x}_{\tau,\mathrm{j}} \in \mathrm{V}$，$\mathrm{x}_{\tau,\mathrm{i}}, \mathrm{x}_{\tau,\mathrm{j}}$的属性感知分子嵌入$\mathrm{p}_{\tau,\mathrm{i}}$，$\mathrm{p}_{\tau,\mathrm{j}}$之间的相似性表明他们的性质是否相似或者说是否同属于一种标签。因此，将初始图节点嵌入表示为$\mathrm{h}_{\tau,\mathrm{i}}^{(0)} = \mathrm{p}_{\tau,\mathrm{i}}$。

在 KNN 子模块中，首先使用当前的分子嵌入来估计$\mathbf{A}_\tau^{(t)}$。其中的元素记录了$\mathrm{x}_{\tau,\mathrm{i}}$和$x_{\tau,j}$之间的相似性，这个相似性是通过公式 4.4 计算。

$$\left[\mathbf{A}_\tau^{(t)}\right]_{ij} = MLP_{\mathbf{W}_a}\left(\exp\left(-\left|h_{\tau,i}^{(t-1)} - h_{\tau,j}^{(t-1)}\right|\right)\right) \tag{4.4}$$

在这里，$\mathbf{W}_a$是这个多层感知机（MLP）的参数。结果得到的$\mathbf{A}_\tau^{(t)}$是一个密集矩阵，它编码了一个完全连接的图$\mathbf{G}_\tau^{(t)}$。

然而，在一个 2-way，K-shot 任务中，一个查询分子在$\mathbf{G}_{\tau}^{(t)}$中只有 K 个真实的邻居。对于二分类问题，选择错误的邻居将严重降低分子嵌入的质量，尤其是当每个类别只提供一个标记分子时。为了避免错误邻居的干扰，根据 KNN 算法，仅选择最相似的 K 个节点作为中心节点的邻居，这样在仅有 K 个正负样例的情况下，不会多选出可能的错误样例，图 4.2 给出 KNN 算法的示例。

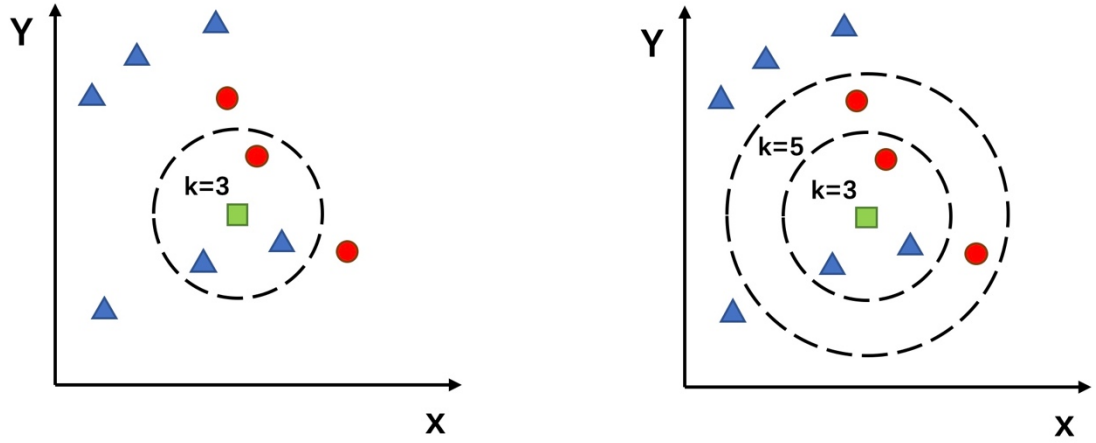


图 4.2 KNN 算法示例

Fig 4.2 Illustration of K-Nearest neighbors

K-最近邻居（KNN）算法是一种基于实例的学习，属于多分类问题的一种解决方案。该方法认为，如果一个样本在特征空间中的 K 个最临近的样本中的大多数属于某一个类别，则该样本也属于这个类别。KNN 算法的步骤包括：

（1）计算待分类项与其他所有项之间的距离；

（2）按照距离的递增关系进行排序；

（3）选取与待分类项距离最小的 K 个点；

（4）确定前 K 个点所在类别的出现频率；

（5）返回前 K 个点中出现频率最高的类别作为待分类项的预测分类。

值得注意的是，KNN 算法中的 K 值的选择、距离度量方式以及分类决策规则都会对最终结果产生重大影响。K 值过小会使模型过于复杂，容易过拟合；K 值过大则会忽略掉数据局部性质，可能导致预测精度下降。常用的距离度量方式有欧氏距离、曼哈顿距离等。

在 GNN 子模块中，邻接矩阵$\mathbf{A}_{\tau}^{(t)}$是一个表示节点间连接关系的矩阵。如果节点之间有连接，则矩阵$\mathbf{A}_{\tau}^{(t)}$中的对应位置元素设置为 1，否则为 0。每个节点嵌入$\mathrm{h}_{\tau,\mathrm{i}}^{(t)}$根据更新的关系图$\mathbf{A}_{\tau}^{(t)}$与其他节点嵌入共同适应。这表明嵌入是在图中邻居的

上下文中进行调整的，可能更有效地捕捉节点间的关系，所有节点嵌入$H_{\tau}^{(t)}$的集合，其中第 i 行对应于$h_{\tau,i}^{(t)}$。$H_{\tau}^{(t)}$通过以下公式更新：

$$\mathbf{H}_{\tau}^{(t)} = \text{LeakyRelu}\left(\mathbf{A}_{\tau}^{(t)}\mathbf{H}_{\tau}^{(t)}\mathbf{W}_{r}\right) \tag{4.5}$$

其中$\mathbf{W}_{r}$是一个可学习的参数。这个过程使用了 LeakyReLU 激活函数，它是 ReLU (Rectified Linear Unit)的一个变种，允许较小的梯度流过，当神经元处于非激活状态时避免了完全的零梯度。

进一步将 $\mathbf{G}_{\tau}^{(t)}$缩减为 K-最近邻（KNN）图，其中 K 被设定为支持集中每个类别的标记分子数量。对于$x_{\tau,i}$来说最大的 K 个$\mathbf{A}_{\tau}^{(t)}$将被记录在$N_{(x_{\tau},i)}^{(t)}$，其中 j = 1,..., 2K-1 的索引，设定：

$$\left[\widehat{\boldsymbol{A}}_{\tau}^{(t)}\right]_{ij} = \begin{cases} \left[\mathbf{A}_{\tau}^{(t)}\right]_{ij} & \text{if } x_{\tau,j} \in N^{(t)}\left(x_{\tau,i}\right) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

$\widehat{\boldsymbol{A}}_{\tau}^{(t)}$中的值通过在每行$\left[\widehat{\boldsymbol{A}}_{\tau}^{(t)}\right]_{i}$上应用 Softmax 函数被归一化到 0 和 1 之间。这样的归一化也可以通过 z-score、min-max、sigmoid 归一化来完成。然后，根据这个更新的关系图编码$\widehat{\boldsymbol{A}}_{\tau}^{(t)}$中的其他节点嵌入来共同适应每个节点嵌入$h^{(t)}$。让$\mathbf{H}_{\tau}^{(t)}$ 表示所有节点嵌入，其中第i行对应$h_{\tau,i}^{(t)}$。$\mathbf{H}_{\tau}^{(t)}$的更新方式如下：

$$\mathbf{H}_{\tau}^{(t)} = \text{LeakyReLU}\left(\hat{A}_{\tau}^{(t)}\mathbf{H}_{\tau}^{(t-1)}\mathbf{W}_{r}\right) \tag{4.7}$$

其中$\mathbf{W}_{r}$是可学习参数。

在进行了 T次迭代后，返回$h_{\tau,i} = \left[H_{\tau}^{(T)}\right]_{i}$作为$x_{\tau}, i$最终分子嵌入，返回 $\widehat{\boldsymbol{A}}_{\tau}^{(t)}$作为最终优化的关系图。

4.2.5 分子描述模块

LangChain 是一个用于开发由语言模型驱动的应用程序的框架。通过将语言模型与提示语、少样本学习等技术结合起来，它为一般应用程序赋予了上下文感知能力。LangChain 的基本框架如图 4.3 所示。模型将用户输入转换为 SQL 查询语句，从数据库中获得信息，然后结合数据库信息及用户问题提交给大语言模型，以获得问题的答案。

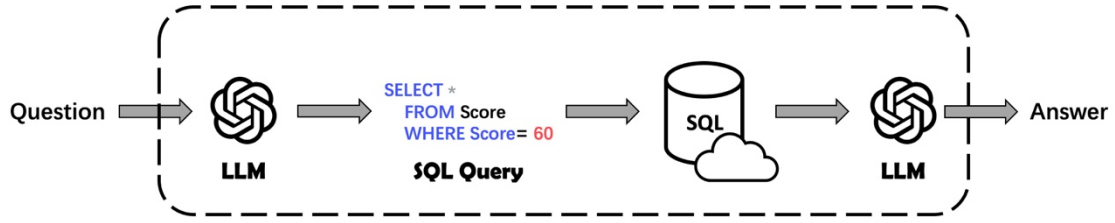


图 4.3 SQL 链和代理的步骤

Fig 4.3 The steps of most SQL chain and agent

相较于 LangChain 检索，分子结构的检索更为复杂。考虑到分子特性，本文基于摩根指纹（Morgan FTS）实现分子检索。分子指纹是分子化学结构的数值表示，可以用于各种计算目标，如相似性搜索、性质预测、虚拟筛选以及聚类分析。最具代表性的分子指纹之一是摩根指纹，它也被称为环形指纹或扩展连接指纹（ECFP）。摩根指纹可表示分子中特定子结构或化学片段的存在或缺失。摩根指纹通过以环形方式表示分子的连接模式，它通过从分子中的中心原子迭代扩展一组原子来生成指纹。在每个扩展步骤中捕获邻近原子及其键类型。这个过程持续进行，直到扩展到预定义的半径。生成的指纹是一个二进制位向量，其中每一位表示特定子结构的存在或缺失。摩根指纹相比其他类型的指纹有几个优势，包括处理不同大小的分子的能力，对小的结构变化的鲁棒性，以及捕获分子间结构相似性的有效性。

在检索分子时，需要计算查询分子与数据库中分子之间的相似性。这里采用 Dice 相似性，其计算方式如公式 4.7 所示。

$$\mathrm{Dice}(\mathrm{A},\mathrm{B}) = \frac{2 * |\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A}| + |\mathrm{B}|} \tag{4.8}$$

其中，A和B是两个分子的摩根指纹。

|A|和|B|代表A和B的基数，也就是子结构的数量。$|\mathrm{A} \cap \mathrm{B}|$表示在A和B中都存在的子结构的数量。Dice 相似性从 0 到 1 变化，其中值 0 表示分子之间没有重叠或相似性，值 1 表示完全重叠。图 4.4 给出了利用两个分子的摩根指纹计算 Dice 相似性的示例。相似性图展示了两个分子的相似性和差异性，绿色部分为相似子结构，紫色部分为差异子结构。根据相似性得分可以找到最相似的 N 个分子和分子描述对作为模型提示语的组成部分。

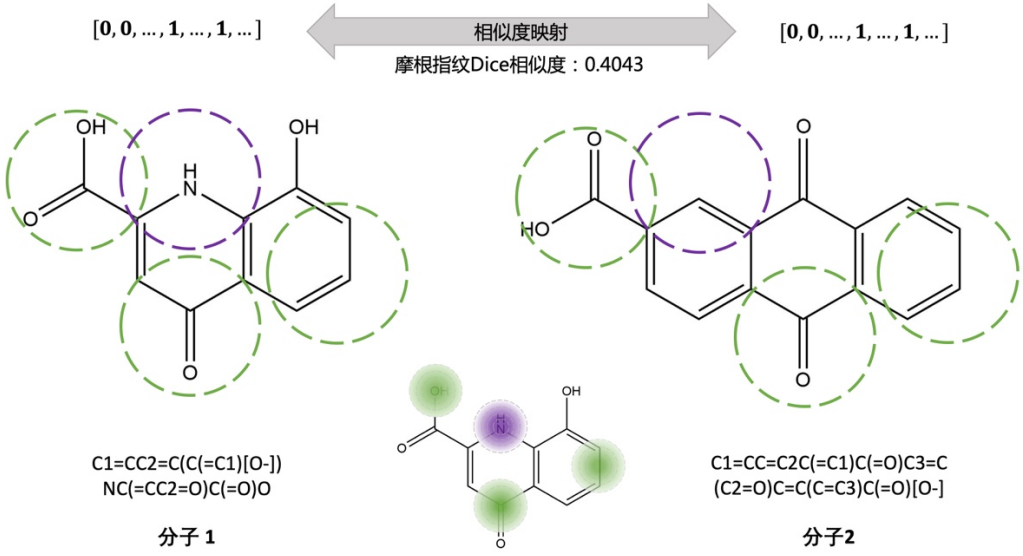


图 4.4 分子相似度图解

Fig 4.4 Molecular similarity visualization

4.2.6 提示语构建

系统提示和用户输入提示是形成任务上下文的两个重要部分。用户提示通常更复杂，并包含解决任务和格式规范化的基本指令，其中用户提示被定义为规范化用户输入。为了帮助大型语言模型理解任务，并要求它生成想要的答案，本章模型设计了提示语模板，这些模板包含了之前分子描述检索的内容。系统提示由四个部分组成：角色识别、任务描述、检索的例子和输出指令。

（1）角色识别（Role Identification）：这部分用于明确定义用户和 AI 的角色。例如，AI 可能被定义为一个专家，用户可能被定义为一个提问者。

（2）任务描述（Task Description）：这部分详细描述了需要完成的任务。例如，如果任务是预测分子的性质，那么任务描述可能包括预测的具体性质和预测的方法。

（3）检索的例子（Retrieved Examples）：这部分提供了一些与任务相关的例子，帮助 AI 理解任务的具体内容和格式。例如，如果任务是预测分子的性质，那么检索的例子可能包括一些已知分子的性质。

（4）输出指令（Output Instruction）：这部分提供了与输出要求相关的指令。例如，输出指令要求 AI 以特定的数据格式提供预测结果。

具体结构和示例如图 4.5 所示。提示管理可以帮助 AI 更好地理解任务，从而生成更符合用户期望的输出。

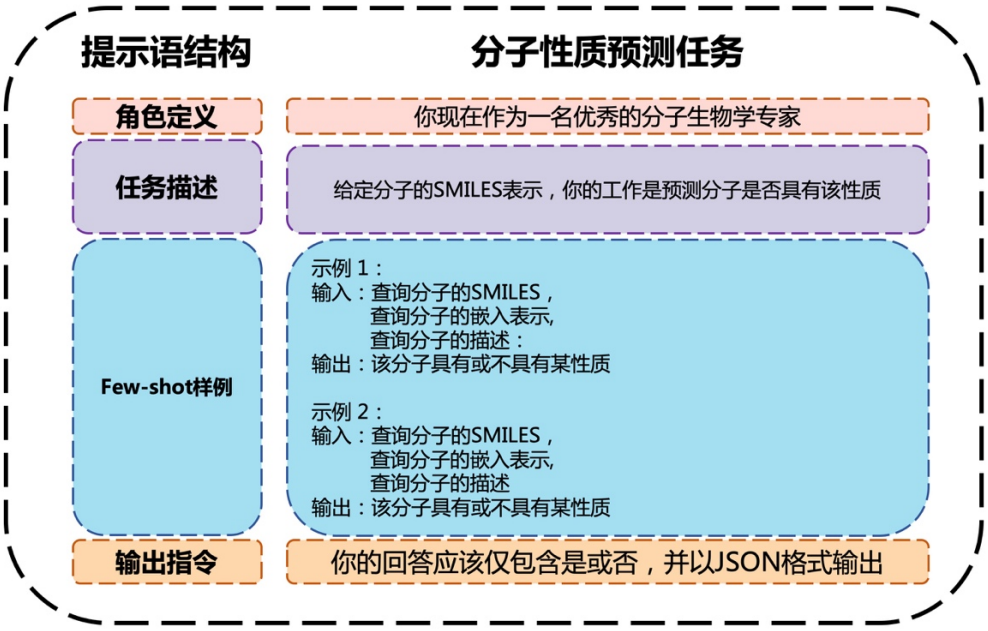


图 4.5 提示语构成

Fig 4.5 Composition of prompt

4.3 实验

4.3.1 实验环境

本章提出的基于 LLM 和分子描述的性质预测模型采用 Python 语言编程，基于 Pytorch 深度学习框架实现，实验环境与第三章的保持一致，具体的实验环境参数配置如表 3.2 所示

4.3.2 实验数据

本章实验主要用于做分类数据集，不再使用第三章的三个回归任务数据集，在此基础上本章添加了两个包含分子数更多的数据集 HIV 和 MUV，数据集详细信息如下：

（1）HIV：该数据集是从公共可用的 AIDS 抗病毒药物筛选数据库中收集的。这个数据库是由美国国家癌症研究所的发展治疗项目部门创建的。HIV 数据集包含了 41127 种化合物，这些化合物都经过了实验室测试，以确定它们对 HIV 复制的抑制效果。每种化合物都被标记为活性或非活性，表示它们是否能有效抑制 HIV 的复制。

（2）MUV：这个数据集包含了 17 个不同的数据集，每个数据集都是为了预测一个特定的蛋白质靶标。MUV 数据集的目标是提供一个具有高度偏差的数据集，用于验证和比较不同的药物发现算法。MUV 数据集包含了 93087 个化合物，每个化合物都被标记为活性或非活性，针对 17 个不同的蛋白质靶标。

表 4.2 数据集统计信息

Table 4.2 Statistics of the datasets

数据集	分子数量	任务数量
HIV	41127	1
Tox21	93087	17

4.3.3 实验设置

由于大模型开源限制的问题，本章根据图表示学习中经常使用的技术选取了 4 个对比模型，大模型所使用的基础模型 Transformer；可以适用于各种任务的改良版编解码器模型 T5-base；在分子和自然语言之间做翻译工作的 MolT5 以及解码器模型 GPT-3.5-turbo。对比模型的详细信息如下：

（1）Transformer：作为大模型的基础模型，未经改良的 Transformer 仍然使用计算效率较低的由正弦余弦生成的位置编码，预训练数据较少，是采用编码器解码器结构的模型。

（2）T5-base：T5（Text-to-Text Transfer Transformer）是 Google 在 2019 年提出的一种基于 Transformer 的模型，更换了位置编码，T5 将所有的 NLP 任务（如分类、翻译、摘要等）都视为文本到文本的转换问题，从而实现了一种统一的模型架构。T5-Base 模型的参数数量约为 220M，编码器和解码器都是 12 层的 Transformer 块组成。

（3）MolT5：MolT5 是一个基于 T5 模型的化学分子生成模型。它被训练来生成和解析化学分子的 SMILES，他的预训练数据库包含大量的化学生物领域分子数据，MolT5 可以被用于多种化学相关的任务，比如预测分子的性质，生成新的分子，或者优化已有分子的性质。由于 T5 模型本身是一个通用的文本到文本的转换模型，所以通过适当的训练，它可以被应用于这种特定的任务。

（4）GPT-3.5-turbo：GPT-3.5-turbo 是 OpenAI 发布的一款语言模型，它是 GP 第三代生成预训练 Transformer 的增强版本，具有更高效的性能。GPT-3.5-turbo 旨在提供更快的响应速度和更好的用户体验，同时保持 GPT-3 的强大语言理解和生成能力。该模型通过大量的文本数据预训练，能够执行各种语言任务，如文本生成、翻译、摘要、问答等。GPT-3.5-turbo 在处理用户输入时进行了优化，以便更快地产生高质量的输出，同时减少了和计算资源的消耗。

表 4.3 微调超参数

Table 4.3 The fine-tuning hyperparameters

参数名称	值/范围
学习率	1e-6~1e-5
Batch 大小	32
Epoch	20
优化器	Adam
嵌入维度	768
注意力头数	8
编码器层数	6

4.3.4 实验结果与分析

本节将分两部分对实验结果进行分析，其中包括分类实验，以及消融实验。与当前先进的模型以及和本章模型使用了相似方法的模型在分子性质预测数据集上进行了实验效果对比，分类任务使用的评估方法是 AUC-ROC。

（1）分类任务实验

本模型为预训练提示语微调模型，在实验过程中，选取了四个模型也为预训练模型，分别为 Transformer，T5-base，MolT5，以及 GPT-3.5-turbo。使用第三章提出的多模态编码器为分子进行编码，并不再更新多模态编码器参数，固定大语言模型参数，仅通过更新适配器和自注意力两个模块的参数来进行分子性质预测。该分类实验采用 4-shot 即两个正例和两个负例和查询分子构架 KNN 关系图，并通过 GNN 迭代更新查询分子嵌入表示。

在所提到的七个分类数据集上进行了分类实验，HIV 和 MUV 为第三章未出现数据集，表格中加粗字体表示最佳效果模型，括号内数字代表上下数据上下浮动的阈值，所有模型采用相同数量的 shot 进行实验，未开源模型则不再进行微调，结果如下表所示：

表 4.4 预训练模型分类数据集实验结果（ROC-AUC）

Table 4.4 Classification dataset experimental results for pretrained（ROC-AUC）

算法	数据集						
	BBBP	HIV	MUV	Tox21	ClinTox	BACE	SIDER
Transformer	0.002	0.069	0.077	0.040	0.001	0.045	0.014
T5-base	0.032	0.176	0.065	0.206	0.024	0.089	0.186
Mol-T5	0.497	0.636	0.377	0.427	0.798	0.143	0.410
GPT-3.5-turbo	0.463	0.807	0.697	0.529	0.924	0.406	0.666
My model	**0.970**	**0.784**	**0.886**	**0.831**	**0.978**	**0.904**	**0.688**

总体来看，本章模型在七个分类数据集上全部达到了最好水平，相比于 Transformer 来说，由于 Transformer 的训练数据可能不包含化学领域的分子序列 SMILES 和相关的领域知识因此很多并不能得到模型的有效回答，该模型并不能从上下文中提取有用的信息，尤其是化学领域。同理，T5-base 知识采用了更精简的旋转位置编码等措施，训练数据中也没有相对应的领域数据，从中也可以看出数据是驱动模型有更好表现的关键原因。对比 Mol-T5 由于训练数据基本都是分子数据，在七个分类任务中准确率有了明显改善，但该模型最初由于用做分子翻译，相对于分子性质预测来说，该模型的效果并不尽如人意。对比于 GPT-3.5-turbo，他有着很强的上下文学习能力，该模型由于是闭源模型，没有做微调的前提下也有着非常不错的学习能力，在 few-shot 下有着不错的预测结果，甚至在 HIV 数据集中，超过了本章模型。相较于上一章没有借助大模型和分子描述特征的帮助，在添加了特征并且应用大模型对分子性质预测进行了进一步的提升改进，在预测

结果上也有所体现，预测准确率基本都有所提升，在新加的两个大分子数据集上的表现也超过了大部分对比模型。数据可视化见下图：

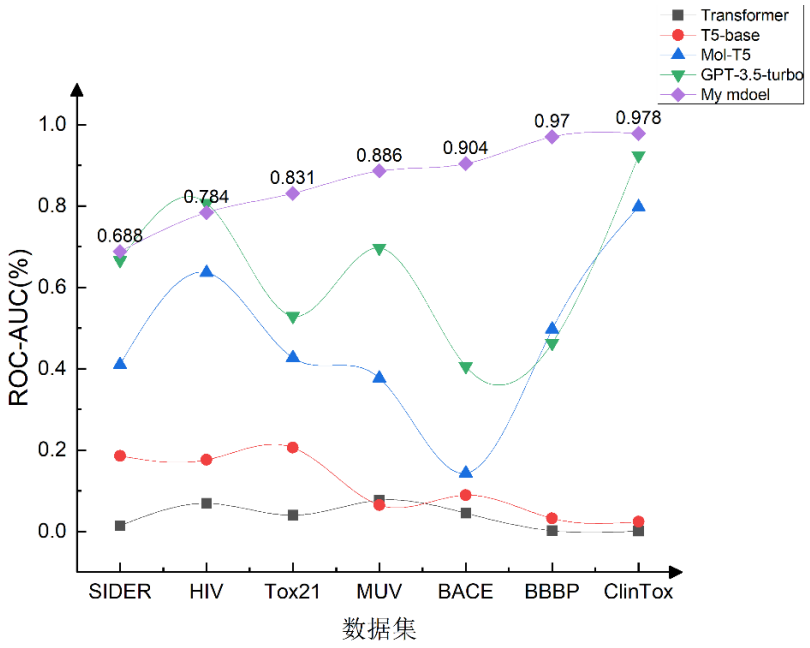


图 4.6 分类实验可视化

Fig 4.6 Classification experiment visualization

（2）消融实验

对于不同数量的示例对实验结果是否有影响上，做了进一步的探究，模型的 few-shot 数量选择由 2，4，10，50 来进行实验，由于 GPT-3.5-turbo 有 token 数量限制，仅在本模型上进行了不同数量的 shot 对模型结果的影响，实验结果如下：

表 4.5 Few-shot 数量影响结果（ROC-AUC）

Table 4.5 Few-shot learning quantity affects the outcome（ROC-AUC）

Few-shot	数据集						
	BBBP	HIV	MUV	Tox21	ClinTox	BACE	SIDER
2-shot	0.832	0.749	0.806	0.813	0.960	0.875	0.623
4-shot	**0.970**	0.784	0.886	0.831	0.978	**0.904**	**0.688**
10-shot	0.956	0.774	**0.909**	**0.837**	**0.983**	0.898	0.645
50-shot	0.907	**0.792**	0.899	0.784	0.922	0.800	0.633

从上述数据结果来看，采用 2-shot 也就是一正例一反例时结果相对于 4-shot 有较小差距，主要原因来自于构建的关系图中，三个节点的关系图并不能为节点提供更丰富的信息，相比于 4-shot 和 10-shot，可以看出是更为合适的超参数，在 BBBP，BACE，SIDER 四个数据集上 4-shot 有着不错的表现，在 MUV，TOX21，ClinTox 三个数据集上 10-shot 也有着较好的效果，综合来看，4shot 和 10shot 可以有效的构建信息较丰富的关系图，并且对大模型预测有着较大帮助。50-shot 本身有着大量的样本应该有不错的效果，但实验结果来看，50-shot 虽然有着较大量的样本，但实验结果反而不如 4-shot 和 10-shot，而且相较于前者，过大的关系图对资源占用以及推理速度来说都有着较大的压力，训练速度也比较慢，但是在数据量较大的 HIV 和 MUV 数据集上，反而有着不错的效果，由此也可以推断，样例数量占总支持集的比例对模型推理和预测有着比较密切的关联，综上所述，小样本数据集可以采用 4-shot 或 10-shot 会有较好的结果，对于大数据量的数据集，根据 5%的比例选择 few-shot 会有着不错的效果。

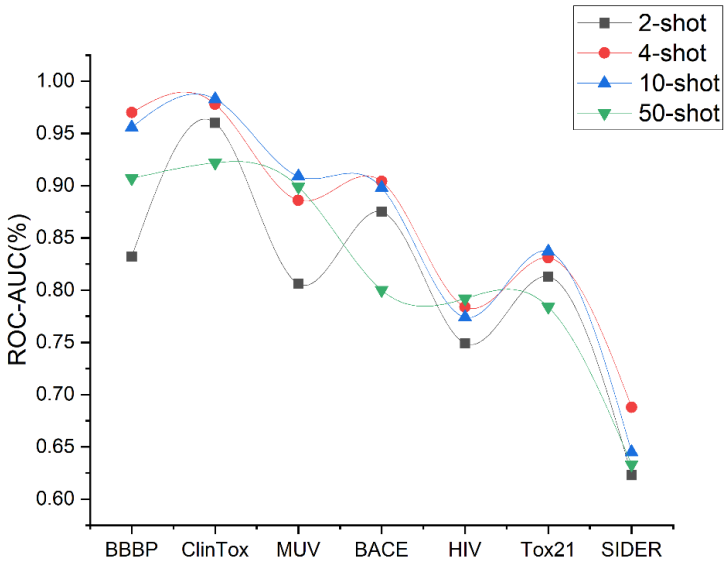
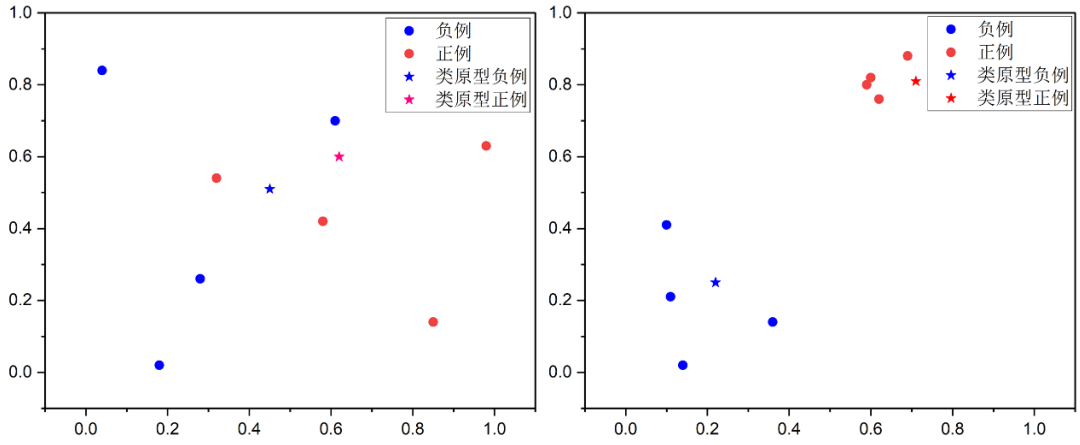


图 4.7 Few-shot 参数实验结果可视化

Fig 4.7 Visualization of few-shot parameter experiment results

为了验证属性感知所做的分子表示融合的重要性，将分子在做属性嵌入前后的表示均使用 T-SNE 可视化，并观察其在分类任务 BBBP 当中的分类效果，可视化两个分子嵌入表示，第一个是由多模态编码器编码得到的$g_{\tau,i}$，另一个是经过属性感知嵌入的$p_{\tau,i}$，挑选了十个分子进行可视化，其中包含五个正例五个负例，可视化实验结果如下：



4.8 有无 few-shot 对分类结果的影响

Fig 4.8 The impact of few-shot learning on classification results

从实验结果图可以看出，在未经 few-shot 提示仅根据多模态编码器编码后的药物分子嵌入分类情况，正负例并不能很好的聚类类原型分子，并且类原型分子也不能在向量空间中有很好的区分，经过属性感知编码后，不同样例可以在向量空间中有效地趋近于类原型分子，并形成聚类，证明属性感知分子编码是有效的。

在消融实验中，我测试了标签在提示语中是否对预测结果有直接影响，实验模型为 GPT-3.5-turbo 和本实验 10-shot 模型，在 HIV 和 ClinTox 一个大分子数据集和一个小分子数据集上进行测试，提示语分为包含标签数据和不包含标签数据，其他不变实验结果如下：

表 4.6 提示语中标签影响结果（ROC-AUC）

Table 4.6 Removing label context information from the in-context learning prompts（ROC-AUC）

模型	数据集	
	HIV	ClinTox
GPT-3.5-turbo (10-shot）	0.784	0.578
GPT-3.5-turbo (10-shot, unlabeled)	0.554	0.438
Our model (10-shot）	0.774	0.983
Our model (10-shot, unlabeled)	0.707	0.922

从表中数据可以看出，将标签数据从提示语中移除后，模型的预测准确率有明显的下降，甚至准确程度低于不使用大模型时的效果，是否有关键性提示信息对于大模型的上下文学习能力（in-context learning）有着重要影响。

4.4 本章小结

为了减少对不同下游任务训练微调时所占用的计算资源和存储空间，本章提出了一种适用于不同下游任务的大模型提示语微调方法。该模型从结构上改变了预训练微调模式，不再需要为每个下游任务对模型进行全量微调，只需要针对不同分子性质构建相应的提示语即可实现不同分子性质的预测。提示语的结构包含两部分，第一部分为分子描述，通过摩根指纹相似度匹配算法找到数据库中最相关的分子描述。第二部分为分子嵌入，不同于第三章的分子嵌入，本章采用少样本学习，在待预测与正样本和负样本之间构建关系图谱，将其充分联系起来。通过图神经网络学习该图谱得到最终的分子嵌入。本章使用已训练好的大语言模型，通过微调适配器和隐藏层参数即可进行对应的性质预测，在利用大语言模型的上下文理解能力的同时，减少的模型的计算资源和存储空间。本章在第三章分类数据集的基础上增加了两个大型分类数据集，与第三章相比，改进后的模型效果相对提升 1.7%，与对比模型相比，所有数据集的实验指标都达到了最优，这也表明了本章改进模型的有效性。

5 总结与展望

5.1 本文工作总结

在当下人工智能兴起的时代，药物发现和药物预测等行业逐渐也有了人工智能的身影。它能够帮助处理和分析大规模的复杂数据集；预测和发现新的生物标记物、药物靶点和潜在的治疗途径；可以通过更有效对候选药物进行筛选和优化，减少研发早期的资源浪费；甚至能够通过分析历史数据和相似化合物的信息，更早地预测药物副作用，从而在药物设计阶段就避免可能的问题。当前的人工智能方法用于分子属性预测任务面临着数据表示不一致和模型泛化能力不足等挑战。为应对这些挑战，本文在分子属性预测方法中引入多模态对比技术和大语言模型提示学习技术。本文的主要工作如下：

（1）本文首先提出了一种多模态对比学习模型。本文利用图信息和文本信息来学习药物分子的融合表示，以捕捉两种信息中丰富的特征。为了在嵌入空间中对齐图和文本特征并最大化它们的互信息，引入一种对比损失，在传统 Transformer 编码器中整合带有交叉注意力的多模态融合编码器，使得多模态编码器能够更有效地进行跨模态学习。在预训练阶段，提出多粒度预训练策略，涵盖原子、官能团和分子等不同级别的粒度。多模态多粒度的预训练任务使模型能充分利用分子中不同形态和不同层次的信息，从而提升了分子的表示能力。最后在几个通用分子性质预测数据集中进行分类和回归两种任务的实验，验证了模型的有效性。

（2）本文提出一种基于提示语微调的分子性质预测模型，为了节约计算资源及其存储空间，提高模型泛化能力，采取提示语微调模式。该模型不再需要针对每个下游任务去进行全量微调，而是使用少样本学习，根据有限的分子数据进行泛化学习并训练适配器。其次，根据分子性质对分子之间的关系进行实例化，构建 KNN 关系图谱，并使用图神经网络更新目标分子表示。最后，将目标分子的嵌入表示以及相关分子性质描述进行拼接，构建输入 LLM 的提示语，利用大语言模型的上下文理解能力，对分子性质进行推理和预测。本模型在 7 个分类数据集上与 4 种基线方法进行了对比实验，并进行了参数敏感性分析以及消融实验，验证了本文提出模型的有效性。

5.2 未来研究展望

本文的工作主要研究的是药物分子一维和二维数据的表示问题以及在大模型中提示语的构建。在分子领域中还有更多特征值得被挖掘和使用，比如分子极性，带电粒子对分子结构的影响以及像蛋白质基因等信息。在技术方面，提示语的构建不仅仅包括人工构建的提示语，还可采用可学习的提示语来帮助模型自动生成更加有效地提示语信息。具体而言，可从以下三方面进行优化和改进。

（1）融合分子的三维结构特征。分子的三维特征中包含着丰富的结构信息，比如键角，空间结构等，它们对分子的化学性质有重要的影响。将这些信息编码进入分子表征中可能会提高药物性质预测的准确性。

（2）可学习的提示语构建。人工构建的提示语不利于用一个量化指标去评价其优劣，将提示语的构建交给计算机或许是更好的选择。具体地，可将提示语抽象化成可学习的向量，从而生成合适于当下任务的提示语。

（3）更广泛的物质研究：在生物信息学领域，除分子表征外，其他组学相关的物质（如蛋白质、基因）等也值得深入探索和研究。对测序基因的功能以及蛋白空间结构对应的功能的测定一直是生物学领域的难题，人工智能技术可以有效地缩减测序流程和功能测定的时间，从而促进更深入的生物学研究。

参考文献

[1] Patel L, Shukla T, Huang X, et al. Machine Learning Methods in drug Discovery[J]. 2020, 25(22):5277.

[2] Wieder O, Kohlbacher S, Kuenemann M, et al. A compact review of molecular property prediction with graph neural networks[J]. 2020, 37:1-12.

[3] Gilmer J, Schoenholz S S, Riley P F, et al. Neural message passing for quantum chemistry[C], International Conference on Machine Learning. Sydney, Australia: PMLR, 2017:1263-1272.

[4] Kearnes S, McCloskey K, Berndl M, et al. Molecular graph convolutions: moving beyond fingerprints[J]. Journal of Computer-aided Molecular Design, 2016, 30:595-608.

[5] Laub V, Devraj K, Elias L, et al. Bioinformatics for wet-lab scientists: practical application in sequencing analysis[J]. 2023, 24(1):382.

[6] Walters W P, Barzilay Regina Accounts of chemical research. Applications of deep learning in molecule generation and molecular property prediction[J]. 2020, 54(2):263-270.

[7] Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules[J]. Journal of chemical information, sciences computer, 1988, 28(1):31-36.

[8] Floridi L, Chiriatti M. GPT-3: Its nature, scope, limits, and consequences[J]. Minds and Machines, 2020, 30:681-694.

[9] Devlin J, Chang M, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[10] Schütt K T, Arbabzadah F, Chmiela S, et al. Quantum-chemical insights from deep tensor neural networks[J]. Nature Communications, 2017, 8(1):13890.

[11] Gasteiger J, Groß J, Günnemann S. Directional message passing for molecular graphs[J]. arXiv preprint arXiv:2003.03123, 2020.

[12] Yun S, Jeong M, Kim R, et al. Graph transformer networks[J]. Advances in Neural Information Processing Systems, 2019, 32.

[13] Hu Z, Dong Y, Wang K, et al. Heterogeneous graph transformer[C], Proceedings of the Web Conference. New York, USA: Association for Computer Machinery, 2020:2704-2710.

[14] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in Neural Information Processing Systems, 2014, 27.

[15] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30.

[16] Zhang Xiaoyu. Towards End-to-end Semi-supervised Deep Learning for Drug Discovery[D]. 2018.

[17] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33:1877-1901.

[18] Zhao W, Zhou K, Li J, et al. A survey of large language models[J]. arXiv preprint arXiv:2303.18223, 2023.

[19] Veerasamy R, Rajak H, Jain A, et al. Validation of QSAR models-strategies and importance[J]. Int. J. Drug Des. Discov, 2011, 3:511-519.

[20] Uyanık G K, Güler N. A study on multiple linear regression analysis[J]. Procedia-Social and Behavioral Sciences, 2013, 106:234-240.

[21] Sperandei S. Understanding logistic regression analysis[J]. Biochemia Medica, 2014, 24(1):12-18.

[22] Noble W S. What is a support vector machine?[J]. Nature Biotechnology, 2006, 24(12):1565-1567.

[23] Qi Y. Random forest for bioinformatics[J]. Ensemble Machine Learning: Methods and Applications, 2012:307-323.

[24] Hollingsworth S A, Dror R O. Molecular dynamics simulation for all[J]. Neuron, 2018, 99(6):1129-1143.

[25] Durrant J D, McCammon J A. Molecular dynamics simulations and drug discovery[J]. BMC Biology, 2011, 9:1-9.

[26] Jorgensen W L. The many roles of computation in drug discovery[J]. Science, 2004, 303(5665):1813-1818.

[27] Case D A. Molecular dynamics and NMR spin relaxation in proteins[J]. Accounts of Chemical Research, 2002, 35(6):325-331.

[28] Cramer C J. Essentials of computational chemistry: theories and models[M]. Hoboken: John Wiley & Sons, 2013.

[29] Dirac P, Adrien M. Quantum mechanics of many-electron systems[J]. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 1929, 123(792):714-733.

[30] Rogers D, Hahn M. Extended-connectivity fingerprints[J]. Journal of Chemical Information and Modeling, 2010, 50(5):742-754.

[31] Rupp M, Tkatchenko A, Müller K, et al. Fast and accurate modeling of molecular atomization energies with machine learning[J]. Physical Review Letters, 2012, 108(5):058301.

[32] Behler J, Parrinello M. Generalized neural-network representation of high-dimensional potential-energy surfaces[J]. Physical Review Letters, 2007, 98(14):146401.

[33] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[34] Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks[J]. Stat, 2017, 1050(20):10-48550.

[35] Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks?[J]. arXiv preprint arXiv:1810.00826, 2018.

[36] Wang Y, Wang J, Cao Z, et al. Molecular contrastive learning of representations via graph neural networks[J]. Nature Machine Intelligence, 2022, 4(3):279-287.

[37] Hu W, Liu B, Gomes J, et al. Strategies for pre-training graph neural networks[J]. arXiv preprint arXiv:1905.12265, 2019.

[38] You Y, Chen T, Sui Y, et al. Graph contrastive learning with augmentations[J]. Advances in Neural Information Processing Systems, 2020, 33:5812-5823.

[39] Doersch C, Gupta A, Efros A A. Unsupervised visual representation learning by context prediction[C], Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile: IEEE Computer Society, 2015:1422-1430.

[40] Sun F, Hoffmann J, Verma V, et al. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization[J]. arXiv preprint arXiv:1908.01000, 2019.

[41] Maziarka Ł, Danel T, Mucha S, et al. Molecule attention transformer[J]. arXiv preprint arXiv:2002.08264, 2020.

[42] Chithrananda S, Grand G, Ramsundar B. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction[J]. arXiv preprint arXiv:2010.09885, 2020.

[43] Ahmad W, Simon E, Chithrananda S, et al. Chemberta-2: Towards chemical foundation models[J]. arXiv preprint arXiv:2209.01712, 2022.

[44] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.

[45] Duvenaud D K, Maclaurin D, Iparraguirre J, et al. Convolutional networks on graphs for learning molecular fingerprints[J]. Advances in Neural Information Processing Systems, 2015, 28.

[46] Bagal V, Aggarwal R, Vinod P, et al. MolGPT: molecular generation using a transformer-decoder model[J]. Journal of Chemical Information and Modeling, 2021, 62(9):2064-2076.

[47] Wang S, Guo Y, Wang Y, et al. Smiles-bert: large scale unsupervised pre-training for molecular property prediction[C], Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. New York, USA: ACM, 2019:429-436.

[48] Chen Y, Wu L, Zaki M J. Reinforcement learning based graph-to-sequence model for natural question generation[J]. arXiv preprint arXiv:1908.04942, 2019.

[49] Pareja A, Domeniconi G, Chen J, et al. Evolvegcn: Evolving graph convolutional networks for dynamic graphs[C], Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA: AAAI, 2020:5363-5370.

[50] Zhang C, Huang C, Yu L, et al. Camel: Content-aware and meta-path augmented metric learning for author identification[C], Proceedings of the 2018 World Wide Web Conference. Lyon, France: IW3C2, 2018:709-718.

[51] Li P, Wang J, Qiao Y, et al. An effective self-supervised framework for learning expressive molecular global representations to drug discovery[J]. Briefings in Bioinformatics, 2021, 22(6):bbab109.

[52] Wu Z, Jiang D, Wang J, et al. Knowledge-based BERT: a method to extract molecular features like computational chemists[J]. Briefings in Bioinformatics, 2022, 23(3):bbac131.

[53] Li J, Tang T, Zhao W X, et al. Pretrained language models for text generation: A survey[J]. arXiv preprint arXiv:2201.05273, 2022.

[54] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692, 2019.

[55] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[J]. arXiv preprint arXiv:2307.09288, 2023.

[56] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. Journal of Machine Learning Research, 2020, 21(140):1-67.

[57] Du Z, Qian Y, Liu X, et al. Glm: General language model pretraining with autoregressive blank infilling[J]. arXiv preprint arXiv:2103.10360, 2021.

[58] Christofidellis D, Giannone G, Born J, et al. Unifying molecular and textual representations via multi-task language modelling[C], International Conference on Machine Learning. Hawaii, USA: PMLR, 2023:6140-6157.

[59] Liang Y, Zhang R, Zhang L, et al. DrugChat: towards enabling ChatGPT-like capabilities on drug molecule graphs[J]. arXiv preprint arXiv:2309.03907, 2023.

[60] Wu T, He S, Liu J, et al. A brief overview of ChatGPT: The history, status quo and potential future development[J]. IEEE/CAA Journal of Automatica Sinica, 2023, 10(5):1122-1136.

[61] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in Neural Information Processing Systems, 2017, 30.

[62] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[63] Zhang Z, Liu Q, Wang H, et al. Motif-based graph self-supervised learning for molecular property prediction[J]. Advances in Neural Information Processing System, 2021, 34:15870-15882.

[64] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning[J]. arXiv preprint arXiv:2104.08691, 2021.

[65] Liu P, Yuan W, Fu J, et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing[J]. ACM Computing Surveys, 2023, 55(9):1-35.

[66] Shin T, Razeghi Y, Logan R, et al. Autoprompt: Eliciting knowledge from language models with automatically generated prompts[J]. arXiv preprint arXiv:2010.15980, 2020.

[67] Huang J, Ling C X. Using AUC and accuracy in evaluating learning algorithms[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(3):299-310.

[68] Nagelkerke N JD. A note on a general definition of the coefficient of determination[J]. Biometrika, 1991, 78(3):691-692.

[69] Lee J, Lee I, Kang J. Self-attention graph pooling[C], International Conference on Machine Learning. California, USA: PMLR, 2019:3734-3743.

[70] Ramsundar Bharath, Kearnes Steven, Riley Patrick, et al. Massively multitask networks for drug discovery[J]. arXiv preprint arXiv:1502.02072, 2015.

[71] Jiang B, Zhang Z, Lin D, et al. Semi-supervised learning with graph learning-convolutional networks[C], Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. California, USA: IEEE Computer Society, 2019:11313-11320.

[72] Schütt K, Kindermans P, Sauceda F, Huziel E, et al. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions[J]. Advances in Neural Information Processing Systems, 2017, 30.

[73] Guo Z, Yu W, Zhang C, et al. GraSeq: graph and sequence fusion learning for molecular property prediction[C], Proceedings of the 29th ACM International Conference on Information & Knowledge Management. Galway, Ireland: ACM, 2020:435-443.

[74] Liu S, Demirel M F, Liang Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules[J]. Advances in Neural Information Processing Systems, 2019, 32.

附 录

B 学位论文数据集:

<table>
<tr><td colspan="2">关键词</td><td colspan="2">密级</td><td colspan="2">中图分类号</td></tr>
<tr><td colspan="2">分子性质预测；多模态对齐；图神经网络；提示语微调；关系图谱</td><td colspan="2">公开</td><td colspan="2">TP</td></tr>
<tr><td>学位授予单位名称</td><td>学位授予单位代码</td><td colspan="2">学位类别</td><td colspan="2">学位级别</td></tr>
<tr><td>重庆大学</td><td>10611</td><td colspan="2">专业学位</td><td colspan="2">硕士</td></tr>
<tr><td colspan="2">论文题名</td><td colspan="2">并列题名</td><td colspan="2">论文语种</td></tr>
<tr><td colspan="2">基于多模态表示和提示语微调的分子性质预测</td><td colspan="2">无</td><td colspan="2">中文</td></tr>
<tr><td>作者姓名</td><td>赵卓然</td><td colspan="2">学号</td><td colspan="2">202114131105</td></tr>
<tr><td colspan="2">培养单位名称</td><td colspan="4">培养单位代码</td></tr>
<tr><td colspan="2">重庆大学</td><td colspan="4">10611</td></tr>
<tr><td>学科专业</td><td>研究方向</td><td colspan="2">学制</td><td colspan="2">学位授予年</td></tr>
<tr><td>计算机技术</td><td>人工智能</td><td colspan="2">3</td><td colspan="2">2024</td></tr>
<tr><td>论文提交日期</td><td>2024 年 5 月</td><td colspan="2">论文总页数</td><td colspan="2">63</td></tr>
<tr><td>导师姓名</td><td>周庆</td><td colspan="2">职称</td><td colspan="2">教授</td></tr>
<tr><td colspan="2">答辩委员会主席</td><td colspan="4">尚明生</td></tr>
<tr><td colspan="6">电子版论文提交格式
文本（ √） 图像（） 视频（） 音频（） 多媒体（） 其他（）</td></tr>
</table>