ChemHTS: Hierarchical Tool Stacking for Enhancing Chemical Agents

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demon-001 strated remarkable potential in scientific research, particularly in chemistry-related tasks such as molecular design, reaction prediction, and property estimation. While tool-augmented LLMs have been introduced to enhance reason-007 ing and computation in these domains, existing approaches suffer from tool invocation errors and lack effective collaboration among diverse tools, limiting their overall performance. To address these challenges, we propose ChemHTS (Chemical Hierarchical Tool Stacking), a novel method that optimizes tool invocation pathways through a hierarchical stacking strategy. 015 ChemHTS consists of two key stages: tool selfstacking warmup and multi-layer decision opti-017 mization, enabling LLMs to refine tool usage dynamically. We evaluate ChemHTS across four classical chemistry tasks and demonstrate 019 its superiority over strong baselines, including GPT-40, DeepSeek-R1, and chemistry-specific models, including ChemDFM. Furthermore, we define four distinct tool-stacking behaviors to enhance interpretability, providing insights into the effectiveness of tool collaboration. Our dataset and code will be released soon.

1 Introduction

027

037

041

In recent years, large language models (LLMs) (Touvron et al., 2023; Achiam et al., 2023; DeepSeek-AI et al., 2025) have made groundbreaking advancements in fields such as natural language processing (NLP) and knowledge reasoning, and are gradually being applied to scientific research, particularly demonstrating immense potential in the domain of chemistry (Guo et al., 2023; Ouyang et al., 2024). LLMs leverage their robust capabilities for knowledge integration and reasoning computation to offer novel solutions for tasks such as molecular design (Noutahi et al., 2023), reaction prediction (Shi et al., 2023), and property prediction (Srinivas and Runkana, 2024). This progress



Figure 1: Using the text-based molecule design task as an example, this analysis examines the issues in model tool usage under Name2SMILES and ChemDFM tools.

significantly enhances the level of automation and intelligence in chemical research.

Despite the promising prospects of LLMs in the field of chemistry, current research faces challenges. Existing models exhibit limitations in their ability to generalize across diverse tasks, and they lack access to external knowledge sources, which constrains their practical utility in scientific applications (Zhao et al., 2024; Liao et al., 2024; Han et al., 2024). Consequently, studies have integrated expert-designed tools to enhance the performance of LLMs in chemical tasks (Bran et al., 2023; Boiko et al., 2024).

However, chemical tool-augmented LLMs still exhibit numerous challenges when confronted with complex and diverse toolsets (Bran et al., 2023). Firstly, due to the limited capabilities of tools or the susceptibility of LLMs to tool calling errors, relying on a single tool invocation may result in biased outcomes (Ye et al., 2024). As illustrated in



Figure 2: Our ChemHTS method framework diagram. For each chemical task, we identify the optimal tool-stacking pathway through the ChemHTS method for subsequent task inference execution.

Fig. 1, if the result of invoking an agent tool once is biased, the model is unable to self-correct and may produce incorrect answers. Secondly, LLMs struggle to collaborate across different types of tools, limiting their capacity to maximize information gain (Cheng and Amiri, 2025). During reasoning, they often depend on a single category of tools, overlooking the synergistic benefits of integrating multiple tool types. As a result, errors can propagate through the reasoning chain, ultimately impairing overall task performance.

Despite the emergence of numerous studies in recent years focused on optimizing tool usage, these efforts have not yet adequately addressed the aforementioned issues. Current research primarily concentrates on optimal tool selection for individual tasks, while overlooking the collaborative interactions between tools and their impact on task inference capabilities. Some studies have attempted to leverage the context learning capabilities of LLMs by employing strategic prompting to optimize tool selection. For instance, frameworks such as Chain of Thought (CoT) (Wei et al., 2023) and ReACT (Yao et al., 2023; Yang et al., 2023) aim to guide model in effectively utilizing tools during the reasoning process. Another category of methods involves fine-tuning LLMs on tool calling datasets to enhance their tool usage capabilities. For instance, approaches such as ToolBench (Qin et al., 2023) utilizes supervised fine-tuning (Shen, 2024) to improve the model's understanding and calling of tools. However, these approaches can only enhance the selection of individual tools to a limited extent and have not succeeded in achieving collaborative optimization among multiple tools.

To address the aforementioned issues, we propose a novel tool calling optimization method termed **ChemHTS** (**Chem**ical **H**ierarchical **T**ool Stacking), which aims to enhance the tool invocation process of LLMs in chemistry-related tasks. The core idea of C he m is to explore the optimal tool invocation pathways through a hierarchical stacking strategy under the constraints of a given chemical task and multiple tools, thereby improving the reasoning and computational capabilities of LLMs in chemical applications. This method encompasses two key stages: tool self-stacking and tool multi-layer decision optimization. 100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

In addition, we apply the ChemHTS method to four classic chemistry tasks: text-based molecular design, molecular description, molecular property prediction, and reaction prediction. We systematically explore the optimal tool invocation pathways for each task and conduct extensive experiments on the test sets. The results demonstrate that the tool invocation paths optimized by ChemHTS outperform several baseline models, such as GPT40 (OpenAI et al., 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025), as well as chemistry-specific models, including UniMol (Ji et al., 2024) and ChemDFM (Zhao et al., 2024), across all tasks. Furthermore, we define four distinct tool stacking behavioral patterns-Correct, Modify, Judge, and Reserve-and provide case studies to elucidate the reasons behind the improvements in task performance due to tool stacking. To summarize, our contributions are mainly three-fold:

- We are the first to highlight the advantages of agent tool collaboration and propose the ChemHTS method to optimize the optimal tool stacking path for agents across various chemical tasks, addressing the issue of low efficiency in the utilization of agent tools.
- Through extensive experiments, we have demonstrated that the optimal tool stacking

paths derived from our method outperform numerous baseline models and chemistryspecific models across four classical chemical tasks, thereby validating the effectiveness and generalizability of our approach.

• We define four behavioural patterns of models during the tool stacking process, to conduct an in-depth analysis and interpretation of the reasons behind the performance improvements in tool stacking, thereby enhancing the interpretability of the experiments.

2 Related Work

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

160

161

162

163

165

166

167

169

170

171

172

2.1 Large Language Models for Chemistry

Large language models (LLMs) have demonstrated significant potential in chemistry, with applications spanning molecular generation, property prediction, reaction modeling, and retrosynthetic analysis (Fang et al., 2024; Tang et al., 2024; Liao et al., 2024). For instance, the ChemDFM (Zhao et al., 2024) pretrained on chemical literature and textbooks and further refined through extensive instruction tuning, has exhibited enhanced performance across various chemical tasks. Similarly, ChemCrow (Bran et al., 2023), an LLM-powered chemistry assistant, integrates multiple expertdesigned chemical tools to improve LLM performance in chemistry-related applications.Despite these advancements, LLMs continue to face challenges in handling complex chemical computations and generalizing across diverse chemical problems (Ouyang et al., 2024; Han et al., 2024). Moreover, they remain inefficient in utilizing existing computational chemistry tools (Shi et al., 2023), and struggle to navigate the combinatorial and hierarchical relationships between these tools.

2.2 Tool-augmented LLMs

LLMs (Anil et al., 2023; Achiam et al., 2023; Tou-173 vron et al., 2023) have demonstrated strong reason-174 ing capabilities in natural language processing and 175 scientific computing. However, they face limita-176 tions in specialized tasks in fields such as chem-177 istry and physics (Yang et al., 2024), including 178 constrained computational accuracy, insufficient 179 numerical reasoning abilities, and a lack of col-181 laboration with external tools. To address these shortcomings, researchers have recently proposed 182 the tool-augmented LLMs approach (Qin et al., 2023; Wang et al., 2024; Yang et al., 2023), enabling LLMs to dynamically call external tools 185

and thereby enhance their task execution capabilities. Representative methods include ReAct (Yao et al., 2023), which combines chain-of-thought reasoning (CoT) (Wei et al., 2023) with tool invocation to allow LLMs to dynamically acquire external information during decision-making, and Toolformer (Schick et al., 2023), which enables LLMs to autonomously decide when to call tools, improving the accuracy of computational tasks. Despite these advancements, existing research primarily focuses on single-tool invocation and has yet to explore hierarchical combinations of tools. A single tool is often insufficient to solve complex scientific problems, whereas the collaborative invocation of multiple tools holds promise for enhancing the reasoning capabilities of LLMs in chemical tasks.

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

226

227

228

229

230

231

232

233

234

3 ChemHTS

Our study proposes a hierarchical tool stacking method, named ChemHTS, to optimize the tool invocation process of LLMs in chemistry-related tasks. ChemHTS iteratively refines and optimizes the tool stacking trajectory, ultimately identifying the optimal sequence of tool calls to enhance computational performance (Fig.2).

3.1 Step 1: Tool Self-stacking Warmup

To evaluate the performance of a single tool after multiple invocations and determine its suitability for stacking, we first conduct a self-stacking warmup phase for the individual tool. Since research in chemical tasks often integrates theoretical computations with experimental knowledge, leveraging computation to predict unknown properties or processes and retrieval to access existing knowledge, we categorize the chemical LLM toolset \mathcal{L} into two types: compute-based tools \mathcal{L}_c and retriever-based tools \mathcal{L}_r . For computational tools, it may be necessary to test whether repeated invocations lead to improved computational accuracy or stability. In contrast, for retriever-based tools, it may be important to evaluate whether multiple retrievals can enhance knowledge coverage.

For a given chemical task \mathcal{T} and its corresponding tool library \mathcal{L} , the performance of an individual tool $(ToolA_i)$ and its self-stacking $(A_i \rightarrow A_i)$ is evaluated. If repeated invocations of the same tool type result in performance degradation, further stacking of that tool is discontinued. From each category, the top-k most effective tool combinations are selected for the next stage of stacking,

as outlined in Algorithm 1.. These refined toolstacking pathways are encapsulated as new tools, reducing the subsequent search space, improving computational efficiency, and providing stronger tool combination strategies for the next phase.

3.2 Step 2: Multi-Layer Optimization

Algorithm 1 ChemHTS Algorithm

```
Require: Training dataset \mathcal{D}, tool library \mathcal{L} (including
    compute-based tools \mathcal{L}_c and retriever-based tools \mathcal{L}_r),
    maximum hierarchy depth n, number of selected paths
    per category k, maximum tool selection per step m.
```

Ensure: Optimized tool invocation strategy \mathcal{P} 1: STEP 1: SELF-STACKING WARMUP

- 2: for each tool $T_i \in \mathcal{L}_c \cup \mathcal{L}_r$ do
- 3: Evaluate self-stacking paths of T_i on \mathcal{D} .
- 4: end for

236

- 5: Select the Top-k best-performing paths for computational tools \mathcal{P}_c and retriever tools \mathcal{P}_r .
- 6: Encapsulate selected tool stacks as new tools in \mathcal{L} .
- 7: STEP 2: MULTI-LAYER OPTIMAZATION
- 8: Initialize hierarchy level $L \leftarrow 1$.
- 9: while $L \leq n$ do
- 10: Sort computational tools \mathcal{L}_c and RAG tools \mathcal{L}_r based on performance metrics.
- 11: Select Top-1 computational tool T_c^* and Top-k RAG tools \mathcal{P}_r
- 12: Select Top-1 RAG tool T_r^* and Top-k computational tools \mathcal{P}_c .
- 13: for each tool subset $\mathcal{L}_L \subseteq (\mathcal{P}_c \cup \mathcal{P}_r), |\mathcal{L}_L| \leq m$ do 14: Evaluate the performance of the tool invocation path.
- 15: if current best performance \leq previous best performance then

16:	STOP	stacking	process
-----	------	----------	---------

- 17: BREAK
- 18: end if
- 19: end for
- 20: Encapsulate best-performing paths as new tools.
- 21: $L \leftarrow L + 1.$
- 22: end while
- 23: Select the best tool invocation path \mathcal{P}^* as the final strategy.

24: return \mathcal{P}^* .

241 To explore the optimal invocation pathways across tool combinations and progressively encap-242 sulate the best tools, ultimately determining the 243 globally optimal tool invocation strategy, we divide 244 the tool-stacking process into multiple layers and 245 perform stacking optimization step by step. In the 246 first layer, the Top-1 computational tool selected 247 in Stage 1 is combined with the Top-k retrieval tools, and the Top-1 retrieval tool is combined with the Top-k computational tools. Each agent is re-251 stricted to selecting the best tool from a pool of mcandidate tools in each step. If the optimal performance in this layer is worse than that of Stage 1, the stacking process is terminated; otherwise, the iteration continues. In subsequent layers, all tools 255

are re-ranked, and the top k optimal pathways in each category are selected and encapsulated as new tools. This process continues until either the maximum number of layers n is reached or no further optimization space remains. Finally, the optimal tool invocation pathway is selected as the general tool invocation strategy for the given task.

256

257

258

259

260

261

262

265

266

267

269

270

271

272

273

274

275

276

277

278

279

281

282

283

284

285

287

288

290

291

292

293

294

295

296

299

300

301

302

303

304

4 **Experiments**

Experiment Setup 4.1

Dataset We evaluate the performance of ChemHTS in the field of chemistry using Chem-LLMBench. ChemLLMBench (Guo et al., 2023) comprises a series of chemistry-related tasks that cover a wide range of chemical topics. In this study, we focus on four representative tasks and select 100 evaluation instances for each task, consistent with the evaluation experiments in ChemDFM, as the test set. Since the evaluation dataset for each task in ChemLLMBench contains only 100 instances, we adopt a similar approach to that in (Guo et al., 2023) to select the training set. For the Text-Based Molecule Design and Molecule Captioning tasks, we randomly sample 100 instances from the ChEBI-20-MM (Liu et al., 2025) dataset, excluding the corresponding test set, as the training set. For the Reaction Prediction task, we randomly sample 100 instances from the USPTO-MIT (Jin et al., 2017) dataset, excluding the corresponding test set, as the training set. For the Molecular Property Prediction task, we randomly sample 50 instances for each dataset from the BBBP, HIV, BACE, Tox21, and ClinTox (Wu et al., 2018) datasets, excluding the corresponding test sets, as the training set. The details of our dataset are shown in Tab. 1.

Models We categorize current models into three primary groups: task-specific specialist models, LLM-based generalist models, and tool-based agent models. Task-specific specialist models refer to non-LLM models designed for specific tasks. Tool-based agent models leverage GPT-40 (OpenAI et al., 2024) as the core agent, augmented with additional tools tailored to the specific task. To ensure fair comparisons, we use the same test set for evaluating different models on each task.

Tools To facilitate the experimental process, for each task, we provide only the most relevant tools for the two categories: computational tools and retrieval tools. Details can be found in Tab. 1.

Ability	Task	Task Type	Dataset	#train	#test	Retrieval Tool	Compute Tool
Understanding	Molecular Property Prediction	Classification	BBBP, HIV, BACE, Tox21, ClinTox	250	100	SMILES2Property	UniMol-v2
Reasoning	Reaction Prediction	Generation	USPTO-MIT	100	100	SMILES2Property	Chemformer
	Text-Based Molecule Design	Generation	ChEBI-20-MM	100	100	Name2SMILES	ChemDFM
Explaining	Molecule Captioning	Generation	ChEBI-20-MM	100	100	SMILES2Description	Text+ChemT5

Table 1: Details of the training and test sets for the four chemistry tasks.

4.2 Results

305

306

307

308

309

311

312

313

315

319

321

322

323

324

331

335

4.2.1 Text-based Molecule Design

In the text-based molecule design task, LLMs predict a molecule's SMILES (Simplified Molecular Input Line Entry System) representation based on a given description, testing their ability to interpret and translate chemical language into valid molecular structures (Zhao et al., 2024). Our study employs two sets of metrics to evaluate the performance of the task. The first set of metrics measures the text-based similarity between the predicted SMILES and the gold standard SMILES, including exact match, BLEU, and Levenshtein distance (Haldar and Mukhopadhyay, 2011). The second set of metrics assesses the chemical similarity between the predicted molecules and the reference molecules, encompassing the validity of the predicted SMILES and the FTS (Fingerprint Tanimoto Similarity) (Tanimoto, 1958), calculated based on MACCS, RDK, and Morgan (Morgan, 1965).

Model	Exact↑	BLEU↑	Dis↑	Validity↓	MACCS↑	RDK↑	Morgan↑
Tasi	k-specific	specialis	st mode	ls			
MolXPT (Liu et al., 2023)	0.22	· -	-	0.98	0.86	0.76	0.67
Text+Chem T5 (Christofidellis et al., 2023)	0.32	0.85	16.87	0.94	0.90	0.82	0.75
Mol-Instruction (Fang et al., 2024)	0.02	0.35	41.40	1.00	0.41	0.23	0.15
LLM-based generalist models							
GPT-4o (OpenAI et al., 2024)	0.01	0.57	52.85	0.91	0.71	0.54	0.38
Deepseek-R1 (DeepSeek-AI et al., 2025)	0.02	0.56	92.29	0.57	0.48	0.38	0.31
Llama3-70b (AI@Meta, 2024)	0.03	0.57	46.63	0.78	0.57	0.40	0.30
Llama3-8b (AI@Meta, 2024)	0.01	0.41	155.17	0.45	0.27	0.15	0.11
ChemDFM-13B (Zhao et al., 2024)	0.32	0.85	<u>11.58</u>	0.94	0.81	0.73	0.67
7	ool-base	d Agent n	nodels				
Agent (1-tool, Name2SMILES)	0.25	0.70	84.81	0.72	0.67	0.61	0.56
Agent (1-tool, ChemDFM)	0.35	0.86	12.66	0.97	0.88	0.79	0.73
Agent (2-tool)	0.34	0.87	12.63	0.94	0.85	0.80	0.74
Ours (Stacking Agent)	0.38	0.93	8.68	0.96	0.92	0.87	0.80

Table 2: Benchmark results of different models in textbased molecule design tasks. All LLM-based generalist models are evaluated on 0-shot.

From the results in Tab. 2, our Stacking Agent outperforms other models in this task. Not only do the generated molecular structures meet the target requirements, but they also maintain high molecular validity. In contrast, models based on large language models perform poorly, particularly in terms of Exact and BLEU scores, suggesting difficulty in accurately generating target molecules in a 0-shot setting. This highlights the challenge of capturing molecular structural information without external tool support. The chemistry-specific ChemDFM performs well in the 0-shot setting, surpassing other task-specific models. However, our Stacking Agent, by combining ChemDFM with the Name2SMILES tool, achieves the best results across multiple key metrics, with a BLEU score of 0.93, outperforming ChemDFM's 0.85.

4.2.2 Molecule captioning

To evaluate the ability of the model to translate complex chemical information into human-readable language descriptions, we introduce the Molecule Captioning task (Guo et al., 2023). This task requires LLMs not only to accurately recognize the molecule represented by a given SMILES string, but also to generate a concise natural language description of the molecule. To assess the model's performance on this task, we employ traditional natural language processing evaluation metrics, such as BLEU and ROUGE, to measure the similarity between the molecule descriptions generated by the model and the reference descriptions in the test set.

Model	BLEU-2↑	BLEU-4↑	ROUGE-1↑	ROUGE-2↑	ROUGE-L↑		
Task-specific specialist models							
Text+Chem T5 (Christofidellis et al., 2023)	0.63	0.54	0.68	0.54	0.62		
MolXPT (Liu et al., 2023)	0.59	0.50	0.66	0.51	0.60		
InstructMol (Cao et al., 2024)	0.48	0.37	0.57	0.39	0.50		
Mol-Instruction (Fang et al., 2024)	0.25	0.17	0.33	0.29	0.27		
LLM-based generalist models							
GPT-40 (OpenAI et al., 2024)	0.26	0.17	0.10	0.00	0.30		
Deepseek-R1 (DeepSeek-AI et al., 2025)	0.40	0.25	0.10	0.02	0.21		
Llama3-70b (AI@Meta, 2024)	0.11	0.07	0.06	0.00	0.12		
Llama3-8b (AI@Meta, 2024)	0.04	0.03	0.05	0.00	0.09		
ChemDFM-13b (Zhao et al., 2024)	0.32	0.27	0.49	0.37	0.48		
To	ol-based A	gent models					
Agent (1-tool, SMILES2Description)	0.59	0.52	0.43	0.29	0.51		
Agent (1-tool, Text+ChemT5)	0.60	0.49	0.39	0.23	0.50		
Agent (2-tool)	0.64	0.56	0.45	0.29	0.55		
Ours (Stacking Agent)	0.73	0.69	0.70	0.58	0.76		

Table 3: Benchmark results of different models in molecule captioning tasks. All LLM-based generalist models are evaluated on 0-shot.

As shown in Tab. 3, Stacking Agent, built using the more efficient ChemHTS method, achieves the best overall performance across all metrics. For text generation tasks, task-specific models still hold a significant advantage in molecule captioning, while , aside from large parameters models like GPT-40 and Deepseek-R1, other large language models still perform poorly. Additionally, a com-

361

362

363

364

365

357

336

337

338

340

341

342

343

344

345

346

347

349

351

352

353

354

355

parison between Agent (1-tool) and Agent (2-tool) shows that increasing the number of tools boosts 367 performance (e.g., BLEU-4 improves from 0.54 to 0.56). However, simply adding more tools doesn't lead to major gains.

4.2.3 Molecular Property Prediction

371

372

373

374

375

378

379

391

400

401

402

403

Molecular property prediction (Guo et al., 2021; Wang et al., 2021) is a fundamental task in computational chemistry that has garnered significant attention in recent years due to its potential applications in drug discovery, materials science, and other areas of chemistry. The task requires models to predict the chemical and physical properties of a given molecule based on its structure. The dataset consists of five tasks drawn from MoleculeNet (Wu et al., 2018), including BACE, BBBP, HIV, ClinTox, and Tox21. Among them, BACE and BBBP each consist of a balanced binary classi-384 fication task. HIV includes an imbalanced binary classification task. ClinTox and Tox21 contain two and twenty-one imbalanced binary classification tasks, respectively. To address the severe label imbalance in these tasks, our study employs the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) as the primary evaluation metric (Tafvizi et al., 2022). However, considering the computational challenges associated with large language models and the balanced binary task, we also incorporate Accuracy as an additional metric 395 to provide a more comprehensive assessment.

Model	BACE		BBBP		ClinTox		HIV		Tox21		Avg	
	ACC↑	AUC↓	ACC↑	AUC↓	ACC↑	AUC↓	ACC↓	AUC↑	ACC↑	AUC↑	ACC↑	AUC.
		Task-	specific	special	ist mod	els						
Uni-Mol-v2 (Ji et al., 2024)	0.75	88.9	0.58	82.6	0.51	85.3	0.96	90.7	0.92	80.0	0.74	85.5
MolXPT (Liu et al., 2023)	-	88.4	-	80.0	-	95.3	-	78.1	-	77.1	-	83.8
InstructMol (Cao et al., 2024)	-	85.9	-	64.0	-	-	-	74.0	-	-	-	-
LLM-based generalist models												
GPT-4o (OpenAI et al., 2024)	0.38	38.5	0.56	57.0	0.51	51.8	0.59	54.7	0.37	36.0	0.48	47.6
Deepseek-R1 (DeepSeek-AI et al., 2025)	0.62	52.7	0.61	63.6	0.48	48.2	0.51	50.5	0.75	60.1	0.59	55.0
Llama3-70B (AI@Meta, 2024)	0.55	50.9	0.59	60.1	0.48	48.8	0.20	58.3	0.59	44.9	0.48	52.6
Llama3-8B (AI@Meta, 2024)	0.50	43.8	0.54	51.4	0.49	49.5	0.05	50.5	0.43	56.1	0.40	50.2
ChemDFM-13B (Zhao et al., 2024)	0.66	78.4	0.57	66.7	0.49	89.9	0.94	73.6	0.83	79.8	0.70	77.7
		To	ol-based	Agent	models							
Agent (1-tool, SMILES2Property)	0.56	55.6	0.58	60.0	0.47	47.8	0.94	60.9	0.91	70.4	0.69	59.0
Agent (1-tool, UniMol-v2)	0.75	78.6	0.54	50.0	0.49	50.0	0.94	96.8	0.93	89.6	0.72	72.4
Agent (2-tool)	0.75	74.2	0.59	58.7	0.49	50.0	0.92	95.8	0.94	91.7	0.74	74.1
Ours (Stacking Agent)	0.79	81.4	0.68	71.1	0.67	72.3	0.95	97.4	0.96	97.8	0.81	84.0

Table 4: Benchmark results of different models in molecular property prediction tasks. All LLM-based generalist models are evaluated on 0-shot.

From the results in Tab. 4, task-specific specialist models demonstrate the best overall performance. Among the models, UniMol-v2 achieves the highest average AUC-ROC and Accuracy values across the five tasks, outperforming the other models. These findings indicate that models specifically optimized for molecular property prediction tasks are more effective at learning the relationships between chemical molecular structures and their biological activities. Moreover, the Stacking Agent structure identified through ChemHTS outperformed LLM-based generalist models on this 407 task, especially in the imbalanced binary classifi-408 cation tasks of HIV and Tox21. This suggests that 409 tool integration methods can, to some extent, compensate for the limitations of generalist models and enhance the accuracy of prediction tasks. Finally, while our ChemHTS method falls 1.5 AUC score behind the state-of-the-art UniMol-v2, it surpasses 414 its average accuracy (0.74) with a score of 0.81. 415

404

405

406

410

411

412

413

416

417

418

419

420

421

422

423

494

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

4.2.4 Reaction Prediction

Reaction prediction is a core task in the field of chemistry, with significant importance for drug discovery, materials science, and the development of novel synthetic pathways. Given a set of reactants, the goal of this task is to predict the most likely products formed during the chemical reaction (Guo et al., 2024; Schwaller et al., 2019). Similarly to the results of the text-based molecule design task, we used the same metrics to measure the task performance.

Model	Exact↓	BLEU↑	Dis↑	Validity↑	MACCS↑	RDK↑	Morgan↑
Tasi	k-specifi	c speciali	st mode	ls			
Chemformer (Irwin et al.)	0.91	96.1	1.26	1.00	0.97	0.97	0.96
Text+ChemT5 (Christofidellis et al., 2023)	0.83	96.0	7.42	0.98	0.96	0.96	0.94
InstructMol (Cao et al., 2024)	0.54	96.7	10.85	1.00	0.88	0.78	0.74
Mol-Instruction (Fang et al., 2024)	0.05	65.4	27.26	1.00	0.51	0.31	0.26
LLM-based generalist models							
GPT-40 (OpenAI et al., 2024)	0.01	65.8	27.24	0.81	0.54	0.39	0.33
Deepseek-R1 (DeepSeek-AI et al., 2025)	0.10	76.2	16.04	0.75	0.60	0.53	0.48
Llama3-70b (AI@Meta, 2024)	0.00	55.2	282.46	0.85	0.48	0.35	0.31
Llama3-8B (AI@Meta, 2024)	0.00	37.6	148.15	0.41	0.18	0.14	0.11
ChemDFM-13B (Zhao et al., 2024)	0.39	80.6	10.38	0.96	0.77	0.69	0.65
7	ool-base	ed Agent i	nodels				
Agent (1-tool, SMILES2Property)	0.05	43.6	33.17	0.83	0.40	0.29	0.27
Agent (1-tool, Chemformer)	0.89	96.4	2.44	1.00	0.97	0.97	0.95
Agent (2-tool)	0.87	97.1	1.6	1.00	0.97	0.97	0.95
Ours (Stacking Agent)	0.90	98.4	0.97	1.00	0.98	0.98	0.96

Table 5: Benchmark results of different models in reaction prediction tasks. All LLM-based generalist models are evaluated on 0-shot.

As shown in the Tab. 5, It can be observed that the Chemformer model performs exceptionally well in this task, achieving a product prediction accuracy of 0.91. It also outperforms other task-specific models across all metrics. In contrast, LLMs face significant challenges, with Deepseek-R1, despite its deep reasoning capabilities, only achieving 0.10 accuracy in product prediction. Similarly, the chemistry-specific ChemDFM struggles under 0-shot conditions. The ChemHTS model excels across all metrics, except for a slightly lower exact score (0.01), surpassing the Chemformer model on all other measures. For Agent (1-tool), relying solely on the SMILES2Property tool leads to

poor performance. However, leveraging the ad-441 ditional information provided by the RAG tool 442 through integration, the agent also achieves sig-443 nificantly better performance. 444

Analysis 5

445

446 447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472 473

474

475

476

477

5.1 Does it improve performance if the agent can choose from more tools?

To further analyze the impact of the number of tools on the performance of chemical tasks, this study investigates the effect of varying tool numbers on the text-based molecule design task. As shown in Tab. 6, under a fixed amount of training data, increasing the number of tools from 2 to 4 results in only minor changes in BLEU-2 scores. Regarding the average BLEU-2 score, both Tool Number = 2and Tool Number = 4 achieve a score of 0.86, while Tool Number = 3 slightly underperforms with a score of 0.85. This indicates that increasing the number of tools has limited benefits for BLEU-2 score improvement, which may be attributed to redundancy in contributions among tools and the fact that the quality of individual tools determines their impact on the final performance.

Tool Number		Train Data Number									
	5			10		20		30			
	Layer	BLEU-2	2 Layer	BLEU-2	Layer	BLEU-2	2 Layer	BLEU-2			
2	0.6	0.79	2.7	0.89	3.0	0.89	3.2	0.87	0.86		
3	0.4	0.81	2.5	0.87	2.8	0.86	3.0	0.85	0.85		
4	0.6	0.80	2.8	0.88	3.0	0.87	3.2	0.86	0.86		

Table 6: Comparison of the performance of different tools (Tool Layerber) on the text-based molecule design task under different training data sizes (Train Data Layerber). And Layer represents the number of layers of stacking agents.

5.2 Does more training data lead to better performance?

Our study also conducts an in-depth analysis of the performance differences in the text-based molecule design task among the optimal stacking structures identified by the ChemHTS method under varying amounts of training data. As shown in Tab. 6, BLEU-2 scores generally exhibit an upward trend with the increase in training data (from 5 to 30). For example, for Tool Number = 2, the BLEU-2score is 0.79 when the training data is 5, and it improves to 0.87 when the training data increases to 30. Similar trends are observed for other tool numbers, indicating that increasing the amount of

training data enhances the model's translation quality. This trend aligns with expectations; however, more training data does not always guarantee better performance. Therefore, further experimental anal-481 ysis is required to determine the optimal amount of training data for different tasks and select the best combination.

478

479

480

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

Layer	1	2	3	4	7	8	10	10+	AVG
BLEU-2	0.88	0.90	0.92	0.93	0.90	0.92	0.91	0.90	0.91

Table 7: Comparison of different tool stacking levels on the performance of text-based molecule design tasks.

5.3 Does increasing the number of stacking layers improve performance?

This section uses the text-based molecule design task as a representative to explore whether increasing the number of stacking layers can improve performance. As shown in the results from Tab. 7, there appears to be an upper limit to the performance improvement achieved by increasing the number of stacking layers. When the number of layers is small (e.g., 1 or 2 layers), the model may fail to sufficiently capture complex features, resulting in lower BLEU-2 scores. As the number of layers increases to 7, the model's learning capacity improves, and the generation quality reaches its peak. However, further increasing the number of layers does not lead to continued performance improvement. This could be attributed to the introduction of redundant information with excessive layers, causing a decline in text generation quality or overfitting to the training data.

5.4 Why tool stacking works?

This chapter explores the behavioral patterns exhibited by agents during the process of tool stacking through case analysis and examines how tool stacking enhances task performance. From the case studies, we identify four distinct stacking behavior patterns—correct, modify, judge and reserve, with detailed information on all cases provided in Appendix. D. Due to space limitations, this paper illustrates only one correct stacking behavior pattern as shown in Tab. 8. In the textbased molecular design task, two callable tools are provided to the model. The experiment first invokes the ChemDFM tool for SMILES prediction, which returns "C1(C)CC1". However, this result does not conform to the standard SMILES repre-

Stacking Pattern	Input	Output
Correct	The molecule is a cycloalkane composed of a three-carbon ring. It functions as an inhalation anesthetic. It is a cycloalkane and a member of the cyclopropane class. What is its predicted SMILES? Gold Answer: C1CC1	According to the provided description, the molecule is cyclopropane, which is a cycloalkane formed by three carbon atoms in a ring, well known for its use as an inhalation anesthetic. To confirm the SMILES notation of cyclopropane, I used the ChemDFM tool. Action: ChemDFM Input: What is the SMILES notation for cyclopropane (a three-carbon cycloalkane used as an inhalation anesthetic)? Output: CC1(C)CC1 × Observation: The returned "CC1(C)CC1" does not correspond to cyclopropane. Cyclopropane should be a simple three-carbon ring, typically represented as "C1CC1". Next, I will call the Name2SMILES tool to further correct the result. Action: Name2SMILES Input: Cyclopropane Output: C1CC1 Observation: Based on the known structure of cyclopropane, the result from this tool is correct. Therefore, I will provide the correct SMILES based on the understanding of the chemical structure. Final Answer: C1CC1 ✓

Table 8: Case analysis of "Correct" behavior patterns of agents during tools stacking.

sentation of cyclopropane. The error likely arises 521 from ChemDFM's misinterpretation of cyclic struc-522 tures, as it fails to correctly identify the single 523 524 three-membered ring and instead erroneously introduces an additional branch. To address this issue, the experiment subsequently invokes the Name2SMILES tool with "Cyclopropane" as the 527 query. This tool correctly returns "C1CC1", which 528 529 is confirmed through subsequent analysis to align with the known standard structure of cyclopropane. 530 Consequently, the final SMILES output is corrected 531 to "C1CC1", meeting the requirements of the stan-532 dard answer. This experiment shows that individual 533 tools may have certain limitations when handling chemical structure prediction tasks, as evidenced 535 by ChemDFM's erroneous prediction. However, 536 by employing the tool stacking strategy, the experiment effectively leverages the complementary strengths of multiple tools, significantly improving the accuracy of the final prediction. 540

5.5 Comparison with LLM-based Multi-Agent Systems

541

542

543

544

545

546

547

548

549

550

552

554

556

560

LLM-based Multi-Agent Systems (MAS) and Toolaugmented LLM share similarities in areas such as task decomposition, tool invocation, and information sharing. Therefore, our study focuses on comparing the performance of six multi-agent systems with different communication structures against the optimal stacking agent path proposed by our ChemHTS method in the text-based molecule design task. For details on the specific multi-agent systems, please refer to Appendix. B, C. As shown in Fig. 3, the BLEU-2 scores of various communication structures improve as the scale of agents increases. However, the growth trends differ significantly. When the agent scale becomes larger, the performance of different structures begins to diverge. Among them, the Full-Connected and Layered structures demonstrate significantly higher quality values compared to other structures. Nev-



Figure 3: Performance comparison of 6 multi-agent systems with different communication structures and our optimal stacking agent path on the text-based molecule design task.

ertheless, the performance of the optimal stacking agent path surpasses the upper performance limit of the multi-agent systems. This is because the optimal path can more effectively leverage task decomposition and tool invocation capabilities while avoiding potential communication overhead and coordination bottlenecks inherent in multi-agent systems. More specific experimental results of multi-agent can be found in Appendix. C. 561

562

563

564

565

566

567

569

570

571

572

573

574

575

577

578

579

580

581

582

583

584

585

6 Conclusion

Our study proposes **ChemHTS**, a hierarchical tool stacking method to enhance tool utilization in LLMdriven chemistry tasks. By addressing key challenges such as bias from single tool calls and lack of tool collaboration, ChemHTS improves reasoning and computational efficiency. Experiments across four chemistry tasks show that ChemHTS outperforms both general LLMs and chemistryspecific models. Additionally, we identify four tool stacking behavioral patterns—Correct, Modify, Judge, and Reserve—which enhance interpretability. In summary, our work highlights the importance of tool collaboration in scientific AI, providing a systematic framework for optimizing LLMagent interactions in chemistry.

Limitations

586

610

612

613

614

615

616

619

621

630

631

632

636

Despite the promising results of ChemHTS in opti-587 mizing tool invocation pathways for chemical tasks, 588 several limitations remain. First, the method relies 589 on predefined toolsets, which may not generalize well to novel or underrepresented chemistry tasks. Expanding the adaptability of ChemHTS to dynamically incorporate new tools remains an open challenge. Second, the hierarchical stacking strat-594 egy assumes that optimal tool combinations can be 595 effectively learned from limited training data, yet real-world chemistry problems often require extensive domain expertise, which LLMs may struggle to acquire solely through tool interactions. Finally, 599 while ChemHTS improves tool collaboration, its computational cost increases with the number of 601 tool-stacking layers, potentially limiting its scalability in large-scale applications. Future work should explore more efficient optimization strategies and adaptive learning mechanisms to enhance both generalizability and efficiency. 606

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- AI@Meta. 2024. Llama 3 model card.
 - Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
 - Daniil A. Boiko, Robert MacKnight, Ben Kline, Gabe Gomes, et al. 2023. Autonomous chemical research with large language models. *Nature*, 624:570–578.
 - Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2023.
 Chemcrow: Augmenting large-language models with chemistry tools. *Preprint*, arXiv:2304.05376.
 - He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. 2024. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *Preprint*, arXiv:2311.16208.
- Jiali Cheng and Hadi Amiri. 2025. Tool unlearning for tool-augmented llms. *Preprint*, arXiv:2502.01083.
- Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. 2023. Unifying molecular and textual representations via multi-task language modelling. In *Proceedings of the 40th International Conference on Machine*

Learning, volume 202 of *Proceedings of Machine Learning Research*, pages 6140–6157. PMLR. 637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

690

691

692

693

694

695

696

697

- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Preprint, arXiv:2501.12948.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.
- Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2024. Mol-instructions: A large-scale

752

- biomolecular instruction dataset for large languagemodels. *Preprint*, arXiv:2306.08018.
 - Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Preprint*, arXiv:2305.18365.
 - Taicheng Guo, Changsheng Ma, Xiuying Chen, Bozhao Nan, Kehan Guo, Shichao Pei, Lu Yu, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024.
 Modeling non-uniform uncertainty in reaction prediction via boosting and dropout.

710

712

714

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

735

736

737

739

740

741

742

743

744

745

746

747

748

- Zhichun Guo, Chuxu Zhang, Wenhao Yu, John Herr, Olaf Wiest, Meng Jiang, and Nitesh V. Chawla. 2021.
 Few-shot graph learning for molecular property prediction. In *Proceedings of the Web Conference 2021*, WWW '21. ACM.
- Rishin Haldar and Debajyoti Mukhopadhyay. 2011. Levenshtein distance technique in dictionary lookup methods: An improved approach. *Preprint*, arXiv:1101.1232.
- Yang Han, Ziping Wan, Lu Chen, Kai Yu, and Xin Chen. 2024. From generalist to specialist: A survey of large language models for chemistry. *Preprint*, arXiv:2412.19994.
- Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pretrained transformer for computational chemistry. 3(1):015022. Publisher: IOP Publishing.
- Xiaohong Ji, Zhen Wang, Zhifeng Gao, Hang Zheng, Linfeng Zhang, Guolin Ke, and Weinan E. 2024. Uni-mol2: Exploring molecular pretraining model at scale. *Preprint*, arXiv:2406.14969.
- Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi Jaakkola. 2017. Predicting organic reaction outcomes with weisfeiler-lehman network. *Preprint*, arXiv:1709.04555.
- Chang Liao, Yemin Yu, Yu Mei, and Ying Wei. 2024. From words to molecules: A survey of large language models in chemistry. *Preprint*, arXiv:2402.01439.
- Pengfei Liu, Jun Tao, and Zhixiang Ren. 2025. A quantitative analysis of knowledge-learning preferences in large language models in molecular science. *Preprint*, arXiv:2402.04119.
- Zequn Liu, Wei Zhang, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Ming Zhang, and Tie-Yan Liu. 2023.
 MolXPT: Wrapping molecules with text for generative pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1606–1616, Toronto, Canada. Association for Computational Linguistics.

- Harry L. Morgan. 1965. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5:107–113.
- Emmanuel Noutahi, Cristian Gabellini, Michael Craig, Jonathan S. C Lim, and Prudencio Tossou. 2023. Gotta be safe: A new framework for molecular design. *Preprint*, arXiv:2310.10773.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason

Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi,

814

815

816

818

825

832

835

841

858

870

871

872

874

875

876

877

Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

878

879

881

882

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

- Siru Ouyang, Zhuosheng Zhang, Bing Yan, Xuan Liu, Yejin Choi, Jiawei Han, and Lianhui Qin. 2024. Structured chemistry reasoning with large language models. *Preprint*, arXiv:2311.09656.
- Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2024. Scaling large-language-model-based multi-agent collaboration. *Preprint*, arXiv:2406.07155.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *Preprint*, arXiv:2307.16789.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Preprint*, arXiv:2302.04761.
- Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. 2019. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583.
- Ming Shen. 2024. Rethinking data selection for supervised fine-tuning. *Preprint*, arXiv:2402.06094.
- Yaorui Shi, An Zhang, Enzhi Zhang, Zhiyuan Liu, and Xiang Wang. 2023. Relm: Leveraging language models for enhanced chemical reaction prediction. *Preprint*, arXiv:2310.13590.
- T. Song, M. Luo, L. Chen, Y. Huang, Q. Zhu, D. Liu, et al. 2024. A multi-agent-driven robotic ai chemist enabling autonomous chemical research on demand. *ChemRxiv*. This content is a preprint and has not been peer-reviewed.
- Sakhinana Sagar Srinivas and Venkataramana Runkana. 2024. Cross-modal learning for chemistry property prediction: Large language models meet graph machine learning. *Preprint*, arXiv:2408.14964.
- Arya Tafvizi, Besim Avci, and Mukund Sundararajan. 2022. Attributing auc-roc to analyze binary classifier performance. *Preprint*, arXiv:2205.11781.

Xiangru Tang, Qiao Jin, Kunlun Zhu, Tongxin Yuan, Yichi Zhang, Wangchunshu Zhou, Meng Qu, Yilun Zhao, Jian Tang, Zhuosheng Zhang, Arman Cohan, Zhiyong Lu, and Mark Gerstein. 2024. Prioritizing safeguarding over autonomy: Risks of Ilm agents for science. *Preprint*, arXiv:2402.04247.

931

932

941

947

949

951 952

954

957

960

963

965

966

967

969

970

971

973

974

975

976

977

978

979

985

- T.T. Tanimoto. 1958. An Elementary Mathematical Theory of Classification and Prediction. International Business Machines Corporation.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
- Hongwei Wang, Weijiang Li, Xiaomeng Jin, Kyunghyun Cho, Heng Ji, Jiawei Han, and Martin D.
 Burke. 2021. Chemical-reaction-aware molecule representation learning. *Preprint*, arXiv:2109.09888.
- Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. 2024. Gta: A benchmark for general tool agents. *Preprint*, arXiv:2407.08713.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.
- Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. 2018. Moleculenet: A benchmark for molecular machine learning. *Preprint*, arXiv:1703.00564.
- Hui Yang, Sifu Yue, and Yunzhong He. 2023. Autogpt for online decision making: Benchmarks and additional opinions. *Preprint*, arXiv:2306.02224.
- Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. 2024. Moosechem: Large language models for rediscovering unseen chemistry scientific hypotheses. *Preprint*, arXiv:2410.07076.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023.
 React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.
- Junjie Ye, Sixian Li, Guanyu Li, Caishuang Huang, Songyang Gao, Yilong Wu, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Toolsword: Unveiling safety issues of large language models in tool learning across three stages. *Preprint*, arXiv:2402.10753.
- Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024. Cut the crap: An economical communication pipeline

for llm-based multi-agent systems. *Preprint*, arXiv:2410.02506.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li,
Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu,
Shuai Fan, Guodong Shen, Kai Yu, and Xin Chen.
2024. Chemdfm: A large language foundation model
for chemistry. *Preprint*, arXiv:2401.14818.988

986

987

993

997

1000

1001

1002

1003

1004

1005

1006

1008

1009

1010

1013

1014

1015

1018

1019

1020

1021

1023

1024

1025

1027

1029

1030

1032

1033

1034

1037

1039

Appendix

A Stacking Agent Details

We conducted several experiments and selected three optimal hierarchical stacking toolsets, and now we will present the stacking results and scores for each stacking agent in the Tab. 9, along with the corresponding prompts:

A.1 Stacking Agent

A.1.1 Prompt

Our Agent framework is based on the ReAct method (Yao et al., 2023) to implement tool and reasoning processes.

A.1.2 Naming Rule

To facilitate the comprehension of Hierarchical-Tool-Stacking, we propose a systematic hierarchical naming rules as follows in Fig. 4. In addition, in order to prevent the tool name from affecting the agent call, we choose to anonymously process the agent tool, that is, {task name}_{num}.

> • Self-Stacking Tools: Hierarchical proxies are constructed through a recursive generation strategy, with the depth of the hierarchy dynamically extendable via the numerical suffix. For instance in Fig. 4, "["Name2SMILES_0"]" denotes the base tool, while "["Name2SMILES_1"]" signifies a firstlayer tool, referred to as an Agent Tool (which encapsulates both the tool and the Agent into a new tool) and "["ChemDFM_2"]".

• Multiple Tool Combinations: The combination of multiple tools within an Agent is represented in a list format, utilizing depth-first traversal to generate sub-tools, thereby forming the final toolset for the agent. For example in Fig. 4, the structure "["Name2SMILES_1", "ChemDFM_2"]" represents a flat structure with tools at the same level ([A, B, ...]), while the structure "[['Name2SMILES_0','ChemDFM_1'],'Nam e2SMILES_1','ChemDFM_0']" illustrates a nested structure ([[A, B], C, D]), where tools A and B are first combined before being integrated with tool C and D.

A.2 Text-based Molecule Design

A.2.1 Task Introduction

The test set of ChEBI-20-MM is exploited for this task in ChemLLMBench. Models are asked to pre-

dict the SMILES of the molecule that fits the given1040description. Considering the low accuracy of the1041models, we use BLEU-2 as the training metric and1042use metrics such as Exact, Dis and others during1043the testing stage.1044

1045

1046

1047

1048

1049

1050

1051

1052

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

A.2.2 Prompt

We use a simpler prompt compared with the prompt introduced in (Guo et al., 2023)

Prompt: Text-based Molecule Design

You are an expert chemist. Given the molecular requirements description, your task is to design a new molecule SMILES: Molecular requirements description::

A.3 Molecule Captioning

A.3.1 Task Introduction

The test set is the same with the Text-based Molecule Design task. Because this is the mirroring task, which generates a detailed description by giving a SMILES to the models. In this task, we also choose the BLEU-2 as the metric in the training stage. When in the test stage, more metrics, like BLEU and ROUGE, are utilized to Measure the performance of the model.

A.3.2 Prompt

We also use a simpler prompt compared with the prompt introduced in (Guo et al., 2023)

Prompt: Molecule Captioning

You are an expert chemist. Given the molecular SMILES, your task is to provide the detailed description((The molecule is ...) of the molecule. Please strictly follow the format, no other information can be provided. Molecular SMILES:

A.4 Molecular Property Prediction

A.4.1 Task Introduction

The molecular property prediction tasks in Chem-
LLMBench consist of five tasks from MoleculeNet1065benchmark (Wu et al., 2018), including BACE,
BBBP, HIV, ClinTox, and Tox21. Among these,
BACE and BBBP are balanced binary classification
tasks, while HIV represents an unbalanced binary
classification task. ClinTox consists of 2 unbal-
anced binary classification tasks, and Tox21 com-1067



Figure 4: The example of the Naming rule, where the icon " 👻 ", " 😒 ", represent the Agent and the icon ' 🤄 ', ' 👗 ' represent the Retrieve tool and Compute tool respectively.

Toolz	Train		Docult
Task	Final tool	score	Result
Text-based	['Name2SMILES_3', 'ChemDFM_0']	0.80	0.90
Molecule	[['ChemDFM_0', 'Name2SMILES_1'], 'ChemDFM_1']	0.81	0.93
Design	[['ChemDFM_1','Name2SMILES_1'],['ChemDFM_1','Name2SMILES_2']]	0.91	0.91
Molecule	['SMILES2Description_2','TextChemT5_0']	0.79	0.73
Contioning	['SMILES2Description_3']	0.70	0.65
Captioning	['TextChemT5_1','SMILES2Description_1']	0.71	0.66
Reaction	['Chemformer_0','SMILES2Property_2']	1.00	0.90
Prediction	['Chemformer_2']	0.90	0.89
Treatenon	['Chemformer_1','SMILES2Property_1']	0.90	0.85
Property	['UniMol_1','SMILES2Property_1']	0.80	72.1
Prediction	[['UniMol_1', 'SMILES2Property_0'], 'SMILES2Property_0']	0.82	81.4
(BACE)	['UniMol_3']	0.75	78.6
Property	['UniMol_1','SMILES2Property_2']	0.68	71.1
Prediction	['SMILES2Property_3']	0.75	69.3
(BBBP)	['UniMol_2']	0.73	70.6
Property	[['UniMol_1', 'SMILES2Property_0'], ['UniMol_1', 'SMILES2Property_1']]	0.70	72.3
Prediction	['SMILES2Property_2']	0.68	61.4
(Clintox)	['UniMol_1','SMILES2Property_1']	0.65	69.1
Property	[['SMILES2Property_1', 'UniMol_0'], 'SMILES2Property_1']	1.00	97.4
Prediction	['UniMol_0', 'SMILES2Property_1']	0.85	90.1
(HIV)	['UniMol_2']	0.90	96.9
Property	['UniMol_2']	0.78	92.3
Prediction	['UniMol_0','SMILES2Property_2']	0.80	79.6
(Tox21)	['SMILES2Property_2']	0.85	74.9

Table 9: Stacking results of different tasks. The **bold** font represents the most suitable toolset obtained in the task experiment.

1073prises 21 unbalanced binary classification tasks.1074In this task, we choose the AUC-ROC as the first1075metric in the training stage. Considering the cal-1076culation method of AUC-ROC for large language1077models, we also introduced Accuracy as a second1078metric for test stage.

A.4.2 Prompt

1079

1080

1081

1083

1084

1085

1086

1087

1088

1089

1090

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

We use the same prompts introduced in (Guo et al., 2023)

A.5 Reaction Prediction

A.5.1 Task Introduction

The reaction prediction task asks the model to predict the product of the given reaction. ChemLLM-Bench utilizes the USPTO-MIT dataset for this task. Since the benchmark metric is Accuracy, we also chose Accuracy as the training metric, and considering that the answer is also SMILES, we adopted the same metric as Molecular Design task for measurement during the testing stage.

A.5.2 Prompt

We reformat the prompt provided (Guo et al., 2023).

Prompt: Reaction Prediction

Given an incomplete chemical reaction equation in SMILES notation (format: reactants»product, where multiple reactants are separated by dots '.'), predict and complete the missing products marked as '___'. The response should only contain the only one SMILES representation of the missing molecule, without any additional explanation (Note: Please only output only one final product).

Please Answer the quetion based on the following Chemical reaction equation:

B Multi-agent Implementation Details

In this section, We will discuss how to implement multi-agent systems and specific ways of information transmissionincluding chain, random, star, fullconnected, layered and debate graph.

B.1 Framework

In order to complete chemical tasks, we divided multi-agent into two types: agents with tools and agents without tools, and tested them on the first task, the Text-based Molecule Design task. The overall of our multi-agent framework is a modifica-
tion of the framework of (Zhang et al., 2024) and
(Qian et al., 2024) that utilized different spatial and
temporal masks to complete in the following six
multi-agent structures: Chain, Random, FullCon-
nected, Layered, Star and Debate mode.1106
1110

However, since we are modifying their approach with a greater focus on information transmission and are also limited by API calls, we can only make a one-sided comparison regarding the recording of tokens and time.

B.2 Implementation Details

In multi-agent systems, information transmission is a critical factor for enhancing performance. By utilizing various structures, information can be conveyed through multiple pathways. To improve the efficiency of information collection, we have adopted the Final decision approach. Specifically, at the end of all structures, we have integrated a FinalRefer Agent to perform the final summary and decision-making. The FinalRefer prompt is followed:

Prompt: FinalRefer

You are a strategic planning and final integration agent. You will be given a graduatelevel question and reasoning outputs from all other agents. Your task is to integrate all the information into a single, cohesive answer with detailed reasoning and evidence. Your final output should: 1. Summarize the contributions from all agents, highlighting key insights.

3. Provide the final answer with a clear and detailed explanation.

4. Conclude with the final answer on a new line with the format: "The final answer is 'SMILES'

Here is the question:question. At the same time, the output of other agents is as follows:

answers

In the implementation of the agents with tools, we modified all agents along the path except for the Final agent, while still following the ReAct framework for tool calling. During this process, due to the constraints of API calls, both the time required and the number of tokens used will be greater compared to agents without tools. 1128

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1130 1131

1132 1133

1136 1137

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

B.3 Spatial Communication Topologies

B.3.1 Chain

1138The chain graph (Fig. 5) is one of the most widely1139utilized communication architectures in contempo-1140rary multi-agent systems. In this architecture, the1141first agent receives input from the user, transforms1142it into new instruction, and subsequently forwards1143it to the next agent. Generally, the final agent in thechain provides a summary and answers.



Figure 5: Demonstration of chain structure

B.3.2 Random

The random graph refers to a sparse graph randomly sampled from a complete graph, as shown in the Fig. 6. They will execute asynchronously in multiple rounds and then randomly transmit information to the target agent. Finally, all the answers and information will transmit to the Final agent to make a final answer.



Figure 6: Demonstration of random structure

B.3.3 FullConnected

The fullconnected graph (Fig. 7) is a directed graphs compared to traditional fully linked undirected graphs, which transmit information in a certain order to complete this topology structure. The final agent summarizes the dialogue and provides a concluding output or reflection.



Figure 7: Demonstration of FullConnected structure

B.3.4 Layered

The layered graph (Fig. 8, (Qian et al., 2024)) refers1161to a stacked configuration similar to a multilayer1162perceptron (MLP). The first layer agents will feed1163to the agents in the second layer, and the final layer1164will make the summary and final-decision.1164



Figure 8: Demonstration of Layered structure

1165

1166

1160

B.3.5 Star

The star grpah (Fig. 9) resembles the tree structure.1167ture. Firstly, the problem will be handed over to1168the external leaf nodes for processing, and the obtained answer will be passed to the central root1169tained answer will be repeated multiple times.1170node, which will be repeated multiple times.1171nally, the root node will give a summary and make1172



Figure 9: Demonstration of Star structure

1173

1174

1175

1176

1177

1178

1179

1180

1181

B.3.6 Debate

The debate graph (Fig. 10, (Du et al., 2023)) is multiple agents to engage in a debate, where in each round, every agent receives the outputs of all agents from the previous round before making their own statements. Generally, the finalRefer agent will help them to make the final descision.

B.3.7 Tree-like&Ours

In order to better understand our stacking structure, we will compare it with a tree like multi-agent1182ture, we will compare it with a tree like multi-agent1183pipeline that is relatively similar. The tree graph1184usually has the root node as the manager to supervise the nodes below to complete various tasks,1186and finally return the results of the leaf nodes to1187the root node for processing. Overall, this is just a1188



Figure 10: Demonstration of Debate structure

top-down process of information transmission. Asshowin in the Fig. 11

As for our stacking structure, from the perspective 1191 of information transmission, the main agent of the 1192 root node also receives the information completed 1193 from below, but there is a difference. For the so-1194 called tool node, it is a bottom-up process. After 1195 continuous information superposition step by step, 1196 it is passed upward through the parent node and 1197 then given to the main agent for processing. It can 1198 be seen from the figure that in each transmission 1199 process, whether it is the root node or the parent 1200 node, they all selectively accept the information 1201 from the child node, and it is not like a tree struc-1202 ture that is passed downward. 1203



Figure 11: Demonstration of **Tree** and **Ours** structure. In the stacking structure, the icon '*.' means the parent node can choose which child node's information to accept.

C Multi-agent Experimental Results

1204

1205

1206

1207

1209

1210

1211

1212

1213

1214

1215

The experimental results are shown in the Tab. 10. From the table, it can be observed that for different structures, there is an initial performance improvement as the value of num increases. However, when num reaches 8, only the Random, Layered, and Star modes show some improvement, with a maximum score of 0.891. In contrast, our best combined structure ([['ChemDFM_0', 'Name2SMILES_1'], 'ChemDFM_1']) achieves a score of **0.925** with a num of only 4, indicating a significant enhancement.

Furthermore, since our multi-agent framework has been modified from others, the number of rounds for information transmission has not been optimized, leading to a substantial increase in both token count and time. This also suggests that even in complex and prolonged reasoning scenarios, relying solely on information transmission does not yield particularly high improvements. 1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

			Text-based Me	olecule E	esign (BLEU-2)	- w/o-1	Fool	
NUM	Chain	Random	FullConnected	Layered	Star	Debate	Avg	Avg_all_tokens	Avg_Time
0	0.548	0.548	0.548	0.548	0.548	0.548	0.548	890.19	8.254
1	0.604	0.604	0.604	0.604	0.604	0.604	0.604	2315.22	15.826
2	0.587	0.587	0.587	0.587	0.587	0.587	0.587	3619.54	21.034
4	0.608	0.642	0.672	0.666	0.661	0.639	0.648	52010.75	74.029
8	0.633	0.701	0.684	0.640	0.649	0.636	0.657	344604.597	227.012
	Text-based Molecule Design (BLEU-2) - Tool (Name2SMILES,ChemDFM)								
NUM	Chain	Random	FullConnected	Layered	Star	Debate	Avg	Avg_all_tokens	Avg_Time
0	0.832	0.832	0.832	0.832	0.832	0.832	0.832	2528.79	21.382
1	0.842	0.842	0.842	0.842	0.842	0.842	0.842	3125.64	30.237
2	0.844	0.844	0.844	0.844	0.844	0.844	0.844	18446.91	108.427
4	0.864	0.888	0.885	0.867	0.884	0.889	0.880	150544.061	2207.641
8	0.841	0.890	0.854	0.887	0.891	0.874	0.873	733540.998	6744.788
	Text-b	ased Mole	ecule Design (B	LEU-2)	- Stacki	ng Tool (1	Name2	SMILES,ChemD	FM)
NUM			Stacking	Tool			Score	Avg_all_tokens	Avg_Time
2			['ChemDF	M_2']			0.898	2801.93	39.103
3		['Nan	e2SMILES 1',	'ChemD	FM 1']		0.918	2821.38	63.795
4		['Nan	e2SMILES_3',	'ChemD	FM_0'		0.904	2745.07	78.144
4	[['Che	emDFM ()', 'Name2SMI	LES 1'].	'Chem	DFM 1']	0.925	2851.43	72.484
8		[['Che ['Cher	mDFM_1','Nar nDFM_1'.'Nan	ne2SMII ne2SMIL	ES_1'] ES_2'1	, , 1	0.907	1830.11	95.623

Table 10: Results of Multi-agent experiment. 0:Final-Refer agent is not included.

D Case Study

As stated in Section 5.4, there are four distinct stacking behavior patterns-**correct**, **modify**, **judge** and **reserve** that make the stacking works. Here are three other cases for these patterns in the Tab. 11:

D.1 Correct

As analyzed in Section 5.4, "Correct" is a very important pattern for agents. It was able to leverage the complementary strengths of multiple tools, significantly improving the accuracy of the final performance.

D.2 Modify

As shown in the table, this is a common approach 1236 to using various tools. First, the problem is decom-1237 posed, and the RAG tool is used to retrieve informa-1238 tion on each sub-question to obtain a preliminary 1239 answer. Then, subsequent processing is carried out 1240 using computational tools. Alternatively, one can 1241 first obtain an answer through computational tools, 1242 then have the agent self-assess the correctness of 1243 that answer, and finally use the RAG tool for cross-1244 verify, thereby refining the answer and improving 1245 accuracy. 1246

Stacking Pattern	Input	Output
		To find the SMILES representation of the peptide anion that is the conjugate base of gamma-Glu-Abu, I first need to
	The molecule is a peptide anion that is the	understand the structure of gamma-Glu-Abu. Gamma-Glu-Abu stands for gamma-glutamyl-aminobutyric acid. I will
	conjugate base of gamma-Glu-Abu, obtained by	start by finding the SMILES of gamma-Glu-Abu.
	removal of protons from the two carboxy	Action: Name2SMILES
	groups as well as protonation of the amino	$\frac{11}{10}$
	group; major species at pH 7.3. It is a conjugate	Observation: The next step is to modify this SMILES to represent the conjugate base of gamma-Glu-Abu at pH 7.3.
Modify	base of a gamma-Glu-Abu. Please try to infer	where the two carboxyl groups are deprotonated and the amino group is protonated. This involves changing the carboxyl
	the SMILES of this molecule.	groups from "-C(=O)O" to "-C(=O)[O-]" and the amino group from "-N" to "-[NH3+]".
		Action: ChemDFM
	Gold Answer:	Input: Modify the "CCC(NC(=O)CCC([NH3+])C(=O)[O-])C(=O)[O-] " from "-C(=O)O" to "-C(=O)[O-]" and the pure form " N " L " DI12 -1
	CCC(NC(=O)CCC([NH3+])C(=O)[O-	amino group from $-N$ to $-[NH3+]$. Output: $CCC(NC(-O)CCC([NH3+])C(-O)[O_1])C(-O)[O_1]$
])C(=O)[O-]	Thought: I now know the final answer
		Final Answer: CCC(NC(=O)CCC([NH3+])C(=O)[O-])C(=O)[O-] ✓
		The molecule described is a complex carbohydrate structure involving alpha-L-Fucp, beta-D-Galp, and D-GlcNAc units.
		The specific linkages and configurations are crucial for determining the correct structure. Given the complexity of the
		molecule, the best approach is to use the Molecule_Design tools, which are designed to handle detailed questions about
		on the detailed description provided
		Action: Molecule Design 1
		Action Input: The molecule is an alpha-L-Fucp-(1->2)-beta-D-Galp-(1->3)-[alpha-L-Fucp-(1->4)]-D-GlcNAc where
		the glucosamine at the reducing end has beta-configuration at its anomeric centre. It has a role as an epitope. Please
	The molecule is a cycloalkane composed of	provide the SMILES of this molecule.
	three carbon atoms to form a ring. It has a role	Observation: CC(=O) N[C@@H]I[C@@H](O]C@@H]2O[C@@H](C)[C@@H](O)[C@@H](O)[C@@H]2O](C@H]
	and a member of cyclopropages Please try to	
i	infer the SMILES of this molecule.	Thought: The SMILES provided by Molecule Design 1 seems to represent a complex carbohydrate structure, but to
		ensure accuracy, I will use Molecule_Design_2 for a second opinion. This will help verify the correctness of the
Judge		SMILES or provide an alternative if there are discrepancies.
	Gold Answer:	Action: Molecule_Design_2
		Action input: The molecule is an alpha-L-Fucp-(1->2)-beta-D-Galp-(1->5)-[alpha-L-Fucp-(1->4)]-D-GicNAC where the glucosamine at the reducing and has beta-configuration at its anomaric centre. It has a role as an enitone Please
-	$O(C_{0})$	includes and the shall be of this molecule.
)[C@H](O)[C@H]2O[C@@H]2O[C@@H](C)	Observation:Nc1ccn([C@@H]2O[C@H](COP(=O)([O-])[O-])[C@@H](O)[C@H]2O)c(=O)n1
	[C@@H](O)[C@@H](O)[C@@H]2O)[C@@	Thought: The SMILES provided by Molecule_Design_2 is significantly different from the one provided by
	H](CO)O[C@H]1O	Molecule_Design_1, indicating a discrepancy. The structure from Molecule_Design_2 seems less likely to represent the
		complex carbohydrate structure described in the question. The SMILES from Molecule_Design_1 appears more
		plausible as it includes multiple sugar units linked together, which aligns better with the description of the molecule involving alpha L Euco beta D Galp, and D GleNAc. Therefore, L will consider the SMILES from Molecule Design 1
		as the more likely correct representation.
		Thought: I now know the final answer
		Final Answer: CC(=O)N[C@@H]1[C@@H](O[C@@H]2O[C@@H](C)[C@@H](O)[C@@H](O)[C@@H]2O]
		[C@H](O[C@@H]2O[C@H](CO][C@H](O)[C@H](O)[C@H]2O[C@@H]2O[C@@H](C)[C@@H](O)[C@@H] (O)[C@@H]2O[C@@H](CO)O[C@H]1O/
		The SMILES representation for the molecule described is not readily available due to the complexity of the glycosidic
		linkages and the specific stereochemistry involved. Further detailed chemical synthesis knowledge or specialized
		software would be required to accurately construct the full SMILES notation of this tetrasaccharide.
		Thought: The initial attempt using the Molecule_Design_2 tool did not yield a direct SMILES result, likely due to the
		complexity of the molecule described. As an alternative approach, I can attempt to construct the molecule step-by-step
	The molecule is a linear amino tetrasaccharide	molecule. However, constructing the full SMILES for the entire molecule including specific linkages manually might
	composed of N-acetyl-beta-D-galactosamine,	not be feasible without specialized chemical software. I will first retrieve the SMILES for each sugar unit.
	alpha-D-galactose, beta-D-galactose and	Action: Name2SMILES
	N-acetyl-beta-D-galactosamine units joined by	Action Input: N-acetyl-beta-D-galactosamine
	sequential $(1>3)$ -, $(1>4)$ - and $(1>4)$ -linkages.	Observation: CC(=O)N[C@@H]1[C@@H](O)[C@@H](O)[C@@H](CO)O[C@H]1O
	tone Please try to give SMILES of this molecule	alpha.D.galactose
	topen lease try to give simillas of this molecule.	Action: Name2SMILES
Keserve		Action Input: alpha-D-galactose
	Gold Answer:	Observation:OC[C@H]10[C@H](O)[C@H](O)[C@@H](O)[C@H]10
	CC(=O)N[C@@H]1[C@@H](O)[C@H](O[C@@H)2O[C@H](O)[C@H](O)[C@H](O]C@H)(O)[C@H](O)O[O)OOOOOOOOOO[Thought: I have obtained the SMILES for alpha-D-galactose. Next, I will retrieve the SMILES for beta-D-galactose.
		Action Input: What is the SMILES of beta-D-galactose
	O(C@H)(O)(C@H)(O)(C@H)(O(C@H)(C))	Observation: OCIC@H]10[C@@H](O)[C@H](O)[C@H](O)[C@H]10
	H]30)[C@H](0)[C@H]20)[C@@H](C0)0[C	Thought: I have obtained the SMILES for beta-D-galactose. Since the molecule starts and ends with
	@H]10	N-acetyl-beta-D-galactosamine, and I already have the SMILES for this unit, I do not need to retrieve it again. However,
		constructing the full SMILES string for the tetrasaccharide with specific glycosidic linkages (1->3, 1->4, 1->4) between
		these units is beyond the capability of the Name2SMILES tool and requires specialized chemical synthesis software or
		Thought: I now know the final answer
		Final Answer: Unable to provide the exact SMILES string for the described tetrasaccharide due to the
		complexity of the glycosidic linkages and the limitations of the tools available.

Table 11: Case analysis of some behavior patterns of agents during tools stacking.

D.3 Judge

1247

Judge refers to the process of selecting between two 1248 candidate answers based on the model's knowledge 1249 in chemistry. This usually happens when two agent 1250 tools are available. When confronted with two 1251 anonymous tools that have the same descriptions, 1252 the model often opts to call both tools simulta-1253 neously before making a judgment. Furthermore, 1254 when using GPT-40 as the agent model, it typically 1255 demonstrates excellent judgment abilities. 1256

D.4 Reserve

Reserve is a relatively uncommon phenomenon for 1258 large language models. Due to the hallucination 1259 problem, they usually choose to cover up knowl-1260 edge blind spots to generate answers. However, 1261 when faced with multiple tools, the agent faces 1262 several answers. After thinking, it does not judge 1263 which one is more suitable, so it chooses to reserve 1264 opinions. Although this will reduce accuracy, in 1265 some ways it is a good thing to emerge this ability. 1266