DIVERGENCE DECODING: TRAINING-FREE CAPABILITY FUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

While large language models excel in reasoning, these generalists often lack knowledge for specialized scientific domains. Conversely, domain models (specialists), while knowledgeable, suffer from specialization side-effects including diminished logic and reduced robustness. To address this dilemma, we introduce ***Divergence Decoding***, a training-free framework for capability fusion. It reconstructs the "draft-and-verify" skeleton of speculative decoding into an adaptive routing mechanism. The core is using Jensen-Shannon divergence to monitor the distributional disagreement between the two models at each token. When the specialist exhibits significant divergence, our method identifies it as a potential reasoning risk and instantaneously routes control to the generalist. This allows the dynamic injection of general reasoning while preserving domain expertise, achieving inference-time policy composition of the generalist and the specialist. We evaluate *Divergence Decoding* across diverse model families (Qwen and Llama series) on challenging scientific benchmarks (GPQA, ChemBench, and ChemCoTBench). Experimental results demonstrate that *Divergence Decoding* outperforms both the domain-specialized and general-purpose models, effectively surpassing the performance of most single-model baseline. This suggests that *Divergence Decoding* provides a general, training-free paradigm for fusing diverse LLM capabilities through adaptive inference-time collaboration.

1 INTRODUCTION

The development of large language models (LLMs) has broadly followed two directions. General-purpose LLMs (Bai et al., 2023; Qwen et al., 2024; Yang et al., 2025; Qwen Team, 2026; Touvron et al., 2023; Grattafiori et al., 2024; Touvron et al., 2023; Jiang et al., 2023; Guo et al., 2025; Ouyang et al., 2022; OpenAI, 2023; Anthropic, 2023; 2024; Gemini Team Google, 2023; Google, 2024) emphasize strong reasoning capabilities, often strengthened through large-scale reinforcement learning, and have demonstrated impressive performance on multi-step problem solving and logical consistency across diverse domains. In parallel, scientific-domain specialist LLMs focus on incorporating expert knowledge through continued pretraining or fine-tuning on discipline-specific corpora in chemistry, biology, and materials science (Zhao et al., 2024; 2025a; Li et al., 2026; Labrak et al., 2024; Bai et al., 2025; Zhang et al., 2024; Hu et al., 2025). While these models excel at scientific terminology and specialized tasks, they often remain less robust than general-purpose models in broad reasoning and out-of-domain generalization. This dichotomy creates a critical bottleneck in scientific reasoning, where a single response necessitates both expert-level knowledge and rigorous multi-step logic. Recent scientific benchmarks (Rein et al., 2023; Phan et al., 2025; Li et al., 2025a; Wang et al., 2024; Hendrycks et al., 2020; Auer et al., 2023; Lu et al., 2024) highlight that even frontier models struggle to navigate this boundary, often failing either at the retrieval of domain knowledge or the maintenance of reasoning consistency. The complementary nature of these failure modes raises a question: *Can we combine these disparate strengths at inference time to outperform the capabilities of individual models?*

Existing paradigms for multi-model collaboration fall short of this goal. *Speculative decoding* (Leviathan et al., 2023), while structurally similar to our work, is strictly constrained by distributional invariance, aiming solely for acceleration. Conversely, model routing and cascades operate at the coarse query level, failing to exploit the dynamic, token-level interplay between expertise and reasoning. To bridge this gap, we propose ***Divergence Decoding***, a training-free framework for

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

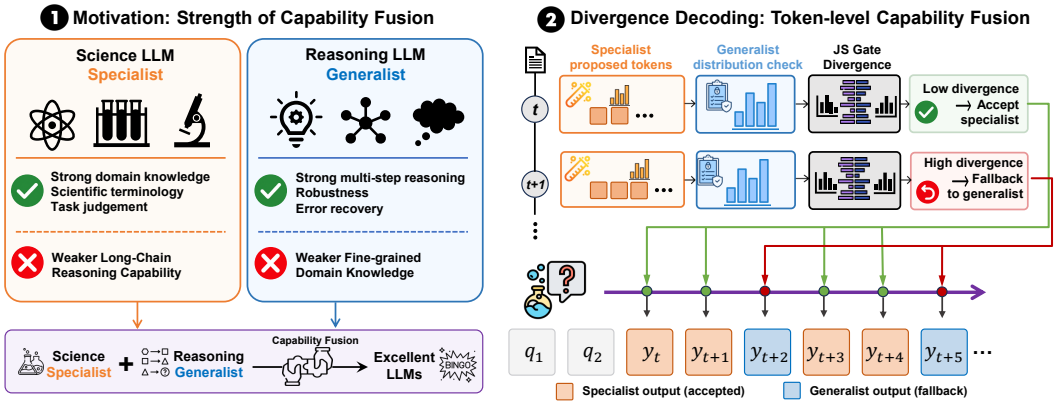

Figure 1: **Illustration of the proposed Divergence Decoding framework.** Left: Scientific tasks often demand both fine-grained domain knowledge and long-chain reasoning, which are typically bifurcated between specialized and generalist LLMs. Right: Our method bridges this gap through a state-dependent routing policy. By measuring the distribution divergence between the two models at each step t, the decoder dynamically navigates between the specialist's expertise and the generalist's consistency without additional training or joint fine-tuning.

capability fusion between a science-specialized LLM and a general reasoning LLM. The specialized model serves as the default generator, while the general model monitors local predictive disagreement through the Jensen-Shannon (JS) divergence between their next-token distributions. When the two models are locally consistent, we preserve the specialized model's token; when they diverge beyond a threshold, we reject it and fall back to the general reasoning model. In this way, the decoder adaptively integrates domain expertise and general reasoning at the token level, without training an additional router, jointly finetuning the models, or committing to either model globally.

A key point is that our method should not be interpreted as a variant of distribution-preserving speculative decoding. The goal of speculative decoding is to reproduce the target model's distribution more efficiently (Leviathan et al., 2023; Chen et al., 2023). Our goal is different: we seek a state-dependent routing policy whose output distribution is neither LLMs, but an adaptive composition of the two. Accordingly, the central theoretical question is not whether the combined trajectory matches any backbone model, but why such a routing policy can outperform either individual model. We answer this question through a theoretical analysis showing that JS divergence is not merely a heuristic disagreement score: under a structural view of complementary model errors, it provides a theoretically justified routing criterion for when the specialist should remain in control and when the generalist should intervene. In particular, low-divergence states can certify regimes where the specialist's domain-specific correction improves over the general model, while high-divergence states identify regimes where falling back to the general reasoning model reduces risk. This establishes that adaptive routing can outperform either constituent model at inference time, and reframes multi-model collaboration as risk-adaptive policy composition rather than acceleration.

We evaluate Divergence Decoding across diverse architectures, including Qwen and Llama-based pairs, on three authoritative benchmarks: ChemBench (Mirza et al., 2024), ChemCoTBench (Li et al., 2025a), and GPQA (Rein et al., 2023). Experimental results demonstrate that our framework outperforms both standalone specialists and generalists in most tasks. Ablation studies confirm that JS divergence serves as a superior gating criterion compared to universal cross-vocabulary or log-prob divergence, providing the most robust signal for beneficial fallbacks. Mechanistic analysis further reveals that resampling primarily occurs in the initial 0%-25% of the token trajectory, where a distinct entropy shift identifies the boundary between confident expertise and logical uncertainty; notably, the resampled tokens are predominantly natural language connectors rather than domain entities, suggesting the Generalist's role in stabilizing the reasoning framework. Finally, Divergence Decoding maintains high inference efficiency with minimal overhead, offering significant practical utility for real-world scientific discovery.

To summarize, our work makes three contributions. **First**, we introduce a training-free token-level fusion mechanism that combines a science-specialized LLM and a general reasoning LLM during decoding using JS-based gating, with no router training, joint finetuning, or extra supervision. **Second**,

we provide a theoretical analysis of this behavior as state-dependent cross-model routing rather than speculative sampling: the analysis shows how JS-based gating can make the combined policy outperform either constituent model when low-divergence states correspond to domain gains and high-divergence states correspond to safer general-model fallback. **Third**, we validate this approach across multiple scientific benchmarks and LLM-series, confirming the $A + B > A$ *or* B effect: the collaboration of two complementary large models can be stronger than either model used alone.

2 Related Work

2.1 General reasoning LLMs and domain-specialized scientific LLMs

Recent progress in large language models has advanced along two largely complementary directions Li et al. (2025b); Lv et al. (2025); Li et al. (2023). On one hand, general-purpose models have improved reasoning ability through large-scale post-training and reinforcement learning: for example, DeepSeek-R1 Guo et al. (2025) demonstrates that RL can elicit stronger long-chain reasoning behaviors such as reflection, verification, and adaptive problem solving, and its distilled variants further transfer such capabilities to smaller dense models. On the other hand, domain-specialized scientific LLMs have focused on improving expert knowledge and reasoning in chemistry and biomedicine. ChemDFM Zhao et al. (2024) shows that chemistry-focused pretraining and instruction tuning improve chemical understanding and dialogue, while ChemDFM-R Zhao et al. (2025a) strengthens chemical reasoning through atomized chemical knowledge and reaction-centered supervision . Chem-R (Zhao et al., 2025b) extends this line with a multi-stage training recipe combining chemical foundation training, reasoning protocol distillation, and multi-task GRPO to induce more deliberative chemical reasoning. Beyond chemistry, TxGemma Wang et al. (2025) similarly highlights the promise of domain-oriented LLMs as interactive and explainable models for biomedical reasoning and prediction. These two lines of work, however, are typically studied in isolation: general models prioritize broad reasoning robustness, whereas domain models emphasize scientific competence. Our work is motivated by the hypothesis that these strengths are complementary and can be fused at inference time without additional training.

2.2 Speculative decoding and token-level routing

Our method is related to both speculative decoding and token-level expert routing. Classical speculative decoding and speculative sampling follow a draft-and-verify paradigm, where a smaller or faster model proposes tokens and a larger target model verifies them to exactly preserve the target distribution; variants such as Medusa Cai et al. (2024) and EAGLE Li et al. (2024a;b) improve the drafting mechanism but retain the same acceleration-oriented objective. A separate line of work, including ETR Zhou et al. (2022); Chai et al. (2024), CITER Zheng et al. (2025), and FusionRoute Xiong et al. (2026), studies token-level routing across experts, typically using learned routers for efficiency or trainable coordination. In contrast, we do not aim to reproduce a fixed target model or rely on learned routing; instead, we combine two full-capability LLMs using a training-free rule based on the Jensen–Shannon divergence between their next-token distributions, so that observable disagreement determines whether to follow the specialist or the general model. This design is simple and architecture-agnostic, and is further motivated by evidence that sparse, high-impact token changes can drive reasoning gains Meng et al. (2026).

3 Method

3.1 Problem Setting

We consider two autoregressive language models: a domain-specialized model A and a general reasoning model B. Model A is adapted to a target scientific domain through domain-specific pretraining, supervised fine-tuning, and/or preference optimization. Model B is a strong general-purpose reasoning model with broader inference robustness. Our goal is to combine these two models *at inference time*, without any additional router training or joint finetuning.

We propose a token-level routing policy based on the Jensen–Shannon (JS) divergence between the two models' next-token predictive distributions. The key intuition is that low distributional divergence

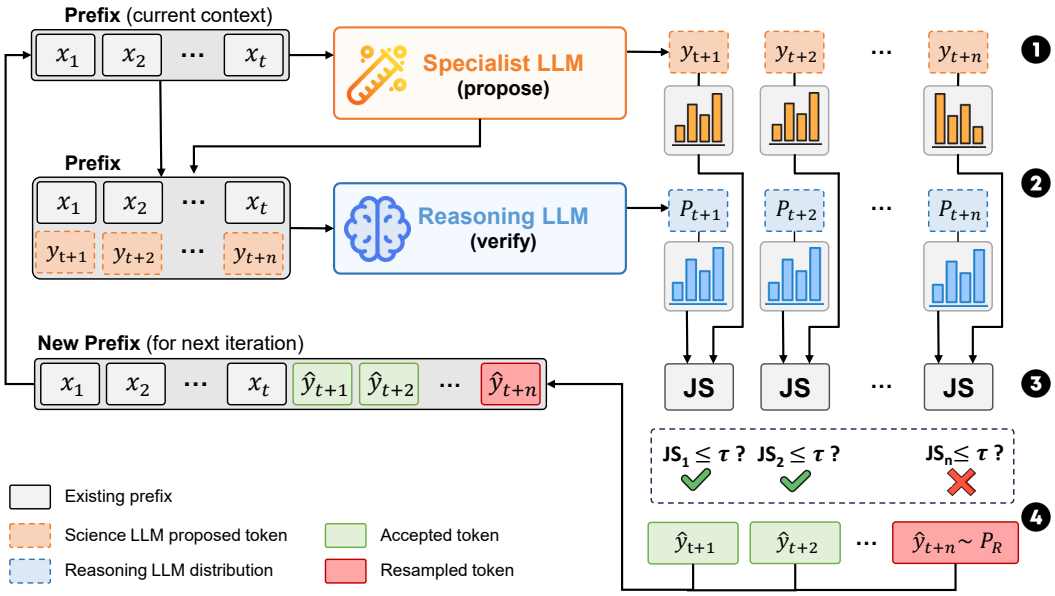

Figure 2: **Divergence Decoding framework. (Step 1)** At each iteration, the domain-specialized model drafts a block of candidate tokens and records the corresponding next-token distributions. **(Step 2)** The reasoning model then evaluates the same draft prefixes to obtain its predictive distributions. **(Step 3)** The decoder computes the JS divergence between the two distributions at each position. **(Step 4)** If the divergence is below the threshold, the drafted token is accepted; otherwise, the decoder samples a token from the reasoning model and starts a new block.

indicates local agreement, in which case the decoder trusts the domain-specialized model. Conversely, high divergence is treated as a warning signal, and the decoder defers to the general reasoning model to continue generation.

This objective differs from classical speculative decoding. Rather than producing a lossless approximation to a target model, our method defines a new adaptive decoding policy that switches between two models according to their local distributional disagreement.

3.2 JS-Gated Block Drafting with Sequential Verification

Autoregressive decoding traditionally generates tokens one at a time by computing next-token predictions from the current prefix. Our method, however, requires comparing two predictive distributions from two separate models. To avoid querying both models at every token, we implement JS-gated routing with a blockwise draft-then-verify procedure.

Let x denote the current generation prefix. At each decoding step, the domain-specialized model A and the reasoning model B define next-token distributions over the same prefix, and the routing decision is made according to the JS divergence between these two distributions. Starting from prefix x, model A drafts a block of n candidate tokens $\hat{\mathbf{y}}_{1:n} = (\hat{y}_1, \ldots, \hat{y}_n)$. For each position $1 \le j \le n$, define the corresponding draft prefix as $x^{(j)} = x \,\|\, \hat{y}_{1:j-1}, x^{(1)} = x$. The two models then induce the next-token distributions $p_A^{(j)} = p_A(\cdot \mid x^{(j)}), p_B^{(j)} = p_B(\cdot \mid x^{(j)})$. While model A produces these distributions during drafting, model B can compute $p_B^{(1)}, \ldots, p_B^{(n)}$ in a single teacher-forced forward pass over the drafted block.

For each position, the decoder computes the Jensen–Shannon divergence $s_j = D_{\mathrm{JS}}\big(p_A^{(j)}, p_B^{(j)}\big)$, where

$$D_{\mathrm{JS}}(p, q) = \frac{1}{2} KL(p \,\|\, m) + \frac{1}{2} KL(q \,\|\, m), \; m = \frac{1}{2}(p + q).$$

We use JS divergence because it is symmetric, bounded, and compares the local predictive distributions rather than only the probability of a realized token, making it suitable for measuring model disagreement during decoding.

Table 1: Performance on molecule understanding and editing tasks in ChemCoTBench. Results are reported as mean ± population standard deviation over three runs. For accuracy and Tanimoto similarity, higher is better (↑); for MAE, lower is better (↓).

Models	Func-Group		Scaffold		SMILES	Molecule-Edit		
	FG↓	Ring↓	Murcko↑	Ring-sys↑	Eq.↑	Add↑	Delete↑	Sub↑
Backbone: Qwen								
R1-distill-Qwen-32B	0.23 ± 0.012	0.87 ± 0.131	0.25 ± 0.085	0.59 ± 0.079	0.70 ± 0.116	43.33 ± 2.36	68.33 ± 2.36	29.67 ± 1.25
ChemDFM-R	0.23 ± 0.017	0.73 ± 0.075	0.77 ± 0.110	0.62 ± 0.035	$\mathbf{0.86 \pm 0.012}$	45.00 ± 4.08	63.33 ± 2.36	58.33 ± 2.05
Divergence Decoding	$\mathbf{0.18 \pm 0.012}$	$\mathbf{0.52 \pm 0.024}$	$\mathbf{0.89 \pm 0.005}$	$\mathbf{0.69 \pm 0.009}$	0.85 ± 0.008	$\mathbf{66.67 \pm 2.36}$	$\mathbf{76.67 \pm 2.36}$	$\mathbf{67.33 \pm 0.94}$
Backbone: Llama								
R1-distill-Llama-70B	0.29 ± 0.035	0.80 ± 0.050	0.30 ± 0.075	$\mathbf{0.64 \pm 0.020}$	$\mathbf{0.60 \pm 0.015}$	45.00 ± 5.00	$\mathbf{78.33 \pm 0.94}$	52.00 ± 1.00
Chem-R	0.17 ± 0.030	$\mathbf{0.38 \pm 0.075}$	0.47 ± 0.115	0.60 ± 0.040	0.56 ± 0.015	81.67 ± 2.36	73.33 ± 1.25	$\mathbf{62.00 \pm 1.00}$
Divergence Decoding	$\mathbf{0.12 \pm 0.015}$	0.43 ± 0.020	$\mathbf{0.60 \pm 0.010}$	$\mathbf{0.64 \pm 0.010}$	0.56 ± 0.009	$\mathbf{83.33 \pm 2.36}$	$\mathbf{78.33 \pm 2.36}$	57.33 ± 0.94

A single threshold τ determines whether to trust model A or defer to model B:

$$g_j = \mathbf{1}[s_j > \tau], \qquad \pi_\tau^{(j)}(\cdot \mid x^{(j)}) = (1 - g_j)p_A^{(j)} + g_j p_B^{(j)}.$$

Since g_j is deterministic, the decoder selects either $p_A^{(j)}$ or $p_B^{(j)}$ at each position rather than averaging the two distributions. If $s_j \leq \tau$, the drafted token $\hat{y}_j$ is accepted. Otherwise, a replacement token $y' \sim p_B^{(j)}$ is appended, verification of the current block stops, and the remaining drafted tokens are discarded because they were generated under a prefix that no longer matches the actual output. This asymmetric rule makes model A the default generator, preserving domain-specific priors, while invoking model B only when distributional disagreement indicates potential unreliability. The full procedure is shown in Figure 2. In all experiments, both models use the same sampling configuration: temperature $= 1.0$, top-$p = 0.7$, and top-$k = 10$, ensuring that routing decisions are not confounded by model-specific decoding settings.

Thresholds were selected using label-free statistics on calibration data disjoint from final evaluation, without optimizing benchmark accuracy. We subsequently formalized this procedure by estimating the switching advantage as $D_i = Q_r(i) - Q_d(i) + A_{\mathcal{B}}$, where Q_r and Q_d denote symmetric predictive support from both models for their respective continuations ($K = 4$ per model), and $A_{\mathcal{B}}$ corrects the self-consistency difference relative to the low-divergence region $J \in [0, 0.05]$. A non-decreasing isotonic fit to JS-binned observations (J_i, D_i) defines the threshold at its zero crossing. This estimator yields 0.20 for ChemBench and 0.65 for ChemCoTBench. With all estimator components fixed, label-free estimation on GPQA-Chemistry yields 0.62, achieving 37.50% accuracy ($27/72$).

We provide a concise justification for why JS-gated routing can outperform using either model alone in Appendix A.2.

4 Main Experiments on Scientific Benchmarks

This section provides a systematic evaluation of Divergence Decoding across diverse scientific domains and model architectures. We first establish the effectiveness of our method on scientific reasoning benchmarks, including ChemCoTBench Li et al. (2025a), ChemBench Mirza et al. (2024), and GPQA Rein et al. (2023). Furthermore, we conduct ablation studies to justify the selection of JS divergence as our core metric. We also analyse the unique token-level patterns emergent during the resampling process and provide a quantitative analysis of inference latency to demonstrate the method's computational efficiency.

4.1 Performance in Chemistry Domain

To evaluate the efficacy and generalizability of our proposed Divergence Decoding strategy, we conduct comprehensive experiments on two domain-specific benchmarks: ChemCoTBench and ChemBench. While ChemCoTBench focuses on specialized molecular editing and understanding through expert-annotated reasoning chains, ChemBench encompasses a broader chemical spectrum, ranging from organic synthesis to materials science. To verify whether Divergence Decoding consistently synergizes domain expertise with logical reasoning across diverse architectures, we investigate two distinct configurations: the Qwen backbone (ChemDFM-R paired with R1-Distill-

Table 2: Performance on different chemistry-related datasets in ChemBench (Part I). Results are reported as mean ± population standard deviation over three runs. Higher is better (↑).

Method	Analytical Chemistry	Chemical Preference	General Chemistry	Inorganic Chemistry	Materials Science
Backbone: Qwen					
R1-distill-Qwen-32B	0.435 ± 0.014	0.563 ± 0.007	0.681 ± 0.003	0.719 ± 0.012	0.547 ± 0.021
ChemDFM-R	0.377 ± 0.021	0.553 ± 0.017	0.663 ± 0.031	0.633 ± 0.017	$\mathbf{0.597 \pm 0.012}$
Divergence Decoding	$\mathbf{0.457 \pm 0.005}$	$\mathbf{0.567 \pm 0.021}$	$\mathbf{0.777 \pm 0.026}$	$\mathbf{0.767 \pm 0.012}$	0.580 ± 0.014
Backbone: Llama					
R1-distill-Llama-70B	0.413 ± 0.012	0.543 ± 0.009	0.733 ± 0.011	0.733 ± 0.025	0.635 ± 0.011
Chem-R	0.310 ± 0.029	0.513 ± 0.012	0.280 ± 0.054	0.380 ± 0.024	0.361 ± 0.026
Divergence Decoding	$\mathbf{0.470 \pm 0.008}$	$\mathbf{0.553 \pm 0.005}$	$\mathbf{0.760 \pm 0.014}$	$\mathbf{0.780 \pm 0.008}$	$\mathbf{0.683 \pm 0.033}$

Table 3: Performance on different chemistry-related datasets in ChemBench (Part II). Results are reported as mean ± population standard deviation over three runs. Higher is better (↑).

Method	Organic Chemistry	Physical Chemistry	Technical Chemistry	Toxicity and Safety
Backbone: Qwen				
R1-distill-Qwen-32B	0.647 ± 0.005	0.718 ± 0.013	0.701 ± 0.007	$\mathbf{0.378 \pm 0.006}$
ChemDFM-R	0.660 ± 0.024	0.563 ± 0.009	0.643 ± 0.009	0.367 ± 0.005
Divergence Decoding	$\mathbf{0.710 \pm 0.008}$	$\mathbf{0.740 \pm 0.010}$	$\mathbf{0.720 \pm 0.030}$	0.375 ± 0.005
Backbone: Llama				
R1-distill-Llama-70B	0.734 ± 0.039	$\mathbf{0.784 \pm 0.017}$	0.643 ± 0.005	0.397 ± 0.012
Chem-R	0.434 ± 0.011	0.236 ± 0.044	0.345 ± 0.034	0.207 ± 0.025
Divergence Decoding	$\mathbf{0.747 \pm 0.005}$	0.760 ± 0.016	$\mathbf{0.720 \pm 0.029}$	$\mathbf{0.413 \pm 0.005}$

Qwen-32B), and the Llama backbone (ChemR combined with R1-Distill-LLaMA-70B). This dual-backbone setup allows us to assess the strategy's cross-modal alignment performance across varying model scales and pretraining paradigms.

Experimental results in Table. 1, Table. 2 and Table. 3 indicate that Divergence Decoding outperforms individual constituent models across most tasks, establishing a new performance upper bound. A pivotal observation is the emergent synergy: by bridging the domain-specialized model with a reasoning-intensive counterpart, the collaborative framework not only mitigates the specialized model's logical deficiencies but also surpasses the performance of the standalone reasoning model. This demonstrates that our method effectively leverages the divergence between models to achieve a more precise and grounded chemical reasoning.

Comparison with Alternative Collaboration Strategies. To evaluate whether the gains of Divergence Decoding can be explained by stronger sampling or generic model routing, we additionally compare against two representative collaboration baselines on the Qwen-based ChemCoTBench setting. *Self-Resample* tests whether additional sampling from either constituent model can reproduce the observed improvements, while *CITER* represents a learned token-routing approach that requires training an auxiliary router. In contrast, Divergence Decoding is entirely training-free and performs token-level model selection directly from the predictive divergence between the two models. These comparisons provide stronger references beyond single-model and speculative-decoding baselines.

4.2 Performance in Multiple Science Domains

To comprehensively evaluate our divergence decoding method across various scientific domains, we apply the GPQA-diamond, a challenging benchmark consisting of expert-level questions across domains such as physics, chemistry, and biology. Solving these questions requires the combination of scientific knowledge, problem understanding, and multi-step reasoning. For each discipline, we

Table 4: Comparison with alternative sampling and model-routing strategies on ChemCoTBench using the Qwen-based model pair.

Method	Router Training	Add↑	Delete↑	Sub↑	Ring System↑	SMILES Eq.↑	Murcko Scaffold↑	Ring Count↓	FG Count↓
Divergence Decoding	No	**70**	**80**	**68**	**0.71**	0.84	0.91	0.57	**0.17**
Self-Resample (ChemDFM-R)	No	65	75	63	0.66	0.82	0.90	**0.40**	0.24
Self-Resample (R1-Distill-Qwen-32B)	No	65	**80**	61	0.64	0.88	**0.92**	0.70	0.35
CITER	Yes	55	70	50	0.69	**0.90**	0.7039	1.68	0.20

Table 5: GPQA accuracy (%) on chemistry, biology, and physics.

Domain	Science model	Acc.	Target model	Acc.	Divergence Decoding
Chemistry	ChemDFM-R	15.28	R1-distill-Qwen-32B	23.61	**37.50**
Biology	TxGemma	34.72	R1-distill-Qwen-32B	52.63	**63.16**
Physics	Raman-1.7B	26.74	R1-distill-Qwen-32B	61.62	**70.93**

select a corresponding domain-specialized expert model (ChemDFM-R, TxGemma, and Raman-1.7B Diddigam (2025)) to provide scientific knowledge, while using the same R1-Distill-Qwen-32B model to guide the reasoning process. Three combinations are tested on domain tasks, respectively. The experimental results show that inference with our divergence decoding strategy, which combines a domain model with a reasoning model, consistently outperforms using either model alone. This trend holds across different scientific domains, suggesting that our method can effectively leverage complementary strengths from domain expertise and reasoning-oriented models.

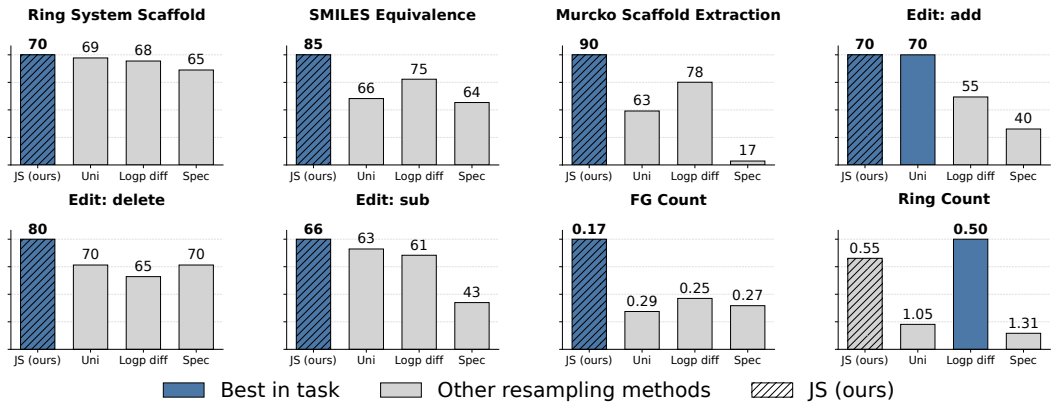

Figure 3: Comparison of different divergence methods during Divergence Decoding.

To further study why sparse token injection affect multi-domain performance, we also conducted mechanistic analysis in Appendix C.

4.3 Ablation Study

The ablation experiments are based on the ChemDFM-R (specialist) and R1-distill-Qwen-32B (generalist) on ChemCoTBench. We focus on (1) Divergence Type Comparison; (2) Token analysis during the resampling process in divergence decoding; (3) Hyperparameter analysis.

Divergence Type Comparison: We first compare different divergence types in the resampling stage, including universal cross-vocabulary divergence Patiño et al. (2025), top-10 vocabulary JS divergence, and single-token log-probability difference. These three signals all measure the discrepancy between the predictive distributions of two models at a given decoding position, but differ in the amount of distributional information they exploit. Specifically, *universal cross-vocabulary JS divergence* uses the full vocabulary distribution, retaining the richest discrepancy information while introducing noise and computational cost; *top-10 vocabulary JS divergence* approximates this discrepancy using only the most likely candidate tokens; and *single-token log-probability difference* provides the most limited signal. In Figure. 3, all three divergence types consistently outperform standard speculative

sampling, whose objective is to obtain an unbiased estimate of the target model distribution. This suggests that explicitly leveraging inter-model disagreement can improve decoding quality.

In Figure. 3, our divergence decoding using top-10 JS-divergence achieves the best performance on most tasks among all these divergence types. Overall, these results lead to two observations. First, incorporating divergence types generally improves model performance. Second, top-10 JS divergence provides a favorable trade-off between effectiveness and efficiency, as it is competitive with, and often superior to, full-vocabulary divergence while requiring substantially less computation.

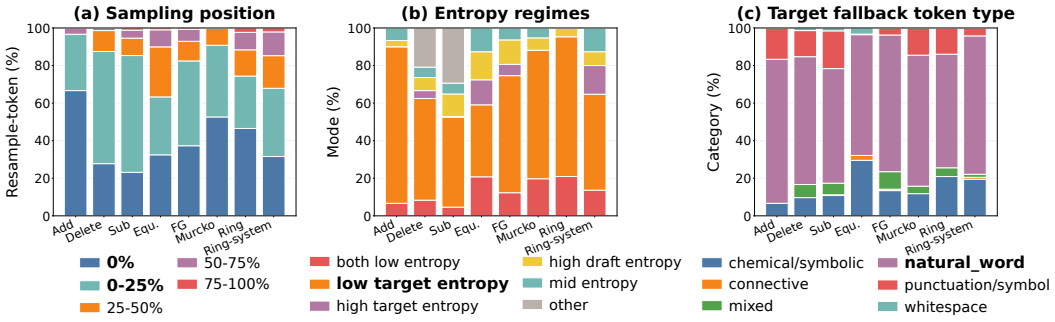

Figure 4: ***Resampling Analysis:*** Resampling patterns by position, uncertainty, and token type.

Token Analysis during Resampling: In Figure. 4, we further analyze the resampling tokens to investigate how divergence-based decoding affects the reasoning process. First, effective resampling events are highly concentrated in the early stage of the reasoning trajectory, with approximately 77% occurring at the initial position (0%-25%) of the generated reasoning sequence. This suggests that resampling primarily intervenes during the formation of the reasoning direction, where early errors in identifying functional groups, ring scaffolds, or structural equivalence may otherwise propagate throughout the entire trajectory. Second, the dominant uncertainty regime corresponds to high JS divergence and low entropy of the reasoning LLM, indicating that resampling is mostly triggered by confident disagreement from the reasoning model rather than by reasoning-side uncertainty. The correction signal often comes from a sharp predictive distribution that decisively redirects the domain model. Token-type analysis further shows that the injected fallback tokens are mainly general reasoning words, with a smaller but non-negligible fraction of chemical or symbolic tokens. This implies that resampling adjusts both the global reasoning path and key domain-specific anchors.

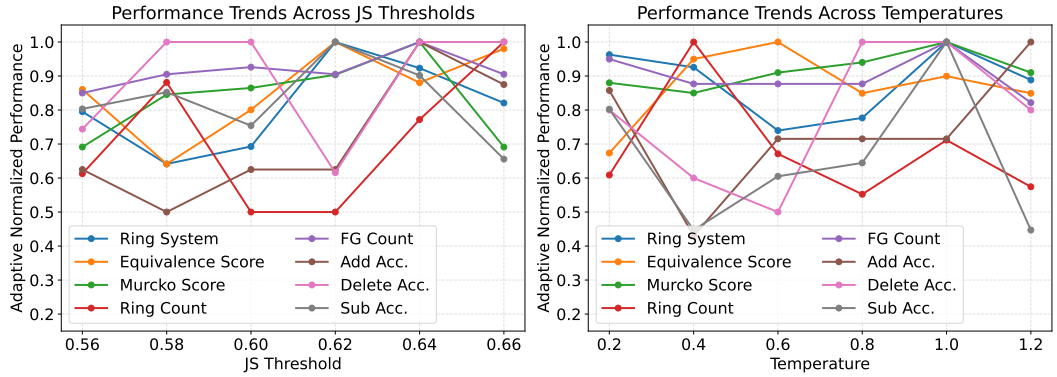

Figure 5: ***Sensitivity Analysis:***For visualization only, metrics with different scales and optimization directions are transformed into task-wise normalized performance scores(higher-is-better). The y-axis reflects normalized relative trends.

Hyperparameter Analysis: We also conduct a sensitivity analysis on key hyperparameters. The results show that both the JS-divergence threshold and the temperature affect downstream performance. The JS-divergence threshold directly controls the frequency with which reasoning-model tokens are injected, while the temperature changes the sharpness of the predictive distributions, thereby influencing the computed JS divergence and the resulting resampling decisions. Overall, injecting reasoning tokens under a broad range of JS-divergence thresholds brings consistent gains by

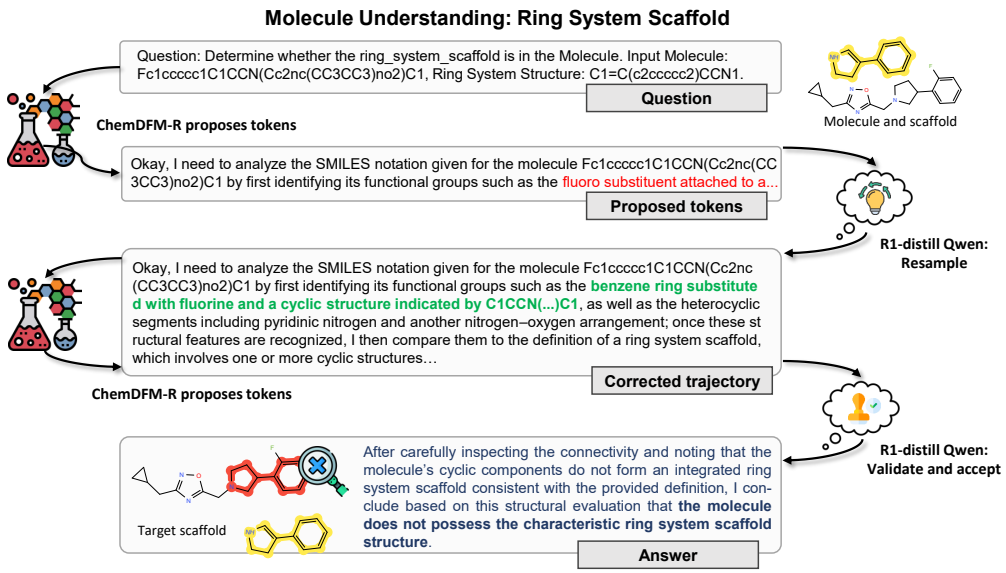

Figure 6: ***Case study:*** the divergence decoding between ChemDFM and R1-Distill Qwen affects the reasoning process in the molecule understanding task.

facilitating the integration of domain knowledge and reasoning ability. Detailed number is presented in Appendix 9.

Efficiency Analysis. We also compare the end-to-end generation time on ChemCoTBench. As shown in Table 6, Divergence Decoding achieves faster inference on Mol-Edit tasks but is slower on Mol-Und tasks, resulting in a slight overall latency increase. This task-dependent behavior may stem from different levels of agreement between the specialist and reasoning models: when their predictions align, blockwise drafting and early token acceptance reduce decoding cost; otherwise, frequent fallback to verification interrupts drafting and increases latency. Overall, Divergence Decoding maintains comparable efficiency while enabling stronger capability fusion.

Table 6: Average end-to-end generation time comparison on ChemCoTBench.

Methods	Mol-Und Avg.↓	Mol-Edit Avg.↓	Overall Avg.↓
Single LLM	**189.02**	218.74	**200.16**
Diver. Decode	233.04	**169.43**	209.19
Single / DD	0.81×	**1.29**×	0.96×

Case Study from ChemCoTBench. We provide a case study to illustrate how reasoning-model token injection influences the model's early focus and guides the decoding process toward a chemically correct final answer. With the reasoning guidance of R1-Distill Qwen, the specialist model (ChemDFM) is able to generate high-quality reasoning trajectories with grounded scientific knowledge. More case studies are provided in the Appendix F.

5 Conclusion

This paper presents *Divergence Decoding*, a training-free inference-time framework designed to fuse the expertise of domain-specialized LLMs with the logical rigor of general reasoning models. By formulating decoding as a state-dependent routing process, our method adaptively transfers control based on JS-divergence, enabling precise token-level integration without the need for additional fine-tuning or supervision. Experimental results across chemistry-centric benchmarks and broader scientific tasks demonstrate that Divergence Decoding surpasses its constituent models in most situations, yielding a synergistic effect where domain knowledge and reasoning capabilities are effectively composed at inference time. Our analyses further reveal that selective, early-stage interventions are pivotal for steering generations toward reliable scientific trajectories. Ultimately, divergence-based adaptive decoding offers a robust and scalable paradigm for constructing high-performance scientific AI systems from existing heterogeneous LLMs.

AI USE STATEMENT

We used a large language model solely to polish the language and improve the readability of the manuscript. The authors reviewed and edited all AI-assisted revisions and take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

The anonymous source code for reproducing our experiments is available at: `https://anonymous.4open.science/r/JS-Gated-Cross-Model-Routing-2816/`

REFERENCES

Anthropic. Model card and evaluations for claude models, 2023. URL `https://www.anthropic.com/news/claude-2`.

Anthropic. Introducing the next generation of claude, 2024. URL `https://www.anthropic.com/news/claude-3-family`.

Sören Auer, Dante AC Barone, Cassiano Bartz, Eduardo G Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, et al. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports*, 13(1): 7240, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

Ziwei Chai, Guoyin Wang, Jing Su, Tianjie Zhang, Xuanwen Huang, Xuwu Wang, Jingjing Xu, Jianbo Yuan, Hongxia Yang, Fei Wu, et al. An expert is worth one token: Synergizing multiple expert llms as generalist via expert token routing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11385–11396, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Sai Praneeth Diddigam. Raman-01: Compact rl-enhanced physics solver. `https://huggingface.co/think-a-tron/raman-01-1.7B`, 2025. RL (GRPO) finetuned on Qwen3-1.7B.

Gemini Team Google. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Google. Gemma open models, 2024. URL `https://ai.google.dev/gemma`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Ming Hu, Chenglong Ma, Wei Li, Wanghan Xu, Jiamin Wu, Jucheng Hu, Tianbin Li, Guohang Zhuang, Jiaqi Liu, Yingzhou Lu, et al. A survey of scientific large language models: From data foundations to agent frontiers. *arXiv preprint arXiv:2508.21148*, 2025.

Albert Q Jiang, A Sablayrolles, A Mensch, C Bamford, D Singh Chaplot, Ddl Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. arxiv. *arXiv preprint arXiv:2310.06825*, 10:3, 2023.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the association for computational linguistics: acl 2024*, pp. 5848–5864, 2024.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Hao Li, Jinfa Huang, Peng Jin, Guoli Song, Qi Wu, and Jie Chen. Weakly-supervised 3d spatial reasoning for text-based visual question answering. *IEEE Transactions on Image Processing*, 32: 3367–3382, 2023.

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, Da Long, Li Yuan, Yu Wang, Yonghong Tian, Xinchang Wang, and Fanyang Mo. Decoupled peak property learning for efficient and interpretable electronic circular dichroism spectrum prediction. *Nature Computational Science*, 5(3):234–244, 2025b.

Hao Li, He Cao, Shenyao Peng, Zijing Liu, Bin Feng, Yu Wang, Zhiyuan Yan, Yonghong Tian, Yu Li, and Li Yuan. Agentic reinforcement learning empowers next-generation chemical language models for molecular design and synthesis. *arXiv preprint arXiv:2601.17687*, 2026.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-2: Faster inference of language models with dynamic draft trees, 2024b. *URL https://arxiv. org/abs/2406.16858*, 1(2), 2024b.

Xingyu Lu, He Cao, Zijing Liu, Shengyuan Bai, Leqing Chen, Yuan Yao, Hai-Tao Zheng, and Yu Li. Moleculeqa: A dataset to evaluate factual accuracy in molecular comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3769–3789, 2024.

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. Prollama: A protein large language model for multi-task protein language processing. *IEEE Transactions on Artificial Intelligence*, 2025.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in rlvr fine-tuning of llms. *arXiv preprint arXiv:2603.22446*, 2026.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family, 2025.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

A Yang Qwen, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengpeng Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Eric Wang, Samuel Schmidgall, Paul F Jaeger, Fan Zhang, Rory Pilgrim, Yossi Matias, Joelle Barral, David Fleet, and Shekoofeh Azizi. Txgemma: Efficient and agentic llms for therapeutics. *arXiv preprint arXiv:2504.06196*, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Nuoya Xiong, Yuhang Zhou, Hanqing Zeng, Zhaorun Chen, Furong Huang, Shuchao Bi, Lizhu Zhang, and Zhuokai Zhao. Token-level llm collaboration via fusionroute. *arXiv preprint arXiv:2601.05106*, 2026.

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.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. A comprehensive survey of scientific large language models and their applications in scientific discovery. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8783–8817, 2024.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, et al. Chemdfm: A large language foundation model for chemistry. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024.

Zihan Zhao, Bo Chen, Ziping Wan, Lu Chen, Xuanze Lin, Shiyang Yu, Situo Zhang, Da Ma, Zichen Zhu, Danyang Zhang, et al. Chemdfm-r: A chemical reasoning llm enhanced with atomized chemical knowledge. *arXiv preprint arXiv:2507.21990*, 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.

Wenhao Zheng, Yixiao Chen, Weitong Zhang, Souvik Kundu, Yun Li, Zhengzhong Liu, Eric P Xing, Hongyi Wang, and Huaxiu Yao. Citer: Collaborative inference for efficient large language model decoding with token-level routing. *arXiv preprint arXiv:2502.01976*, 2025.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

A THEORETICAL ANALYSIS

In this section, we provide sufficient conditions under which a JS-thresholded routing policy outperforms both constituent models. This could be an explaintion for why js-routed divergence decoding outperforms both models under general conditions.

A.1 THEORETICAL SETUP

Let the decoding state at step t be the prefix

$$h_t = (x_1, \ldots, x_{t-1}).$$

Consider a domain-specialized model A and a general-purpose model B. We denote by $p_A(\cdot \mid h_t)$ and $p_B(\cdot \mid h_t)$ the next-token distributions of models A and B, respectively, and by $r_t(\cdot)$ an unknown target distribution that represents the desired conditional distribution at state h_t. Throughout, we suppress the conditioning on h_t when no ambiguity arises.

Model assumptions. We consider general assumptions on the domain-specialized model A and the general-purpose model B. For a powerful general-purpose model, we assume that it is capable enough to fit the domain-specific task with a small bounded error. Furthermore, we assume there is an in-domain state subspace where the domain-specialized model fundamentally outperforms the general-purpose model.

Assumption A.1 (Robustness of the General-Purpose Model)**.** *There exists a constant $\epsilon_B > 0$, such that for every decoding state h_t, the general-purpose model B maintains a bounded distance to the target distribution r_t:*

$$\sqrt{D_{\mathrm{JS}}(p_B, r_t)} \le \epsilon_B.$$

Assumption A.2 (In-Domain Subspace Advantage)**.** *There exist a constant $\delta > 0$ and an in-domain state subspace $\mathcal{H}_{\mathrm{in}}$ for the domain-specialized model A such that for any decoding state $h_t \in \mathcal{H}_{\mathrm{in}}$, model A maintains a strict precision advantage over model B, such that the difference in their JS distances to the target distribution satisfies:*

$$\sqrt{D_{\mathrm{JS}}(p_B, r_t)} - \sqrt{D_{\mathrm{JS}}(p_A, r_t)} \ge \delta.$$

Moreover, there exists a constant ϵ_A depending on δ, $\forall h_t \notin \mathcal{H}_{\mathrm{in}}$,

$$\sqrt{D_{\mathrm{JS}}(p_A, r_t)} \ge \epsilon_A.$$

JS-gated cross-model routing. Define the observable disagreement signal as the JS divergence between $p_A(\cdot \mid h_t)$ and $p_B(\cdot \mid h_t)$

$$s_t = D_{\mathrm{JS}}\left(p_A(\cdot \mid h_t), p_B(\cdot \mid h_t)\right).$$

Given a threshold $\tau > 0$, we define the hard gate

$$g_t = \mathbf{1}[s_t > \tau].$$

Hence, the routed policy is defined as

$$\pi_\tau(\cdot \mid h_t) = (1 - g_t)p_A(\cdot \mid h_t) + g_t p_B(\cdot \mid h_t). \qquad \text{(JS-gated routing)}$$

Thus, the decoder accepts the specialized model in low-disagreement states and falls back to the general model in high-disagreement states.

Goal. We measure the risk with respect to the target distribution using

$$\ell_i(t) = D_{\mathrm{JS}}(r_t, p_i), \quad i \in \{A, B\}.$$

We further define the expected risk

$$L(i) = \mathbb{E}[\ell_i(t)].$$

Hence, the routed loss is

$$\ell_\tau(t) = (1 - g_t)\ell_A(t) + g_t \ell_B(t).$$

The expected risk for the JS-gated routing is

$$L(\tau) = \mathbb{E}[\ell_\tau(t)].$$

The goal is to show that under general conditions, $L(\tau)$ can be smaller than $L(A)$ and $L(B)$. Namely, JS-gated cross-model routing can perform better than specialized model A and general model B.

A.2 THEORETICAL PROOF

To connect the risk metric with our assumptions, we define the risk difference $\Delta_t = \ell_B(t) - \ell_A(t)$. Remark that we only consider the task is non-trivial, i.e., $\mathbb{P}(s_t > \tau) > 0$, $\mathbb{P}(s_t \leq \tau) > 0$. If the task were entirely within the specialized domain ($\mathbb{P}(s_t > \tau) = 0$) or entirely out-of-domain ($\mathbb{P}(s_t \leq \tau) = 0$), routing would be unnecessary, as a single model would suffice.

Theorem A.1. *Under Assumption A.1 and Assumption A.2, if $\tau, \epsilon_B, \epsilon_A$ satisfy*

$$\tau \geq 4\epsilon_B^2, \quad \epsilon_A > \sqrt{\tau} + \epsilon_B,$$

then the expected risk of the JS-gated cross-model routing policy is strictly less than that of both individual models:

$$L(\tau) < \min(L(A), L(B)).$$

Proof. By definition of the routed policy, the loss at step t is:

$$\ell_\tau(t) = \ell_A(t) + g_t(\ell_B(t) - \ell_A(t)) = \ell_A(t) + g_t\Delta_t.$$

Taking the expectation yields the risk difference relative to model A:

$$L(\tau) - L(A) = \mathbb{E}[g_t\Delta_t] = \mathbb{P}(s_t > \tau)\mathbb{E}[\Delta_t \mid s_t > \tau].$$

Similarly, expressing the routed loss in terms of model B gives $\ell_\tau(t) = \ell_B(t) - (1 - g_t)\Delta_t$, which implies:

$$L(\tau) - L(B) = -\mathbb{E}[(1 - g_t)\Delta_t] = -\mathbb{P}(s_t \leq \tau)\mathbb{E}[\Delta_t \mid s_t \leq \tau].$$

To prove $L(\tau) < \min(L(A), L(B))$, it suffices to show that $\mathbb{E}[\Delta_t \mid s_t > \tau] < 0$ and $\mathbb{E}[\Delta_t \mid s_t \leq \tau] > 0$. We mainly leverage the fact that the square root of the Jensen-Shannon divergence, $\sqrt{D_{\mathrm{JS}}(P, Q)}$, is a true mathematical metric and therefore satisfies the triangle inequality.

- **Case 1: High-disagreement region ($s_t > \tau$)**

  Noticing that

  $$\sqrt{s_t} = \sqrt{D_{\mathrm{JS}}\left(p_A(\cdot \mid h_t), p_B(\cdot \mid h_t)\right)} \leq \sqrt{D_{\mathrm{JS}}(p_A, r_t)} + \sqrt{D_{\mathrm{JS}}(p_B, r_t)},$$

  and by Assumption A.1,

  $$\sqrt{D_{\mathrm{JS}}(p_B, r_t)} \leq \epsilon_B,$$

  we have

  $$\sqrt{D_{\mathrm{JS}}(p_A, r_t)} \geq \sqrt{s_t} - \sqrt{D_{\mathrm{JS}}(p_B, r_t)} > 2\epsilon_B - \epsilon_B = \epsilon_B,$$

  where we use $s_t > \tau \geq 4\epsilon_B^2$. This strictly guarantees $\sqrt{D_{\mathrm{JS}}(p_A, r_t)} > \sqrt{D_{\mathrm{JS}}(p_B, r_t)}$. Consequently, model A diverges further from the target distribution than model B, yielding a negative risk difference $\Delta_t < 0$. Therefore, $\mathbb{E}[\Delta_t \mid s_t > \tau] < 0$.

- **Case 2: Low-disagreement region ($s_t \leq \tau$)**

  We analyze the state space by bounding the divergence in the out-of-domain region. For any out-of-domain state $h_t \notin \mathcal{H}_{\mathrm{in}}$, applying the reverse triangle inequality on the JS divergence metric yields:

  $$\sqrt{s_t} = \sqrt{D_{\mathrm{JS}}(p_A, p_B)} \geq \left|\sqrt{D_{\mathrm{JS}}(p_A, r_t)} - \sqrt{D_{\mathrm{JS}}(p_B, r_t)}\right|.$$

  By Assumption A.2, the specialized model degrades significantly out-of-domain such that $\sqrt{D_{\mathrm{JS}}(p_A, r_t)} \geq \epsilon_A$, and by Assumption A.1, the general model maintains $\sqrt{D_{\mathrm{JS}}(p_B, r_t)} \leq \epsilon_B$. Under the condition that the parameter ϵ_A satisfies $\epsilon_A > \sqrt{\tau} + \epsilon_B$, we obtain the lower bound for the out-of-domain disagreement signal:

  $$\sqrt{s_t} \geq \epsilon_A - \epsilon_B > \sqrt{\tau}.$$

  This strictly implies $s_t > \tau$ for all $h_t \notin \mathcal{H}_{\mathrm{in}}$. By contraposition, observing a low-disagreement signal mathematically guarantees that the decoding state resides within the in-domain subspace:

  $$s_t \leq \tau \implies h_t \in \mathcal{H}_{\mathrm{in}}.$$

Consequently, within this low-disagreement region, the strict precision advantage of model A from Assumption A.2 definitively applies. For any $h_t \in \mathcal{H}_{\text{in}}$, we have $\sqrt{D_{\text{JS}}(p_B, r_t)} - \sqrt{D_{\text{JS}}(p_A, r_t)} \geq \delta > 0$. The risk difference expands as:

$$\Delta_t = \ell_B(t) - \ell_A(t) = \left(\sqrt{\ell_B(t)} - \sqrt{\ell_A(t)}\right)\left(\sqrt{\ell_B(t)} + \sqrt{\ell_A(t)}\right) \geq \delta \cdot (\delta + 0) = \delta^2 > 0.$$

Since Δ_t is strictly bounded below by $\delta^2 > 0$ for all states satisfying $s_t \leq \tau$, its conditional expectation is strictly positive:

$$\mathbb{E}[\Delta_t \mid s_t \leq \tau] \geq \delta^2 > 0.$$

Together with Case 1 and Case 2, the proof is completed. □

B Implementation Details.

First, we set the block size to 10 in all experiments. That is, the science model samples a block of ten tokens at each drafting step, and the reasoning LLM then computes the predictive distributions at the corresponding ten token positions.

Second, we set the JS-divergence threshold to 0.20 for ChemBench and 0.65 for ChemCoTBench using label-free calibration statistics, without optimizing final evaluation accuracy. Calibration samples are disjoint from the evaluation sets: we use non-overlapping samples from ChemCoTDataset for ChemCoTBench and reserve a fixed 10% of each constituent subset of ChemBench, excluding these samples from final evaluation.

To characterize the threshold selection, we estimate the relative advantage of reasoning-model intervention at different divergence levels. For each sampled prefix i, we compute the next-token JS divergence J_i and generate $K = 4$ short continuations from each model. Both models score these continuations through teacher forcing, yielding symmetric predictive support $Q_r(i)$ and $Q_s(i)$ for the reasoning-model and science-model continuations, respectively. We define

$$D_i = Q_r(i) - Q_s(i) + A_{\mathcal{B}}, \tag{1}$$

where $A_{\mathcal{B}}$ is the benchmark-level difference in repeated-generation self-consistency between the reasoning and science models, corrected by subtracting the corresponding difference in the low-divergence region $J \in [0, 0.05]$. We aggregate (J_i, D_i) into fixed JS-divergence bins and fit a non-decreasing isotonic curve $\hat{g}_{\mathcal{B}}(J)$, whose zero crossing defines the estimated switching threshold:

$$\hat{\tau}_{\mathcal{B}} = \inf\{J : \hat{g}_{\mathcal{B}}(J) \geq 0\}. \tag{2}$$

This estimate is 0.20 for ChemBench and 0.64 for ChemCoTBench; the latter lies in the same locally stable calibration region as the adopted value of 0.65. All estimator components are shared across datasets, with only the unlabeled calibration observations varying. As an additional generalization check, applying the fixed estimator to GPQA-Chemistry without accessing answer labels yields a threshold of 0.62, at which Divergence Decoding achieves 37.50% accuracy $(27/72)$.

Third, for the computation of top-10 JS divergence, the distributions proposed by the two models, denoted as p and q, are almost always not exactly identical. We therefore first construct an intermediate support m that contains all tokens appearing in either p or q. For tokens that appear in m but are absent from either p or q, we extend the corresponding distribution and assign probability zero to those missing tokens. We then compute the KL divergence from p to m and from q to m, respectively, to obtain the JS divergence.

When the prefixes differ, feeding the same text into another model may lead to inconsistent tokenization. To make the distributions from the two models comparable, we first perform sequence-level alignment. We then merge and normalize the probabilities over the aligned segments, and compute the JS divergence only on the aligned parts.

We note that this sequence-alignment procedure is highly similar to part of the method used in universal cross-vocabulary JS divergence Patiño et al. (2025), so we include it as a comparison in the ablation study in the main paper. Universal cross-vocabulary JS divergence is designed for model distillation when tokenizers are inconsistent. It first performs sequence alignment and probability

merging. Then, for the two distributions, it directly computes JS divergence over the alignable segments, pads the unaligned tails into equal-length vectors, computes an additional distance over these tails, and adds the two terms as the final distributional distance. In practice, to avoid excessive computation, we use the top-200 distributions to compute this universal JS divergence and compare its performance with our method.

Similarly, the thresholds for the log-probability difference baseline and the universal JS divergence baseline are chosen so that only a very small fraction of tokens are resampled. Their thresholds are set to 5.0 and 0.11, respectively.

"'latex

C Mechanistic Analysis of Sparse Interventions on GPQA

The improvements on GPQA may appear surprising given that Divergence Decoding resamples only a small fraction of tokens from the general reasoning model. For example, on GPQA-Chemistry, the domain-specialized model achieves 15.28% accuracy, the general reasoning model achieves 23.61%, while Divergence Decoding reaches 37.50%, despite only approximately 3–5% of generated tokens being sampled from the generalist. To better understand this behavior, we conduct additional matched-rate, per-question, and trajectory-level analyses. We focus primarily on GPQA-Chemistry, which contains 72 questions and exhibits the largest relative improvement.

Why can sparse interventions have a large effect? The intervention rate should not be interpreted as the fraction of the final reasoning directly contributed by the generalist. Autoregressive generation is inherently path dependent: changing even a single token at a consequential position modifies the prefix on which all subsequent tokens are conditioned.

After an intervention, the specialist remains the default generator, but it continues generation from the modified context. Consequently, a small number of resampled tokens can redirect a much longer specialist-generated continuation. For instance, replacing a transition such as *"Next"* with *"Wait"* can induce the model to reconsider an intermediate result, verify a previous step, or enter an alternative reasoning branch. Hundreds of subsequent tokens may then again be generated by the specialist.

Thus, Divergence Decoding does not improve performance by directly inserting 3–5% of a correct solution from the generalist. Instead, the generalist provides sparse trajectory-level interventions at positions where the two models strongly disagree. The resulting effect is therefore nonlinear and depends on *where* an intervention occurs and how it changes the subsequent autoregressive trajectory, rather than simply on the total number of resampled tokens.

Matched-rate random-intervention control. To determine whether the improvement can be explained merely by injecting a small number of generalist tokens, we introduce a matched-rate random-intervention baseline. At each decoding step, this baseline randomly selects approximately the same fraction of positions for generalist resampling as Divergence Decoding, corresponding to an intervention rate of approximately 3% on GPQA-Chemistry. The total amount of generalist participation is therefore comparable, while the intervention positions are chosen independently of the JS divergence.

Table 7: Matched-rate intervention analysis on GPQA-Chemistry. Random intervention uses approximately the same intervention rate as Divergence Decoding, but selects intervention positions uniformly at random rather than according to predictive divergence.

Decoding Strategy	Position Selection	Accuracy
Specialist only	No intervention	15.28% (11/72)
Generalist only	No intervention	23.61% (17/72)
Random intervention	Random positions, matched $\sim$3% rate	16.67% (12/72)
Divergence Decoding	High-divergence positions	**37.50% (27/72)**

Random intervention improves the specialist by only one additional correct question, from 11/72 to 12/72. In contrast, divergence-guided intervention increases the number of correct answers to

27/72. This result indicates that the gain cannot be explained simply by introducing a small number of generalist-generated tokens. Rather, *where* these interventions occur is critical.

Full paired per-question analysis. We further compare specialist-only decoding and Divergence Decoding on all 72 GPQA-Chemistry questions using identical prompts, generation limits, answer-extraction procedures, and evaluation scripts. Relative to the specialist-only model, Divergence Decoding:

- corrects **17** questions that are answered incorrectly by the specialist;
- changes only **1** specialist-correct question into an incorrect answer; and
- leaves the correctness of all remaining questions unchanged.

The resulting net improvement is therefore

$$17 - 1 = 16$$

additional correct questions, exactly matching the increase from 11/72 to 27/72. The improvement is thus distributed across a substantial fraction of the evaluation set rather than arising from a small number of selected examples. Moreover, because both decoding strategies use the same answer extraction and evaluation pipeline, this paired analysis also rules out answer-extraction differences as an explanation for the observed gain.

Case study 1: triggering carbon recounting. The case illustrates a representative example in which a sparse intervention redirects the specialist toward verification. For GPQA example 0, the specialist predicts **C (10 carbons)**, whereas Divergence Decoding produces the correct answer **B (11 carbons)**. Only 22 out of 671 generated tokens (3.28%) are sampled from the generalist.

At one high-divergence position, the specialist strongly favors the token *"Next"*, whereas the generalist favors *"Wait"*. After resampling, the trajectory continues with:

> "Wait, I need to count the carbons in product 1."

A later intervention triggers another verification step:

> "But wait, I made a mistake in counting the carbons in the reagent."

Importantly, the generalist does not directly provide the final carbon count. Instead, these sparse interventions alter the local reasoning trajectory, after which the specialist itself performs the recounting, revises the intermediate result, and reaches the correct answer. This example illustrates how a single intervention can influence many subsequent specialist-generated tokens through the modified autoregressive prefix.

Case study 2: switching to NMR feature matching. A second example demonstrates a change in reasoning strategy. For GPQA example 29, the specialist predicts **D**, whereas Divergence Decoding correctly selects **B, 2-(4-methylstyryl)oxirane**. Only 12 out of 514 generated tokens (2.33%) are sampled from the generalist.

Several high-divergence interventions introduce tokens such as *"Looking"*, *"H"*, and *"groups"*. Following the first intervention, the continuation shifts toward explicit spectrum-based comparison:

> "Looking at the ^{1}H-NMR spectrum, we see several signals..."

The subsequent reasoning is again predominantly generated by the specialist, but it now organizes the evidence around NMR features, including proton multiplicities and the numbers of methyl and substituent groups. This alternative reasoning trajectory enables the model to eliminate the competing structures and select the correct answer.

These examples are intended only as trajectory-level illustrations of the mechanism. The primary evidence is provided by the full paired analysis over all 72 GPQA-Chemistry questions and the matched-rate random-intervention control.

Summary. Taken together, these analyses provide a mechanistic explanation for why a low token-level intervention rate can nevertheless yield a large accuracy improvement. Divergence Decoding does not rely on the generalist to directly generate a substantial fraction of the final solution. Instead, it uses high-divergence tokens as sparse interventions that modify the autoregressive prefix and can redirect much longer specialist-generated trajectories. Specifically, we find that (i) matched-rate random interventions produce almost no improvement, (ii) divergence-guided interventions produce 17 beneficial correctness changes while causing only one harmful change, and (iii) individual interventions can trigger qualitatively different reasoning behaviors, such as recounting, verification, or alternative evidence-matching strategies. These results suggest that the effectiveness of the method depends primarily on the *location* and downstream trajectory impact of the interventions rather than on their frequency alone. "'

D Sensitivity Analysis

We compare performance on molecule understanding and editing tasks in ChemCoTBench under different setting of js-thresholds and different temperatures. Detailed results are presented in table 8 and 9.

Table 8: Sensitivity analysis of JS-threshold selection.

JS threshold	Func-Group		Scaffold		SMILES	Molecule-Edit		
	FG↓	Ring↓	Murcko↑	Ring-sys↑	Eq.↑	Add↑	Delete↑	Sub↑
Backbone: Qwen								
0.56	0.24	0.89	0.74	0.62	0.78	55	70	58
0.58	0.21	0.63	0.82	0.56	0.67	50	**80**	60
0.60	0.20	1.05	0.83	0.58	0.75	55	70	56
0.62	0.27	1.05	0.85	**0.68**	0.85	55	65	**62**
0.64	**0.19**	0.72	**0.88**	0.67	0.79	60	**80**	60
0.66	0.21	**0.55**	0.74	0.63	0.84	**65**	70	52

Table 9: Sensitivity analysis of temperature selection.

Temperature	Func-Group		Scaffold		SMILES	Molecule-Edit		
	FG↓	Ring↓	Murcko↑	Ring-sys↑	Eq.↑	Add↑	Delete↑	Sub↑
Backbone: Qwen								
0.2	0.1978	0.80	0.84	0.66	0.70	65	70	55
0.4	0.21	**0.55**	0.83	0.65	0.81	50	60	46
0.6	0.21	0.75	0.85	0.60	**0.83**	60	55	50
0.8	0.21	0.85	0.86	0.61	0.77	60	**80**	51
1.0	**0.19**	0.72	**0.88**	**0.67**	0.79	60	**80**	**60**
1.2	0.22	0.83	0.85	0.64	0.77	**70**	70	46

E Compute Resources and Model/Asset Details.

All experiments were conducted on a single machine equipped with four NVIDIA A800 GPUs, each with 80GB memory. The detailed running time is listed in Table 10.

We report the default checkpoint precision of all HuggingFace models used in our experiments in Table 11. Unless otherwise specified, the precision corresponds to the tensor type or `torch_dtype` reported by the corresponding HuggingFace model repository. In addition to model precision, all the benchmarks and evaluation protocols used in our experiments are existing public evaluation datasets. ChemCoTBench is used to evaluate molecule understanding and editing abilities, ChemBench is used to evaluate broader chemistry knowledge across multiple subdomains, and GPQA-diamond is used to evaluate expert-level scientific reasoning in chemistry, biology, and physics.

Table 10: End-to-end generation time in ChemCoTBench.

Methods	Func-Group		Scaffold		SMILES	Molecule-Edit		
	FG↓	Ring↓	Murcko↓	Ring-sys↓	Eq.↓	Add↓	Delete↓	Sub↓
Single LLM	**129.30**	194.47	252.93	**165.20**	**203.19**	224.47	247.57	**184.17**
Diver. Decode	369.77	**73.72**	**186.29**	210.11	325.33	**117.17**	**109.04**	282.08
Single / DD	0.35×	**2.64**×	**1.36**×	0.79×	0.62×	**1.92**×	**2.27**×	0.65×

Table 11: Default checkpoint precision of HuggingFace models used in our experiments.

Model	HuggingFace repository	Default precision
R1-Distill-Qwen-32B	`deepseek-ai/DeepSeek-R1-Distill-Qwen-32B`	BF16
R1-Distill-LLaMA-70B	`deepseek-ai/DeepSeek-R1-Distill-Llama-70B`	BF16
ChemDFM-R	`OpenDFM/ChemDFM-R-14B`	BF16
Chem-R	`weidawang/Chem-R-8B`	BF16
TxGemma	`google/txgemma-2b-predict`	FP32
Raman-1.7B	`think-a-tron/raman-01-1.7B`	FP32

F Case Study

We present more cases to further demonstrate that divergence decoding can correct the reasoning process at an early stage, thereby helping the model focus more directly on the chemical facts relevant to the question.

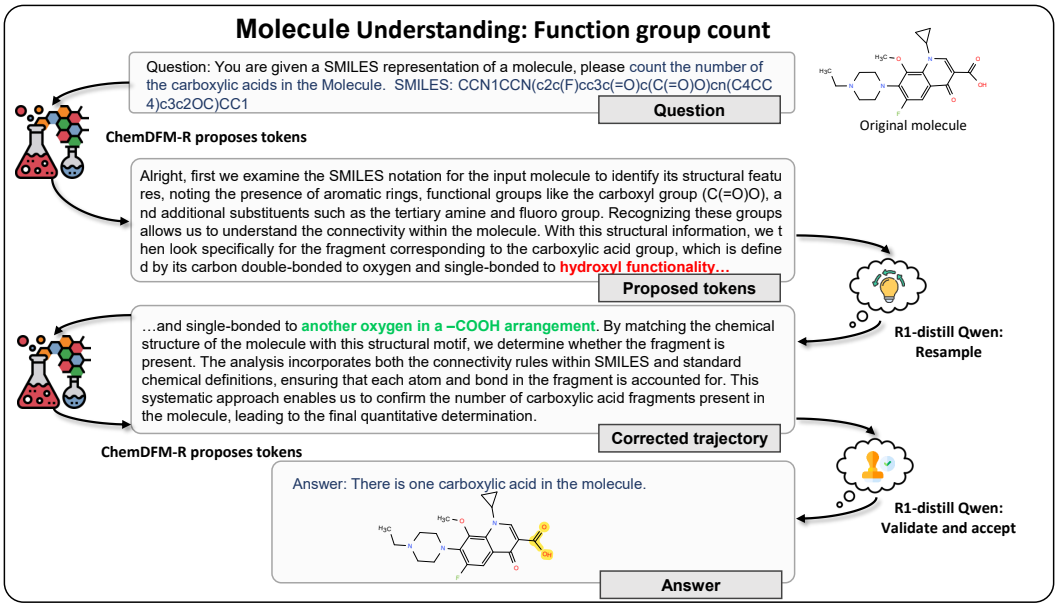

Figure 7: ***Case study:*** divergence decoding between ChemDFM and R1-Distill Qwen affects the reasoning process in the function group counting task.

G Code Availability

The anonymous source code for reproducing our experiments is available at: `https://anonymous.4open.science/r/JS-Gated-Cross-Model-Routing-2816/`

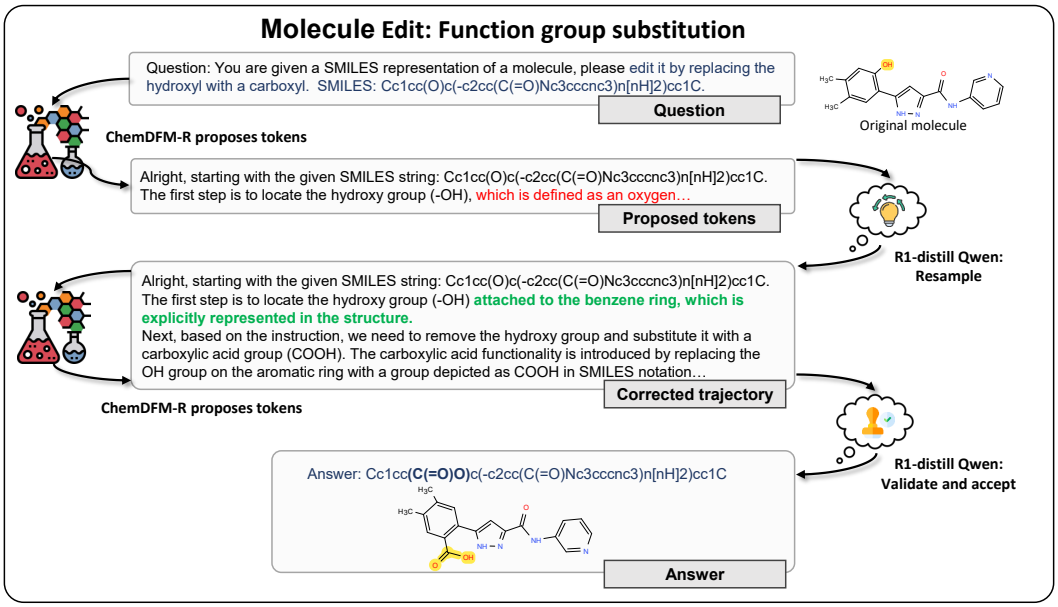

Figure 8: ***Case study:*** divergence decoding between ChemDFM and R1-Distill Qwen affects the reasoning process in the molecule editing task.

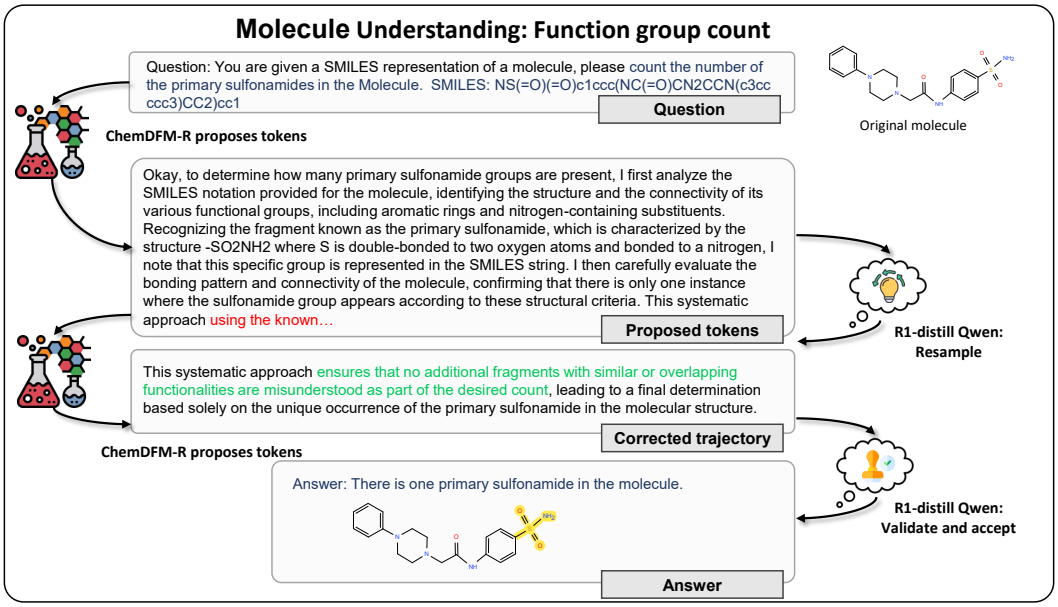

Figure 9: ***Case study:*** divergence decoding between ChemDFM and R1-Distill Qwen affects the reasoning process in the function group counting task.

H Limitations and Broader Impacts

Limitations. Although Divergence Decoding provides a training-free way to fuse domain expertise and general reasoning, it still has several limitations. First, the method relies on the complementarity between the specialist model and the general reasoning model. If the two models make highly correlated errors, or if the generalist does not provide a reliable fallback signal, the benefit of token-level routing may be limited. Second, the method introduces additional hyperparameters, including the JS threshold, block size, temperature, and the top-k support used for divergence computation. While our experiments show stable improvements under the chosen settings, different model pairs

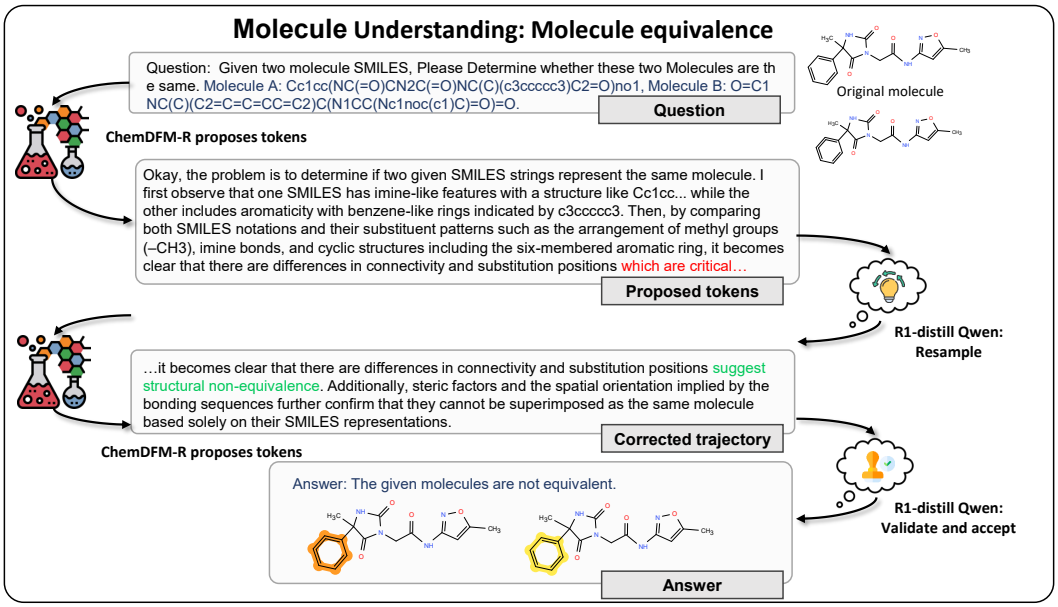

Figure 10: ***Case study:*** divergence decoding between ChemDFM and R1-Distill Qwen affects the reasoning process in the function group counting task.

or domains may require further calibration. Third, although the draft-then-verify design can offset part of the overhead, Divergence Decoding still requires loading and querying two LLMs, leading to higher memory requirements than single-model decoding.

Broader Impacts. Divergence Decoding may have positive impacts by enabling stronger scientific reasoning systems without additional training or fine-tuning. Since it composes existing specialist and generalist LLMs at inference time, it can reduce the cost of building domain-specific reasoning systems and may support scientific tasks such as chemistry understanding, molecular editing, and expert-level scientific question answering. More broadly, the method provides a practical framework for reusing heterogeneous LLM capabilities and improving reliability through adaptive collaboration.

At the same time, stronger scientific LLM systems may also introduce risks. Users may over-rely on generated reasoning traces or final answers, especially in high-stakes scientific, biomedical, or engineering settings where incorrect conclusions can have real-world consequences. In addition, improved scientific reasoning ability should not be interpreted as a substitute for expert validation. We therefore recommend using Divergence Decoding as an assistive tool rather than an autonomous decision maker, and outputs should be verified by domain experts before being used in safety-critical applications. Future work should further evaluate the robustness, uncertainty calibration, and safety behavior of divergence-based model collaboration before deployment in real-world scientific workflows.