大语言模型智能体技术及其在 AI4Chemistry 中的应用进展

张迪

23110240066

计算机科学技术学院

上海人工智能实验室 AI4Science 实验室物质科学组

【摘要】本文介绍了大语言模型智能体技术及其在 AI4Chemistry 中的应用进展。大语言模型智能体是一种基于 Transformer 架构的深度学习模型，能够处理复杂的自然语言生成和理解任务。本文分析了大语言模型智能体的三项核心能力：指令遵循、上下文学习和思维推理，以及它们在化学领域的应用场景。文章还介绍了大语言模型智能体的主要组成部分，如分词器、词嵌入、解码器、预训练、有监督学习、偏好对齐等，以及它们的技术原理和发展趋势。

【关键词】大语言模型；AI4Science；智能体技术；

介绍

随着深度学习技术的发展，自然语言处理（NLP）领域出现了一种新的技术范式，即大语言模型（LLM）。LLM 是一种能够从大规模的文本数据中学习语言的统计规律和语义知识，从而实现对自然语言的理解和生成的技术。

LLM 作为一种强大的多模态智能体，不仅能够与人类进行自然语言交互，还能够调用各种工具和资源，完成复杂的任务和目标。LLM 在各个领域都有广泛的应用和影响，其中化学领域是一个典型的例子。化学是一门基于实验和理论的科学，涉及到大量的文献阅读、实验设计、数据分析等工作，这些工作都需要高度的专业知识和技能。

LLM 能够通过自然语言交互，协助化学家完成这些工作，提高化学研究的效率和质量，这就是 AI4Chemistry 的概念。本文旨在介绍 LLM 的概念、发展历程和在 AI4Chemistry 中的应用进展，为读者提供一个全面的视角和参考。本文的结构如下：第一部分介绍了 LLM 的基本概念和发展历史；第二部分介绍了 LLM 的主要技术和方法；第三部分介绍了 LLM 在 AI4Chemistry 中的应用案例和效果；第四部分展望了 LLM 的未来发展方向和挑战；第五部分总结了本文的主要内容和贡献。希望本文能够对读者有所帮助和启发。

LLM 的进展

语言建模技术属于序列建模的一种，传统上有基于特征提取的方法，如 N-gram，语法树和 TF-IDF 等和基于自回归的方法，如 GRU，LSTM，Transformer 等。当前主流的语言模型是以 Transformer 自回归模型为主的深度学习模型。

长短期记忆网络（LSTM）由 Sepp Hochreiter 和 Jürgen Schmidhuber 于 1997 年提出，是自然语言序列建模领域的里程碑工作。LSTM 旨在解决传统递归神经网络（RNN）中出现的梯度消失和梯度爆炸问题。其核心创新在于引入了“门”结构，这些结构有效地控制信息在网络中的流动，从而使网络能够学习长期依赖关系。LSTM 在序列数据建模，特别是语言建模和时间序列预测方面，表现出显著的性能提升。词袋模型（Bag of Words，BoW）是文本处理的一种早期简单方法。它通过将文本转换为单词出现次数的表示，忽略了语法和单词顺序。尽管这种方法有其局限性，它为文本分类和搜索引擎优化提供了基础的表示方法。GloVe（Global Vectors for Word Representation）模型，由 Jeffrey Pennington, Richard Socher, 和 Christopher Manning 于 2014 年提出，旨在结合全局矩阵分解和局部上下文窗口的优势。GloVe 通过共现矩阵和矩阵分解来学习词向量，有效地捕捉了单词间的全局统计信息，为词嵌入提供了一种高效方法。

序列到序列（Seq2Seq）模型，由 Ilya Sutskever 等于 2014 年提出，处理从一个序列到另一个序列的转换任务，如机器翻译。Seq2Seq 模型使用一个编码器来理解输入序列，然后用解码器生成输出序列。该模型使机器能够处理复杂的序列转换任务，如翻译和文本摘要。Transformer 模型，由 Ashish Vaswani 等人于 2017 年提出，旨在寻找一种比循环神经网络更有效的方法处理序列数据。Transformer 的主要创新在于使用自注意力机制来处理输入数据的不同部分，并能够并行处理。该模型在机器翻译等任务中显著提高了效率和性能，并为后续模型如 Bert 和 GPT 奠定了基础，更彻底改变了机器学习和深度学习的主流范式。Bert（Bidirectional Encoder Representations from Transformers），由 Jacob Devlin 等人于 2018 年提出，旨在利用 Transformer 模型更有效地学习语言表示。Bert 使用双向 Transformer 来预训练深度双向表示，在广泛的自然语言处理任务中取得显著进步。

Transformer 模型的独特之处在于，它完全放弃了传统的循环网络（RNN）和卷积网络（CNN），转而依赖自注意力机制来处理序列数据。Transformer 由编码器和解码器组成，每个编码器包含自注意力层和前馈神经网络，而解码器则增加了一个额外的注意力层以关注编码器的输出。

在自注意力机制中，Transformer 通过 Query（查询），Key（键）和 Value（值）的交互来计算权重分布，这样模型在处理一个单词时就能够同时关注句子中的其他单词。由于 Transformer 不像 RNN 那样可以处理序列中的位置信息，因此它通过向每个输入嵌入添加位置编码来提供这种信息，其中位置编码通常是基于正弦和余弦函数。

此外，Transformer 的每个编码器和解码器层都包含一个全连接的前馈网络，它对每个位置的表示进行独立处理。为了帮助避免深层网络中的梯度消失问题，模型还增加了层归一化和残差连接。而多头注意力机制则允许模型将注意力分成多个头，每个头专注于序列的不同方面，从而增强模型关注不同位置的能力并提高信息捕获的全面性。Transformer 模型因其优异的性能和灵活性，在自然语言处理领域得到了广泛应用，尤其是在机器翻译、文本摘要等任务中表现卓越。

常用于文本序列建模的 Transformer 模型中，以应用于序列到序列学习范式的 Encoder2Decoder 模型发展来的主要变体 Encoder2Decoer 类，Encoder-only 类和 Decoder-only 类为主。

其中 Encoder-only 类常用于掩码语言模型（Masked Language Model, MLM）建模的自监督预训练学习任务中，适用于进行语言特征的建模与提取，其中应用较广的有 BERT、ALBERT 等；

Decoder-only 类模型常用于下一字符预测（Next token prediction）类语言模型建模的生成式自监督预训练学习任务，其中应用较广的是以 GPT（Generative Pre-Training）系列为代表的因果语言模型（Casual Language Model， CasualLM），如 GPT-3，GPT-4，LLaMA 等；GPT（Generative Pre-trained Transformer，由 OpenAI 于 2018 年提出，基于 Transformer 架构的预训练和微调方法。GPT 在大规模数据集上进行预训练，然后在特定任务上进行微调，在文本生成、语言理解等多种任务上取得优异成果。2022 年发布了通过指令微调和人类偏好对齐强化学习算法训练的生成式对话模型 ChatGPT 系列，其以强大的内容理解能力和零样本学习能力深刻改变了自然语言领域研究。

Encoder2Decoer 类模型尝试在生成任务中进行掩码等监督信号的训练，以期增强模型的非局部建模能力，其中应用较广的有 BART，GLM 等。

在语言模型技术的发展历程中，Decoder-only 类模型，尤其是 GPT-3 表现出了强大的文本生成能力、逻辑推理能力和零样本、小样本学习能力，因此逐渐成为了应用最广、影响最大的语言模型技术。

对话模型技术发展自对话系统技术，传统的对话系统技术涵盖模板式、检索式和生成式，前两者通过特征提取或规则对用户的输入进行处理，在知识库中进行相关内容检索，通过模板方法合成最终回复，常用于客服系统场景；后者则基于生成式语言模型技术，通过自回归或端到端的方式进行回复文本的生成；随着语言模型技术的发展，基于 LLM 的生成式对话系统成为主流。

因此，我们将重点讨论以 GPT 为主的 LLM，及其对话微调方法的发展情况。

分词器

分词器（Tokenizer）是用于预处理语句，如补充结束符、切分语句到分立次元（Token）并编码到 Token ID 的模型组件。其中常用的有：

字节对编码(BPE)是对自然语言处理(NLP)数据压缩技术，它的工作原理是迭代地合并数据集中最频繁的字节或字符对，从而创建固定大小的子词表。这种方法不仅有助于通过将未收录词分解为已知子词来有效对应它们，而且还确保词汇表适应数据集的语言或领域细节。

SentencePiece 是一个分词器和解码库，它将输入视为原始字节。该方法统一支持多种语言和符号，包括 BPE 语言模型和 unigram 语言模型。它的主要优势在于语种的不相关性，使其对没有明确单词边界的语料特别有效，并且与端到端神经网络模型兼容。

主要用于中文文本的 Jieba 采用了基于词典的方法和隐马尔可夫模型。这种组合可以有效地分割中文文本，尤其适合处理复杂的中文语言，提供快速的处理能力，并通过用户自定义的词典适应特定的领域。

词嵌入

词嵌入技术的发展是自然语言处理（NLP）领域的重要里程碑，它关注于如何将离散的文本次元有效地转换成计算机可以处理的连续数值形式。最初，词袋模型（Bag of Words, BoW）作为一种简单的词表示方法，通过构建一个所有单词的索引，在文本中的每个单词被转换为对应索引的概率向量。然而，这种方法存在一个重大缺陷，即无法捕捉到词与词之间的上下文关系，也不能表达词之间的语义相似性。

为了解决这一问题，Word2Vec 技术应运而生。它通过两种主要的架构——连续词袋（CBOW）和跳跃式 gram（Skip-Gram）模型——来训练词向量。CBOW 模型预测目标词基于其上下文，而 Skip-Gram 模型则反过来，使用目标词来预测上下文。这两种模型都通过神经网络实现，能够捕捉到更丰富的词语间的关联性和语义信息。

随着深度学习的兴起，Embedding 神经网络层成为了流行的深度学习框架（如 PyTorch）中的一个关键组件。它提供了一种简单而有效的方式来将每个词映射到一个高维空间中，使得模型能够在训练过程中学习到词的嵌入表示。这种方法不仅提高了模型处理词汇的灵活性，而且因其在内存和计算上的高效性，被广泛应用于各种 NLP 任务中。

最近，稀疏嵌入技术的出现旨在解决词嵌入中的存储和计算效率问题。与传统的密集嵌入（如 Word2Vec 生成的嵌入）不同，稀疏嵌入生成的是一个稀疏的向量，其中大部分元素为零。这种表示方式减少了存储空间的需求，并能在保留重要语义信息的同时提高计算效率。它在处理大规模词汇表或需要高度优化的 NLP 应用中表现出了显著的优势。

解码

随着大型预训练语言模型的兴起，解码层的技术不断演化，以期提高文本生成的准确性和相关性。LMHead，或语言模型头，是 Pytorch 实现中的一个关键组件，通常位于模型的顶层。它负责将模型的复杂内部表示转换为最终的词汇输出。这一转换通常通过一个线性层实现，它将隐藏状态映射到一个与词汇表大小相同的输出空间，然后通过 softmax 函数转化为词汇概率分布。这种结构的灵活性使得 LMHead 能够适应多种不同的 NLP 任务，如文本生成、摘要和翻译。

Beam Search 是一种启发式图搜索算法，经常用于自然语言处理中的序列生成任务，特别是在大型语言模型的解码阶段。与传统的贪婪搜索不同，Beam Search 在每一步中保留了多个候选解决方案（称为“beam”），这使得算法能够探索更多可能的序列组合，从而提高生成文本的质量。

投机解码（Speculative Decoding）方法通过同时考虑多个可能的解码路径，以此减少必须计算的路径数量，从而加速解码过程。这种方法在模型需要快速响应的场景（如实时对话系统）中尤为有价值。它通过预先计算和缓存可能的路径选择，减少了等待时间。

对比学习解码（Contrastive Decoding）利用了对比学习的原理，通过比较不同的解码候选来改善解码过程。这种方法强调上下文的理解，使模型能够生成更加符合给定上下文的文本。它通过对比正面和负面样本来优化解码过程，从而提高生成文本的准确性和相关性。

MCTS（蒙特卡洛树搜索）解码是一种复杂的启发式搜索技术，用于在决策过程中评估多个可能的路径。这种方法在长文本生成或维护复杂对话系统中的一致性和相关性方面表现出色。MCTS 通过模拟不同的解码路径并评估其结果来选择最佳路径，这使得它在处理需要高度非确定性和创造性的任务时特别有效。

预训练

早期的预训练模型，如 Word2Vec 和 GloVe，侧重于学习词汇的向量表示，但这些模型通常忽略了上下文信息，导致在复杂的语言理解任务上性能有限。随后，出现了以 ELMo 为代表的上下文敏感型预训练模型，这些模型能够生成考虑了周围单词的词嵌入，从而在语义理解上取得了显著进步。

随后，随着 Transformer 架构的引入，预训练技术迈入了一个新的时代。BERT（Bidirectional Encoder Representations from Transformers）是一个里程碑，它通过双向 Transformer 编码器学习上下文相关的表示。BERT 及其变体（如 RoBERTa、ALBERT 等）利用掩码语言模型（MLM）和下一句预测（NSP）等任务进行预训练，大大提高了对语言深层次理解的能力。

GPT 系列（Generative Pretrained Transformer）则采取了另一种方法，其核心特点是使用自回归的方式进行训练。在自回归训练中，模型的任务是预测下一个词，即所谓的“下一个词预测”任务。具体来说，给定一个文本序列的部分或全部单词，GPT 需要预测序列中下一个单词的概率分布。例如，给定序列“the cat sat on the”，GPT 的任务是预测紧接着“the”的最可能的词。

在技术实现上，GPT 系列使用单向 Transformer。这意味着在预测一个词时，模型只能使用该词之前的词作为上下文。每个词都被映射到一个高维空间中，产生一个词嵌入向量。这些词嵌入通过多层 Transformer 网络进行处理，网络的每一层都使用自注意力机制和前馈神经网络。自注意力机制允许模型在生成每个词的表示时考虑到前面的所有词，这有助于捕获长距离依赖关系。

语言模型损失（LM Loss）是用于训练 GPT 模型的关键。LM Loss 是一个基于最大似然估计的损失函数。它计算的是模型预测正确单词的概率的负对数似然损失。换句话说，对于给定的文本序列，LM Loss 度量的是模型预测每个真实单词的概率。训练过程中，模型的目标是最小化这个损失函数，这样可以使模型更准确地预测下一个词，这一过程可以用以下公式表示：

$$\text{LM Loss} = -\sum_{i=1}^{N} \log P(w_i|w_1, w_2, \ldots, w_{i-1}; \Theta)$$

其中：

- (N) 是序列中的单词总数。
- (w_i) 是序列中的第 (i) 个单词。
- ($P(w_i|w_1, w_2, \ldots, w_{i-1}; \Theta)$) 是在给定之前的单词 ($w_1, w_2, \ldots, w_{i-1}$) 和模型参数 (Θ) 的情况下，模型预测第 (i) 个单词 (w_i) 的概率。
- (log) 表示对数函数，通常使用自然对数。

这个损失函数的目标是最大化模型预测序列中每个真实单词的概率，或等价地，最小化预测概率的负对数似然。在实际训练过程中，通过反向传播算法和梯度下降方法来调整模型参数Θ，以最小化整个训练集上的 LM Loss。

GPT 系列模型的这种训练方式使得它们在生成连贯、流畅且上下文相关的文本方面表现出色。每一次迭代，模型都会根据自身已生成的文本继续生成下一个词，这种方式在自然语言生成任务中尤其有效。但是这种生成方式也使得模型生成文本的开销与历史文本长度呈现出二次方的较高复杂度。

有监督学习

LLM 的有监督学习常用于进行指令微调或对话微调，一般 SFT 数据集中的样本由{指令，输入，输出，对话历史}组成，四项内容通过模板和特殊 Token 组合成最终输入文本输入到模型中，模型需要根据输入，通过自回归范式生成输出文本直到到达终止条件，如遇到终止符等。训练过程可以通过以下公式来表示：

$$\text{Loss}_{\text{fine-tune}} = -\sum_{(x,y)\in D} \log P(y|x; \Theta)$$

其中：

- D 是微调过程中使用的有监督数据集，包含输入-输出对 (x, y)。
- x 是输入数据，如用户指令或对话上下文。
- y 是期望的输出，如正确的响应或执行指令的结果。
- $P(y|x;\Theta)$ 是在给定输入 x 和模型参数 Θ 的情况下，模型预测输出 y 的概率。
- log 表示对数函数，通常使用自然对数。

在微调阶段，目标是调整预训练的模型参数 Θ，使得模型更好地适应特定的任务或数据集。通过最小化微调损失（$\text{Loss}_{\text{fine-tune}}$），模型学习如何根据输入 x 生成正确的输出 y。这种方法特别有效于增强模型在特定任务上的性能，比如执行特定的用户指令或在对话系统中生成更加准确和自然的响应。

偏好对齐

“基于人工反馈的强化学习”是最经典的偏好对齐方法，简称为 RLHF。它将基于策略的强化学习方法与人在环过程相结合，通过合并从人类反馈建模的奖励信号，将模型奖励与人类在复杂任务中的偏好相结合。OpenAI 在 InstructGPT 中引入了近端策略优化(PPO)的 RLHF 方法，在 ChatGPT 系列产品的应用中取得了重大进展。

人工反馈可以用奖励函数表示，该函数反映了人类对模型生成的文本的满意度。人类反馈可以通过不同的方式收集，例如比较、评分、标记、纠正或语言反馈。不同类型的反馈具有不同的优缺点，需要根据任务的特点和目标选择合适的反馈类型。然后这些数据可以用于训练奖励模型，该模型可以在给定输入和输出的情况下预测人工反馈分数。奖励模型的训练可以使用不同的损失函数和正则化技术来减少噪声和偏差的影响。最后，我们可以优化一个语言模型，该模型可以生成输出，使给定输入的奖励模型分数最大化。语言模型的优化可以使用不同的强化学习算法来完成，如策略梯度、actor-critic 或自然梯度。

直接偏好优化是 PPO RLHF 的进一步扩展，假设偏好是由底层奖励模型产生的，$r^*(y, x)$,这对我们来说是不可访问的。有多种方法可以对偏好进行建模，其中布拉德利-特里(Bradley-Terry, BT)模型是一种流行的选择(尽管更通用的 Plackett-Luce 排名模型也可以与框架兼容，如果可以获得多个排名答案)。BT 模型规定人类的偏好分布p^*可以表示为:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp(r^*(x, y_1))}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}.$$

假设我们从偏好函数分布p^*中采样静态比较数据三元组$\mathcal{D} = \left\{x^{(i)}, y_w^{(i)}, y_l^{(i)}\right\} i = 1^N$，我们可以将奖励模型参数化为$r\phi(x, y)$并通过最大似然估计参数。假设是一个二分类问题，我们得到负对数似然损失:

$$\mathcal{L}_R\left(r_\phi, \mathcal{D}\right) = -\mathbb{E}_{(x, y_w, y_l)\sim\mathcal{D}}\left[\log\sigma\left(r_\phi(x, y_w) - r_\phi(x, y_l)\right)\right]$$

其中$\color{red}{\sigma}$表示 logistic 函数。

从与现有方法相同的强化学习目标出发，即在 KL 散度约束下最大化奖励。

$$\max_{\pi_\theta}\mathbb{E}_{x\sim D,y\sim\pi_\theta(y|x)}r_\phi(x,y)-\beta D_{KL}(\pi_\theta\left(y|x\right)||\pi_{ref}(y|x)\Big)$$

我们可以推导出最优解的解析解，并将奖励函数表示为策略、参考策略和未知的划分函数。

$$\pi_r(y|x)=\frac{1}{Z(x)}\pi_{ref}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

$$Z(x)=\sum_y\pi_{\text{ref}}(y\mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

$$r(x,y)=\beta\log\frac{\pi_r(y|x)}{\pi_{ref}(y|x)}+\beta\log Z(x)$$

将这种重新表达式应用于实际奖励函数$r^\star$和最优策略$\pi^\star$，我们观察到配分函数的取消，允许根据最优策略和参考策略表示人类偏好概率。

$$r^*(x,y)=\beta\log\frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)}+\beta\log Z(x)$$

$$p^*(y_1\succ y_2|x)=\frac{\exp\big(r^*(x,y_1)\big)}{\exp\big(r^*(x,y_1)\big)+\exp\big(r^*(x,y_2)\big)}$$

$$p^*(y_1\succ y_2|x)=\frac{1}{1+\exp\left(\beta\log\frac{\pi^*(y_2|x)}{\pi_{ref}(y_2|x)}-\beta\log\frac{\pi^*(y_1|x)}{\pi_{ref}(y_1|x)}\right)}$$

因此，可以使用最大似然目标来优化参数化策略，这构成了 DPO 训练过程的优化目标。

$$\mathcal{L}_{DPO}\big(\pi_\theta;\pi_{ref}\big)=-\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}-\beta\log\frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\right)\right]$$

大语言模型的三项能力

观察大语言模型的应用与发展，我们能够发现使得 LLM 取得巨大成功的因素中，三类能力在下游任务中应用最多，训练过程和模型评测中受到的重视也最多，

指令遵循

指令遵循能力是模型理解并执行用户输入的指令的能力，一个良好的对话模型既要能够准确地理解用户输入的指令，生成指令要求的对应输出内容，还应在执行用户指令时遵循较对话指令更高一级的约束，考虑质量道德、法律、用户使用协议的约束，避免生成幻觉或有害内容。

上下文学习

上下文学习（In-Context Learning，ICL）是模型从历史语境上下文中学习到预训练阶段较少或不存在的知识和信息的能力，LLM 以其强大的上下文学习能力，表现出了良好的零样本学习和小样本学习能力，可以在提供少量示例或不提供示例的提示词工程（Prompt Engineering）方法场景中，完成复杂任务的学习和泛化，如逻辑推理，编程，翻译，文学创作和角色扮演等。

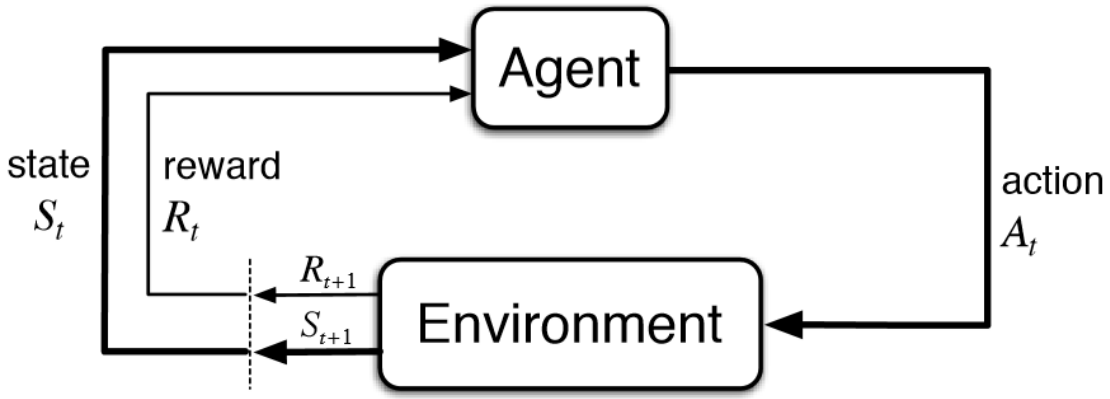
思维推理能力

思维推理能力与上下文学习能力密不可分，但更侧重于复杂概念，如谓词逻辑，数学，符号逻辑等的综合、分析推理能力，长跳数的多轮推理能力。以思维链为代表的简易提示词工程方法的辅助下，LLM 能表现出近似人类水平的数学和逻辑推理解题水平。

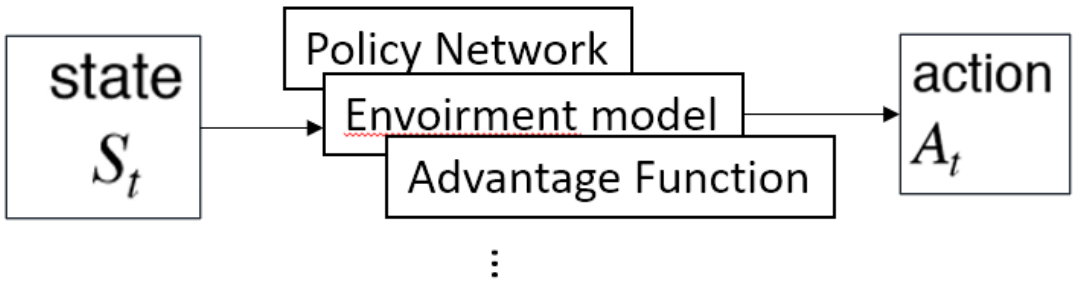
LLM Agent 的进展

Agent

Agent 是来自于强化学习领域的经典概念，一个 Agent 是一个可以与环境进行互动，感知环境的状态和反馈作为输入，以一定方法产生并执行 Action 影响环境，使得环境的状态变化的对象。一个 Agent 的典型工作流程如下图所示，

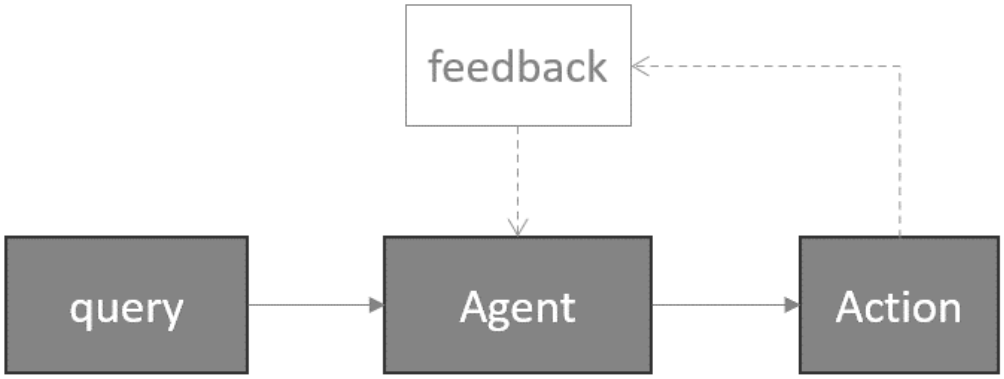


基于 RL 方法的 Agent 可概括为以下结构：



其中 RL Agent 的核心是产生由 State 到 Action 映射的 RL 方法，一般由策略网络，世界模型，值函数等方法组成。

而以 LLM 为中心的 Agent 的结构则可以概括为以下形式：



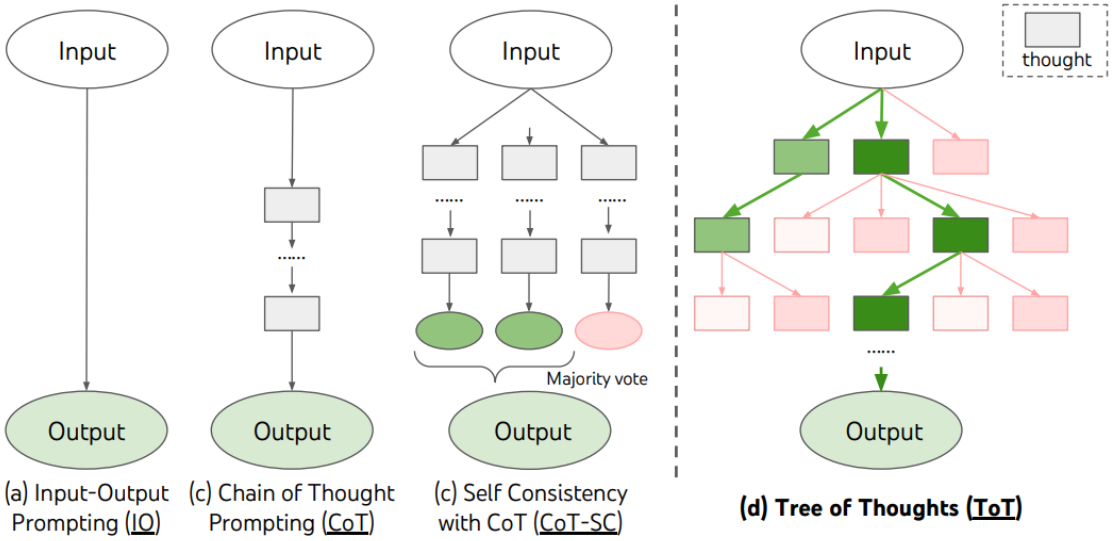
其中 Agent 一般是一个具有可拓展能力的对话模型，Query 来自于环境状态的变化或用户的请求、传感器的事件等，Agent 同样会产生一个相对应的 Action 影响环境。但是其中有两点不同：一是 LLM Agent 的环境反馈是可选的，二是 LLM Agent 工作的环境可以由用户输入或者状态描述组件所替代，来提供 Agent 的 query 输入。

推理规划

LLM Agent 的推理能力，往往侧重于对历史状态和新的 query 输入的分析、思考过程，和产生候选行动的思维逻辑，这类能力来自于模型的上下文学习能力，可以通过思维链等提示词方法进行辅助强化。

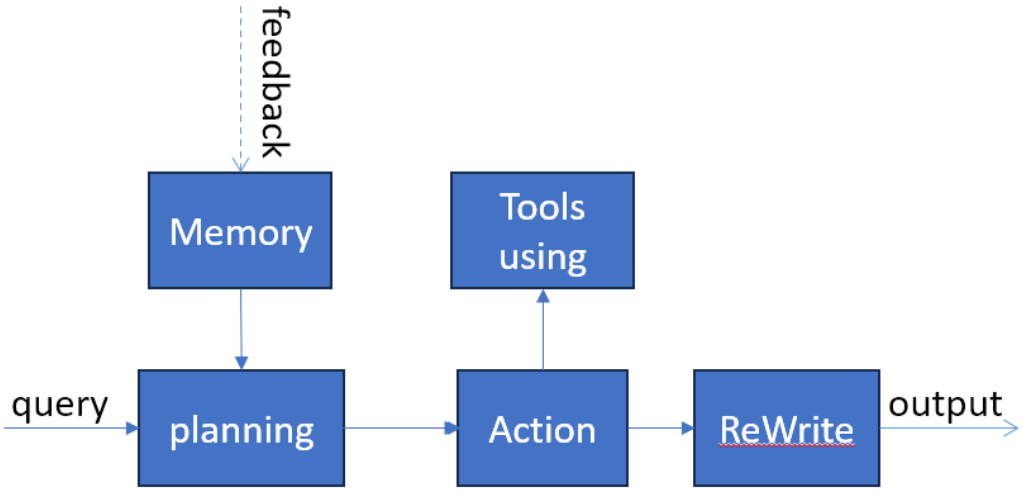
思维链的策略较为直观，类比于人类的思考过程，我们会天然的倾向于将复杂的因果推理过程分解为多步的、较简单的子推理过程，这一思维方法被称为“Step by step”，通过“Step by step”的提示词方法和 Fewshot 示例（或无示例），可以引导模型主动的将复杂生成任务分解为多步子任务，并逐步解决任务，输出最终的结果。比模型没有中间步骤的”一步(Input-Output Prompt)”产生结果有着更好的准确率和稳定性。

沿着思维链方法的思想，之后产生了 Tree of Thought、Monte Carlo Tree of Thought、Graph of thought 等系列工作。这些工作的核心思想都是通过启发式方法，引导模型在一个由思维步骤组成的搜索空间中搜索出一条价值更高的思维链路，具体搜索方法则由任务的复杂程度和搜索空间的建模所决定，如 UCB、蒙特卡洛方法、最短路径和启发式树搜索等。



工作流程

LLM Agent 可以通过组合多个简单 Action 完成更复杂的 Action，因此其具体工作流程的规划和对环境反馈的利用方法就具有了更强的多样性。观察一个 LLM Agent 的典型构成：



记忆

LLM 的“记忆“包括以下几个层级（Hierarchy）：预训练权重知识、上下文窗口记忆、外部检索输入，其中外部检索输入也被称作检索增强的生成（RAG）。

预训练权重知识是 LLM 最稳固、最鲁棒的知识存储位置，通过结合合适的提示词设计，可以充分探索和利用 LLM 预训练权重中存储的海量知识，这也是模型推理、学习能力和先验知识的来源。

上下文窗口记忆（In-context Memory）是模型上下文窗口长度内存储的历史输入和输出结果的产出，以 LLM 主流的旋转位置编码（Rotary positional Embedding）而言，由于距离的增加，距离当前轮次较久远的历史可能会逐渐衰减，直到溢出上下文窗口长度并被移除。目前存在很多拓展模型上下文窗口长度的方法，如 Tangent-kernel aware interpolation,Longlora 和 StreamingLLM 等，但是仍然无法避免长度增加带来的长距关联衰减和记忆检索时的失真。上下文窗口记忆是最可靠的信息输入方式，也是模型利用信息最直接的信息存储方式，但是其受限于上下文窗口长度而存在无法避免的上限。

检索增强生成中可用的信息检索方式常有三类来源：向量数据库，网络检索，知识图谱。向量数据库技术主要涵盖两类核心组件，向量化嵌入算法（Embeddings）和向量检索算法（Retrieval）；向量化嵌入算法主要有 OpenAI 的 Ada 嵌入模型、Setence Transformer、BGE、Spacy 的语义相似度系列模型等；向量检索算法主要是通过余弦相似度或欧氏距离度量，在数据库检索最相近的条目，常用的有 Faiss,KNN,Mrpt 等；较成熟的商业化向量数据库解决方案有: pinecone,GPTCache;

网络检索方法有着更强和更广泛的适用性,通过合适的提示词设计和搜索结果筛选策略，可以直接获取可靠且可用的信息，如 New Bing 等应用中已经取得了巨大成功。

工具调用

LLM Agent 可以通过 API 调用方式直观的接入不同的工具，进行更加复杂的信息处理来完成任务。一个典型的 LLM 工具调用过程涉及到三个话题：工具选择和组织，工具函数参数生成，工具调用时机选择。模型需要能够在恰当的时机出发函数调用，选择适当的工具（或多个工具组成工具链），组织生成合法的函数参数结构体，并对函数返回值进行解析利用。

工具函数参数结构体的生成与结构化约束生成问题有关，微软的 guidance 通过终止规则设计和令牌修复技术，可以约束模型合成合法有效的 JSON 结构体。

ControlLLM 利用 Graph of Thought 技术对工具链使用进行编排组合，利用简单的第三方工具完成涵盖多模态解析、生成在内的复杂任务。

输出时增强

输出时增强也被称作重写（Re-Write）策略，主要有三类，

第一种是训练时修正，比如 Claude 就是先生成回答，然后自我批评，再修改原来的回答，修正后的问答再回流重新微调模型；

第二种是生成时修正，通过自动反馈引导生成过程，比如 tree of thought 就是对推理过程的每一步都打分，找到最优的推理路径；

第三种是生成后修正，生成回答后基于人类反馈即时修改回答。

工作流程设计

在以上组件的基础上，LLM Agent 的工作流程设计，特别是对环境反馈信息的利用，可以通过结合高级语言处理能力和特定的功能性增强，使得 LLM 更加智能和适应性强。

ReAct 是一种先进的 LLM Agent 技术，旨在提高模型的反应能力和交互质量。它通过实现更快的响应时间和更高效的信息检索，优化了用户与模型的互动。ReAct 的关键在于它能够快速识别用户的意图，并提供即时且相关的反馈，使得对话更加自然和流畅。

ReFlexion 则是一种专注于反思和自我评估的技术。它使 LLM 能够在生成回答后进行自我反思，从而提高了回答的准确性和相关性。ReFlexion 的核心是其能力在生成回答后评估其自身的输出，通过内部机制来判断回答的准确性和可能的改进方式。

MRKL（Memory, Reasoning, Knowledge, and Learning）是另一种重要的 LLM Agent 技术，它集成了记忆、推理、知识和学习的能力。这种技术使 LLM 不仅能够存储和检索大量信息，还能进行逻辑推理和持续学习。MRKL 的特点在于其综合能力，能够在复杂的问题解决和决策制定中模拟人类的思维过程。

多模态感知和具身控制等技术前沿

CLIP 模型作为现代视觉-语言模型技术的里程碑式工作，CLIP 通过对大量的图像和文本对进行对比学习，能够理解图像和文本之间的关系。它在多种视觉任务上展现出强大的零样本泛化能力，为后续工作奠定了基础。作为 CLIP 的姊妹模型，DALL-E 则可以根据文本描述生成相应的图像，进一步显示了视觉-语言模型在创造性任务上的潜力。Flamingo 则将预训练语言模型和预训练视觉模型进行组合，通过处理任意交错顺序的文本和图像序列完成多模态理解和交互任务。随着 2022 年以来大语言模型（LLM）技术的蓬勃发展，大语言模型表现出了强大的表征学习、零样本泛化和模态迁移学习能力，视觉语言模型的研究热点逐渐从模态融合对齐技术过渡到以自然语言文本为中心的视觉大语言模型（VLLM）和视觉-文本指令微调训练的范式上来，在这一时期中，涌现了 LLaVa、BLIP 和 GPT4V 等优秀工作。

当前多模态基础模型的研究逐渐呈现出以 LLM 为中心的发展趋势，这其中涌现出一批值得关注的优秀工作，如 LLaVa、Fuyu、OneLLM 和 GPT4V。

Fuyu 使用了与 LLaVa 不同的架构，它没有使用冻结参数的 LLM，而是在预训练阶段将视觉数据与文本数据一同训练；在进行编码处理时，文本数据首先被 tokenizer 处理，并通过特殊的占位符 Token 为图像数据预留一定数量的空位，原始图像在被划分为 patch 后，经过维度变换填入这些空位，与文本数据在同一个编码器中进行统一化的处理。

OneLLM 是 Han 等人前作 Meta-Transformer 的后续工作，在 Meta-Transformer 中作者尝试设计了一个以前缀引导的统一多模态词向量编码器网络，能将图像、图、点云和脑电等模态数据编码到统一的空间中，在 OneLLM 中，作者使用 Meta-

Transformer 网络作为文本数据以外其他模态的统一特征提取器，并在模型架构中采用了 MoE 思想，针对不同模态前缀引导的数据针对性微调了不同的 Expert 和路由权重，将 LLM 的多模态处理能力从文本-图像扩展到了图、点云和脑电等十余种不同模态数据。

当前在视觉-语言模态融合任务上，主流的办法有特征投影、对比学习和跨模态注意力融合等，特征投影方法仅需训练少量适配器层（Adapter）即可将新模态的信息对齐到大语言模型的文本表征空间，因此被广泛用于扩展 LLM 的多模态信息处理能力，其中较有影响力的工作有 BLIP-2、LLaVa 等；通过对比学习进行特征语义对齐的方法的主要代表是 CLIP 和 BLIP-1 等工作；跨模态注意力融合方法的主要代表工作有 ProVALA 和 Muffin 等。

CLIP 是首个处理计算机视觉的多模态模型。它利用大约 4 亿个图像-文本对进行自监督的对比学习训练，展现了出色的“零样本学习”能力，能准确预测未见过的类别。BLIP 模型由 Junnan Li 等人提出，BLIP 在视觉-语言理解和生成任务中转换灵活。它通过多任务学习的引导和过滤利用网络数据，高效的实现了图文检索、图像描述和视觉问答等多项任务的 SOTA。

LLaVa 模型由 Haotian Liu 等人提出，LLaVa 是结合视觉编码器和 Vicuna 的大型多模态模型。它通过自回归语言模型和 Transformer 架构实现强大的对话能力，为多模态语言模型提供了近似 GPT-4 的性能，并在多项任务指标上达到了 SOTA。
ProVLA 采用基于 Transformer 的视觉和语言模型来生成多模态嵌入，专注于理解时尚电子商务环境中复杂的用户意图，其具有两步学习过程、基于交叉注意力的融合编码器以及动量队列式硬负采样机制，以处理噪声训练数据。在 Fashion 200k 和 Shoes 基准数据集上，ProVLA 展现了超越现有方法的卓越性能。

Muffin 代表了多模态大型语言模型（MLLM）的一种新方法，有效地作为视觉和语言模块之间的桥梁。这个框架采用预训练的视觉-语言模型（VLM）作为视觉信号的提供者，从而消除了额外的特征对齐预训练的需要。Muffin 使用 VLMs 的嵌入空间中的一组查询向量来感知视觉表示，优化视觉模块与大型语言模型（LLM）的连接，无需进行额外的对齐过程。Muffin 在多种视觉-语言任务中取得了最新技术水平的性能，超越了 LLaVA 和 InstructBLIP 等模型。

随着机器人，尤其是仿人机器人和大语言模型技术的发展，持乐观观点的研究者认为大语言模型引导的机器人行动规划是通向具身智能（Embodied intelligence）的重要途径，而这其中最受关注的就是视觉-语言导航和多模态感知技术。

机器人学多模态感知技术以谷歌在 2023 年提出的 PaLM 2 和 Q transformer 最受关注，它们各自解决了机器学习和人工智能领域中的一些独特挑战。

PaLM E（Pathways Language Model-E）是一个通用多模态语言模型，用于体现推理、视觉语言任务和语言任务。它能将视觉语言领域的知识融入体现推理中，例如在复杂环境下的机器人规划和回答关于可观察世界的问题。PaLM E 通过结合图像、神经 3D 表示或状态等多种模态输入与文本令牌，创建了一种新的"体现语言模型"，直接将实体代理的连续感测输入融入语言模型中。

Q Transformer 则介绍了一种可扩展的离线强化学习方法，用于从大型离线数据集中训练多任务策略。该方法利用 Transformer 为通过离线时间差异备份训练的 Q 函数提供可扩展表示。通过离散化每个动作维度并将每个动作维度的 Q 值表示为单独的令牌，Q Transformer 能有效地应用高容量序列建模技术进行 Q 学习。它在复杂的真实世界机器人操作任务中表现出色，超越了以前的离线 RL 算法和模仿学习技术。

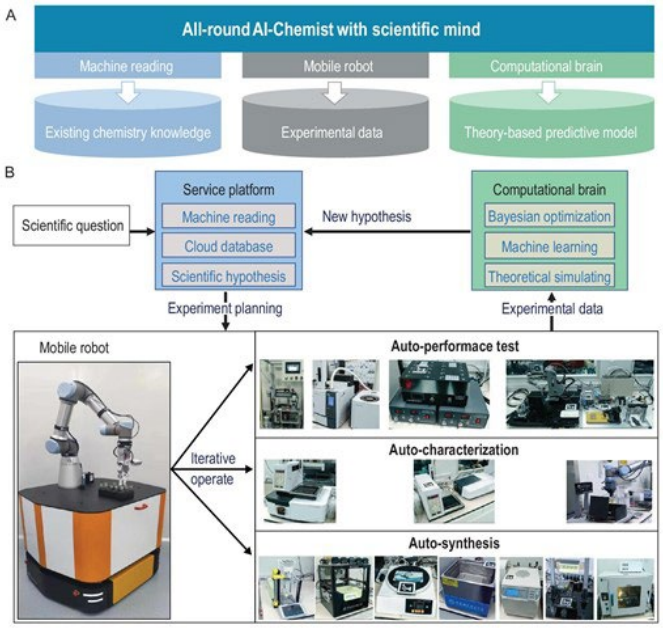
这两个项目代表了各自领域的重大进步，突显了谷歌在推动 AI 和机器学习领域边界方面的持续承诺。PaLM E 专注于整合多样化的模态以实现全面理解和推理，而 Q Transformer 则解决了在复杂、真实世界场景中可扩展和有效强化学习的挑战。

LLM Agent 在 AI4Chemistry 中的应用进展

随着 AI4Science 领域的发展，LLM 的应用前景逐渐收到重视，尤其在有着海量数据库资源得 AI4Chemistry 领域，LLM 结合应用的前景逐渐明朗，近年来也有一批优秀工作出现。

中国科大的江俊等人设计了一台全栈 AI 化学家，包括一个服务平台、一个移动机器人、14 个智能化学工作站平台和一个计算大脑。服务平台利用自然语言处理技术从大量文献中提取化学知识，并根据科学假设生成实验计划。

该移动机器人配备了六自由度机械臂、双激光定位系统和视觉识别系统，能够在不同的工作站之间自主移动和操作。计算机大脑模块利用分子动力学和密度泛函理论进行理论模拟，利用机器学习和贝叶斯优化进行预测模型的建立和优化。作者以非贵金属氧化物电催化剂为例，展示了 AI 化学家能够进行全方位的化学研究，包括文献阅读、假设提出、实验设计、实验执行、数据分析、模型训练和反馈新假设。



ChemCrow 是一个基于 MRKL 理论的，支持多种计算化学工具的 Agent 框架，作者使用 LangChain 框架来实现 ChemCrow，这是一个支持多种模块的语言模型应用开发框架。作者选择了 GPT-4 作为 LLM，并集成了 17 个化学工具，包括网页搜索，文献搜索，分子和反应相关的工具等。作者使用了 Thought， Action， Action Input， Observation（TAIO）格式来指导 LLM 的生成过程，要求 LLM 根据当前任务的状态和目标来进行推理、选择工具、输入参数、观察结果，并迭代直到完成任务。

“GPT-4 for MoF researching“的作者提出了一种使用 ChatGPT 从科学文献的各种文本和表格中挖掘 MOF 的合成条件的新方法。本文还采用 ChatGPT 生成 Python 代码进行数据处理和分析。其开发了一个工作流程,由 ChatGPT 本身编程实现三个不同的文本挖掘过程:

过程 1:摘要。给 ChatGPT 预选的实验部分,要求其总结和制表合成参数到一个具有 11 列的表格中。在这一步中,作者提出了三个约束要求:最小化幻觉、实现详细指令和请求结构化输出。

过程 2:分类。给 ChatGPT 整篇研究论文,要求其将每个段落分类为“Yes”或“No”,根据段落是否包含合成参数。只有标记为“Yes”的段落传递给过程 1 进行摘要。

过程 3:过滤。给 ChatGPT 整篇论文,要求其使用文本嵌入和余弦相似性过滤掉最不相关的部分。只传递相似度高的部分给过程 2 进行分类。

最后作者使用构建的数据集训练了一个基于随机森林的机器学习模型,以预测 MOF 实验结晶的结果(单晶或多晶),并识别 MOF 结晶的重要因素。

Boiko 等人设计了一套名为 Coscientist 的 LLM Agent，一个由 GPT-4 驱动的人工智能系统，通过整合由互联网和文档搜索、代码执行和实验自动化等工具支持的大型语言模型，自主设计、计划和执行复杂的实验。他们结合 OT-2 机器人展示了其加速六种不同任务研究的潜力，包括钯催化交叉偶联的成功反应优化，同时展示了（半）自主实验设计和执行的先进能力。

结论

本文综述了大语言模型智能体技术及其在 AI4Chemistry 中的应用进展，展示了 LLM Agent 在多种复杂任务中的强大能力和潜力。然而，LLM Agent 仍然面临着一些挑战和问题，如数据质量、可解释性、安全性、伦理性等，需要进一步的研究和探索。此外，LLM Agent 也可以与其他领域的技术进行结合和创新，如神经符号系统、知识图谱、强化学习、元学习等，以提高其智能水平和适应性。未来，我们期待 LLM Agent 能够成为人类的智能伙伴，协助人类完成更多的科学探索和创造性工作。

【参考文献】

[1] LAKE B M, BARONI M. Human-like systematic generalization through a meta-learning neural network[J/OL]. Nature, 2023: 1-7. DOI:10.1038/s41586-023-06668-3.

[2] XIAO G, LIN J, SEZNEC M, 等. SmoothQuant: accurate and efficient post-training quantization for large language models[M/OL]. arXiv, 2023[2023-10-26]. http://arxiv.org/abs/2211.10438.

[3] CUI T, TANG C, SU M, 等. GPIP: geometry-enhanced pre-training on interatomic potentials[M/OL]. arXiv, 2023[2023-10-26]. http://arxiv.org/abs/2309.15718.

[4] RAJESWAR S, MAZZAGLIA P, VERBELEN T, 等. Mastering the unsupervised reinforcement learning benchmark from pixels[M/OL]. arXiv, 2023[2023-10-25]. http://arxiv.org/abs/2209.12016.

[5] LI S, XIA Y, XU Z. Simultaneous perturbation stochastic approximation: towards one-measurement per iteration[M/OL]. arXiv, 2022[2023-10-25]. http://arxiv.org/abs/2203.03075.

[6] YANG S, NACHUM O, DU Y, 等. Foundation models for decision making: problems, methods, and opportunities[M/OL]. arXiv, 2023[2023-10-25]. http://arxiv.org/abs/2303.04129.

[7] HOFFMANN J, BORGEAUD S, MENSCH A, 等. Training compute-optimal large language models[M/OL]. arXiv, 2022[2023-10-18]. http://arxiv.org/abs/2203.15556.

[8] BAGAL V, AGGARWAL R, VINOD P K, 等. MolGPT: molecular generation using a transformer-decoder model[J/OL]. Journal of Chemical Information and Modeling, 2022, 62(9): 2064-2076. DOI:10.1021/acs.jcim.1c00600.

[9] MUKHERJEE S, MITRA A, JAWAHAR G, 等. Orca: progressive learning from complex explanation traces of GPT-4[M/OL]. arXiv, 2023[2023-10-17]. http://arxiv.org/abs/2306.02707.

[10] ARIAFAR S, COLL-FONT J, BROOKS D. ADMMBO: bayesian optimization with unknown constraints using ADMM[J].

[11] LI C, GAN Z, YANG Z, 等. Multimodal foundation models: from specialists to general-purpose assistants[M/OL]. arXiv, 2023[2023-10-17]. http://arxiv.org/abs/2309.10020.

[12] RADFORD A, KIM J W, HALLACY C, 等. Learning transferable visual models from natural language supervision[M/OL]. arXiv, 2021[2023-10-17]. http://arxiv.org/abs/2103.00020.

[13] ZHANG Z, FANG M, CHEN L, 等. How do large language models capture the ever-changing world knowledge? A review of recent advances[M/OL]. arXiv, 2023[2023-10-15]. http://arxiv.org/abs/2310.07343.

[14] SONG Y, MIRET S, ZHANG H, 等. HoneyBee: progressive instruction finetuning of large language models for materials science[M/OL]. arXiv, 2023[2023-10-15]. http://arxiv.org/abs/2310.08511.

[15] KARPAS E, ABEND O, BELINKOV Y, 等. MRKL systems: a modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning[M/OL]. arXiv, 2022[2023-09-08]. http://arxiv.org/abs/2205.00445.

[16] AHMAD W, SIMON E, CHITHRANANDA S, 等. ChemBERTa-2: towards chemical foundation models[M/OL]. arXiv, 2022[2023-10-08]. http://arxiv.org/abs/2209.01712.

[17] BRAN A M, COX S, WHITE A D, 等. ChemCrow: augmenting large-language models with chemistry tools[M/OL]. arXiv, 2023[2023-08-09]. http://arxiv.org/abs/2304.05376. DOI:10.48550/arXiv.2304.05376.

[18] An all-in-one multipurpose robotic platform for the self-optimization, intensification and scale-up of photocatalysis in flow | chemical engineering and industrial chemistry | ChemRxiv | cambridge open engage[EB/OL]. [2023-10-15]. https://chemrxiv.org/engage/chemrxiv/article-details/64809b97e64f843f41767eac.

[19] WEN Y, WANG Z, SUN J. MindMap: knowledge graph prompting sparks graph of thoughts in large language models[M/OL]. arXiv, 2023[2023-10-15]. http://arxiv.org/abs/2308.09729.

[20] BEAINI D, HUANG S, CUNHA J A, 等. Towards foundational models for molecular learning on large-scale multi-task datasets[M/OL]. arXiv, 2023[2023-10-15]. http://arxiv.org/abs/2310.04292.

[21] SUI Y, GOTOVOS A, BURDICK J W, 等. Safe exploration for optimization with gaussian processes[J].

[22] MCLEOD M, ROBERTS S, OSBORNE M A. Optimization, fast and slow: optimally switching between local and bayesian optimization[C/OL]//Proceedings of the 35th International Conference on Machine Learning. PMLR, 2018: 3443-3452[2023-10-12]. https://proceedings.mlr.press/v80/mcleod18a.html.

[23] KIRSCHNER J, MUTNÝ M, HILLER N, 等. Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces[J].

[24] TAYLOR R, KARDAS M, CUCURULL G, 等. Galactica: a large language model for science[EB/OL]//arXiv.org. (2022-11-16)[2023-10-09]. https://arxiv.org/abs/2211.09085v1.

[25] GROSSE R, BAE J, ANIL C, 等. Studying large language model generalization with influence functions[J].

[26] Paper page - direct preference optimization: your language model is secretly a reward model[EB/OL]. (2023-10-13)[2023-10-13]. https://huggingface.co/papers/2305.18290.

[27] A deep learning framework for accurate reaction prediction and its application on high-throughput experimentation data | journal of cheminformatics | full

text[EB/OL]. [2023-10-15]. https://jcheminf.biomedcentral.com/articles/10.1186/s13321-023-00732-w.

[28] SUI Y, GOTOVOS A, BURDICK J W, 等. Safe exploration for optimization with gaussian processes[J].