LATENTCHEM: FROM TEXTUAL COT TO LATENT THINKING IN CHEMICAL REASONING

Xinwu Ye[1,2,*], Yicheng Mao[3,*], Jia Zhang[4], Yimeng Liu[5], Hao Li[3], Fang Wu[6], Zhiwei Li[7], Yuxuan Liao[8], Zehong Wang[9,2], Zhenfei Yin[10,2], Li Yuan[3], Xiangxiang Zeng[4], Zhiyuan Liu[11,2], Mengdi Wang[12], Le Cong[6], Philip Torr[10], Huan Sun[13], Shenghua Gao[1], Xiangru Tang[8,2,†]
[1]The University of Hong Kong [2]Haven [3]Peking University
[4]Hunan University [5]University of Toronto [6]Stanford University
[7]Hong Kong University of Science and Technology (Guangzhou) [8]Yale University
[9]University of Notre Dame [10]University of Oxford [11]National University of Singapore
[12]Princeton University [13]The Ohio State University

ABSTRACT

Chemical large language models (LLMs) predominantly rely on explicit Chain-of-Thought (CoT) in natural language to perform complex reasoning. However, chemical reasoning is inherently continuous and structural, and forcing it into discrete linguistic tokens introduces a fundamental representation mismatch that constrains both efficiency and performance. We introduce **LatentChem**, a latent reasoning interface that decouples chemical computation from textual generation, enabling models to perform multi-step reasoning directly in continuous latent space while emitting language only for final outputs. Remarkably, we observe a consistent emergent behavior: when optimized solely for task success, models spontaneously internalize reasoning, progressively abandoning verbose textual derivations in favor of implicit latent computation. This shift is not merely stylistic but computationally advantageous. Across diverse chemical reasoning benchmarks, LatentChem achieves a **59.88%** non-tie win rate over strong CoT-based baselines on ChemCoTBench, while delivering a **10.84**$\times$ average reduction in reasoning overhead. Our results provide empirical evidence that chemical reasoning is more naturally and effectively realized as continuous latent dynamics rather than discretized linguistic trajectories.The code and model weights are publicly available at GitHub and HuggingFace.

1 INTRODUCTION

LLMs have emerged as transformative tools for scientific discovery, facilitating tasks ranging from molecule generation to reaction synthesis Guo et al. (2023); Zhang et al. (2025); Xia et al. (2025). In this context, chemical reasoning is typically mediated through explicit CoT Zhang et al. (2024); Zhao et al. (2025b;a). In current chemical LLM paradigms, models are trained to linearize a complex physicochemical intuitions, such as electron delocalization or steric hindrance, into a discrete sequence of natural language tokens before arriving at a solution. While this CoT approach has unified scientific tasks under a generative framework Zuo et al. (2025), it relies on the implicit assumption that natural language is an adequate vehicle for the continuous physicochemical dynamics inherent to chemistry.

However, it remains unclear whether such discrete symbolic reasoning is optimal, for complex chemical reasoning tasks Li et al. (2025b); Zhang et al. (2025). We hypothesize that forcing chemical logic into a linguistic bottleneck results in a fundamental continuity–discretization gap in chemical reasoning (Figure 1). Much like an expert chemist who manipulates abstract 3D structures mentally before verbalizing the result, we posit that the core of chemical reasoning, such as navigating manifolds, optimizing properties, and identifying substructures, is more naturally performed in a continuous latent space. In this view, natural language should serve merely as the input/output interface rather than the computational substrate for the reasoning process itself.

*Co-first authors.
†Corresponding author: xiangru.tang@yale.edu.

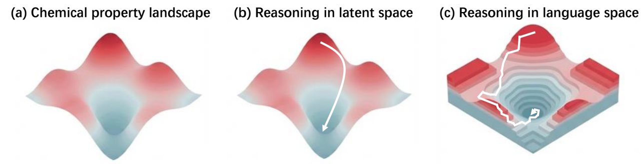

Figure 1: Conceptual illustration of the continuity–discretization gap in chemical reasoning. (a) The intrinsic chemical property landscape is continuous and high-dimensional. (b) We posit that a continuous latent space can theoretically offer a smoother optimization surface akin to the property landscape, avoiding the jagged trajectories of discrete tokens. (c) Linguistic tokenization fragments chemical state transitions into discrete symbolic steps, inducing staircase-like landscapes and inefficient reasoning paths.

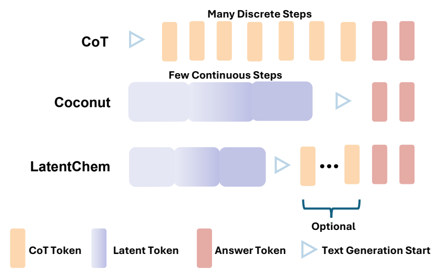

Figure 2: Comparison of reasoning paradigms in chemical LLMs. (Top) Explicit CoT relies on discrete linguistic steps, forcing high-dimensional chemical intuition into a constrained textual bottleneck. (Middle) Generic latent reasoning (e.g., CoconutHao et al. (2025)) shifts reasoning to a continuous latent space but treats molecular embeddings as a static context, limiting the ability to focus on different substructures during reasoning. (Bottom) LatentChem (Ours) introduces a dynamic perception-reasoning loop. Leveraging the ChemUpdater mechanism, latent thoughts actively re-query and refine the molecular representation at each step, ensuring structure-aware reasoning. Depending on the complexity of the query and the allocated latent thinking budget, the model exhibits dynamic reasoning behaviors, ranging from fully latent computation (no textual CoT) to hybrid reasoning with varying lengths of textual CoT.

To study this hypothesis, we introduce LatentChem, a latent reasoning interface Deng et al. (2024); Hao et al. (2025) designed to decouple chemical reasoning from explicit natural language generation. As illustrated in Figure 2, unlike standard models that rigidly bind reasoning to token output, LatentChem injects a lightweight sequence of continuous thought vectors between perception and generation. Crucially, we equip this interface with a ChemUpdater mechanism, allowing these latent thoughts to actively re-query molecular features in a dynamic loop. This architecture serves as an experimental instrument, allowing us to observe whether the model prefers to reason via explicit text or through internal latent dynamics when optimized for task success.

Our investigation reveals an intriguing emergent behavior: the model exhibits spontaneous internalization of chemical logic. Despite being initialized with explicit CoT data, when optimized via reinforcement learning with rewards only on format adherence, validity, and correctness, LatentChem voluntarily discards verbose textual reasoning chains in the majority of tasks. Instead, it predominantly collapses the reasoning process into the continuous latent space, outputting solutions directly after a sequence of "silent" thought vectors. While this shift sacrifices the explicit readability of intermediate steps, it strongly suggests that the model perceives natural language as a low-bandwidth constraint, identifying the continuous latent reasoning as a more native and expressive mode for chemical logic.

Crucially, this spontaneous shift does not imply that the model is merely taking a shortcut to minimize generation effort, as such a behavior would typically degrade task performance. On the contrary, our experimental results demonstrate that this internalization represents a superior computational strategy. Quantitatively, by choosing to compress linguistic steps into compact latent states, LatentChem achieves a dramatic efficiency gain, reducing the reasoning token overhead by an average of **10.84**$\times$ across all benchmarks. In molecule optimization and reaction tasks, this reduction factor even exceeds **28**$\times$. This efficiency is achieved alongside dominant performance, as evidenced by extensive evaluations across four diverse benchmarks, ChemCoTBench Li et al. (2025a), Mol-Instructions Fang et al. (2023), ChEBI-20 Edwards et al. (2021), and ChemLLMBench Guo et al. (2023). Notably, LatentChem achieves a **59.88%** non-tie win rate against the strong explicit CoT baseline on the reasoning-intensive ChemCoTBench. The simultaneous improvement in both reasoning-efficiency and accuracy confirms that decoupling reasoning from language unlocks a more native and effective reasoning paradigm for chemical LLM.

In summary, our contributions are as follows:

- We propose a reasoning mode to investigate chemical LLMs, challenging the necessity of natural language CoT for chemical tasks.
- We introduce LatentChem, a system that empowers chemical LLMs to perform latent thinking via continuous thought vectors and a dynamic perception loop.
- We report the phenomenon of spontaneous internalization, where the model autonomously absorbs explicit reasoning into the latent space, validating the efficiency of continuous representations.
- We demonstrate that this paradigm shift yields superior performance and token efficiency across diverse chemical benchmarks compared to explicit CoT approaches.

2 Related Work

LLMs for chemical tasks. While general LLMs struggle with the specialized logic required for scientific discovery Hatakeyama-Sato et al. (2023); Jiang et al. (2025a); Sallam et al. (2024); Tang et al. (2025b), the field has progressed through domain-specific fine-tuning Zhang et al. (2024); Zhao et al. (2025b), multimodal architectures Tan et al. (2025); Jiang et al. (2025b); Li et al. (2025c); Bai et al. (2025); Lv et al. (2025); Liu et al. (2023) that integrate 2D/3D geometries to capture chemical topology Pei et al. (2024); Xia et al. (2025), and agentic decomposition of tasks Tang et al. (2025a). Furthermore, reinforcement learning is increasingly utilized to enforce logical consistency and physical validity Narayanan et al. (2025); Zhuang et al. (2025); Wang et al. (2025c), often within unified training frameworks Zuo et al. (2025). Despite these algorithmic strides, current systems remain prone to hallucination Li et al. (2025b) and often lack the robust planning capabilities necessary for complex chemical reasoning tasks Zhang et al. (2025).

Latent thinking in LLMs. An emerging paradigm shifts reasoning from explicit token generation to implicit processing within high-dimensional latent spaces, bypassing the linguistic bottleneck of sequential text. Pioneering works demonstrate that models can internalize reasoning steps by feeding hidden states directly as input and benefit from multiple reasoning paths Deng et al. (2024); Hao et al. (2025); Zhu et al. (2025). To structure these internal processes, research has compressed reasoning to "contemplation tokens" Cheng and Durme (2024) or "capsules" Shan et al. (2025), and employs differentiable caches for "soft thoughts" and memories Xu et al. (2025a); Liu et al. (2024). These latent states are further refined through auxiliary supervision, synthetic target for alignment, self-rewarding mechanisms, and variance optimization Wei et al. (2025); Wang et al. (2025a); Chen et al. (2024); Wang et al. (2025b). Crucially, this non-sequential nature unlocks test-time compute scaling: models can deepen reasoning through recursive unrolling Geiping et al. (2025); Aleksandrov et al. (2025), leverage native parallelism via Jacobi iteration, Wen et al. (2025); Wu et al. (2025) or scaled through diverse initializations Xu et al. (2025b). These approaches establish the foundation for the iterative, structure-aware computation that our method extends to dynamically refine molecular features.

3 LatentChem: A Latent Reasoning Interface

We present LatentChem, a framework designed to decouple chemical reasoning from linguistic generation. As illustrated in Figure 3, the system comprises three core components: (1) a latent thinking architecture that establishes the structural foundation for non-linguistic reasoning; (2) an active perceptual refinement mechanism that enables the model to dynamically update its molecular understanding; and (3) a progressive training protocol that evolves the model from simple alignment to outcome-driven policy optimization.

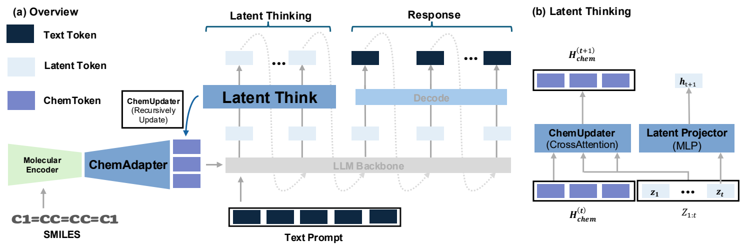

Figure 3: Overview of the LatentChem architecture and the latent thinking mechanism. (a) LatentChem decouples reasoning from generation via a dedicated latent thinking phase. A ChemAdapter aligns molecular features with the LLM space. During the reasoning phase, the model generates a sequence of continuous latent thought vectors. The ChemUpdater allows these thoughts to recursively refine the ChemToken based on the current reasoning state before the final response is decoded. (b) In latent thinking, the ChemUpdater refines the ChemTokens from current latent states, while the Latent Projector produces the next-step input embedding for continued reasoning.

3.1 Latent Thinking Neural Architecture

The architecture of LatentChem introduces three specialized modules integrated into a general-purpose LLM backbone: the ChemAdapter for initial perception, the ChemUpdater for dynamic refinement, and the Latent Projector for reasoning continuity.

ChemAdapter. To bridge the modality gap between continuous molecular features and the discrete embedding space, we employ a query-based attention projector (a Perceiver Resampler architecture Alayrac et al. (2022)). Given a molecule, we first extract a variable-length sequence of dense features $\mathbf{H}_{mol} \in \mathbb{R}^{L \times d_{enc}}$ using the pre-trained SMI-TED encoder Soares et al. (2024). Then, to characterize the molecule comprehensively, we initialize N learnable latent queries $\mathbf{Q} \in \mathbb{R}^{N \times d_{llm}}$. These queries serve as semantic anchors, extracting specific chemical attributes via cross-attention:

$$\mathbf{H}_{chem} = W(\mathrm{LN}(\mathbf{Q} + \mathrm{MHA}(\mathbf{Q}, \mathbf{H}_{mol}, \mathbf{H}_{mol}))),$$

where W projects dimensions to the LLM size d_{llm}. This mechanism compresses variable-length molecular information into a fixed number of "ChemTokens" $\mathbf{H}_{chem}$, which are prepended to the textual instruction embeddings as a prefix soft prompt.

ChemUpdater and Latent Projector. To enable dynamic interaction between reasoning and perception, we incorporate two complementary modules. First, unlike standard models that treat encoders as static feature extractors, the ChemUpdater enables dynamic perceptual refinement. It utilizes a cross-attention layer where the current ChemTokens $\mathbf{H}^{(t)}_{chem}$ serve as queries and the accumulated reasoning history $\mathbf{Z}_{1:t}$ serves as keys and values, allowing the model to shift its focus on molecular substructures as reasoning evolves. Second, to close the loop, we employ a Latent Projector, implemented as a lightweight residual feed-forward network (FFN). This module maps the raw hidden state $\mathbf{z}_t$ output by the LLM backbone back into the input embedding space, creating the continuous input vector required for the subsequent step and effectively bypassing the discrete tokenization bottleneck.

3.2 Latent Thinking with Active Perceptual Refinement

The core of LatentChem is the continuous latent thinking loop, which transforms the generation process into a dual-pathway system of thought generation and active perceptual updating. At each reasoning step t, the LLM produces a high-dimensional hidden state $\mathbf{z}_t$. Instead of decoding this state into a text token, the model treats it as a "thought vector" that drives two parallel processes.

On the perceptual pathway, the raw state $\mathbf{z}_t$ is appended to the history $\mathbf{Z}_{1:t}$ to trigger the ChemUpdater. The model actively refreshes its molecular representation via:

$$\mathbf{H}^{(t+1)}_{chem} = \mathrm{LN}\left(\mathbf{H}^{(t)}_{chem} + \mathrm{CrossAttn}(\mathbf{H}^{(t)}_{chem}, \mathbf{Z}_{1:t}, \mathbf{Z}_{1:t})\right).$$

This ensures the model does not just "see" a static structure but continuously refines its focus based on the latest thought. Simultaneously, on the reasoning pathway, the state $\mathbf{z}_t$ is processed by the Latent

Table 1: Overview of training strategies. We detail the module status and supervision signals across four stages.

Stage	Module		Supervision		
	Adapter + LLM	Latent Thinking	CoT	Answer	Reward
1	Trainable	Disabled	×	✓	×
2	Trainable	Disabled	✓	✓	×
3	Frozen	Trainable	✓	✓	×
4	Trainable	Frozen	×	×	✓

Projector to form the input for the next step: $\mathbf{h}_{t+1} = \mathbf{z}_t + \text{FFN}(\text{LN}(\mathbf{z}_t))$. The projected vector $\mathbf{h}_{t+1}$ is fed back into the backbone LLM to trigger $\mathbf{z}_{t+1}$. This cycle continues until the model predicts a discrete termination token <end_latent> or reaches a budget limit T_{max}, at which point the final refined context is used to decode the explicit response.

3.3 Training Protocol: From Explicit Alignment to Implicit Reasoning

We design a progressive four-stage training protocol to instill latent reasoning capabilities (Table 1). The first three stages focus on supervised structural alignment and mind activation, establishing the prerequisites for reasoning. Crucially, the final stage employs Group Relative Policy Optimization (GRPO) Shao et al. (2024), a reinforcement learning paradigm that allows the model to autonomously explore and optimize its reasoning trajectory based on task feedback. We provide an expanded elaboration of the protocol in Appendix C and detailed implementation hyperparameters in Appendix D.

Training data. Our model is trained on ChemCoTDataset Li et al. (2025a), a specialized corpus designed for step-by-step chemical reasoning. We utilize the snapshot from November 2025,[1] which comprises approximately 14k high-quality CoT-annotated samples covering diverse tasks such as molecular understanding, editing, optimization, and reaction prediction.

Stage 1: Establishing the molecular-linguistic mapping. The primary goal is to align the projected ChemTokens $\mathbf{H}_{chem}$ with the LLM's semantic space. We use "answer-only" supervision, suppressing intermediate reasoning steps to force the ChemAdapter to compress all necessary properties into the ChemTokens. To prevent textual overfitting, we employ a counterfactual alignment strategy, optimizing a hinge loss $\mathcal{L}_{CF}$ that maximizes the margin between the likelihood of the answer given clean versus perturbed molecular inputs. The total objective is $\mathcal{L}_{total}^{(1)} = \mathcal{L}_{clean} + \lambda\mathcal{L}_{CF}$.

Stage 2: SFT for molecule-aware CoT. We then transition to full sequence training, requiring explicit CoT derivations $\mathbf{y}_{cot}$ prior to the answer $\mathbf{y}_{ans}$. To ensure the generated reasoning is strictly grounded in molecular structure rather than textual priors, we extend the counterfactual alignment strategy to the entire sequence $\mathbf{y}_{full} = [\mathbf{y}_{cot}, \mathbf{y}_{ans}]$.

Stage 3: Chemistry-aware latent mind activation. In this phase, we activate the latent thinking modules (updater and projector). To prevent the massive LLM backbone from treating the lightweight latent signals as noise, we freeze both the ChemAdapter and the LLM backbone, updating only the updater and projector. This constraint forces the latent modules to adapt to the frozen semantic space of the LLM, compelling them to generate "legible" thought vectors that guide the fixed decoder toward the correct CoT path.

Stage 4: GRPO with latent thinking budget. With the latent mind activated, we employ GRPO to refine the model's policy. Reversing the freezing strategy from the previous stage, we now freeze the latent thinking modules and fine-tune the remaining parameters. This strategy treats the learned latent dynamics as a stable "internal simulator," optimizing the backbone LLM to effectively utilize these latent thoughts for decision-making. The optimization is guided by a composite reward function comprising four terms: format adherence, answer validity, and answer correctness (detailed in Appendixe D.4). Critically, by removing the supervision for explicit CoT generation and rewarding only the final result, we enable the model to explore the most efficient reasoning pathway.

[1]The dataset was accessed on November 25, 2025. Since ChemCoTDataset is continuously updated, we explicitly state the access date to ensure reproducibility.

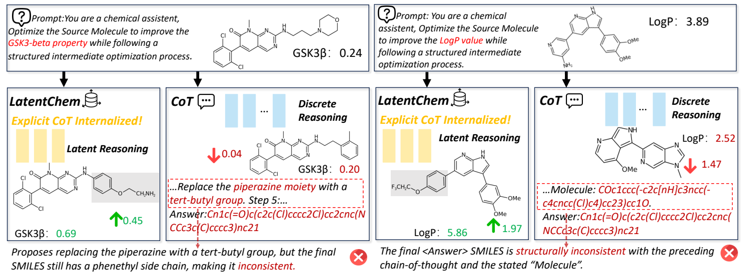

Figure 4: Case study: spontaneous internalization vs. explicit CoT. While the standard CoT model generates verbose reasoning chains that fail to execute the planned modification, LatentChem spontaneously internalizes these logics. It bypasses textual output, utilizing the latent space to perform structural optimization.

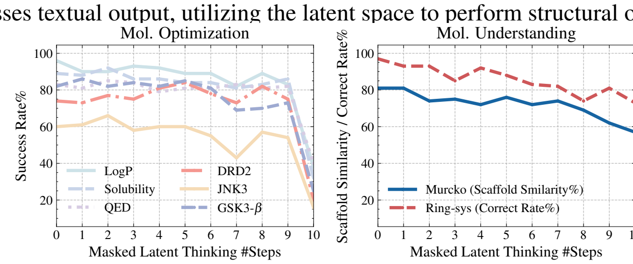

Figure 5: Causal necessity analysis. We measure task performance (success rate, scaffold similarity, or correct rate) as the first k latent tokens are replaced with Gaussian noise. The monotonic degradation observed in both Molecule Optimization (left) and Understanding (right) confirms that early latent states encode critical precursors for the final solution rather than redundant noise.

4 Emergent Properties of Latent Chemical Reasoning

Having established the LatentChem interface, we now investigate the nature of the reasoning process that emerges from the training protocol. To dissect these behaviors, we utilize ChemCoTBench Li et al. (2025a) as our primary testbed. Specifically, our analysis focuses on two representative tasks: molecule optimization and molecule understanding. Unless otherwise stated, all comparisons in this section are conducted between LatentChem and the explicit CoT baseline (Stage 1+2+4).

4.1 Spontaneous CoT Internalization

A pivotal discovery in our experiments is the emergent behavior observed during the transition from SFT to reinforcement learning (Stage 4). Despite being heavily supervised with explicit textual CoT data in before Stage 4, LatentChem spontaneously internalized the entirety of explicit textual reasoning after GRPO training.

The phenomenon. Remarkably, despite being optimized using rewards constrained only by format adherence, validity, and correctness, the model ceased generating intermediate textual steps. Instead, the generation process follows a distinct pattern: the model performs a sequence of latent thinking steps (typically using the full 10-step budget), outputs a trivial artifact (typically a single "." or ":") as a transition token, and immediately generates the target XML tags. Two cases are presented in Figure 4, where LatentChem bypasses the verbose and often hallucinated textual plan generated by the CoT baseline, utilizing the latent space to perform accurate structural optimization.

Causal necessity verification. To confirm that this "silent" phase performs actual computation rather than acting as a passive delay, we conduct causal ablation experiments. As shown in Figure 5, replacing the initial latent steps with Gaussian noise leads to monotonic performance degradation. This rigorously establishes that the internalized latent states encode critical structural precursors and are functionally essential for the final generation.

Insight: optimization over imitation. This internalization was not explicitly enforced by any loss penalty on output length. Rather, it represents a spontaneous strategy shift driven purely by the reward signal. This observation serves as powerful empirical validation for our core hypothesis regarding the continuity–discretization gap in chemical reasoning. It suggests that when the model is granted the freedom to reason for chemical tasks, it identifies the high-dimensional, continuous latent space as a more native and structurally consistent workspace than the discrete, low-bandwidth channel of natural language.

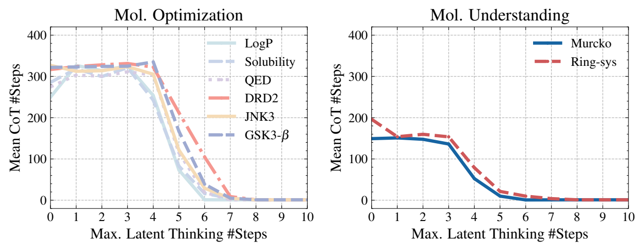

Figure 6: Budget stress testing. We monitor the length of generated explicit CoT (y-axis) as the maximum allowed latent thinking steps (x-axis) decrease. A distinct hydraulic trade-off is observed across both Molecule Optimization (left) and Understanding (right) tasks: as the latent budget is restricted (*e.g.,* < 5 steps), the model spontaneously externalizes reasoning into text to compensate for the reduced internal capacity.

4.2 Emergent Compensatory Dynamics and Causal Necessity

Building on the discovery of CoT internalization, a critical question arises: *is this behavior a rigid pattern or a flexible strategy?* To investigate this, we conduct budget stress testing on the molecule optimization and understanding tasks, monitoring the length of generated explicit CoT as the maximum allowed latent thinking steps decrease.

As illustrated in Figure 6, we observe a remarkable hydraulic trade-off. When the latent budget is sufficient (*e.g.,* $T \geq 6$), the model maintains the "internalized" state with near-zero CoT length. However, as we progressively starve the model of latent thinking steps ($T < 6$), it spontaneously reactivates explicit CoT without any external instruction. This dynamic behavior suggests that the model treats implicit and explicit reasoning as communicating vessels, revealing that LatentChem has autonomously learned a robust mechanism for arbitrating between implicit and explicit reasoning pathways based on its computational capacity.

4.3 Unveiling the Latent Manifold

To investigate the internal dynamics of the latent thinking process, we examine the evolutionary trajectories of ChemTokens across different optimization tasks. We first project the high-dimensional latent states into a 2D space using t-SNE to visualize the global distribution shift. As illustrated in Figure 7(a), the dynamics reveal a phenomenon of immediate functional partitioning rather than gradual evolution. Initially, at Step 0, representations from all tasks are densely entangled, indicating a generic, task-agnostic encoding. However, within the first two latent steps, the trajectories undergo a dramatic divergence, rapidly snapping into distinct, task-specific clusters. From Step 3 to 10, these spatial configurations remain stable. This rapid convergence suggests that the model possesses an intrinsic efficiency; it does not require a prolonged period to locate the optimization manifold but instead reconfigures the molecular representation almost instantaneously to align with the specific chemical goal.

To determine whether these drastic latent updates compromise the physical validity of the representation, we further employ representational similarity analysis (RSA) to quantify the alignment between the latent geometry and the true chemical topology. Specifically, we track the Spearman correlation between the pairwise cosine similarity of ChemTokens and the Tanimoto similarity of their corresponding molecular fingerprints. As shown in Figure 7(b), the structural correlation remains remarkably stable across all reasoning steps. It implies that the massive updates occurring in the first few steps are orthogonal to the direction of structural information. The latent thinking process acts as a lossless reservoir of structural information throughout the reasoning chain.

5 Benchmarking Latent vs. Explicit Paradigms

To substantiate that the observed latent reasoning dynamics translate into superior performance, we conduct comprehensive evaluations across four diverse chemical benchmarks.

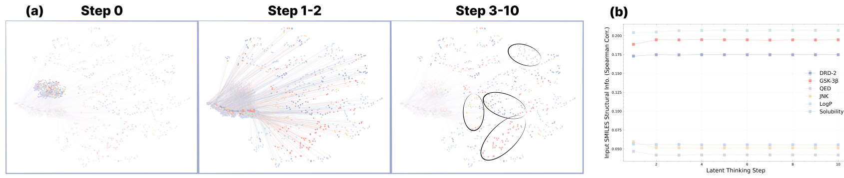

Figure 7: Visualization of latent dynamics during the reasoning process. (a) t-SNE projections of ChemTokens at different steps show a transition from initial entanglement to rapid disentanglement. The representations diverge significantly within the first two steps to form task-specific clusters and remain stable thereafter. (b) RSA results showing the correlation between latent geometry and chemical topology. The flat trajectories indicate that despite the significant spatial updates for task adaptation, the model preserves structural fidelity throughout the reasoning chain, suggesting that optimization occurs orthogonally to topological encoding.

Table 2: Main results on chemical benchmarks. We report the non-tie win rate ($\mathcal{R}^*_{win}$ ↑) of each method compared to the reference CoT baseline (Stage 1+2+4). The "All" column denotes the aggregated win rate on each benchmark. Bold and underlined values indicate the best and second-best performance, respectively. * denotes statistical significance with $p < 0.01$ according to the sign test.

Category	Method	ChemCoTBench			Mol-Instructions			ChemLLMBench			ChEBI-20
		Open	Closed	All	Open	Closed	All	Open	Closed	All	Open
Text-only	Qwen-3-8B	13.98^*	2.80^*	8.61^*	58.65^*	21.87^*	23.97^*	66.67^*	19.58^*	22.78^*	74.86^*
	Qwen-3-8B (SFT)	28.08^*	20.63^*	25.75^*	$\mathbf{73.13}^*$	35.92^*	38.39^*	$\underline{83.14}^*$	37.51^*	41.11^*	$\underline{80.85}^*$
Chem. LLMs	Stage 1	$\underline{43.84}^*$	$\underline{36.76}^*$	$\underline{41.73}^*$	63.00^*	$\mathbf{50.49}$	$\mathbf{51.25}^*$	68.47^*	$\underline{50.56}$	$\underline{51.57}$	64.80^*
	Stage 1+2	38.35^*	28.44^*	35.50^*	50.36	43.69^*	44.10^*	59.77	45.12^*	46.21^*	54.60^*
Latent Chem. LLMs	Coconut-Chem	34.71^*	26.32^*	32.11^*	57.94^*	39.45^*	40.57^*	75.00^*	42.46^*	44.82^*	65.58^*
	LatentChem (Ours)	$\mathbf{63.73}^*$	$\mathbf{50.00}$	$\mathbf{59.88}^*$	$\underline{72.00}^*$	$\underline{48.36}^*$	$\underline{49.88}$	$\mathbf{87.37}^*$	**52.87**	$\mathbf{55.58}^*$	$\mathbf{85.26}^*$

5.1 Experimental Setup

Task taxonomy. We categorize the target capabilities into two paradigms based on the solution space. Open-ended generative tasks (*e.g.,* molecule optimization) lack a unique optimal solution, requiring the model to explore the chemical manifold to generate valid candidates. In contrast, precise closed-ended tasks (*e.g.,* reaction prediction) demand unique, deterministic answers, functioning as rigorous mapping problems. We leverage this dichotomy to investigate the differential impact of latent reasoning on creative exploration versus deterministic execution.

Benchmarks. We evaluate our method on four established benchmarks: ChemCoTBench Li et al. (2025a), Mol-Instructions Fang et al. (2023), ChEBI-20 Edwards et al. (2021), and ChemLLMBench Guo et al. (2023). To rigorously assess the molecular reasoning and generation capabilities targeted by our framework, the evaluation suite is curated to focus on chemical competencies aligned with the instruction-tuning phase. Consequently, for ChemCoTBench, the evaluation covers 1,120 samples covering molecule optimization, understanding, editing, and reaction prediction. The molecule-oriented component of Mol-Instructions provides 4,000 samples across forward reaction prediction, retrosynthesis, reagent prediction, and description generation. ChEBI-20 contributes its standard test set of 3,297 molecule-description pairs. Finally, 600 samples from ChemLLMBench are included, spanning seven relevant subtasks, including molecule captioning and various reaction prediction challenges, to verify performance consistency across diverse evaluation sources.

Baselines. We compare LatentChem against three categories of baselines. All molecule-aware models (LatentChem, explicit chemical LLMs, and Coconut-Chem) share the identical Qwen-3-8B Yang et al. (2025) backbone equipped with the SMI-TED encoder and ChemAdapter. The categories include: (1) text-only LLMs (Qwen-3-8B and Qwen-3-8B SFT) which establish a performance floor; (2) explicit chemical LLMs, which utilize the exact same ChemAdapter+Qwen-3-8B neural architecture as LatentChem but are constrained to standard CoT reasoning (reported at Stage 1, Stage 1+2, and the GRPO-optimized Stage 1+2+4); and (3) generic latent models, specifically Coconut-Chem, which adapts the Coconut paradigm Hao et al. (2025) to our encoder-adapter setup.

Metrics. We employ diverse task-specific metrics (detailed in Appendix E). To facilitate a rigorous statistical comparison, we employ the sign test Dixon and Mood (1946). Aligning with this framework's theoretical premise that ties are non-informative, we quantify performance using the non-tie win rate ($\mathcal{R}^*_{win}$), which is computed as: $\mathcal{R}^*_{win} = \frac{N_{win}}{N_{win}+N_{loss}}$.

Table 3: Detailed performance on open-ended generative tasks. Metrics include property improvement ($\Delta\uparrow$) and success rate (SR%$\uparrow$) for molecule optimization, alongside METEOR$\uparrow$ scores for description. Results for closed-ended tasks are detailed in Appendices F.

Category	Method	Molecule Optimization (ChemCoTBench)												Molecule Description		
		LogP		Solubility		QED		DRD2		JNK3		GSK3-β		Mol-Instr.	ChEBI-20	ChemLLM.
		Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	METEOR	METEOR	METEOR
Text-only	Qwen-3-8B	0.00	3	0.00	4	0.00	4	0.00	4	-0.01	0	0.00	2	0.09	0.12	0.09
	Qwen-3-8B (SFT)	0.15	47	0.48	52	0.10	48	0.04	38	-0.02	20	0.02	36	**0.12**	0.12	0.13
Chem. LLMs	Stage 1	0.35	61	0.87	78	0.14	73	0.20	66	0.02	40	0.13	61	0.10	0.08	0.11
	Stage 1+2	0.51	60	0.82	69	0.14	72	0.12	53	0.01	32	0.08	39	0.07	0.05	0.06
	Stage 1+2+4	0.67	77	1.17	86	**0.19**	82	0.19	70	0.03	44	0.13	67	0.07	0.05	0.07
Latent Chem. LLMs	Coconut-Chem	0.17	44	0.57	58	0.15	67	0.04	45	-0.00	35	0.06	47	0.09	0.07	0.04
	LatentChem (Ours)	**1.37**	**96**	**1.53**	**89**	0.18	**83**	**0.26**	**74**	**0.08**	**60**	**0.17**	**82**	**0.12**	**0.15**	**0.16**

5.2 Main Results

Overall superiority. Table 2 reports the non-tie win rate ($\mathcal{R}^*_{win}$) against the strong explicit baseline (Stage 1+2+4). LatentChem demonstrates robust superiority, dominating on ChemCoTBench (**59.88%**) and ChEBI-20 (**85.26%**), while generalizing well to ChemLLMBench (**55.58%**). On Mol-Instructions, it maintains competitive parity (**49.88%**). These results confirm that the latent reasoning paradigm offers a superior substrate for complex chemical logic, enhancing capabilities without compromising foundational proficiency on standard tasks.

Generative capabilities. The advantages of the latent paradigm are most pronounced in open-ended generative tasks (Table 3), surpassing all baselines on 14 out of 15 metrics. In molecule optimization, reasoning in the continuous latent manifold establishes a substantial lead over discrete planning. Furthermore, the 8B-parameter LatentChem consistently outperforms the state-of-the-art results from Claude 3.7 Sonnet reported in ChemCoTBench (see Appendix J for detailed comparison). Notably, on the challenging GSK3-β task, LatentChem achieves a success rate of **82%** (vs. 67% for explicit CoT). This validates that high-dimensional latent exploration unlocks a level of "generative creativity" and navigational precision that is inaccessible to explicit token-based methods.

Robustness on deterministic mappings. The performance contrast between task categories offers critical insight into the mechanism of latent thinking. As shown in the "Closed" columns of Table 2, while the latent approach maintains a leading position, the margin is narrower compared to open-ended tasks. This aligns with the nature of closed-ended tasks as deterministic mapping problems, where the benefits of creative exploration are naturally saturated. Crucially, however, internalized reasoning preserves foundational precision, allowing the system to maintain competitive parity with explicit baselines even on these rigid tasks, effectively bypassing the typical trade-off between creativity and accuracy.

Inference efficiency. Beyond capability, we quantify efficiency by contrasting the total reasoning overhead (baseline CoT tokens vs. LatentChem's combined latent and textual steps). As shown in Figure 8, by compressing verbose linguistic thought, LatentChem reduces this overhead by factors ranging from **5.4**$\times$ to **29.9**$\times$ (average **10.84**$\times$). This confirms that the latent paradigm is not only effectively superior but also computationally transformative, breaking the inference latency bottleneck inherent to traditional CoT.

We attribute this dramatic acceleration to the inherent compressibility of chemical logic in continuous space compared to language. Consider a routine optimization step, such as appending a functional group. In the latent manifold, the model can theoretically execute this structural shift via a single or few vector transitions, effectively "gliding" across the smooth landscape visualized in Figure 1 (b). Conversely, explicit CoT is forced to discretize this operation into a verbose textual chain (e.g., analyzing binding sites, verifying valency, and describing the addition), triggering dozens of redundant autoregressive forward passes to achieve the exact same chemical move (the jagged "staircase" trajectory in Figure 1 (c)). By bypassing this language bottleneck, LatentChem aligns computational expenditure with the actual complexity of the chemical task. Further analysis is presented in Appendix A.

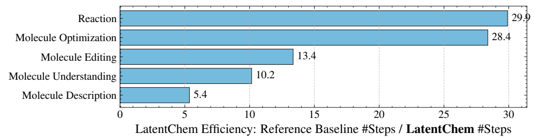

Figure 8: Reasoning-overhead reduction. We report the efficiency ratio (Explicit CoT Steps / LatentChem Steps) across all five task categories. LatentChem achieves up to **29.9**× efficiency on reaction tasks, demonstrating that latent thinking significantly accelerates inference by bypassing verbose textual generation.

5.3 Ablation Study

To isolate the drivers of performance, we conduct a component-wise ablation (Table 4). Removing Latent Thinking causes a precipitous performance drop, confirming that continuous vectors capture chemical nuances far more effectively than discrete tokens. Furthermore, the removal of ChemUpdater leads to a 12.0% decline in optimization success, underscoring that dynamic perception is essential for chemical reasoning. Finally, the Latent Projector proves indispensable for bridging the semantic gap; without it, the model fails to map internal thought vectors back to the input space, disrupting the reasoning trajectory.

Table 4: Ablation study. We report the success rate (SR%) for molecule optimization and METEOR scores for molecule description across all applicable tasks from the four evaluated benchmarks.

Method	Mol. Optimization (SR%↑)	Mol. Description (METEOR↑)
w/o Latent Thinking	71.00	0.052
w/o Latent Projector	69.83	0.087
w/o ChemUpdater	68.67	0.068
LatentChem (Full)	**80.67**	**0.143**

6 Discussion and Conclusion

This study investigates the hypothesis of continuity–discretization gap in chemical reasoning, proposing that the discontinuity of natural language inhibits the modeling of continuous chemical manifolds. By introducing LatentChem, we demonstrate that decoupling reasoning from linguistic generation offers a superior computational substrate for chemical logic. Our analysis reveals the mechanism behind this success: the model exhibits spontaneous internalization, voluntarily abandoning verbose textual derivations for compact latent computation that correlates with physical chemical topology. This structural advantage translates directly into performance, with LatentChem achieving a **59.88%** non-tie win rate against strong explicit baselines on ChemCoTBench and a **10.84**× efficiency gain, effectively breaking the trade-off between reasoning depth and inference latency.

Limitations and future directions. The transition to latent reasoning changes the interface through which intermediate reasoning is exposed. Because computation unfolds in continuous latent space rather than explicit tokens, intermediate steps are not automatically available as human-readable traces. Although we provide empirical analyses of the latent dynamics, the current framework does not yet offer a native mechanism for systematically externalizing these internal trajectories in symbolic form. In addition, our experiments are conducted primarily on a Qwen-3-8B backbone and SMI-TED Light, so their extent of transfer across other backbone families, model scales, and encoder choices remains to be established. Future work should focus on hybrid cognitive architectures Kahneman (2011): systems that leverage efficient latent thinking for heavy structural computation (System 1) while retaining the ability to decode these thoughts into explicit language when justification is required (System 2). Another promising direction is to explore more efficient latent-space navigation strategies that can capitalize on the rapid early organization of representations and reduce the overall number of thinking steps without sacrificing performance. Ultimately, this work establishes latent thinking not merely as an acceleration trick, but as a foundational reasoning modality for the next generation of scientific AI.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Preslav Aleksandrov, Meghdad Kurmanji, Fernando Garcia Redondo, David O'Shea, William Shen, Alex Iacob, Lorenzo Sani, Xinchi Qiu, Nicola Cancedda, and Nicholas D. Lane. Abbie: Autoregressive block-based iterative encoder for efficient sequence modeling, 2025. URL https://arxiv.org/abs/2507.08567.

Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, et al. Intern-s1: A scientific multimodal foundation model. *arXiv preprint arXiv:2508.15763*, 2025.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Haolin Chen, Yihao Feng, Zuxin Liu, Weiran Yao, Akshara Prabhakar, Shelby Heinecke, Ricky Ho, Phil Mui, Silvio Savarese, Caiming Xiong, and Huan Wang. Language models are hidden reasoners: Unlocking latent reasoning capabilities via self-rewarding, 2024. URL https://arxiv.org/abs/2411.04282.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations, 2024. URL https://arxiv.org/abs/2412.13171.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step, 2024. URL https://arxiv.org/abs/2405.14838.

Wilfrid J Dixon and Alexander M Mood. The statistical sign test. *Journal of the American Statistical Association*, 41(236):557–566, 1946.

Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6): 1273–1280, 2002.

Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, 2021.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*, 2023.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach, 2025. URL https://arxiv.org/abs/2502.05171.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688, 2023.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space, 2025. URL https://arxiv.org/abs/2412.06769.

Kan Hatakeyama-Sato, Naoki Yamane, Yasuhiko Igarashi, Yuta Nabae, and Teruaki Hayakawa. Prompt engineering of gpt-4 for chemical research: what can/cannot be done? *Science and Technology of Advanced Materials: Methods*, 3(1):2260300, 2023.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, et al. Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. *arXiv preprint arXiv:2502.09621*, 2025a.

Lei Jiang, Shuzhou Sun, Biqing Qi, Yuchen Fu, Xiaohua Xu, Yuqiang Li, Dongzhan Zhou, and Tianfan Fu. Chem3dllm: 3d multimodal large language models for chemistry. *arXiv preprint arXiv:2508.10696*, 2025b.

Daniel Kahneman. Thinking, fast and slow. *Farrar, Straus and Giroux*, 2011.

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, Sriniker, Gedeck, Gareth Jones, Nadine Schneider, Eisuke Kawashima, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Vincent F. Scalfani, Rachel Walker, Kazuya Ujihara, Daniel Probst, Guillaume Godin, Axel Pahl, Tadhurst-cdd, Juuso Lehtivarjo, Francois Berenger, and Jason D Biggs. Rdkit: Open-source cheminformatics and machine learning. Software, May 2024. Open-source cheminformatics and machine learning.

Hao Li, He Cao, Bin Feng, Yanjun Shao, Xiangru Tang, Zhiyuan Yan, Li Yuan, Yonghong Tian, and Yu Li. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *arXiv preprint arXiv:2505.21318*, 2025a.

Hao Li, Liuzhenghao Lv, He Cao, Zijing Liu, Zhiyuan Yan, Yu Wang, Yonghong Tian, Yu Li, and Li Yuan. How to detect and defeat molecular mirage: A metric-driven benchmark for hallucination in llm-based molecular comprehension, 2025b. URL `https://arxiv.org/abs/2504.12314`.

Junxian Li, Di Zhang, Xunzhi Wang, Zeying Hao, Jingdi Lei, Qian Tan, Cai Zhou, Wei Liu, Yaotian Yang, Xinrui Xiong, et al. Chemvlm: Exploring the power of multimodal large language models in chemistry area. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 415–423, 2025c.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

Luyang Liu, Jonas Pfeiffer, Jiaxing Wu, Jun Xie, and Arthur Szlam. Deliberation in latent space via differentiable cache augmentation, 2024. URL `https://arxiv.org/abs/2412.17747`.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*, 2023.

Liuzhenghao Lv, Hao Li, Yu Wang, Zhiyuan Yan, Zijun Chen, Zongying Lin, Li Yuan, and Yonghong Tian. Navigating chemical-linguistic sharing space with heterogeneous molecular encoding, 2025. URL `https://arxiv.org/abs/2412.20888`.

Siddharth M Narayanan, James D Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G Rodriques, and Andrew D White. Training a scientific reasoning model for chemistry. *arXiv preprint arXiv:2506.17238*, 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Qizhi Pei, Zhimeng Zhou, Kaiyuan Gao, Jinhua Zhu, Yue Wang, Zun Wang, Tao Qin, Lijun Wu, and Rui Yan. Leveraging biomolecule and natural language through multi-modal learning: A survey. *arXiv preprint arXiv:2403.01528*, 2024.

Malik Sallam, Khaled Al-Salahat, Huda Eid, Jan Egger, and Behrus Puladi. Human versus artificial intelligence: Chatgpt-4 outperforming bing, bard, chatgpt-3.5 and humans in clinical chemistry multiple-choice questions. *Advances in Medical Education and Practice*, pages 857–871, 2024.

Nadine Schneider, Roger A Sayle, and Gregory A Landrum. Get your atoms in order—an open-source implementation of a novel and robust molecular canonicalization algorithm. *Journal of chemical information and modeling*, 55(10):2111–2120, 2015.

Hongyu Shan, Mingyang Song, Chang Dai, Di Liang, and Han Chen. R-capsule: Compressing high-level plans for efficient large language model reasoning, 2025. URL `https://arxiv.org/abs/2509.22131`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Eduardo Soares, Emilio Vital Brazil, Victor Yukio Shirasuna, Dmitry Zubarev, Renato Cerqueira, and Kristin Schmidt. Smi-ted: A large-scale foundation model for materials and chemistry. 2024.

Qian Tan, Dongzhan Zhou, Peng Xia, Wanhao Liu, Wanli Ouyang, Lei Bai, Yuqiang Li, and Tianfan Fu. Chemmllm: Chemical multimodal large language model. *arXiv preprint arXiv:2505.16326*, 2025.

Xiangru Tang, Tianyu Hu, Muyang Ye, Yanjun Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou, Pan Lu, Zhuosheng Zhang, Yilun Zhao, et al. Chemagent: Self-updating library in large language models improves chemical reasoning. *arXiv preprint arXiv:2501.06590*, 2025a.

Xiangru Tang, Zhuoyun Yu, Jiapeng Chen, Yan Cui, Daniel Shao, Weixu Wang, Fang Wu, Yuchen Zhuang, Wenqi Shi, Zhi Huang, et al. Cellforge: agentic design of virtual cell models. *arXiv preprint arXiv:2508.02276*, 2025b.

Jianwei Wang, Ziming Wu, Fuming Lai, Shaobing Lian, and Ziqian Zeng. Synadapt: Learning adaptive reasoning in large language models via synthetic continuous chain-of-thought, 2025a. URL `https://arxiv.org/abs/2508.00574`.

Jiaqi Wang, Binquan Ji, Haibo Luo, Yiyang Qi, Ruiting Li, Huiyan Wang, Yuantao Han, Cangyi Yang, jiaxu Zhang, and Feiliang Ren. Lta-thinker: Latent thought-augmented training framework for large language models on complex reasoning, 2025b. URL `https://arxiv.org/abs/2509.12875`.

Weida Wang, Benteng Chen, Di Zhang, Wanhao Liu, Shuchen Pu, Ben Gao, Jin Zeng, Xiaoyong Wei, Tianshu Yu, Shuzhou Sun, et al. Chem-r: Learning to reason as a chemist. *arXiv preprint arXiv:2510.16880*, 2025c.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. Sim-cot: Supervised implicit chain-of-thought, 2025. URL `https://arxiv.org/abs/2509.20317`.

Hao Wen, Yifan Su, Feifei Zhang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. Parathinker: Native parallel thinking as a new paradigm to scale llm test-time compute, 2025. URL `https://arxiv.org/abs/2509.04475`.

Haoyi Wu, Zhihao Teng, and Kewei Tu. Parallel continuous chain-of-thought with jacobi iteration, 2025. URL `https://arxiv.org/abs/2506.18582`.

Yingce Xia, Peiran Jin, Shufang Xie, Liang He, Chuan Cao, Renqian Luo, Guoqing Liu, Yue Wang, Zequn Liu, Yuan-Jyue Chen, et al. Nature language model: Deciphering the language of nature for scientific discovery. *arXiv preprint arXiv:2502.07527*, 2025.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms, 2025a. URL `https://arxiv.org/abs/2502.12134`.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot++: Test-time scaling with soft chain-of-thought reasoning, 2025b. URL `https://arxiv.org/abs/2505.11484`.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024.

Yu Zhang, Yang Han, Shuai Chen, Ruijie Yu, Xin Zhao, Xianbin Liu, Kaipeng Zeng, Mengdi Yu, Jidong Tian, Feng Zhu, et al. Large language models to accelerate organic chemistry synthesis. *Nature Machine Intelligence*, pages 1–13, 2025.

Zihan Zhao, Bo Chen, Ziping Wan, Lu Chen, Xuanze Lin, Shiyang Yu, Situo Zhang, Da Ma, Zichen Zhu, Danyang Zhang, et al. Chemdfm-r: An chemical reasoner llm enhanced with atomized chemical knowledge. *arXiv e-prints*, pages arXiv–2507, 2025a.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, et al. Developing chemdfm as a large language foundation model for chemistry. *Cell Reports Physical Science*, 6(4), 2025b.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Emergence of superposition: Unveiling the training dynamics of chain of continuous thought, 2025. URL https://arxiv.org/abs/2509.23365.

Jiaxi Zhuang, Yaorui Shi, Jue Hou, Yunong He, Mingwei Ye, Mingjun Xu, Yuming Su, Linfeng Zhang, Ying Qian, Guolin Ke, et al. Reasoning-enhanced large language models for molecular property prediction. *arXiv preprint arXiv:2510.10248*, 2025.

Chenyang Zuo, Siqi Fan, and Zaiqing Nie. Biomedgpt-mol: Multi-task learning for molecular understanding and generation. *arXiv preprint arXiv:2512.04629*, 2025.

A Efficiency Analysis

To analyze the efficiency gains reported in the main text, we formalize reasoning as a trajectory discretization problem. Crucially, curvature-limited discretization arises in the *representation space* of a modality (where discrete steps are taken).

A.1 Preliminaries and Assumptions

Definition A.1 (Chemical Manifold)**.** Let $\mathcal{M} = (\mathcal{S}, g)$ be a smooth d-dimensional Riemannian manifold representing the chemical state space, where g encodes physicochemical properties.

Definition A.2 (Optimal Reasoning Path)**.** The logical derivation from a problem state x_{start} to a solution x_{end} is modeled as a unit-speed geodesic $\gamma : [0, L] \to \mathcal{M}$, parameterized by manifold arc length s, such that $\nabla_{\dot{\gamma}}\dot{\gamma} = 0$ and $\|\dot{\gamma}(s)\|_g = 1$.

Assumption A.3 (Perfect Manifold Learning)**.** We assume an idealized scenario where both the Explicit CoT model and the LatentChem model attempt to approximate the same optimal path γ. This isolates modality-dependent efficiency limits from model capacity and optimization.

Definition A.4 (Modality Representation Trajectory)**.** Each reasoning modality mod induces a differentiable representation map $\Phi_{\mathrm{mod}} : \mathcal{M} \to \mathbb{R}^{m_{\mathrm{mod}}}$. Along the optimal path, the modality operates on the represented trajectory

$$c_{\mathrm{mod}}(s) := \Phi_{\mathrm{mod}}(\gamma(s)) \subset \mathbb{R}^{m_{\mathrm{mod}}}.$$

Definition A.5 (Valid Discrete Approximation)**.** A discrete reasoning chain is specified by a partition $0 = s_0 < s_1 < \cdots < s_N = L$ and induces a polygonal approximation of c_{mod} by connecting consecutive endpoints $c_{\mathrm{mod}}(s_k)$ and $c_{\mathrm{mod}}(s_{k+1})$ with a line segment in $\mathbb{R}^{m_{\mathrm{mod}}}$. The approximation is *valid* if for each segment

$$\sup_{s\in[s_k,s_{k+1}]} \mathrm{dist}\Big(c_{\mathrm{mod}}(s),\, \overline{c_{\mathrm{mod}}(s_k)c_{\mathrm{mod}}(s_{k+1})}\Big) \le \delta,$$

where dist is Euclidean distance and $\delta > 0$ is a fixed tolerance.

Definition A.6 (Effective Curvature in Representation Space)**.** Let $\tilde{s}$ denote arc-length along the represented curve c_{mod} in $\mathbb{R}^{m_{\mathrm{mod}}}$. Assuming c_{mod} is C^2, define curvature along the path by

$$\kappa_{\mathrm{mod}}(s) := \left\| \frac{d^2 c_{\mathrm{mod}}}{d\tilde{s}^2}(s) \right\|.$$

Definition A.7 (Intrinsic Semantic Resolution (Path-Advance))**.** The intrinsic semantic resolution of modality mod is the maximum *advance along the optimal path* that a single discrete step can make:

$$s_{k+1} - s_k \le \rho_{\mathrm{mod}} \qquad \text{for all } k. \tag{1}$$

Equivalently, one step can traverse at most ρ_{mod} units of manifold arc-length along γ.

Assumption A.8 (Non-degenerate Representation Speed)**.** Along the optimal path, the representation does not collapse locally: there exists $v_{\min} > 0$ such that

$$\|c'_{\mathrm{mod}}(s)\| \ge v_{\min} \qquad \text{for all } s \in [0, L]. \tag{2}$$

A.2 The Curvature–Resolution Theorem

Theorem A.9 (Curvature–Resolution Trade-off (Qualitative Lower Bound))**.** *Let ρ_{mod} be the intrinsic semantic resolution (Definition A.7), and let $\kappa_{\mathrm{mod}}(s)$ be the effective curvature of $c_{\mathrm{mod}}(s) = \Phi_{\mathrm{mod}}(\gamma(s))$ in representation space. Under Assumption A.8, any valid discrete approximation requires at least*

$$N \;\ge\; \int_0^L \max\left(\frac{1}{\rho_{\mathrm{mod}}},\, v_{\min}\sqrt{\frac{\kappa_{\mathrm{mod}}(s)}{8\delta}}\right) ds. \tag{3}$$

Consequently, the efficiency ratio between LatentChem and Explicit CoT can be estimated by

$$\eta = \frac{N_{txt}}{N_{lat}} \;\approx\; \frac{\int_0^L \max\left(\frac{1}{\rho_{txt}},\, v_{\min}\sqrt{\frac{\kappa_{txt}(s)}{8\delta}}\right) ds}{\int_0^L \max\left(\frac{1}{\rho_{lat}},\, v_{\min}\sqrt{\frac{\kappa_{lat}(s)}{8\delta}}\right) ds}. \tag{4}$$

A.3 PROOF OF THEOREM A.9

Step 1: Geometric constraint in representation arc length. Fix an interval $[s_k, s_{k+1}]$ and let its manifold-arc-length width be $\Delta s_k := s_{k+1} - s_k$. Let the representation-space arc length of the curve segment be

$$\tilde{\ell}_k := \int_{s_k}^{s_{k+1}} \|c'_{\text{mod}}(u)\| \, du.$$

Define $\kappa_k := \sup_{u \in [s_k, s_{k+1}]} \kappa_{\text{mod}}(u)$. A standard chord–arc deviation (sagitta) bound for C^2 curves, stated in terms of representation arc length, gives

$$\sup_{s \in [s_k, s_{k+1}]} \text{dist}\Big(c_{\text{mod}}(s), \, \overline{c_{\text{mod}}(s_k) c_{\text{mod}}(s_{k+1})}\Big) \; \leq \; \frac{1}{8} \, \kappa_k \, \tilde{\ell}_k^2. \tag{5}$$

Validity implies $\tilde{\ell}_k \leq \sqrt{\frac{8\delta}{\kappa_k}}$.

Step 2: Converting to a bound on manifold-arc-length advance. By Assumption A.8, $\tilde{\ell}_k \geq v_{\min} \Delta s_k$, hence

$$\Delta s_k \leq \frac{1}{v_{\min}} \sqrt{\frac{8\delta}{\kappa_k}}. \tag{6}$$

Independently, Definition A.7 gives the modality advance constraint

$$\Delta s_k \leq \rho_{\text{mod}}. \tag{7}$$

Combining equation 6–equation 7 yields

$$\Delta s_k \leq \min \left(\rho_{\text{mod}}, \, \frac{1}{v_{\min}} \sqrt{\frac{8\delta}{\kappa_k}} \right).$$

Step 3: Aggregation into a global lower bound. Define

$$w(s) := \max \left(\frac{1}{\rho_{\text{mod}}}, v_{\min} \sqrt{\frac{\kappa_{\text{mod}}(s)}{8\delta}} \right).$$

On each interval $[s_k, s_{k+1}]$, we have $\kappa_{\text{mod}}(s) \leq \kappa_k$, so

$$w(s) \leq \max \left(\frac{1}{\rho_{\text{mod}}}, v_{\min} \sqrt{\frac{\kappa_k}{8\delta}} \right) \quad \text{for all } s \in [s_k, s_{k+1}].$$

Therefore,

$$\int_{s_k}^{s_{k+1}} w(s) \, ds \leq \Delta s_k \cdot \max \left(\frac{1}{\rho_{\text{mod}}}, v_{\min} \sqrt{\frac{\kappa_k}{8\delta}} \right) \leq 1,$$

and summing over k yields $\int_0^L w(s) \, ds \leq N$, proving equation 3. $\square$

A.4 EFFICIENCY SCALING VIA MANIFOLD RECTIFICATION

Theorem A.9 implies a qualitative trade-off: to keep chord deviation within tolerance δ, high representation-space curvature forces smaller advances Δs_k (hence more steps), while a rectified representation reduces curvature and permits coarser discretization.

Latent Space ($\kappa_{lat} \to 0$). If representation learning rectifies the latent trajectory so that $\kappa_{lat}(s) \approx 0$, then the curvature term vanishes and

$$N_{lat} \gtrsim \int_0^L \frac{1}{\rho_{lat}} \, ds = \frac{L}{\rho_{lat}}.$$

Explicit CoT Space (curvature remains large). If the text-space representation retains high curvature $\kappa_{txt}(s)$, then for sufficiently complex tasks the curvature term can dominate, leading to

$$N_{txt} \gtrsim \frac{1}{v_{\min,txt}} \int_0^L \sqrt{\frac{\kappa_{txt}(s)}{8\delta}} \, ds,$$

and hence a larger step count than the rectified latent case. This supports the qualitative hypothesis that LatentChem's advantage grows with task complexity when it reduces representation-space curvature.

B Benchmark Details and Task Definition

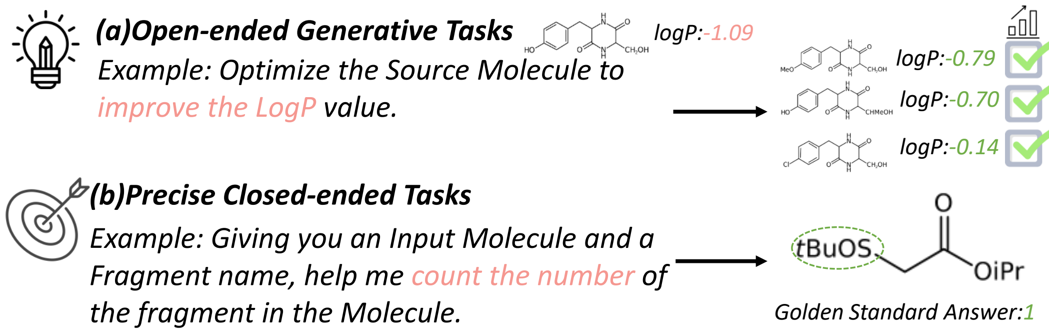

Figure 9: Taxonomy of chemical reasoning tasks. The evaluated tasks consist of two categories: (a) open-ended generative tasks (*e.g.,* molecule optimization), characterized by the non-determinism of the optimal solution; (b) precise closed-ended tasks (*e.g.,* fragment counting), which demand a unique, deterministic standard answer.

B.1 ChemCoTBench

ChemCoTBench is a step-by-step, application-oriented, and high-quality benchmark for assessing LLM reasoning in chemical applications. ChemCoTBench comprises molecule understanding, molecule editing, molecule optimization, and reaction prediction, covering 22 subtasks and 1,495 examples. These samples are derived from authoritative public databases (*e.g.,* PubChem, ChEMBL, ZINC) and patent corpora, carefully reviewed by the combined effort of LLM and 13 chemists for chemical validity and diversity.

Molecule understanding. As shown in Figure 10(a), we evaluate models on progressively more demanding levels of structural comprehension. First, models must recognise and count two fundamental molecular features: (i) functional groups (FGs), which are atom clusters that largely determine a molecule's physicochemical properties and reactivity; and (ii) rings, which impose fixed conformations and serve as common building blocks in drug design, crystal engineering, and polymer synthesis. Accurate identification and counting of functional groups and rings requires both lexical and syntactic understanding of SMILES and remains challenging for LLMs that lack explicit topological awareness. Second, we assess recognition of higher-order scaffolds: Murcko scaffolds—obtained by systematically removing side chains and used for structural analysis in medicinal chemistry—and ring systems, including fused and bridged motifs that present substantial synthetic challenges. These tasks probe hierarchical and topology-aware representations. Finally, we introduce SMILES-equivalence tasks (permutations and mutations) to test whether models can identify chemically equivalent structures despite surface-level SMILES variability, thereby evaluating robustness to representational changes.

Molecule editing. This suite (Figure 10(b)) measures whether models can execute basic molecular-editing operations (add, delete, substitute functional groups) when guided by natural-language instructions. These operations are analogous to elementary arithmetic in mathematics and form the primitive actions required for more complex design workflows. Many optimization and synthesis problems can be decomposed into sequences of such edits; thus this task evaluates two core capabilities: (i) preserving chemical validity after each edit, and (ii) correctly applying the requested modification.

Molecule optimization. Here we test whether models can propose structural modifications that improve a specified target (Figure 10(c)). We consider two classes of objectives. At the physicochemical level, targets include LogP, solubility, and QED to assess improvements in drug-likeness. At the target-specific level, we consider binding-affinity–driven objectives for receptors such as DRD2, GSK3β, and JNK3, which require reasoning about drug–target interactions. Success on these tasks requires not only syntactic parsing of molecular structure but also inference about how local and global modifications affect complex chemical and biological properties.

Molecule description. This task (Figure 10(d)) evaluates whether models can generate accurate and informative natural-language descriptions of molecular structures. Given a molecular representation

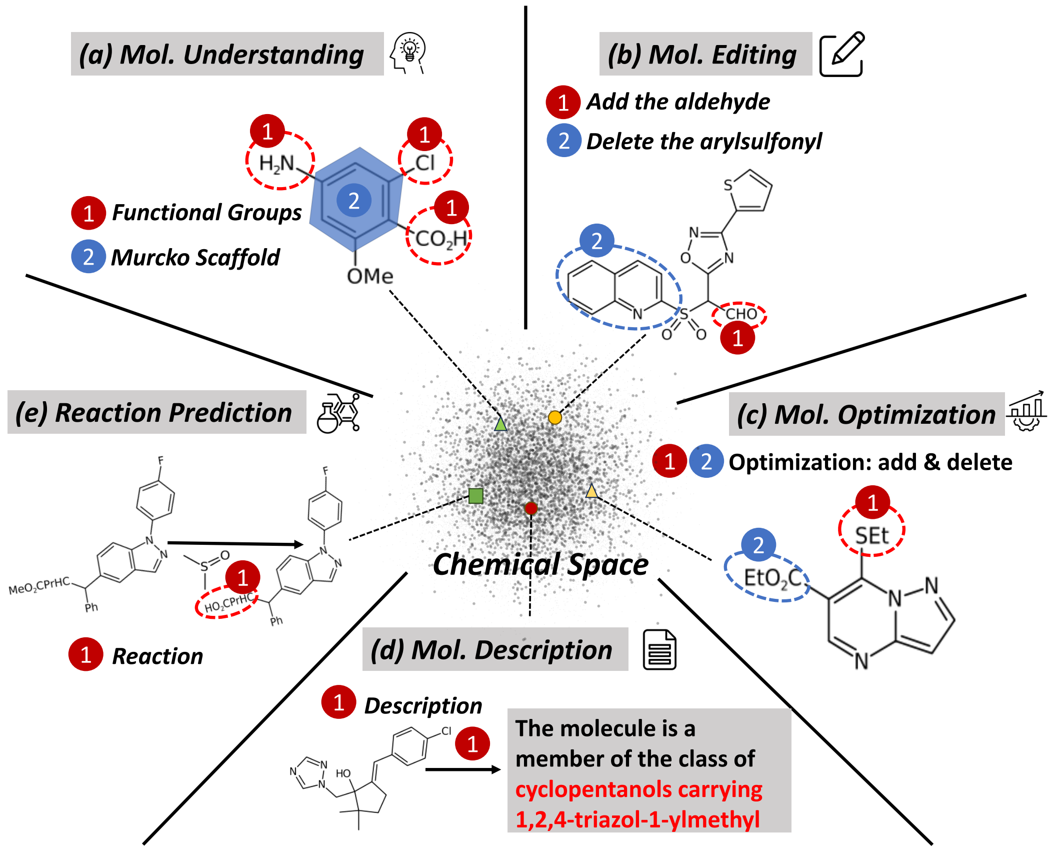

Figure 10: Task overview of ChemCoTBench

(*e.g.,* SMILES or graph-based encoding), models are required to describe key chemical features such as functional groups, ring systems, heteroatom composition, stereochemistry, and overall structural motifs, and in some cases relate these features to physicochemical or biological properties. Successful performance demands precise parsing of molecular structure, abstraction of salient substructures, and coherent translation of symbolic chemical information into human-readable descriptions, reflecting a deep alignment between molecular representation and chemical semantics.

Reaction prediction. As shown in Figure 10(e), this set of tasks evaluates chemical reasoning across multiple granularities: (i) *Forward prediction*: given reactants and reagents, predict the major products and plausible by-products, reflecting knowledge of reactivity, selectivity, and thermodynamic/kinetic considerations; (ii) *Single-step retrosynthesis*: propose reactants and disconnections for a target product, identifying key bond cleavages and feasible transformations; (iii) *Reaction-condition recommendation*: suggest catalysts, solvents, and reagents appropriate for a specified transformation, accounting for solvent effects and catalyst mechanisms that influence yield and selectivity; and (iv) *Mechanistic understanding*: include next-elementary-step prediction (intermediate species and their formation) and mechanism-route selection (choose the most plausible pathway among alternatives). Together, these tasks span coarse product-level prediction to fine-grained mechanistic reasoning, providing a comprehensive benchmark of an LLM's ability to act as a chemical-reasoning agent.

B.2 MOL-INSTRUCTIONS

Mol-Instructions is a large instruction dataset for biomolecular tasks. It is organized into three components—molecule-oriented , protein-oriented, and biomolecular-text; data were derived from various licensed biochemical resources and generated via template conversion and human–AI augmentation. Among all of the three components, only the molecule-oriented subset is related to our work.

Molecular description generation. Molecular description generation evaluates the model's high-level comprehension of molecules. Given a molecule's SMILES string, the model is required to infer the molecule's structure, properties, biological activities, and potential applications. This task is evaluated using standard natural-language metrics (*e.g.,* BLEU, ROUGE, METEOR).

Description-guided molecule generation. This task requires the model to design new molecules from natural-language descriptions. Given descriptions of desired structural features, properties, biological activity, or applications, the model should generate the corresponding molecular SMILES representation.

Property prediction tasks. Property prediction refers to the inference or estimation of a molecule's intrinsic physicochemical properties using information derived from its structural features. This task supports the high-throughput assessment of a wide range of molecular properties, thereby enabling efficient virtual screening of compound libraries. Moreover, it allows for the prediction of previously unknown properties of novel molecules, enhancing research productivity and substantially shortening development timelines.

Reaction prediction tasks. Reaction prediction tasks comprise three independent problems: forward reaction prediction, retrosynthesis, and reagent prediction. In *forward reaction prediction*, the model is given reactant and reagent SMILES and must predict the major product SMILES. In *retrosynthesis*, the model is given a product SMILES and must predict plausible reactant SMILES. In *reagent prediction*, the model is given both reactant and product SMILES and must propose suitable catalysts, solvents, or ancillary substances required to effect the specified chemical reaction.

B.3 CHEBI-20

ChEBI-20 was constructed by extracting PubChem compound records annotated with ChEBI and pairing each compound with its ChEBI textual description. To increase informational content and exclude trivial entries, only records whose descriptions exceed 20 tokens were retained. Compounds whose SMILES could not be parsed by RDKit were removed to ensure consistent molecular representations. The final curated dataset contains 33,010 molecule–text pairs and is referred to as ChEBI-20, which has been commonly used as a benchmark for text–molecule retrieval, molecule captioning, and text-guided molecular generation.

Molecule caption generation. We evaluate models on *molecule caption generation* to assess molecular understanding. Given a molecular SMILES string, the model must generate a detailed natural-language description that accurately reports salient structural features, functional groups, and chemically relevant properties present in the input. Evaluation combines standard natural-language quality metrics (BLEU, ROUGE, METEOR).

B.4 CHEMLLMBENCH

ChemLLMBench is a comprehensive benchmark that evaluates large language models on eight practical chemistry tasks—including name prediction, molecular property prediction, yield and reaction prediction, retrosynthesis, text-based molecule design, molecule captioning, and reagent selection—using widely recognized datasets (*e.g.,* MoleculeNet, USPTO, PubChem), assessing different abilities such as understanding, reasoning, and explaining.

Molecule captioning. Molecule captioning evaluates the model's ability to explain molecular structures. Given a molecule's SMILES string, the model is required to generate a textual description of its key features, properties, biological activities, and potential applications. This task is evaluated using standard natural-language metrics (*e.g.,* BLEU, ROUGE, METEOR).

Text-based molecule design. This task requires the model to design new molecules from natural-language descriptions. Given textual descriptions of desired properties, structural features, biological activity, or applications, the model should generate the corresponding molecular SMILES representation.

Reaction-related tasks. Reaction-related tasks comprise four independent problems: yield prediction, forward reaction prediction, retrosynthesis, and reagents selection. In *yield prediction*, the model estimates if a reaction is high-yielding given reactants and reagents. In *forward reaction prediction*, the model is given reactant and reagent SMILES and must predict the major product SMILES. In *retrosynthesis*, the model is given a product SMILES and must predict plausible reactant SMILES. In *reagents selection*, the model selects appropriate reagents, ligands, or solvents from candidates for a given reaction.

C Detailed Training Protocol

Stage 1: Establishing the Molecular-Linguistic Mapping The primary goal of this stage is to align the projected molecular representations $\mathbf{H}_{chem}$ with the semantic space of the pre-trained LLM. To establish a robust mapping, we utilize instruction-tuning data containing reasoning chains but intentionally *suppress* the intermediate reasoning steps during this phase. Specifically, given a sample comprising an instruction $\mathbf{x}_{prompt}$, a chain-of-thought (CoT), and a final answer $\mathbf{y}_{ans}$, we train the model to directly generate the final answer. This "answer-only" supervision acts as an information bottleneck, compelling the projector to compress all necessary chemical properties into the ChemTokens.

To ensure the model genuinely utilizes these molecular features rather than memorizing textual patterns, we employ a **Counterfactual Alignment** strategy. Let $\mathcal{L}_{CE}(\mathbf{y}|\mathbf{H})$ denote the standard cross-entropy loss. We define the loss on the original clean input as $\mathcal{L}_{clean} = \mathcal{L}_{CE}(\mathbf{y}_{ans}|\mathbf{H}_{chem}, \mathbf{H}_{prompt})$. Simultaneously, we perform a corrupted forward pass using perturbed ChemTokens $\tilde{\mathbf{H}}_{chem} = \Phi(\mathbf{H}_{chem})$ (via dropout, shuffling, or noise) to compute $\mathcal{L}_{corrupt} = \mathcal{L}_{CE}(\mathbf{y}_{ans}|\tilde{\mathbf{H}}_{chem}, \mathbf{H}_{prompt})$. The counterfactual alignment loss is defined as a hinge loss:

$$\mathcal{L}_{CF} = \max(0, \gamma - (\mathcal{L}_{corrupt} - \mathcal{L}_{clean})) \tag{8}$$

where γ is a margin hyperparameter. The total objective for Stage 1 is:

$$\mathcal{L}^{(1)}_{total} = \mathcal{L}_{clean} + \lambda \mathcal{L}_{CF} \tag{9}$$

Stage 2: SFT for Molecule-aware CoT In this stage, we transition from direct mapping to complex reasoning. We unlock the full potential of the data by training the model on the complete sequences, requiring it to generate the explicit CoT derivations prior to the final answer. This enables the model to decompose complex molecular problems into intermediate steps.

To prevent "hallucinated reasoning" and ensure that the generated logic remains grounded in the molecular structure, we continue to employ the counterfactual strategy. The target sequence is extended to include both the reasoning chain and the answer, denoted as $\mathbf{y}_{full} = [\mathbf{y}_{cot}, \mathbf{y}_{ans}]$. Accordingly, the clean and corrupted losses are updated to $\mathcal{L}_{clean} = \mathcal{L}_{CE}(\mathbf{y}_{full}|\mathbf{H}_{chem}, \mathbf{H}_{prompt})$ and $\mathcal{L}_{corrupt} = \mathcal{L}_{CE}(\mathbf{y}_{full}|\tilde{\mathbf{H}}_{chem}, \mathbf{H}_{prompt})$. The total objective for Stage 2 becomes:

$$\mathcal{L}^{(2)}_{total} = \mathcal{L}_{clean} + \lambda \mathcal{L}_{CF} \tag{10}$$

This formulation ensures that the model not only generates the correct answer but also derives it through a valid, molecule-dependent reasoning path.

Stage 3: Chemistry-aware Latent Mind Activation In this phase, we activate the latent thinking modules during training. A major challenge in training these lightweight modules alongside a massive LLM backbone is the risk of optimization imbalance: the LLM's overwhelming parameter count can easily lead it to bypass the latent signals, treating them as noise rather than meaningful thought vectors. To mitigate this, we employ a **Strict Freezing Strategy**. We freeze both the ChemAdapter and the LLM backbone, updating *only* the parameters of the thinker and updater modules. This constraint forces the latent modules to adapt to the frozen semantic space of the LLM, compelling them to generate "legible" latent thoughts that effectively guide the LLM's frozen decoder toward the correct reasoning path defined by the CoT data.

Stage 4: GRPO with Latent Thinking Budget With the latent mind activated, we employ GRPO to refine the model's policy. Reversing the freezing strategy from the previous stage, we now **freeze** the latent thinking modules and fine-tune the remaining parameters. This strategy treats the learned latent dynamics as a stable "internal simulator," optimizing the backbone LLM to effectively utilize these latent thoughts for decision-making. The optimization is guided by a composite reward function comprising four terms: format adherence, answer validity, and answer correctness.

D Experimental Implementation Details

D.1 Dataset and Preprocessing

Prompt and Label Formatting. The dataloader structures each training example to enforce specific output formats. Every question is appended with the instruction: "*Your final answer must be formatted as* `<answer> ... </answer>`". Ground-truth labels are formatted similarly. The prompt and response are tokenized separately and concatenated.

D.2 Model Architecture

Base Language Model. We use Qwen-3-8B as the backbone LLM.

Molecule Encoder. We utilize SMI-TED Light as the molecule encoder. The encoder is kept frozen during all training stages and operates in inference mode (no gradient computation).

ChemAdapter. We employ a Querying Transformer (Q-Former) style projector with 128 learnable queries, an input dimension of 768, and 8 attention heads. For each input molecule, the adapter produces a sequence of 128 projected tokens, encapsulated by special tokens `<mol_start>` and `<mol_end>`.

Latent Modules. We add specific control tokens to the vocabulary, including `<latent>`, `<start_latent>`, and `<end_latent>`.

D.3 Supervised Fine-Tuning (SFT)

We adopt a three-stage SFT pipeline. All stages are trained using distributed data parallel (DDP) across 8 H200 GPUs using `bfloat16` precision.

Common Training Configuration. We use the AdamW (8-bit) optimizer with a cosine learning rate scheduler, a weight decay of 0.01, and gradient checkpointing enabled. The maximum sequence length is set to 8192 tokens. The regularization weight $\lambda = 0.2$, and $m = 0.1$ is the margin.

Stage 1. We train the LoRA adapters on the Qwen backbone and the ChemAdapter for 3 epochs with a learning rate of 2×10^{-4} and a batch size of 4 per device.

Stage 2. Weights are initialized from the Stage I checkpoint. We continue training the LoRA adapters and the ChemAdapter for 3 epochs with a learning rate of 2×10^{-4}.

Stage 3. This stage incorporates the continuous reasoning loop. We insert a block of latent tokens between the prompt and the response: `<start_latent>` $+ N \times$ `<latent>` $+$ `<end_latent>`. The number of latent tokens N is dynamic, calculated based on the length of the ground-truth CoT (capped at 4 tokens). Training proceeds for 3 epochs with a learning rate of 2×10^{-4}. The latent tokens are masked in the loss, except for the boundary token.

D.4 Reinforcement Learning via GRPO (Stage 4)

In the final stage, we employ Group Relative Policy Optimization (GRPO) to further refine the model's reasoning capabilities.

Algorithm and Environment. We adapted the TRL GRPO implementation with vLLM for efficient rollout generation (tensor parallel size 1, GPU memory utilization 0.3). We sample a group size of $G = 8$ completions per prompt. Training runs for 1 epoch with a learning rate of 1×10^{-5} and an effective batch size of 128 prompts per step. We use a KL penalty of $\beta = 0.0$ and a clipping epsilon of 0.2.

Reward Functions. We utilize a composite reward signal consisting of three equally weighted components:

1. **Format Reward:** +1.0 if the completion contains non-empty <answer> tags.
2. **Type Validity Reward:** +0.5 if the content within the tags matches the expected data type (e.g., valid SMILES string, float, or boolean).
3. **Correctness Reward:** +2.0 if the answer matches the ground truth. Correctness is determined via task-specific oracles, including exact canonical SMILES matching for structure generation and functional group verification for editing tasks.

Generation Settings. During rollouts, we use a temperature of 1.5 and top-p sampling of 0.9, with a maximum completion length of 2048 tokens.

E METRICS

Mean Absolute Error: The mean of the absolute deviations between predicted and true values, quantifies the typical magnitude of prediction error in regression tasks. Employed on *Functional Groups (FGs)* and *Rings*.

Scaffold Similarity: Structural conservation between generated and reference molecules is measured via the Tanimoto coefficient computed on molecular scaffolds. Scores range from 0 to 1, with higher values indicating stronger preservation of the core framework. Used in *Murcko scaffolds*.

Accuracy: The proportion of correctly predicted outcomes, serving as a baseline measure of overall correctness. For *Molecule Editing* tasks (e.g. add, delete, substitute functional groups), an outcome is correct only when the edited molecule stays valid, and the functional group count matches the instruction. Binary classification accuracy is calculated on *Ring System*. For reaction-prediction tasks (e.g. *Forward Prediction* and *Retrosynthesis*), we use Top-1 accuracy, i.e., a prediction is counted as correct only when the model's highest-ranked output exactly matches the ground-truth molecule(s).

Improvement: Absolute gains in the target property induced by *Molecule Optimization*. We mainly report the mean improvement across samples to indicate typical uplift.

success rate: The fraction of optimized molecules that exceed a predefined target threshold (for example, solubility over 0.8), reflecting the practical utility in *Molecule Optimization* tasks.

Fingerprint-based Similarities (FTS): Morgan Schneider et al. (2015), MACCS Durant et al. (2002), and RDKit Landrum et al. (2024) fingerprints are used to reflect correctness and structural similarity between the predicted molecule and the reference molecule in reaction-prediction tasks (e.g. *Forward Prediction* and Retrosynthesis). Additionally, the FTS metric refers to the average of Morgan, MACCS and RDKit fingerprint-based similarities.

Validity: The proportion of generated SMILES strings that are syntactically valid and can be parsed into chemical structures (*e.g.,* successfully converted to RDKit molecule objects), indicating the chemical feasibility of model outputs. Reported explicitly in reaction-prediction tasks (e.g. *Forward Prediction* and Retrosynthesis).

Text Similarities: Text similarity metrics like BLEU Papineni et al. (2002), ROUGE Lin (2004) and METEOR Banerjee and Lavie (2005) are utilized to assess the quality of generated molecule descriptions in *Molecular Description Generation* tasks by comparing them with the reference answers.

F Results for Closed-ended Tasks

We report the results for closed-ended tasks on ChemCoTBench, Mol-Instructions and ChemLLM-Bench to complement the results discussed in Section 5.2. Specifically, results on ChemCoTBench, including Molecule Understanding, Molecule Editing, and Reaction-related tasks are presented in Table 5. In addition, results on Mol-Instructions and ChemLLMBench, which comprise broad reaction-related subtasks, are shown in Table 6.

Table 5: Detailed performance on closed-ended tasks from ChemCoTBench. Metrics include mean absolute error (MAE↓) for functional group, Scaffold Similarity for Murcko scaffolds, correct rate (CR%) for ring system count, alongside Top-1 accuracy and fingerprint-based similarities (FTS) for reaction.

Category	Method	Molecule Understanding				Molecule Editing			Reaction			
		Functional Group		Scaffold		Add	Delete	Sub	Fwd $_{major}$		Fwd $_{by}$	
		FG↓	Ring↓	Murcko↑	Ring-sys↑	CR%	CR%	CR%	Top1↑	FTS↑	Top1↑	FTS↑
Text-only	Qwen-3-8B	7.03	1.03	0.02	36	4	52	1	0.00	0.02	0.00	0.00
	Qwen-3-8B (SFT)	0.12	0.45	0.40	65	55	76	37	0.09	0.35	0.02	0.07
Chem. LLMs	Stage 1	0.14	**0.10**	0.71	83	62	61	47	0.15	0.57	**0.12**	0.18
	Stage 1+2	0.08	0.20	0.66	63	63	80	49	0.10	0.44	0.03	0.11
	Stage 1+2+4	**0.07**	0.15	0.65	83	70	**88**	60	**0.20**	**0.63**	**0.12**	**0.20**
Latent Chem. LLMs	Coconut-Chem	0.08	0.30	0.64	60	54	68	44	0.08	0.42	0.02	0.09
	LatentChem (Ours)	0.09	0.15	**0.81**	**97**	**71**	84	**61**	0.17	0.52	0.08	0.17

Table 6: Detailed performance on closed-ended tasks from Mol-Instructions and ChemLLMBench. Metrics include Top-1 accuracy and fingerprint-based similarities (FTS) for reaction prediction, alongside correct rate (CR%) for reagent selection.

Category	Method	Mol-Instructions						ChemLLMBench						
		Reagent		Fwd_{Major}		Retro		Reagent Sel.			Fwd $_{major}$		Retro	
		Top1↑	FTS↑	Top1↑	FTS↑	Top1↑	FTS↑	Ligand%	Reactant%	Solvent%	Top1↑	FTS↑	Top1↑	FTS↑
Text-only	Qwen-3-8B	**0.00**	0.02	0.00	0.04	**0.00**	0.05	9	11	14	0.00	0.02	**0.00**	0.07
	Qwen-3-8B (SFT)	**0.00**	0.06	0.02	0.26	**0.00**	0.30	31	43	23	0.02	0.29	**0.00**	0.31
Chem. LLMs	Stage 1	**0.00**	**0.07**	0.05	**0.55**	**0.00**	0.47	28	**51**	34	0.01	0.49	**0.00**	0.47
	Stage 1+2	**0.00**	**0.07**	0.05	0.42	**0.00**	0.42	34	37	31	**0.04**	0.38	**0.00**	0.46
	Stage 1+2+4	**0.00**	**0.07**	**0.09**	0.52	**0.00**	0.49	**47**	39	33	**0.04**	0.48	**0.00**	0.45
Latent Chem. LLMs	Coconut-Chem	**0.00**	**0.07**	0.02	0.36	**0.00**	0.36	31	48	31	0.01	0.35	**0.00**	0.38
	LatentChem (Ours)	**0.00**	**0.07**	0.03	0.50	**0.00**	**0.52**	13	46	**49**	0.02	**0.53**	**0.00**	**0.49**

G Additional Results for Latent Thinking Budget Testing and Causal Study

This appendix contains supplementary figures supporting the analyses reported in Section 4.2. Omitted from the main text for reasons of space, these plots present additional results that are consistent with the trends reported in the main paper. Specifically, Figure 11 supplements Figure 6 by providing additional evidence of the performance trend, and Figure 12 supplements Figure 5 by providing additional measurements of generated CoT length.

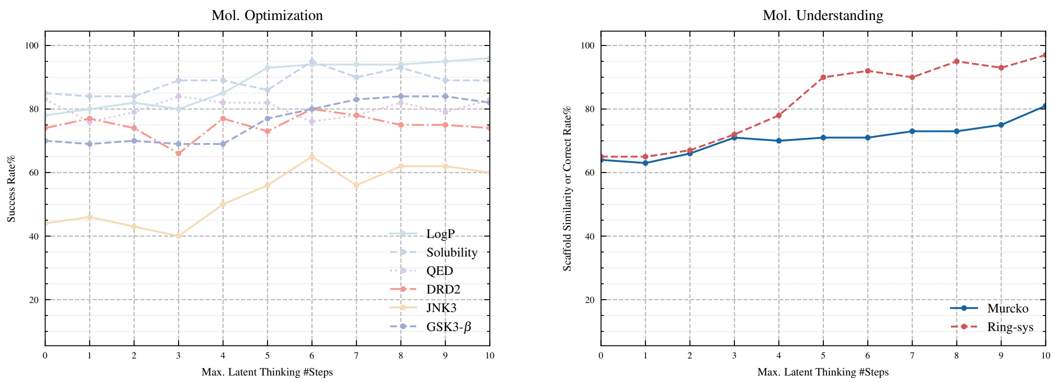

Figure 11: Additional results for budget stressing testing. We measure task performance (success rate, scaffold similarity, or correct rate) as the maximum allowed latent thinking steps (x-axis) decrease.

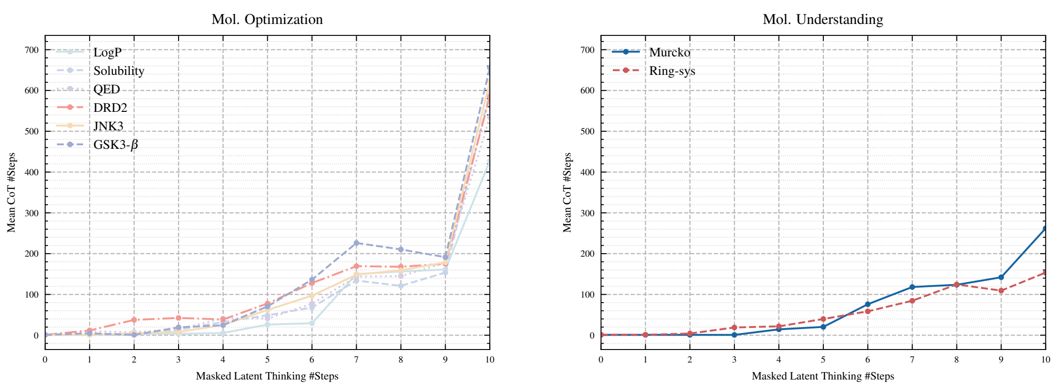

Figure 12: Additional results for causal necessity analysis. We monitor the length of generated explicit CoT (y-axis) as the first k latent tokens are replaced with Gaussian noise.

H Additional Case Study on Latent Thinking Budget Test

To better demonstrate the spontaneous CoT internalization ability of our model discussed in Section 4.1, we present the full, unedited model outputs with different latent thinking budget for a representative case from the solubility optimization subtask alongside with the textual prompt, as illustrated in Figure 13.

User:
You are an expert chemist. Below is a molecule structure: `CCNc1ccc(S(=O)(=O)c2ccc(N)cc2)cc1`. Optimize the Source Molecule to improve the compound's ability to dissolve in water (Solubility) while following a structured intermediate optimization process. Your final answer must be formatted as `<answer> SMILES </answer>`.

Latent Thinking Budget 10:

```
.<answer> Nc1ccc(S(=O)(=O)c2ccc(N)cc2)cc1 </answer>
```

Latent Thinking Budget 0:
Step 1: Structural Analysis of Source Molecule: Structural Analysis of Source Molecule: The source molecule contains a sulfonamide group (`S(=O)(=O)`) attached to a benzene ring, which is further substituted with a methyl-piperazine group (`CCN`). The sulfonamide is linked to another benzene ring with an amino (`-NH2`) substituent. The molecule has hydrophobic (methyl-piperazine and aromatic rings) and moderately polar (sulfonamide, amino group) regions.
Step 2: Property Analysis: The solubility of the molecule is limited by its hydrophobic methyl-piperazine group and the overall size of the aromatic systems. The sulfonamide and amino groups provide some water solubility but are insufficient for optimal aqueous dissolution.
Step 3: Limitation in Source Molecule for Property: The methyl-piperazine group (`CCN`) is a major contributor to low solubility due to its hydrophobic character and potential steric hindrance. The single amino group on the secondary benzene ring offers limited hydrogen bonding capacity.
Step 4: Optimization for Source Molecule: 1. Replace the methyl-piperazine group (`CCN`) with a smaller, polar cyclopropylamine group (`NCC1CC1`) to reduce hydrophobicity and steric bulk. 2. Maintain the sulfonamide and amino groups for polarity. 3. Simplify the substituent structure to enhance solvation.
Step 5: Final Target Molecule: `Nc1ccc(S(=O)(=O)c2ccc(NCC3CC3)cc2)cc1`

```
<answer> Nc1ccc(S(=O)(=O)c2ccc(NCC3CC3)cc2)cc1 </answer>
```

Figure 13: Comparison of model output with maximum latent thinking steps set to 10 versus 0.

I Additional Statistics on Inference Effeciency Gain

In this appendix, Figure 14 serves as a per-benchmark per-task breakdown of the aggregated results on LatentChem's efficiency ratio reported in Figure 8. Within each task category, the measured efficiency remains reasonably consistent across the evaluated benchmarks, supporting the use of averaged results in the main paper.

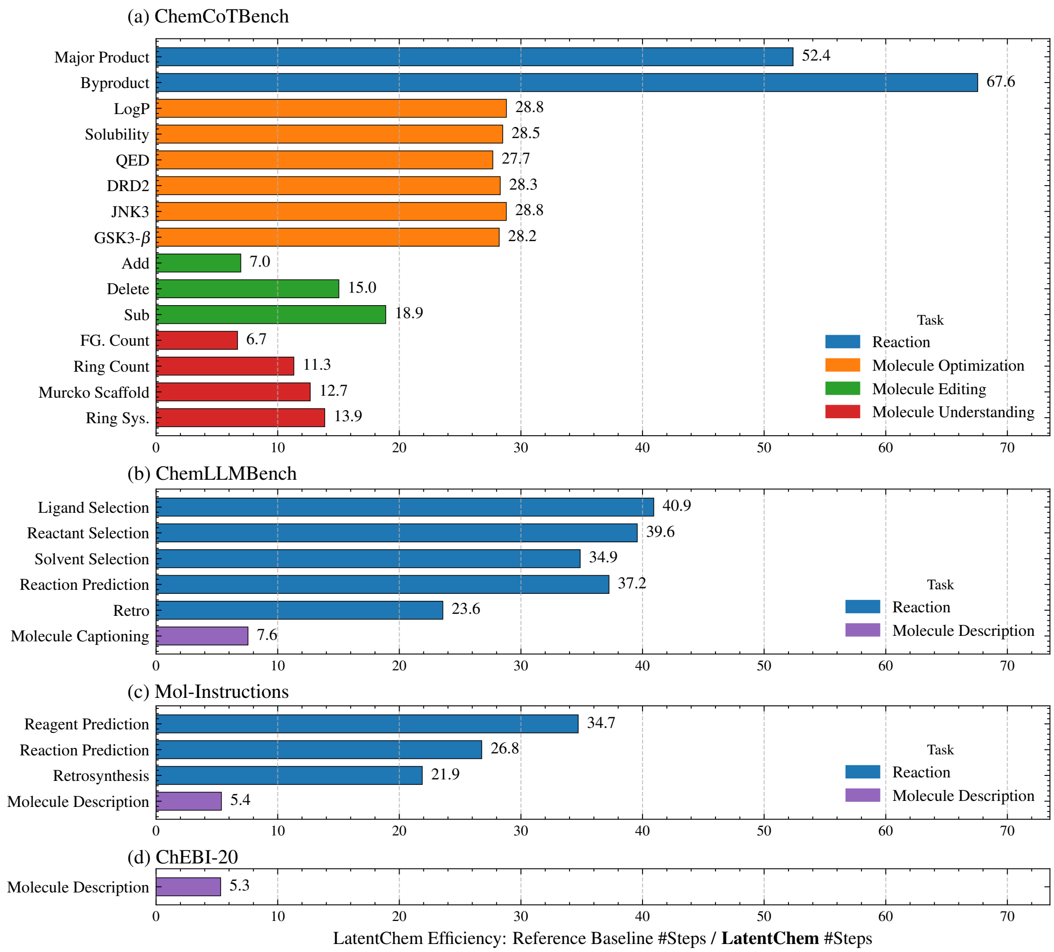

Figure 14: Detailed efficiency statistics grouped by benchmarks we used. Task categories are sorted by decreasing *LatentChem Efficiency*. Within each category, the task order remains consistent with that of other tables and figures in this study.

J Comparison with SOTA Models on ChemCoTBench

We adapt selected evaluation tables from ChemCoTBench Li et al. (2025a), limiting the reported sub-tasks to those evaluated in our experiments. Specifically, we omit the equivalence subtask from Molecule Understanding, as well as Retrosynthesis, RCR, NEPP, and MechSel from the Chemical Reaction task. Alongside the results reported in the original benchmark, we present the performance of our model on the same sub-tasks for reference. The results are shown in Tables 7, 8, and 9 respectively.

Table 7: Comparison with results reported on the ChemCoTBench foundational tasks, including molecule understanding, molecule editing, and their correlated subtasks. For the functional-group counting task (FG) and ring counting task (Ring) in the functional-group level molecule understanding, we apply the mean absolute error (MAE) as the evaluation metric. Tanimoto molecule similarity is applied as the evaluation for the Murcko scaffold extraction task (Murcko). The accuracy (%) metric is applied to other subtasks.

Models	Func-Group		Scaffold		Molecule-Edit		
	FG↓	Ring↓	Murcko↑	Ring-sys↑	Add	Delete	Sub
W/ Thinking							
Gemini-2.5-pro-think	0.11	0.60	0.51	87.5	100	85	81.7
Claude3.7-sonnet-think	0.21	1.60	0.40	80.0	85	80	83.4
DeepSeek-R1	0.27	1.55	0.34	45.0	70	70	68.3
o3-mini@20250103	0.13	0.60	0.39	75.0	65	55	80.0
o1-mini@20240912	0.21	1.25	0.25	61.7	55	80	58.3
Qwen3-235B-A22B-think	0.42	1.00	0.38	82.5	40	75	71.7
Qwen3-32B-think	0.25	0.95	0.21	75.0	20	55	20.0
Llama-Nemo-49B-think	0.80	1.90	0.09	86.8	0	80	8.0
W/o Thinking							
GPT-4o@20241120	0.17	1.35	0.21	80.0	80	80	65.0
Deepseek-V3	0.15	1.50	0.24	76.7	70	75	76.7
Gemini-2.0-flash	0.19	1.65	0.43	75.0	65	75	66.7
Qwen3-235B-A22B	0.42	1.00	0.34	82.5	40	75	66.7
Qwen3-32B	0.26	0.95	0.22	68.3	30	55	25.0
Qwen2.5-72B-Instruct	0.26	0.60	0.24	70.0	70	80	56.7
Qwen2.5-32B-Instruct	0.36	0.65	0.12	53.3	50	50	48.3
Llama-3.1-70B-Instruct	0.52	1.80	0.12	68.3	60	80	50.0
Llama-Nemo-49B	0.72	1.77	0.11	65.0	30	55	30.5
Gemma-2-27b-it	0.19	1.65	0.43	66.7	75	70	35.0
Phi-4-14B	0.28	1.65	0.15	70.0	60	80	38.3
OLMo2-32B-Instruct	0.19	1.05	0.07	63.3	15	30	11.7
Domain Expert Models							
Ether0	Failed	0.35	Failed	Failed	94	76	78
BioMedGPT-7B	1.6	2.43	0.18	53.3	10	12	10
BioMistral-7B	1.0	1.85	0.04	32.5	0	10	0
Latent Chemical LLMs							
LatentChem (Ours)	**0.09**	**0.15**	**0.81**	**96.7**	**71**	**84**	**61**

Table 8: Comparison with results reported on the ChemCoTBench Molecule Optimization task. The optimized targets are categorized into physicochemical properties (QED, LogP, solubility) and protein activity-related properties (JNK3, DRD2, GSK-3β), with the latter posing greater challenges to the model's chemical knowledge and reasoning capabilities. Δ is the mean property improvement, where a negative Δ indicates that most optimizations are property degradations. SR% is the success rate that brings property increase.

Models	LogP		Solubility		QED		DRD2		JNK3		GSK3-β	
	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%
W/ Thinking												
Gemini-2.5-pro-think	-0.22	76	1.06	70	0.28	84	0.36	74	-0.02	35	0.06	68
Claude3.7-sonnet-think	0.41	80	0.37	75	0.12	73	0.17	63	0.01	49	0.02	57
DeepSeek-R1	0.47	69	0.80	80	0.17	72	0.12	62	-0.02	29	0.01	41
o3-mini@20250103	0.26	59	0.81	85	0.21	86	0.19	69	-0.03	23	0.01	45
o1-mini@20240912	-0.42	52	1.78	95	0.07	70	-0.03	37	-0.10	15	-0.08	31
Qwen3-235B-A22B-think	0.05	40	0.20	40	0.02	24	0.03	31	-0.01	23	0.01	31
Qwen3-32B-think	-0.01	1	0.13	19	0.01	9	0.0	4	-0.02	3	-0.02	6
Llama-Nemo-49B-think	-0.24	7	0.25	25.2	0.10	41	0.03	29.9	-0.02	6	-0.01	11.2
W/o Thinking												
GPT-4o@20241120	-0.09	37	0.92	80	0.13	70	0.07	48	-0.02	30	-0.00	39
DeepSeek-V3	0.09	33	0.57	92	0.08	46	0.03	28	0.00	18	-0.01	29
Gemini-2.0-flash	0.37	72	0.28	58	0.13	79	0.15	63	-0.02	34	0.01	38
Qwen3-235B-A22B	0.03	21	0.18	45	0.07	34	0.04	26	-0.01	18	0.02	25
Qwen3-32B	0.0	2	0.08	20	0.02	14	-0.01	6	-0.02	6	-0.02	5
Qwen2.5-72B-Instruct	-0.03	41	0.34	59	0.07	57	0.04	40	-0.02	26	-0.00	40
Qwen2.5-32B-Instruct	0.15	44	0.49	65	0.09	54	0.05	32	-0.02	19	0.01	31
Llama-3.1-70B-Instruct	0.02	35	0.72	81	0.15	61	-0.00	31	-0.01	30	0.01	40
Llama-Nemo-Super-49B	-0.01	24	0.34	40	0.08	43	-0.00	16	-0.00	15	0.01	27
Gemma-2-27b-it	0.01	31	0.39	69	0.07	56	-0.02	15	-0.00	16	-0.00	17
Phi-4-14B	-0.26	44	0.22	53	0.17	74	-0.02	18	-0.03	14	-0.00	22
OLMo2-32B-Instruct	-1.71	11	1.21	46	0.08	40	-0.05	7	-0.03	8	-0.02	12
Domain Expert Models												
Ether0	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0
BioMedGPT-7B	-0.36	17	0.25	63	-0.29	7	-0.09	5	-0.11	6	-0.08	1
BioMistral-7B	0.01	1	0.24	6	0.0	0	0.0	1	-0.01	1	-0.01	0
Latent Chemical LLMs												
LatentChem (Ours)	**1.37**	**96**	**1.53**	**89**	**0.18**	**83**	**0.26**	**74**	**0.08**	**60**	**0.17**	**82**

Table 9: Comparison with results reported on the ChemCoTBench Chemical Reaction task, restricted to forward prediction ($\mathrm{Fwd}_{\mathrm{major}}$: major-product prediction, and $\mathrm{Fwd}_{\mathrm{by}}$: by-product prediction). FTS: molecule fingerprint similarity with reference.

Models	$\mathrm{Fwd}_{\mathrm{major}}$		$\mathrm{Fwd}_{\mathrm{by}}$	
	Top-1	FTS↑	Top-1	FTS↑
W/ Thinking				
Gemini-2.5-pro-think	0.72	0.89	0.20	0.51
Claude3.7-sonnet-think	0.73	0.87	0.25	0.31
DeepSeek-R1	0.48	0.71	0.21	0.45
o3-mini@20250103	0.52	0.71	0.20	0.27
o1-mini@20240912	0.26	0.31	0.11	0.17
Qwen3-235B-A22B-think	0.03	0.54	0.0	0.07
Qwen3-32B-think	0.11	0.33	0.09	0.18
Llama-Nemo-49B-think	0.09	0.18	0.04	0.18
W/o Thinking				
GPT-4o@20241120	0.28	0.58	0.04	0.20
DeepSeek-V3	0.36	0.62	0.04	0.30
Gemini-2.0-flash	0.19	0.56	0.01	0.07
Qwen3-235B-A22B	0.04	0.57	0.0	0.06
Qwen3-32B	0.06	0.57	0.0	0.13
Qwen2.5-72B-Instruct	0.04	0.49	0.0	0.13
Qwen2.5-32B-Instruct	0.01	0.43	0.0	0.12
Llama-3.1-70B-Instruct	0.02	0.35	0.0	0.08
Llama-Nemo-49B	0.04	0.40	0.0	0.08
Gemma-2-27b-it	0.01	0.55	0.0	0.04
Phi-4-14B	0.01	0.27	0.03	0.10
OLMo2-32B-Instruct	0.0	0.10	0.0	0.07
Text+Chem T5	0.44	0.74	0.0	0.07
Latent Chemical LLMs				
LatentChem (Ours)	**0.17**	**0.52**	**0.08**	**0.17**