LATENT-JEPA: ABSTRACT FUTURE PREDICTION FOR LATENT REASONING IN CHEMISTRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models offer a promising foundation for chemical reasoning, bringing together chemical knowledge and multistep problem solving. Chemical intuition can provide an initial sense of plausible outcomes before the details of a solution are fully worked out. Inspired by how such expectations complement explicit analysis, we study how continuous latent thoughts can be trained to anticipate informative aspects of future solutions without verbalizing every intermediate step. We introduce LATENT-JEPA, a framework that combines autoregressive learning with joint-embedding prediction of one or more future views. For chemical reasoning, we develop textual and molecular prediction objectives that connect latent thoughts to both subsequent reasoning and molecular outcomes. Experiments on ChemCoTBench show gains in molecular optimization and on several editing and reaction metrics. Representation analyses show that future prediction makes latent thoughts more informative about molecular outcomes and strengthens their correspondence with chemical structure. These findings support abstract future prediction as a learning principle for connecting continuous latent reasoning with scientific outcomes.

1 INTRODUCTION

Scientific discovery often begins before a complete explanation is available. Researchers form hypotheses, recognize promising directions, and develop expectations about possible outcomes while important details remain unresolved. This interplay between provisional understanding and explicit analysis has long been discussed in accounts of scientific reasoning. Poincaré & Maitland (2003) emphasized the complementary roles of intuition and logical demonstration in mathematical invention, while Polanyi & Sen (2009) argued that scientific knowledge contains an important tacit component that cannot always be fully articulated. Cognitive studies likewise suggest that people can become sensitive to meaningful coherence before they can explicitly identify it (Bowers et al., 1990). Such expectations do not replace formal reasoning or empirical validation; they help identify and organize possibilities that can subsequently be worked out and tested. We ask how a scientific reasoning model might learn an analogous capability: forming useful expectations about a solution before that solution is fully articulated.

In chemistry, intuition can provide an initial sense of plausible outcomes before the details of a solution are fully worked out. An experienced chemist may, for example, have an initial sense of what transformations are plausible or how a change in molecular structure might affect an outcome before working through all of the details. Such judgments draw on accumulated chemical knowledge and patterns learned through experience (Talanquer, 2022). They complement explicit analysis and can be refined and tested as reasoning proceeds. We draw inspiration from this anticipatory aspect of chemical intuition to study how models can learn useful expectations about future solutions.

Continuous latent reasoning provides a computational setting in which to study this idea. Chain-of-thought (CoT) expresses intermediate reasoning in language (Wei et al., 2022), whereas continuous reasoning allows models to update internal thoughts without verbalizing every step. Coconut implements hidden-state feedback (Hao et al., 2024), CODI learns continuous reasoning through self-distillation (Shen et al., 2025), and LatentChem incorporates dynamic molecular representations into chemical latent reasoning (Ye et al., 2026). We ask a complementary learning question: *can*

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.

predicting representations of future solutions improve both the information retained in latent thoughts and the reasoning they support?

An informative prediction need not specify every detail of what comes next. When a glass tips over the edge of a table, one may anticipate that it could break without predicting the position of every shard (Battaglia et al., 2013). Similarly, an improvising poet may be guided by an implicit sense of rhyme without explicitly planning the line in advance. These examples motivate studying how internal representations can anticipate meaningful aspects of a future before its full realization. Joint-Embedding Predictive Architectures (JEPAs) offer a learning principle aligned with this distinction: predict an encoded view rather than reconstruct its observable details (Dawid & LeCun, 2024; Assran et al., 2023; Bardes et al., 2024). Predictive video models have also shown sensitivity to violations of physical expectations (Garrido et al., 2025). For continuous reasoning, this motivates training latent thoughts to predict representations of subsequent reasoning and outcomes, while leaving their explicit realization to autoregressive generation.

We introduce LATENT-JEPA, which augments continuous latent reasoning with joint-embedding prediction of future representations. *The reasoner determines how latent computation unfolds, while future views specify what aspects of a developing solution it is trained to anticipate.* For chemical reasoning, we instantiate this principle with two complementary views. A textual view represents the remaining reasoning and answer in the language model's representation space, while a molecular view uses a pretrained molecular encoder to represent the chemical outcome (Soares et al., 2025). The two views capture different aspects of the same solution: different explanations can describe the same molecular outcome, and similar forms of reasoning can lead to different structures. A shared latent thought predicts both views in their respective spaces, allowing linguistic and chemical outcomes to provide complementary supervision without requiring a common target representation.

We evaluate LATENT-JEPA on ChemCoTBench (Hao et al., 2026). LATENT-JEPA improves molecular optimization and several editing and reaction metrics. Our analyses further show that predictive supervision makes individual molecular outcomes more recoverable from latent thoughts and brings recurrent representations into closer correspondence with molecular structure. Together, these results support abstract future prediction as a way to train latent thoughts to form useful expectations about future reasoning and chemical outcomes before explicit generation. More broadly, LATENT-JEPA takes a step toward the anticipatory aspect of scientific intuition, where internal computation forms useful expectations about possible outcomes before an explicit solution is produced.

Our contributions are:

- We introduce LATENT-JEPA, a framework that augments continuous latent reasoning with prediction of one or more future views, separating the reasoning architecture from the choice of target representations.
- We develop textual and molecular JEPA objectives that train latent thoughts to predict representations of subsequent reasoning and molecular outcomes, and demonstrate their complementary benefits for chemical reasoning.
- We show that future prediction makes latent thoughts more informative about molecular outcomes and strengthens their correspondence with molecular structure.

2 RELATED WORK

Predictive representation learning. Predictive representation learning provides a way to supervise abstractions of an observation. JEPA predicts embeddings of withheld views rather than reconstructing their low-level details (Dawid & LeCun, 2024; Assran et al., 2023; Bardes et al., 2024). Related non-contrastive methods study target asymmetry and representation collapse (Grill et al., 2020; Bardes et al., 2021), while LLM-JEPA combines embedding prediction with language modeling (Huang et al., 2026). Future Summary Prediction and Semformer extend prediction to representations of a future text segment (Mahajan et al., 2026; Yin et al., 2024). We apply this principle to thoughts produced during continuous reasoning, with language and molecular representations describing complementary futures.

Continuous latent reasoning. Continuous reasoning changes how a model computes before answering. Coconut feeds hidden states back into the language model and progressively replaces explicit

CoT with continuous thoughts (Hao et al., 2024); CODI and SoftCoT develop related latent reasoning methods (Shen et al., 2025; Xu et al., 2025). Looped Transformers increase computation through repeated shared transformations (Saunshi et al., 2025).

Chemical reasoning and representations. Chemical reasoning connects linguistic explanations with molecular structure and task outcomes (Narayanan et al., 2026; Wang et al., 2026; Song et al., 2026; Feng et al., 2026; Zhu et al., 2026). LatentChem incorporates molecular representation updates into latent reasoning (Ye et al., 2026), while ChemCoTBench evaluates chemical transformations and reasoning tasks (Hao et al., 2026). A complementary line of work learns molecular representations from structure and text (Chithrananda et al., 2020; Ross et al., 2022; Liu et al., 2023; Soares et al., 2025; Zhao et al., 2026). Together with language-model embeddings (BehnamGhader et al., 2024), these representations provide ways to describe both a solution and its molecular outcome. We use them as future prediction targets; Appendix A further connects target design with scientific representation learning.

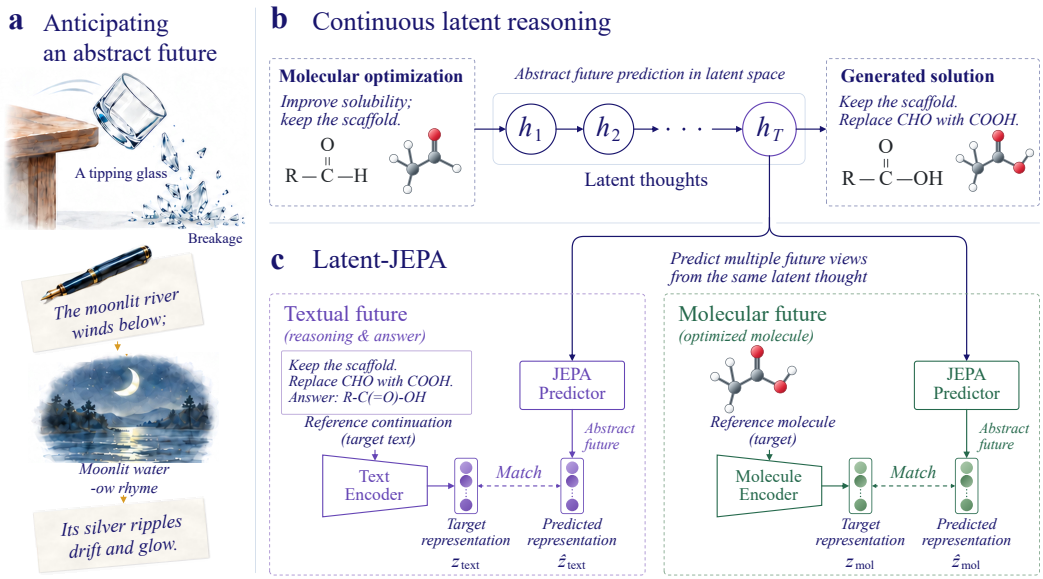


Figure 1: Overview of LATENT-JEPA. **(a)** Anticipating breakage or a poem's theme and rhyme illustrates prediction at the level of useful abstractions. The verse gives one possible continuation of the first line. **(b)** A continuous reasoner updates latent thoughts before generating the remaining reasoning and molecular outcome. **(c)** The final latent thought also predicts textual and molecular representations of the reference solution during training. The molecular optimization illustration replaces an aldehyde with a carboxyl group; R denotes the unchanged scaffold.

3 LATENT-JEPA

We aim to train latent thoughts to form useful expectations about a solution before it is fully expressed. To this end, LATENT-JEPA combines autoregressive learning with prediction of representations of subsequent reasoning and outcomes (Figure 1). Let W_t denote the reasoner's current workspace and h_t a continuous representation of intermediate reasoning read from that workspace:

$$W_0 = I_\theta(x), \qquad h_t = R_\theta(W_t), \qquad W_{t+1} = U_\theta(W_t, h_t). \tag{1}$$

Here x is the input, I_θ initializes the workspace, R_θ reads the thought, and U_θ advances the reasoning process for $0 \le t < K$. A decoder generates the answer from W_K. The workspace can include the reasoner's attention cache or domain-specific state, allowing this formulation to cover different continuous reasoning architectures.

Definition 1 (Latent-JEPA). *A Latent-JEPA architecture consists of a continuous latent reasoner and, at each supervised step $t \in \mathcal{I} \subseteq \{0, \dots, K\}$, a finite family of future-view maps, target encoders, and predictors $\{(\phi_{t,j}, E_j, P_j)\}_{j=1}^{N_t}$, where $N_t \ge 1$. Given a reference continuation or outcome $F_{>t}$*

available during training, $\phi_{t,j}$ selects a future view, E_j encodes it, and P_j predicts its representation from h_t:

$$v_{t,j} = \phi_{t,j}(F_{>t}), \qquad z_{t,j} = E_j(v_{t,j}), \qquad \widehat{z}_{t,j} = P_j(h_t, q_{t,j}), \tag{2}$$

where $q_{t,j}$ optionally specifies a prediction horizon or query. The latent thought h_t depends only on computation preceding the future view. Each predictor maps the thought into its target encoder's representation space, where a discrepancy $d_j(\widehat{z}_{t,j}, z_{t,j})$ supplies the predictive learning signal.

A future view can describe an intermediate reasoning step, the answer, or a domain-specific outcome. Different views can use different encoders and representation spaces; multiple horizons can also share an encoder and a query-conditioned predictor. The framework leaves these choices open so that new target representations and reasoning architectures can extend the same predictive structure. The definition also leaves the use of the components after training open. Encoders and predictors may support downstream planning or evaluation using information available at inference. In this work, we use the predictive branches to supervise latent thoughts during training, without changing the reasoner's generation procedure.

Predictive learning. We train the predictors and latent reasoner with

$$\mathcal{L}_{\mathrm{LJ}} = \mathbb{E}\left[\sum_{t\in\mathcal{I}}\sum_{j=1}^{N_t} w_{t,j}\, d_j(P_j(h_t, q_{t,j}), \mathrm{sg}(E_j(v_{t,j})))\right], \qquad w_{t,j} \geq 0, \tag{3}$$

where the expectation is over training examples and $w_{t,j}$ controls the contribution of each view and reasoning step. Stop-gradient prevents the prediction loss from directly updating the target branch. The encoder may remain frozen, follow a momentum update, or share parameters with the reasoner; these choices determine how the targets evolve during learning. Target calibration, when used, forms part of the map E_j. A simple choice is to supervise the final latent thought, which can integrate information from preceding latent steps without accessing the future continuation. We adopt this choice, setting $\mathcal{I} = \{K_s\}$, and combine the predictive loss with the existing training objective:

$$\mathcal{L}_{\mathrm{adapt}} = \mathcal{L}_{\mathrm{AR}} + \mathcal{L}_{\mathrm{base}} + \lambda_{\mathrm{LJ}}\mathcal{L}_{\mathrm{LJ}}. \tag{4}$$

Here $\mathcal{L}_{\mathrm{base}}$ denotes any architecture-specific auxiliary losses. The chemical construction below specifies the future views, encoders, and normalization of their contributions.

3.1 Target asymmetry and geometry

The target branch observes information withheld from the latent thought. Stop-gradient makes the prediction asymmetric: the predictor and reasoner learn to approach the target representation without directly optimizing the target through the same loss. When encoder parameters are shared with the reasoner, they still evolve through autoregressive and prediction-side updates. A frozen encoder instead provides a fixed representation space. Target geometry also determines which prediction errors are informative. If embeddings share a dominant common direction, high cosine similarity may reveal little about the corresponding example. We center target representations using means estimated from training data and apply ℓ_2 normalization to expose variation across examples. We evaluate this geometry through pairwise similarity and effective rank, and test sample-specific prediction using shuffled targets. These are diagnostic controls; the training objective in Equation (3) is non-contrastive.

4 Latent-JEPA for Chemical Reasoning

We use Coconut-Chem, the chemical adaptation of Coconut introduced by Ye et al. (2026), as a representative latent reasoner. We first describe its latent computation, then define textual and molecular prediction objectives. Additional instantiations of Latent-JEPA are given in Appendices C and F.

4.1 Recurrent chemical reasoning

Let x denote the input prompt and any observable molecular information. A demonstrated solution consists of reasoning steps $c_{1:M}$ and a final answer y, where M is the number of explicit reasoning steps. Following Coconut (Hao et al., 2024), Coconut-Chem progressively replaces an increasing

prefix of these steps with continuous thoughts. At curriculum stage s, the first s reasoning steps are replaced by a latent sequence $h_{1:K_s}$, where K_s denotes the latent reasoning length at that stage. The model then autoregressively generates the remaining reasoning $c_{s+1:M}$ followed by y. For stages with $K_s > 0$, we supervise the final latent thought,

$$h_T \equiv h_{K_s}, \tag{5}$$

where h_{K_s} is the final-layer hidden state at the last continuous-thought position, before the end-of-reasoning delimiter. The causal attention mask prevents this state from accessing the remaining reasoning or answer tokens. The autoregressive objective is

$$\mathcal{L}_{\mathrm{AR}} = -\sum_{t \in \mathcal{S}} \log p_\theta(w_t \mid x, h_{1:K_s}, w_{<t}), \tag{6}$$

where w_t is a continuation token and $\mathcal{S}$ indexes the supervised tokens in $c_{s+1:M}$ and y. Gradients from Equation (6) train latent computation through continuation prediction. LATENT-JEPA additionally supervises h_T with the textual and molecular targets available for each example. Thus, $\mathcal{I} = \{K_s\}$ at stages with continuous thoughts. The curriculum schedule is specified in Appendix B.

4.2 DECODER-NATIVE CONTINUATION VIEW

The textual view trains the final latent thought to anticipate the remaining solution in the generator's own representation space. We construct these decoder-native targets with a separate causal forward pass over the reference continuation $(c_{s+1:M}, y)$. This pass shares the current decoder parameters with the reasoner, but receives only the continuation tokens, without the prompt, source-molecule features, or latent thoughts. We apply stop-gradient to the target representations. From this single pass, we read the final-layer hidden state at the last content token of each reasoning step and of the answer. A shared predictor $P_{\mathrm{text}}(h_T, \rho)$ predicts these representations from the same final latent thought, with ρ specifying the relative horizon in the remaining solution. We center the targets using separate running means for reasoning and answers, then apply ℓ_2 normalization. The objective $\mathcal{L}_{\mathrm{reason}}$ minimizes cosine distance $d(a, b) = 1 - \cos(a, b)$, assigning equal total weight to reasoning and answer prediction when both are available. Appendix B gives the encoding details, horizon queries, and loss normalization.

4.3 STRUCTURE-NATIVE OUTCOME VIEW

The textual view trains latent thoughts to anticipate the continuation in the language model's representation space. We complement this supervision with a molecular view of the future outcome, using a pretrained molecular encoder to provide a structural abstraction of the solution. For examples with a molecular outcome, let m_i^{chem} denote the reference molecule or molecular state and $\tilde{e}_i = E_{\mathrm{SMI}}(m_i^{\mathrm{chem}})$ its representation from a frozen SMI-TED-based encoder (Soares et al., 2025). To make differences among outcomes more visible to the cosine objective, we subtract the mean over molecular training targets and apply ℓ_2 normalization:

$$z_i^{\mathrm{chem}} = \mathrm{sg}\left(\frac{\tilde{e}_i - \mu_{\mathrm{train}}}{\|\tilde{e}_i - \mu_{\mathrm{train}}\|_2 + \epsilon}\right). \tag{7}$$

The training mean μ_{train} remains fixed. A separate predictor produces $\hat{z}_i^{\mathrm{chem}} = P_{\mathrm{chem}}(h_T)$, with

$$\mathcal{L}_{\mathrm{chem}}^{(i)} = d(\hat{z}_i^{\mathrm{chem}}, z_i^{\mathrm{chem}}). \tag{8}$$

Examples with text-only answers receive autoregressive and applicable textual prediction supervision, but no molecular prediction objective. Appendix B describes the preparation of molecular targets. The two predictors map the same latent thought into separate target spaces, allowing textual and molecular representations to preserve different aspects of the solution.

4.4 OBJECTIVE AND INFERENCE

We retain Coconut-Chem's autoregressive and counterfactual objectives and add the two predictive losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{AR}} + \mathcal{L}_{\mathrm{CF}} + \lambda_{\mathrm{reason}}\mathcal{L}_{\mathrm{reason}} + \lambda_{\mathrm{chem}}\mathcal{L}_{\mathrm{chem}}, \tag{9}$$

Table 1: Molecule editing and optimization on ChemCoTBench. Editing reports correctness (%); optimization reports mean property improvement. All metrics are higher-is-better. Dark-red bold marks LATENT-JEPA scores equal to or better than Coconut-Chem. Optimization success rates appear in Table 11.

	Molecule editing (%)			Property improvement					
Method	Add	Delete	Sub.	LogP	Solubility	QED	DRD2	JNK3	GSK3-β
General-purpose language models									
GPT-4o	80.0	80.0	65.0	-0.09	0.92	0.13	0.07	-0.02	0.00
o1-mini	55.0	80.0	58.3	-0.42	1.78	0.07	-0.03	-0.10	-0.08
o3-mini	65.0	55.0	80.0	0.26	0.81	0.21	0.19	-0.03	0.01
Gemini 2.5 Pro (thinking)	100.0	85.0	81.7	-0.22	1.06	0.28	0.36	-0.02	0.06
Claude 3.7 Sonnet (thinking)	85.0	80.0	83.4	0.41	0.37	0.12	0.17	0.01	0.02
Open and chemistry-aligned language models									
Qwen3-8B	4.0	52.0	1.0	0.00	0.00	0.00	0.00	–	–
Qwen3-8B (SFT)	55.0	76.0	37.0	0.15	0.48	0.10	0.04	–	–
Molecular alignment	62.0	61.0	47.0	0.35	0.87	0.14	0.20	–	–
Molecular alignment + CoT	63.0	80.0	49.0	0.51	0.82	0.14	0.12	–	–
Molecular alignment + CoT + GRPO	70.0	88.0	60.0	0.67	1.17	0.19	0.19	–	–
Latent reasoning models									
LatentChem	65.0	78.3	56.7	0.7051	0.7829	0.1809	0.1732	0.0358	0.1481
Coconut-Chem	71.7	76.7	65.6	0.7404	0.9259	0.2181	0.2636	0.1032	0.2285
Coconut-Chem + LATENT-JEPA	**78.3**	**86.7**	61.1	**0.8318**	**0.9659**	**0.2296**	**0.2848**	**0.1296**	**0.2515**

where $\mathcal{L}_{\text{CF}}$ is Coconut-Chem's molecular-input counterfactual objective. Each predictive loss is averaged over its eligible examples and uses unperturbed inputs; Appendix B gives the details.

At inference, the model follows the underlying reasoner's recurrent computation and decoding procedure. Prediction heads and target encoding serve only as training supervision.

5 EXPERIMENTS

We first evaluate whether anticipating future solutions improves chemical reasoning. We then examine what latent thoughts learn about molecular outcomes and which target choices provide useful predictive supervision.

5.1 EXPERIMENTAL SETUP

We implement Coconut-Chem with Qwen3-8B-Base (Yang et al., 2025), initialized from LatentChem's chemistry instruction-tuned model and retaining its SMI-TED molecular input encoder and adapter. Coconut-Chem and its dual-view LATENT-JEPA variant share initialization, training data, and a curriculum with $0, 2, 4$, and 6 continuous thoughts, trained for three epochs per phase. We also evaluate answer prediction in LatentChem and compare future views in Vanilla Coconut, which receives molecular SMILES as text without source ChemTokens. Architecture-specific training settings are detailed in Appendix B. We evaluate on ChemCoTBench (Hao et al., 2026; Ye et al., 2026), which covers molecular optimization, editing, molecular understanding, scaffold abstraction, and forward-reaction prediction. We use the corresponding property, accuracy, count-error, and structural-similarity metrics, including fingerprint Tanimoto similarity (FTS) for reactions. Each checkpoint generates three responses per problem at temperature 0.7 and top-$p = 0.9$; we average their scores without best-of-three selection. Our within-architecture comparisons share evaluation settings, while published baselines retain their original protocols. Appendix B documents baseline sources, and Appendix L specifies the tasks, metrics, and scoring procedure.

5.2 MAIN RESULTS

Future prediction improves chemical reasoning. We first test the central premise of LATENT-JEPA by comparing chemical task performance with and without future prediction. Tables 1 and 2 show improvements on most reported metrics. The gains are most consistent in molecular optimization,

Table 2: Molecular understanding and forward-reaction prediction. FG and Ring report count MAE; Murcko and FTS report similarity; ring-system accuracy and reaction validity are percentages. Dark-red bold marks LATENT-JEPA scores equal to or better than Coconut-Chem. Ring-system scoring and the treatment of published reference scores are specified in Appendix L.2.

	Molecular understanding				Major product		Byproduct	
Method	FG ↓	Ring ↓	Murcko ↑	Ring-sys. (%) ↑	FTS ↑	Valid (%) ↑	FTS ↑	Valid (%) ↑
General-purpose language models								
GPT-4o	0.17	1.35	0.21	80.0	0.58	–	0.20	–
o1-mini	0.21	1.25	0.25	61.7	0.31	–	0.17	–
o3-mini	0.13	0.60	0.39	75.0	0.71	–	0.27	–
Gemini 2.5 Pro (thinking)	0.11	0.60	0.51	87.5	0.89	–	0.51	–
Claude 3.7 Sonnet (thinking)	0.21	1.60	0.40	80.0	0.87	–	0.31	–
Open and chemistry-aligned language models								
Qwen3-8B	7.03	1.03	0.02	36.0	0.02	–	0.00	–
Qwen3-8B (SFT)	0.12	0.45	0.40	65.0	0.35	–	0.07	–
Molecular alignment	0.14	0.10	0.71	83.0	0.57	–	0.18	–
Molecular alignment + CoT	0.08	0.20	0.66	63.0	0.44	–	0.11	–
Molecular alignment + CoT + GRPO	0.07	0.15	0.65	83.0	0.63	–	0.20	–
Latent reasoning models								
LatentChem	0.0967	0.1667	0.7220	73.89	0.5099	74.00	0.1728	42.67
Coconut-Chem	0.0667	0.1500	0.7783	80.00	0.6408	86.67	0.1518	45.33
Coconut-Chem + LATENT-JEPA	**0.0667**	**0.1333**	**0.8065**	**81.11**	**0.6492**	**88.33**	**0.1806**	**45.33**

where mean property improvement increases on all six objectives. Improvements also extend to structural analysis and reaction prediction, while editing results are more mixed. The separate optimization success rates improve on most objectives (Table 11 in Appendix M). Together, these findings support training latent thoughts to anticipate abstract outcomes as a useful complement to autoregressive learning for chemical reasoning.

The two future views contribute differently. A solution's textual continuation and molecular outcome describe different aspects of its future. We therefore compare textual-only, molecular-only, and dual prediction to understand the contribution of each view (Table 8 in Appendix F). Molecular prediction is particularly effective for editing, whereas adding textual prediction improves most property objectives and the reaction metrics relative to molecular prediction alone. The combined objective thus extends the benefits across a broader range of chemical tasks without being best on every metric. These results motivate choosing predictive views according to the aspects of a solution that a task requires; the loss coefficients for each comparison are specified with the full results.

Future prediction extends beyond Coconut-Chem. To examine whether the benefits depend on Coconut-Chem's particular reasoning setup, we also apply the objective to LatentChem and Vanilla Coconut. In LatentChem, predicting the answer representation improves most evaluated metrics under the same supervised adaptation budget (Table 4 in Appendix C). In Vanilla Coconut, which has no source molecular encoder, textual prediction gives the strongest average rank, while the other views benefit different subsets of tasks (Table 9 in Appendix F). Future prediction can therefore support more than one reasoning setup, with different setups favoring different predictive views.

5.3 LEARNING PREDICTIVE LATENT THOUGHTS

To understand what accompanies the task gains, we examine whether latent thoughts distinguish individual molecular outcomes, reflect their structural relationships, and change through recurrent computation.

Latent thoughts distinguish molecular outcomes. Anticipating an outcome requires more than recognizing the type of problem. We therefore test whether a pre-answer thought contains information about its particular molecular outcome. We fit a separate ridge probe to each model and recurrent depth, then evaluate predictions on examples excluded from both reasoner training and probe fitting (Appendix K). Figure 2 shows higher prediction cosine and within-task retrieval with LATENT-JEPA; final-update cosine increases from 0.186 to 0.397. Predictions also remain more similar to their corresponding targets than to targets shuffled within the same task. This comparison indicates that the improvement concerns individual outcomes, rather than task membership alone. The example-wise

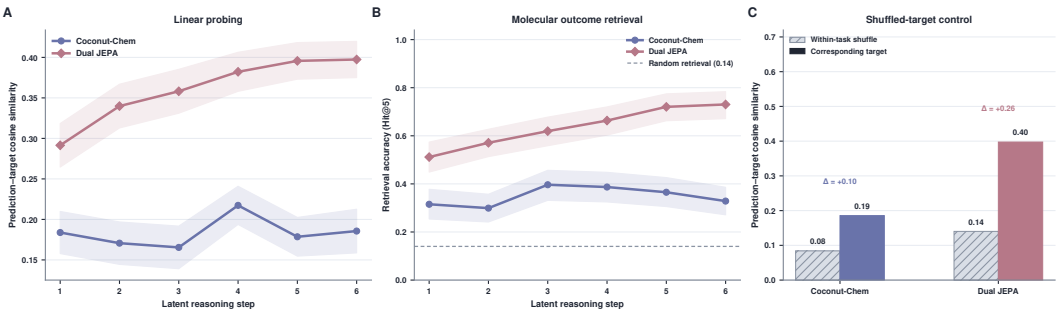


Figure 2: Molecular-outcome prediction on examples excluded from reasoner training and probe fitting. **A:** Ridge-probe cosine similarity across recurrent updates. **B:** Within-task Hit@5; the dashed line denotes uniform retrieval. **C:** Corresponding-target similarity versus within-task shuffled targets. Bands summarize evaluation-example uncertainty; Appendix K specifies the protocol.

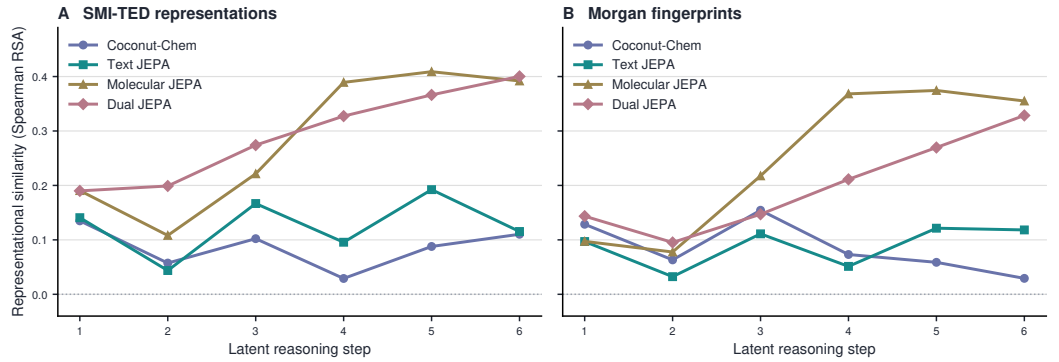


Figure 3: Representational similarity between recurrent feedback embeddings and molecular outcomes. **A:** Agreement with calibrated SMI-TED target distances. **B:** Agreement with Morgan fingerprint Tanimoto distances. Curves are task-averaged within-task Spearman RSA and single-run point estimates. Appendix K specifies representations, configurations, and distance measures.

and task-wise analyses show that these gains extend across most analyzed examples and multiple task families (Figures 10 and 12 in Appendix I).

Recurrent representations reflect molecular relationships. Recovering an outcome with a learned probe does not reveal how the recurrent representations themselves are organized. We therefore compare pairwise distances between recurrent feedback embeddings and molecular outcomes using within-task representational similarity analysis (RSA), without fitting a mapping (Appendix K.3). In Figure 3, molecular-only and dual prediction show substantially stronger agreement with SMI-TED geometry than Coconut-Chem or textual prediction. Molecular-only prediction also gives the strongest agreement with Morgan fingerprints, which are not used as prediction targets. These results suggest that structure-native future targets can shape recurrent representations toward molecular relationships that extend beyond the target space itself. More generally, the choice of JEPA target determines which aspects of a future solution the latent state is encouraged to emphasize. To examine the contribution of the molecular view more directly, we compare final thoughts while holding each active view's coefficient fixed. In this training-cohort analysis, molecular-only prediction gives the highest outcome retrieval and dual prediction the strongest agreement with molecular task-centroid relations (Figure 7 in Appendix H). Molecular and dual prediction also improve retrieval in Vanilla Coconut (Figure 8). These results motivate using scientific encoders to supervise future outcomes, alongside their role in encoding model inputs.

Outcome representations develop through recurrent computation. We further examine how these representations change as latent reasoning proceeds. The dual model's probe cosine and Hit@5 generally increase with depth, with smaller changes near the final updates (Figure 2). Its structural agreement also increases overall (Figure 3), while the task-wise RSA gains follow different depth profiles (Figure 11 in Appendix I). We also examine the explicit continuations and find task-dependent length changes rather than a general shortening (Figure 13 in Appendix J).

Table 3: Effect of target centering on molecular optimization. All entries are mean property improvements (↑). Full task results are reported in Table 6.

Target calibration	LogP	Solubility	QED	DRD2	JNK3	GSK3-β
Both views centered	0.8318	0.9659	0.2296	0.2848	0.1296	0.2515
Without molecular centering	0.8701	0.9703	0.2151	0.2645	0.1163	0.2329
Without text centering	0.7854	0.8311	0.2036	0.2644	0.1174	0.2232

Constructing informative prediction targets. The preceding analyses concern what latent thoughts learn. We finally examine how the target construction affects this supervision. A useful predictive objective should distinguish different outcomes, rather than reward a component shared by nearly every target. We inspect the molecular target geometry and compare predictions using original and shuffled latent sources. We find that raw SMI-TED targets share a dominant direction, so uncentered targets can yield small prediction distances for both source conditions. Centering and normalization make differences among outcomes more visible to the cosine objective (Figure 4 in Appendix E). Retaining both centers also gives the strongest average rank in the task ablation, with improvements on most metrics over either removal (Table 3 and the complete results in Table 6). These observations support considering target geometry together with the prediction loss when constructing scientific supervision.

The textual targets raise a related design question because their encoder shares parameters with the reasoner and changes during training. We examine whether a momentum encoder or a change in target dropout offers better supervision, retaining both predictive views in each variant (Table 5 in Appendix D). Each variant improves a majority of the reported metrics over Coconut-Chem, and the online encoder with dropout has the strongest average rank among them. Thus, useful predictive supervision is not confined to one target-update recipe. The shared online encoder provides an effective source of textual futures, complementing the fixed molecular representation. Together with the target-space analysis in Appendix G, these findings motivate studying which scientific representations provide informative targets for reasoning.

Discussion. Our results suggest a computational perspective on intuitive reasoning in continuous latent models. Scientific intuition often involves forming an expectation of a plausible outcome before the details of a solution are fully worked out; LATENT-JEPA instantiates this idea by training latent computation to anticipate abstract representations of future reasoning and outcomes before explicit generation. The representation analyses show that such supervision makes molecular outcomes more accessible from latent thoughts and reshapes recurrent computation toward chemically meaningful structure, suggesting that intermediate states can be trained not only to support eventual token generation, but also to encode useful expectations about what comes next. From this perspective, the choice of prediction target determines which aspects of the future are emphasized, while the predictive objective provides a more general mechanism for shaping the internal computation that precedes generation. More broadly, future prediction may influence not only what information is present in individual latent thoughts, but also how latent thoughts evolve through recurrent computation and how the underlying latent manifold is organized around possible future outcomes. Understanding these dynamics may provide a useful path toward intuitive reasoning in scientific models, where useful expectations emerge before a solution is explicitly worked out. Future work can further study how such latent trajectories develop across reasoning depth and whether different stages of computation benefit from different prediction horizons or levels of abstraction.

6 Conclusion

We introduced LATENT-JEPA, a framework for learning latent thoughts through prediction of future representations. Its formulation connects a continuous reasoner to one or more future views and allows different encoders and predictors to define the supervision. In chemistry, we train latent thoughts to anticipate both the remaining reasoning and molecular outcomes. Experiments show gains in chemical task performance and stronger molecular information in latent thoughts. These findings motivate future prediction as a learning objective for continuous reasoning and open further study of how future representations can guide reasoning and planning.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pp. 1298–1312. PMLR, 2022.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the national academy of sciences*, 110(45):18327–18332, 2013.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

Kenneth S Bowers, Glenn Regehr, Claude Balthazard, and Kevin Parker. Intuition in the context of discovery. *Cognitive psychology*, 22(1):72–110, 1990.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Anna Dawid and Yann LeCun. Introduction to latent variable energy-based models: a path toward autonomous machine intelligence. *Journal of Statistical Mechanics: Theory and Experiment*, 2024 (10):104011, 2024.

Jinjia Feng, Wenda Wang, and Zhewei Wei. Motifagent: Learning molecular assembly through multi-agent collaboration for chemical language understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40700–40739, 2026.

Quentin Garrido, Nicolas Ballas, Mahmoud Assran, Adrien Bardes, Laurent Najman, Michael Rabbat, Emmanuel Dupoux, and Yann LeCun. Intuitive physics understanding emerges from self-supervised pretraining on natural videos. *arXiv preprint arXiv:2502.11831*, 2025.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Li Hao, He Cao, Bin Feng, Daniel Shao, Robert Tang, Zhiyuan Yan, Yonghong Tian, Li Yuan, and Yu Li. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *Advances in Neural Information Processing Systems*, 38, 2026.

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. *arXiv preprint arXiv:2412.06769*, 2024.

Hai Huang, Yann LeCun, and Randall Balestriero. Llm-jepa: Large language models meet joint embedding predictive architectures. In *International Conference on Learning Representations*, volume 2026, pp. 105717–105737, 2026.

Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023.

Divyat Mahajan, Sachin Goyal, Badr Youbi Idrissi, Mohammad Pezeshki, Ioannis Mitliagkas, David Lopez-Paz, and Kartik Ahuja. Beyond multi-token prediction: Pretraining llms with future summaries. In *International Conference on Learning Representations*, volume 2026, pp. 64342–64364, 2026.

Siddharth Narayanan, James Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Michael Pieler, Sam Rodriques, and Andrew White. Training a scientific reasoning model for chemistry. *Advances in Neural Information Processing Systems*, 38:157671–157710, 2026.

Henri Poincaré and Francis Maitland. *Science and method*. Courier Corporation, 2003.

Michael Polanyi and Amartya Sen. *The tacit dimension*. University of Chicago press, 2009.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, volume 2025, pp. 14855–14881, 2025.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025.

Eduardo Almeida Soares, Emilio Ashton Vital Brazil, Victor Yukio Shirasuna, Dmitry Zubarev, Renato Cerqueira, and Kristin Schmidt. Smi-ted: A large-scale foundation model for materials and chemistry. In *International Conference on Learning Representations*, 2025.

Xiaozhuang Song, Xuanhao Pan, Xinjian Zhao, Hangting Ye, Shufei Zhang, Jian Tang, and Tianshu Yu. Aot*: Efficient synthesis planning via llm-empowered and-or tree search. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34727–34758, 2026.

Vicente Talanquer. The complexity of reasoning about and with chemical representations. *Jacs Au*, 2 (12):2658, 2022.

Weida Wang, Benteng Chen, Di Zhang, Wanhao Liu, Ben Gao, Shuchen Pu, Shuzhou Sun, Jin Zeng, Tianshu Yu, Wanli Ouyang, et al. Chem-r: Learning to reason as a chemist. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 12291–12302, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351, 2025.

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.

Xinwu Ye, Yicheng Mao, Yuxuan Liao, Jia Zhang, Yimeng Liu, Li Hao, Fang Wu, Zhiwei Li, Zehong Wang, Zhiyuan Liu, et al. Latentchem: From textual cot to latent thinking in chemical reasoning. *arXiv preprint arXiv:2602.07075*, 2026.

Yongjing Yin, Junran Ding, Kai Song, and Yue Zhang. Semformer: Transformer language models with semantic planning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18669–18680, 2024.

Xinjian Zhao, Xiangru Jian, Yaoyao Xu, Xiaozhuang Song, Wei Pang, Lei Bai, and Tianshu Yu. Molemb: Multimodal large language models can be strong molecular embedding models. *arXiv preprint arXiv:2608.23646*, 2026.

Yanqiao Zhu, Jingru Gan, Xiaoqi Sun, Fang Sun, Yidan Shi, Md Mofijul Islam, Chao Shang, Wenhao Gao, Connor W Coley, Yizhou Sun, et al. Retroagent: Harnessing llms to search over structured memory for agentic retrosynthesis planning. *arXiv preprint arXiv:2607.14512*, 2026.

Ethics Statement

The work studies models for chemical reasoning. Such models can produce incorrect or unsafe chemical suggestions and should not be treated as a substitute for expert review, laboratory validation, or established safety procedures. Our experiments use benchmark data and computational evaluation; they do not involve human subjects or wet-laboratory execution.

Reproducibility Statement

The appendix specifies the training objectives, target preprocessing, curriculum, and evaluation protocol. The supplementary material provides the corresponding scripts for data processing, training, evaluation, and representation analysis, together with the paper configurations and instructions for the required external assets. Within each architecture, the baseline and predictive variants share initialization and supervised training budget, as detailed in Appendix B.

AI use Statement

Generative AI tools were used to assist code review, experiment documentation, and language editing. Generated code and analyses were reviewed and tested by the authors.

A Additional Related Work

Predictive targets for JEPA. When the prediction target is a representation rather than the observation itself, constructing that target becomes part of the learning problem. The encoder's available context and the level of detail it represents help define what must be predicted. I-JEPA examines target scale and contextual information when predicting representations of image regions (Assran et al., 2023). V-JEPA studies feature prediction in video (Bardes et al., 2024), while data2vec predicts contextualized representations from masked inputs across speech, vision, and language (Baevski et al., 2022). These studies provide a basis for considering what the target encoder observes and how it represents that information. Its update rule, including momentum updates, also determines how the target space evolves during training. For chemical reasoning, we construct targets from two views of a future solution: its textual continuation and the molecular object it denotes. The encoder-update and calibration comparisons in Appendix D, E examine how these design choices affect predictive supervision in our setting.

Scientific and multimodal representations. Scientific representation models offer several ways to describe a molecular outcome. MoleculeSTM relates structures to textual descriptions (Liu et al., 2023), while SMI-TED learns from molecular strings (Soares et al., 2025). Their structural and semantic representations motivate using scientific encoders to define prediction targets, alongside encoding observed inputs. In our molecular branch, a frozen SMI-TED-based encoder represents the reference outcome during training; the language model supplies the textual view of the remaining solution. Separate predictors map the same latent thought to the two spaces. More broadly, this use of domain encoders raises the question of which aspects of a scientific outcome provide useful supervision for a reasoning task. Comparing encoders of the same outcome examines the choice of representation, while selecting different outcome views examines what information to predict.

B Implementation Details

Optimization and hyperparameters. Coconut-Chem uses AdamW with learning rate 2×10^{-4}, eight GPUs, four examples per device, and no gradient accumulation. Each curriculum phase is trained for three epochs. We use $\lambda_{\text{reason}} = 0.025$ and $\lambda_{\text{chem}} = 0.01$, with a decoding temperature of 0.7 and top-$p = 0.9$. The loss-coefficient comparisons vary $\lambda_{\text{reason}} \in \{0.025, 0.05, 0.075\}$ and $\lambda_{\text{chem}} \in \{0.005, 0.01, 0.02\}$ one factor at a time around $(0.05, 0.01)$. Text-predictor hidden widths of 512 and 1536 were also compared with the 912-dimensional reference. These comparisons use the benchmark evaluation protocol described in Appendix L. The representation-analysis configurations are specified in Section K, including the text coefficient of 0.05 used for the task-organization and recurrent-geometry comparisons.

Curriculum. The reported curriculum allocates $c_{\text{thought}} = 2$ continuous thoughts per replaced reasoning step and uses latent lengths $K_s \in \{0, 2, 4, 6\}$. Each phase is trained for three epochs. The baseline and its LATENT-JEPA variant use the same schedule.

Supervised latent state. The predictor reads the final-layer hidden state at the last continuous-thought position, before the end-of-reasoning delimiter. The initial, fully textual curriculum phase has no latent thoughts and receives neither predictive loss.

Initialization and reference results. Coconut-Chem starts from LatentChem's chemistry-aware CoT model, called Stage 2 in its training procedure. LatentChem adaptation starts after latent-module training (its Stage 3) and precedes GRPO. Our adapted baseline and predictive model receive the same additional supervised training. The answer-prediction coefficient is 0.05, and this transfer experiment has no molecular predictive loss. We update the latent reasoning module and decoder adapters while keeping the molecular updater frozen. Published values in Tables 1 and 2 use the ChemCoTBench tables of Hao et al. (2026) for GPT-4o (2024-11-20), o1-mini (2024-09-12), o3-mini (the source labels it 2025-01-03), Gemini 2.5 Pro with thinking, and Claude 3.7 Sonnet with thinking. We retain their original evaluation settings. Closed-model editing scores in ChemCoTBench use Pass@1. For our models, we average scores from three sampled responses per problem, evaluating each response separately as described in Appendix L.2. The Qwen and alignment rows come from the fifth arXiv revision of Ye et al. (2026). Molecular alignment, molecular alignment + CoT, and molecular alignment + CoT + GRPO correspond to that paper's Stage 1, Stage 1+2, and Stage 1+2+4, respectively. Their original evaluation protocol differs from our reproduced comparisons.

Target visibility. The textual prediction branch encodes only the independently tokenized residual CoT and wrapped answer. It cannot attend to the prompt, source molecules, or latent states. Target hidden states are detached before centering and loss computation. The frozen SMI-TED-derived encoder reads only the reference chemical target (a molecule or scaffold-like structure, as applicable).

Textual target construction and loss. At curriculum stage s, the independent target sequence contains the remaining reasoning, the answer enclosed by its answer delimiters, and an end-of-sequence token:

$$f^{\text{text}} = [c_{s+1}, \ldots, c_M, \langle\text{answer}\rangle, y, \langle/\text{answer}\rangle, \text{EOS}]. \tag{10}$$

We encode this sequence in a separate causal forward pass using the current decoder parameters, with gradients disabled and dropout enabled by default. The pass does not reuse the reasoner's attention cache. Each reasoning-step representation therefore depends only on the continuation prefix through that step, while the answer representation also includes the preceding residual reasoning. We read the final-layer hidden representation at the last semantic content token of each reasoning step and the last answer-content token, excluding separators and delimiters. With $R = M - s$ remaining reasoning steps, the predictor queries are $\rho_j = j/(R+1)$ for $j = 1, \ldots, R$ and $\rho_A = 1$ for the answer. All predictions use the same final latent thought, rather than supervising different latent depths.

For each representation e_j^{text}, we subtract the running mean for its target population and apply ℓ_2 normalization:

$$z_j^{\text{text}} = \text{sg}\left(\frac{e_j^{\text{text}} - c_{g(j)}^{\text{text}}}{\left\| e_j^{\text{text}} - c_{g(j)}^{\text{text}} \right\|_2 + \epsilon}\right), \tag{11}$$

where $g(j)$ distinguishes reasoning-step and answer representations. Center initialization, distributed aggregation, and update timing are specified below. With cosine distance $d(a, b) = 1 - \cos(a, b)$, the loss for example i is:

$$\mathcal{L}_{\text{reason}}^{(i)} = \begin{cases} \frac{1}{2} d(\widehat{z}_A, z_A) + \frac{1}{2R} \sum_{j=1}^{R} d(\widehat{z}_j, z_j), & R > 0, \\ d(\widehat{z}_A, z_A), & R = 0. \end{cases} \tag{12}$$

This normalization gives every example a fixed total weight regardless of CoT length; examples without remaining reasoning receive answer prediction alone.

Molecular target preparation. We encode the reference SMILES as provided, preserving component order. The frozen SMI-TED-light encoder produces contextual token representations, which we average over non-padding positions to obtain each molecular target. We precompute these representations, subtract the mean over molecular training targets, and apply ℓ_2 normalization. The resulting targets and training mean remain fixed during training.

Running target means. We maintain separate running centers for reasoning and answer targets, with EMA coefficient $\beta = 0.99$. For reasoning targets, we first average step representations within each example and then average across eligible examples; answer targets are averaged across eligible examples. These statistics are aggregated across distributed workers using sums and counts. Both centers are computed only from training targets and reset at each curriculum phase. The first eligible batch initializes each center from its globally aggregated mean and uses that center immediately. For subsequent batches, let $\bar{e}_{b,g}$ denote the globally aggregated batch mean for target population g. The update is

$$c_{b,g} = \beta c_{b-1,g} + (1 - \beta)\bar{e}_{b,g}, \tag{13}$$

with the current batch using the pre-update center $c_{b-1,g}$ for target normalization and the updated center used for later batches. A center is unchanged when its target population has no eligible examples. This EMA updates target means, not encoder parameters. The molecular branch uses the fixed training-cache mean μ_{train}.

Distributed reduction. We first average each prediction objective within an example, then average over the examples eligible for that objective. Global sums and eligible-example counts determine the normalization across distributed workers.

Counterfactual objective. The counterfactual objective corrupts observable molecular features and is enabled only when such features exist. Predictive losses use the unperturbed input. Examples without molecular inputs receive no molecular counterfactual loss.

C Transfer to LatentChem

This experiment evaluates answer-representation prediction during supervised adaptation, before reinforcement learning. Both arms share the same training budget and trainable modules. Table 4 reports the complete task results; the baseline is the same adapted LatentChem model reported in Tables 1 and 2.

D Target Encoder Ablation

We examine stochasticity and parameter updates in the text-target encoder. The default shares the current language-model parameters and retains dropout. The alternatives disable target dropout or use an exponential moving average (EMA) of the LoRA parameters with momentum 0.995 and dropout disabled. All variants retain the molecular branch, predictors, loss coefficients, and training schedule; generation always uses the current parameters. The online-with-dropout configuration corresponds to the main model. Average ranks are computed within this table over its 17 metrics using unrounded scores and average ranks for ties; they summarize ordering, not effect size.

The variants favor different tasks. Disabling dropout improves LogP but reduces deletion and substitution accuracy. EMA improves substitution, QED, DRD2 over the default, while reducing deletion accuracy and major-product prediction. The online encoder thus provides useful targets in this setting, without a uniform advantage from momentum updates. These results motivate considering target updates together with the information represented by the encoder, rather than treating one update rule as a prerequisite for useful prediction.

E Training and Target Diagnostics

Predicting a molecular future should require distinguishing the outcome associated with each input. We examine whether the target geometry makes these distinctions visible to the cosine objective by removing centering from one view at a time. We retain ℓ_2 normalization, predictor dimensions, loss coefficients, and the four-phase curriculum. Table 6 reports the complete task results. The

Table 4: Answer-representation prediction with LatentChem. Both models receive three epochs of supervised adaptation with the molecular updater frozen and are evaluated at $T = 0.7, p = 0.9, n = 3$. LATENT-JEPA predicts an independently encoded answer representation from the final latent thought. In the LATENT-JEPA column, dark-red bold indicates improvement over the adapted LatentChem baseline. Editing accuracy, ring-system accuracy, and reaction validity are percentages; optimization entries are mean property improvements, count entries are MAE, and Murcko and FTS are similarities.

Metric	LatentChem	LatentChem + LATENT-JEPA
Molecule editing (%) ↑		
Addition	65.0	63.3
Deletion	78.3	**85.0**
Substitution	56.7	**60.0**
Molecular optimization: property improvement ↑		
LogP	0.7051	**0.8040**
Solubility	0.7829	**0.8295**
QED	0.1809	**0.2053**
DRD2	0.1732	**0.1907**
JNK3	0.0358	**0.0500**
GSK3-β	0.1481	0.1320
Molecular understanding		
Functional-group count error ↓	0.0967	**0.0733**
Ring count error ↓	0.1667	**0.1000**
Murcko scaffold similarity ↑	0.7220	0.6912
Ring-system accuracy (%) ↑	73.89	72.78
Reaction prediction		
Major-product FTS ↑	0.5099	**0.5541**
Major-product validity (%) ↑	74.00	**78.33**
Byproduct FTS ↑	0.1728	**0.1735**
Byproduct validity (%) ↑	42.67	**45.33**

calibration figure uses the 16,949 molecular training targets; the spectra are computed before rowwise normalization. Average rank in the task table uses all 17 unrounded metrics with average ranks for ties.

Retaining both centers gives the strongest average rank and outperforms either removal on most task metrics. Removing text centering also lowers all six property-improvement scores. This supports the calibrated targets used in the dual-view model. Centering changes both target geometry and the numerical scale of the cosine loss, so the comparison evaluates their combined effect.

The raw SMI-TED targets share a dominant direction. Centering followed by ℓ_2 normalization reduces mean random-pair cosine from 0.989 to 0.010 (Figure 4). The spectrum likewise shows that much of the raw representation energy lies in a common direction. Calibration makes differences among molecular outcomes more visible to the cosine objective. The resulting task-level relationships are examined in Appendix G.

Autoregressive losses remain similar across Coconut-Chem, text-only, and dual prediction. At the end of the final curriculum phase, the dual model's autoregressive, weighted textual, and weighted molecular losses are approximately 0.0665, 0.0144, and 0.0021. Predictive losses are lower than their shuffled-source counterparts.

The single-view centering comparisons in Table 7 use coefficients 0.05 for textual prediction and 0.01 for molecular prediction; the dual comparison uses 0.025 and 0.01. All four comparisons use all 17 task metrics. Differences below 10^{-4} are counted as ties, accounting for metric direction. Prediction distances are averaged over the final 50 training batches.

Table 5: Text-target encoder ablation. Editing, ring-system accuracy, and validity are percentages; property scores are mean improvements; counts are MAE; Murcko and FTS are similarities. Dark-red bold marks the best task score in each column, including ties at the displayed precision. Shading identifies the default configuration. Lower average rank is better.

	Molecule editing (%) ↑			Property improvement ↑					
Text target	Add	Delete	Sub.	LogP	Solubility	QED	DRD2	JNK3	GSK3-β
Online, dropout	78.3	**86.7**	61.1	0.8318	**0.9659**	0.2296	0.2848	**0.1296**	**0.2515**
Online, no dropout	80.0	80.0	56.1	**0.8613**	0.9398	0.2168	0.2705	0.1104	0.2222
EMA, no dropout	**81.7**	80.0	**64.4**	0.8410	0.9549	**0.2308**	**0.2957**	0.1295	0.2278

	Molecular understanding				Major product		Byproduct	
Text target	FG ↓	Ring ↓	Murcko ↑	Ring-sys. (%) ↑	FTS ↑	Valid (%) ↑	FTS ↑	Valid (%) ↑
Online, dropout	**0.0667**	**0.1333**	**0.8065**	81.11	**0.6492**	**88.33**	0.1806	45.33
Online, no dropout	0.0767	0.1500	0.7515	**81.67**	0.6357	87.33	0.1765	**47.00**
EMA, no dropout	0.0800	0.1500	0.7793	81.11	0.6147	85.33	**0.1823**	46.00

Text target	Avg. rank ↓
Online, dropout	**1.68**
Online, no dropout	2.35
EMA, no dropout	1.97

Table 6: Target-centering ablation with both predictive views and ℓ_2 normalization retained. Editing, ring-system accuracy, and validity are percentages; property scores are mean improvements; counts are MAE; Murcko and FTS are similarities. Dark-red bold marks column-best task scores, including ties at the displayed precision; shading identifies the default configuration. Lower average rank is better.

	Molecule editing (%) ↑			Property improvement ↑					
Target calibration	Add	Delete	Sub.	LogP	Solubility	QED	DRD2	JNK3	GSK3-β
Both views centered	**78.3**	**86.7**	61.1	0.8318	0.9659	**0.2296**	**0.2848**	**0.1296**	**0.2515**
Without molecular centering	75.0	73.3	**62.2**	**0.8701**	**0.9703**	0.2151	0.2645	0.1163	0.2329
Without text centering	**78.3**	75.0	58.9	0.7854	0.8311	0.2036	0.2644	0.1174	0.2232

	Molecular understanding				Major product		Byproduct	
Target calibration	FG ↓	Ring ↓	Murcko ↑	Ring-sys. (%) ↑	FTS ↑	Valid (%) ↑	FTS ↑	Valid (%) ↑
Both views centered	**0.0667**	**0.1333**	0.8065	81.11	**0.6492**	**88.33**	**0.1806**	45.33
Without molecular centering	**0.0667**	**0.1333**	0.7849	**83.89**	0.6396	**88.33**	0.1773	45.00
Without text centering	0.0700	0.1500	**0.8115**	83.33	0.6211	86.67	0.1695	**48.33**

Target calibration	Avg. rank ↓
Both views centered	**1.53**
Without molecular centering	1.97
Without text centering	2.50

Without molecular centering, original- and shuffled-source distances are both small (0.003 and 0.009), consistent with predicting a shared target direction rather than relying strongly on the corresponding input. Text targets retain more source specificity, although its measured separation also decreases without centering. Removing text centering helps some text-only task scores, whereas the dual model benefits on more metrics from retaining both centers. Together, these diagnostics suggest that a

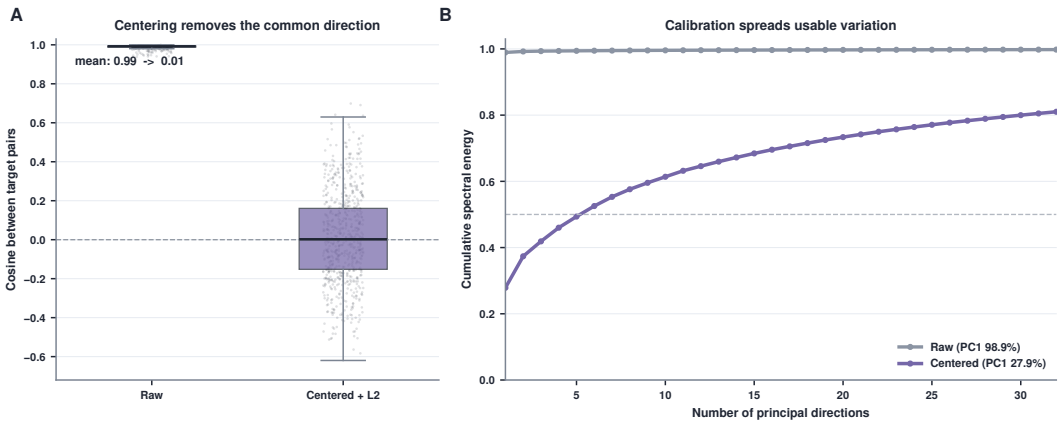


Figure 4: Calibration of molecular prediction targets. **Left:** Pairwise cosine before and after centering and ℓ_2 normalization. **Right:** Spectral energy of the raw second moment and centered covariance, computed before rowwise normalization. Appendix K defines the diagnostics.

Table 7: Target centering and source-specific prediction. Task-score changes count improved, unchanged, and worse metrics relative to the corresponding centered model after removing the indicated center. The final column gives cosine distances using the original and shuffled latent sources. Settings and aggregation are described in the surrounding text.

Objective	Center removed	Metrics	Task-score changes	Original / shuffled distance
Text only	Text	17	10 / 3 / 4	0.102 / 0.330
Molecular only	Molecular	17	4 / 3 / 10	0.003 / 0.009
Dual	Molecular	17	4 / 3 / 10	0.003 / 0.009
Dual	Text	17	3 / 1 / 13	0.095 / 0.217

target's usefulness depends on both its encoded information and the outcome differences emphasized by the prediction loss. This motivates considering the encoder, calibration, and discrepancy together when designing scientific prediction targets (Appendix A).

F Prediction Views and Chemical Task Performance

The two predictive views describe a solution in different spaces. To examine what each contributes, we compare textual prediction, molecular prediction, and their combination in Coconut-Chem (Table 8). All configurations use six latent thoughts, three epochs per curriculum stage, and the ChemCoTBench evaluation specified in Appendix L. Decoding uses temperature 0.7, top-$p = 0.9$, three responses per example. The molecular coefficient is 0.01 in both relevant variants, while the textual coefficient is 0.05 for text-only and 0.025 for dual prediction. This comparison therefore varies textual supervision strength as well as the set of views. Its average rank uses the 17 metrics available for all three models, from each, with unrounded scores and average ranks for ties.

Molecular prediction is particularly effective for editing, while dual prediction gives the strongest average rank over the metrics shared by all three variants. Adding the textual view improves most property objectives and all reported reaction metrics relative to molecular-only prediction. The combined supervision thus benefits a broader set of tasks, while the full table retains the cases favoring a single view. These patterns suggest that the usefulness of a future view depends on the information needed to solve the task.

Transfer to Vanilla Coconut. Vanilla Coconut receives source SMILES through the textual prompt without source ChemTokens. The four variants in Table 9 share a text-only initializer, six latent thoughts, three epochs per curriculum phase, and the 1,120-example evaluation at temperature 0.7,

Table 8: Chemical performance with different prediction views. Editing, ring-system accuracy, and validity are percentages; property scores are mean improvements; counts are MAE. Dark-red bold marks column-best task scores, including ties at the displayed precision; shading identifies dual prediction. Loss coefficients and rank aggregation are specified in the text.

Prediction views	Add ↑	Delete ↑	Sub. ↑	LogP ↑	Solubility ↑	QED ↑	DRD2 ↑	JNK3 ↑	GSK3-β ↑
Textual	81.7	80.0	58.9	0.7839	0.9178	0.2229	0.2684	0.1100	0.2415
Molecular (SMI-TED)	**83.3**	**90.0**	**62.8**	**0.8624**	0.9359	**0.2333**	0.2579	0.1141	0.2442
Textual + molecular	78.3	86.7	61.1	0.8318	**0.9659**	0.2296	**0.2848**	**0.1296**	**0.2515**

Prediction views	FG ↓	Ring ↓	Murcko ↑	Ring-sys. ↑	Major FTS ↑	Major valid ↑	Byproduct FTS ↑	Byproduct valid ↑
Textual	0.0867	0.1500	0.7856	**81.67**	0.6036	83.67	0.1445	44.00
Molecular (SMI-TED)	0.0767	0.1500	0.7552	**81.67**	0.6392	86.67	0.1741	45.00
Textual + molecular	**0.0667**	**0.1333**	**0.8065**	81.11	**0.6492**	**88.33**	**0.1806**	**45.33**

Prediction views	Avg. rank ↓
Textual	2.71
Molecular (SMI-TED)	1.82
Textual + molecular	**1.47**

Table 9: Prediction-view comparison in Vanilla Coconut without source ChemTokens. Editing, ring-system accuracy, and validity are percentages; property scores are mean improvements; counts are MAE. Training and evaluation settings are given in the text. Dark-red bold marks column-best task scores, including ties at the displayed precision; shading identifies textual prediction. Lower average rank is better.

	Molecule editing (%) ↑			Property improvement ↑					
Prediction views	Add	Delete	Sub.	LogP	Solubility	QED	DRD2	JNK3	GSK3-β
Vanilla Coconut	71.7	88.3	56.1	0.7726	0.7250	0.1968	0.2300	0.0861	0.1861
Textual	**76.7**	86.7	56.1	**0.8656**	**0.9001**	0.2048	0.2268	**0.0992**	0.1875
Molecular (SMI-TED)	68.3	85.0	**57.2**	0.7960	0.8107	0.2105	**0.2420**	0.0769	**0.2153**
Textual + molecular	66.7	**90.0**	55.6	0.8429	0.8799	**0.2307**	0.2376	0.0796	0.1972

	Molecular understanding				Major product		Byproduct	
Prediction views	FG ↓	Ring ↓	Murcko ↑	Ring-sys. ↑	FTS ↑	Valid ↑	FTS ↑	Valid ↑
Vanilla Coconut	**0.0600**	**0.1000**	0.7537	**81.67**	0.5673	82.67	0.1648	47.00
Textual	**0.0600**	**0.1000**	0.7485	77.22	**0.6106**	**86.33**	0.1849	**50.00**
Molecular (SMI-TED)	0.0700	**0.1000**	0.7045	77.78	0.5809	81.33	0.1596	46.33
Textual + molecular	0.0667	**0.1000**	**0.7848**	79.44	0.5441	78.00	**0.1893**	47.00

Prediction views	Avg. rank ↓
Vanilla Coconut	2.65
Textual	**2.09**
Molecular (SMI-TED)	2.85
Textual + molecular	2.41

top-$p = 0.9$, with three responses per example. Average rank uses all 17 unrounded metrics and average ranks for ties.

Textual prediction improves 10 of 17 metrics over Vanilla Coconut, with four regressions and three ties; molecular and dual prediction improve seven and eight metrics, respectively. Textual prediction

has the lowest average rank in this comparison, while molecular and dual prediction favor different tasks. For example, molecular prediction performs best on DRD2 and GSK3-β, while dual prediction performs best on QED and Murcko similarity. These results show that predictive supervision can benefit a model whose molecular inputs are provided only as text. The different patterns in Coconut-Chem and Vanilla Coconut motivate choosing future views together with the task and reasoning setup; their source interfaces, initialization, and training histories are not otherwise matched. Appendix H examines the associated representation patterns within each setting.

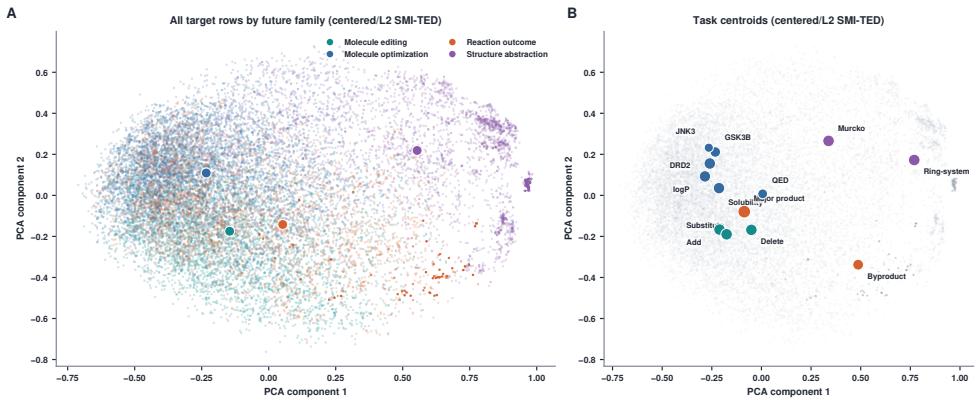


Figure 5: Structure of the calibrated molecular target space. **A:** Target PCA colored by task family. **B:** Task centroids in the same projection. Both panels describe the target encoder's representations, not the reasoner's latent states.

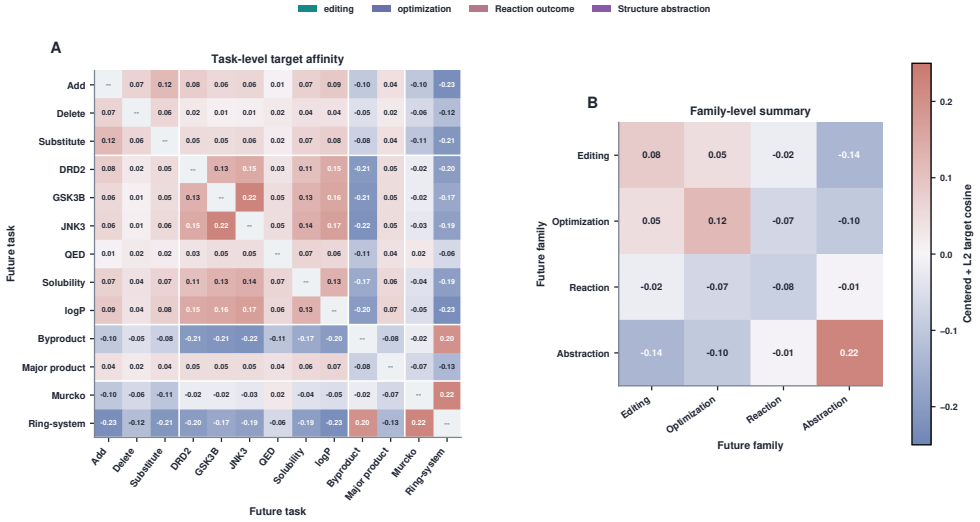


Figure 6: Molecular target relationships across tasks. **Left:** Mean cross-task cosine after centering and ℓ_2 normalization; the within-task diagonal is omitted. **Right:** Aggregation by task family. Definitions are given in Appendix K.4.

G Chemical Relationships in the Prediction Targets

To understand what the molecular encoder supplies as supervision, we examine the relationships among its representations of reference outcomes. We use the calibrated SMI-TED embeddings of 16,949 molecular training targets, before considering the latent states trained to predict them. The shared PCA in Figure 5 shows individual targets and task centroids; Figure 6 summarizes their cross-task cosine relations. Appendix K.4 defines the affinity and task-family aggregation.

The target relations do not simply reproduce the task taxonomy. Murcko and ring-system targets form a close pair, and several optimization tasks have positive affinities. Yet major-product and byproduct targets, which share a reaction-task label, have lower affinity than byproduct and ring-system targets. The encoder thus relates outcomes in ways that differ from grouping them by the requested operation.

These relations help explain the role of the molecular target: it supplies similarities among solutions beyond their task membership. Predicting this view may encourage latent thoughts to reflect such relationships while distinguishing individual outcomes. We examine these possibilities in Section H.

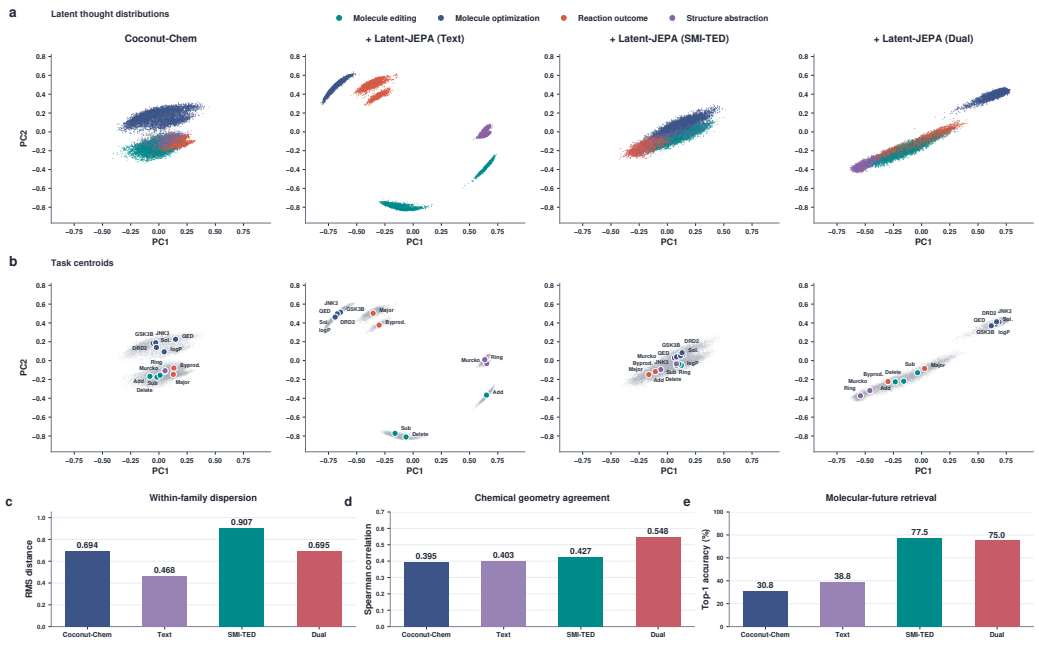


Figure 7: Latent organization under different predictive views. **a–b:** Shared PCA and task centroids for Coconut-Chem and its predictive variants. **c:** Within-family dispersion. **d:** Agreement with molecular task-centroid relations. **e:** Within-task molecular-outcome retrieval on a probe holdout from the training cohort. Dispersion and centroid agreement use full-dimensional calibrated states; settings are given in the surrounding text.

The affinities describe this encoder and target cohort; identifying their specific chemical drivers would require additional molecular annotations.

H Organization of Latent Thoughts

After characterizing the targets, we ask how the choice of predictive view relates to the outcome information in the final latent thought. We compare Coconut-Chem with textual, molecular, and dual prediction on 16,943 training examples across 13 tasks. We analyze final-layer hidden states at the sixth latent thought. The PCA is shared across the four variants, while dispersion and task-centroid agreement use full-dimensional calibrated states. Retrieval uses ridge probes with 3,390 examples from this training cohort excluded from probe fitting, and distinct future-SMILES candidates within each task. This is a separate probe split from Figure 2; preprocessing and metric definitions are specified in Appendix K.

Molecular prediction yields the highest outcome retrieval, while dual prediction has the strongest agreement with molecular task-centroid relations among these configurations (Figure 7). Dual prediction also raises retrieval from 30.8% to 75.0% relative to Coconut-Chem. The dispersion statistic describes a different pattern: textual prediction has the lowest within-family dispersion, and the dual model is close to Coconut-Chem (0.695 versus 0.694). Thus, the ordering by dispersion does not follow the ordering by retrieval in this comparison.

We repeat the analysis in Vanilla Coconut to examine target-side supervision when source molecules enter only as SMILES in the prompt. Dual prediction gives the highest retrieval, increasing it from 5.8% to 78.5%, while molecular prediction gives the strongest agreement with molecular task relations. Textual prediction again has the lowest within-family dispersion. The within-setting comparison shows that outcome information can become more accessible without a dedicated source-molecule interface. It therefore supports studying domain encoders as sources of predictive supervision as well as input representations (Appendix A). Coconut-Chem and Vanilla Coconut

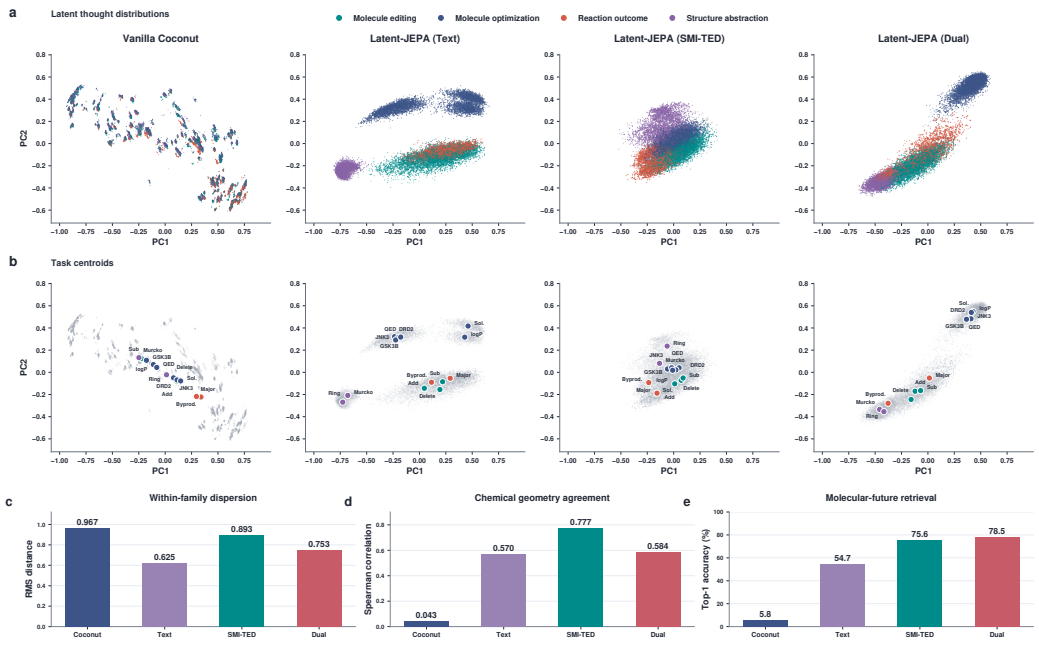


Figure 8: Latent organization in Vanilla Coconut, which receives SMILES as text without source ChemTokens. Panels follow Figure 7. The PCA basis is shared within this comparison, not across architectures. Settings and the probe protocol are specified in the text and Appendix K.

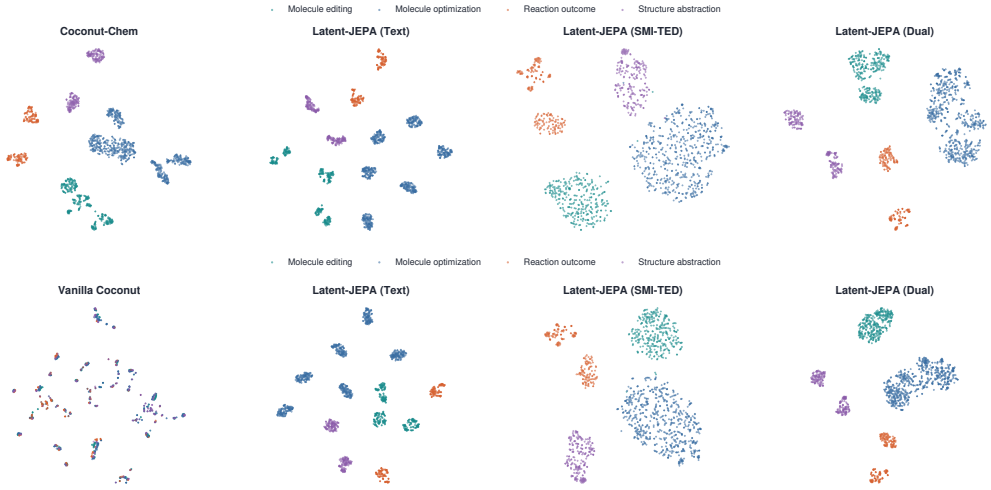


Figure 9: Exploratory local neighborhoods of latent thoughts in Coconut-Chem (top) and Vanilla Coconut (bottom). Figures 7 and 8 provide the full-dimensional quantitative comparisons; preprocessing is given in Appendix K.

also differ in initialization and training history, so their absolute scores do not isolate the role of ChemTokens.

Figure 9 provides a qualitative view of how predictive supervision changes local neighborhood organization. Textual prediction produces clearer task-family separation, while molecular and dual supervision yield different local arrangements that accompany their stronger molecular-outcome retrieval and geometry agreement.

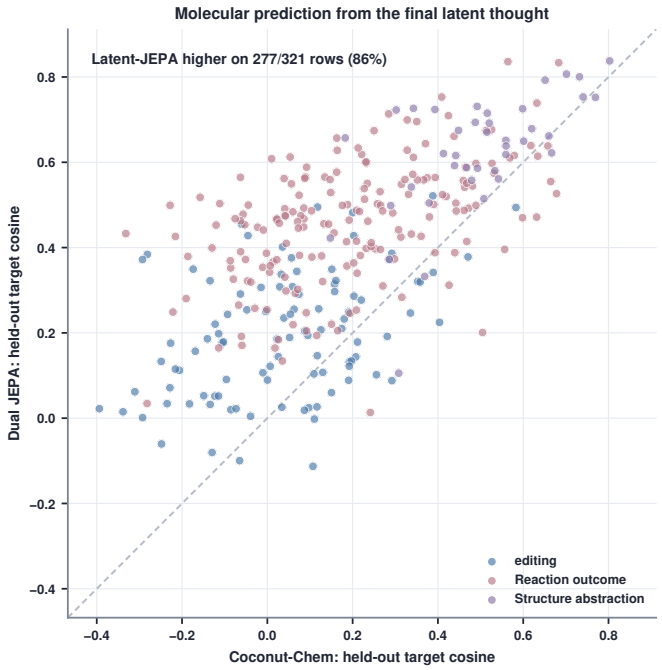


Figure 10: Example-wise molecular-outcome prediction from the final latent thought. Each point compares probe cosine for Coconut-Chem (horizontal) and dual-view LATENT-JEPA (vertical); colors denote task families. Points above the diagonal favor LATENT-JEPA. The probe split is shared with Figure 2.

I PREDICTING MOLECULAR OUTCOMES FROM LATENT THOUGHTS

The average probe gains leave open whether the improvement is widespread or concentrated in a few examples. We examine this distribution, then study how prediction and structural agreement vary across tasks and recurrent depth. For probing, we fit a separate ridge map at each step using 1,224 training examples and evaluate it on 321 molecular-target-eligible ChemCoTBench test examples. We compare corresponding targets with a fixed within-task permutation of evaluation targets. The probes use latent hidden states, whereas RSA uses recurrent feedback embeddings. SMI-TED RSA uses the same 321 test examples, and Morgan RSA uses the subset of 294 with valid outcome fingerprints; Appendix K specifies the protocols and depth indexing.

The improvement extends across most of the probe-evaluation cohort: LATENT-JEPA increases cosine on 277 of 321 examples (Figure 10), with mean final-update cosine rising from 0.186 to 0.397. Corresponding targets remain more predictable than targets reassigned within the same task. The higher mean therefore reflects improved outcome prediction across many examples rather than a small subset.

We next examine whether the aggregate geometry trend extends across tasks. Figure 11 shows positive final-update gains over Coconut-Chem in every analyzed task. The relative gains are larger at later updates for Murcko and major-product prediction, while deletion and byproduct prediction follow different depth patterns.

At the aggregate level, dual-model SMI-TED RSA rises from 0.190 at the first update to 0.400 at the final update. Final-update RSA is 0.110 for Coconut-Chem, 0.115 for text-only prediction, and 0.392 for molecular-only prediction. With Morgan fingerprints, the final-update scores are 0.029, 0.118, and 0.329 for Coconut-Chem, text-only, and dual prediction. The task breakdown therefore adds detail to the overall structural agreement measured in Figure 3.

Figure 12 shows that retrieval gains occur across tasks and are not confined to the final update. In the aggregate analysis, the dual model's probe cosine and Hit@5 generally increase across recurrent steps, with smaller changes near the final updates (Figure 2). Its structural agreement also increases

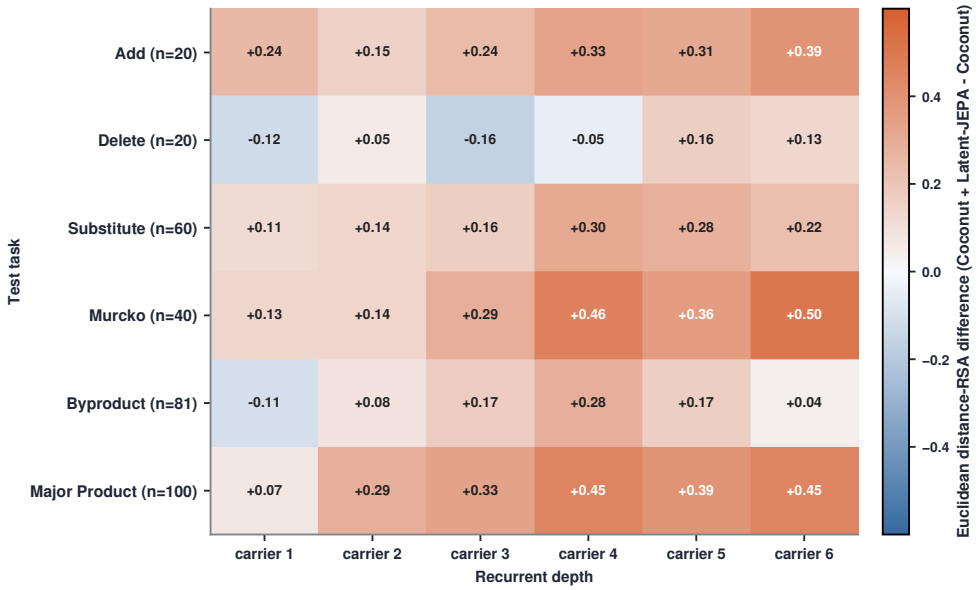


Figure 11: Task-wise changes in molecular geometry across recurrent depth. Each cell gives dual-view LATENT-JEPA minus Coconut-Chem within-task RSA, using feedback-embedding and calibrated molecular-target distances. Positive values indicate greater agreement for LATENT-JEPA; these are between-model differences, not absolute RSA trajectories.

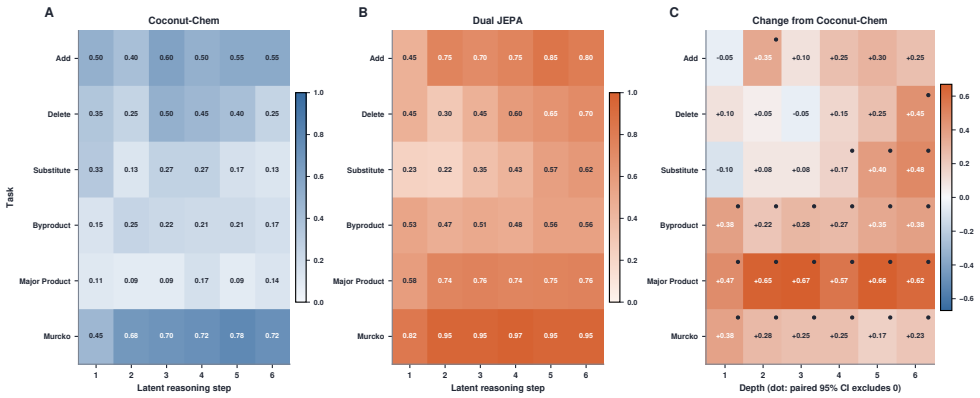


Figure 12: Molecular-outcome retrieval by task and recurrent depth. **A:** Coconut-Chem Hit@5. **B:** Dual-view LATENT-JEPA Hit@5. **C:** Their difference; dots mark 95% bootstrap intervals excluding zero. The heatmaps use the same latent hidden states, probe holdout, and within-task candidates as Figure 2. Averaging each column over tasks recovers the corresponding aggregate Hit@5 curve; Appendix K gives the protocol.

overall (Figure 3). The task-level retrieval gains follow different depth profiles: gains for substitution become larger at later steps, whereas major-product and Murcko-scaffold prediction already show clear gains at the first step. These task-dependent patterns motivate studying predictive objectives whose horizons or levels of abstraction vary across recurrent depth.

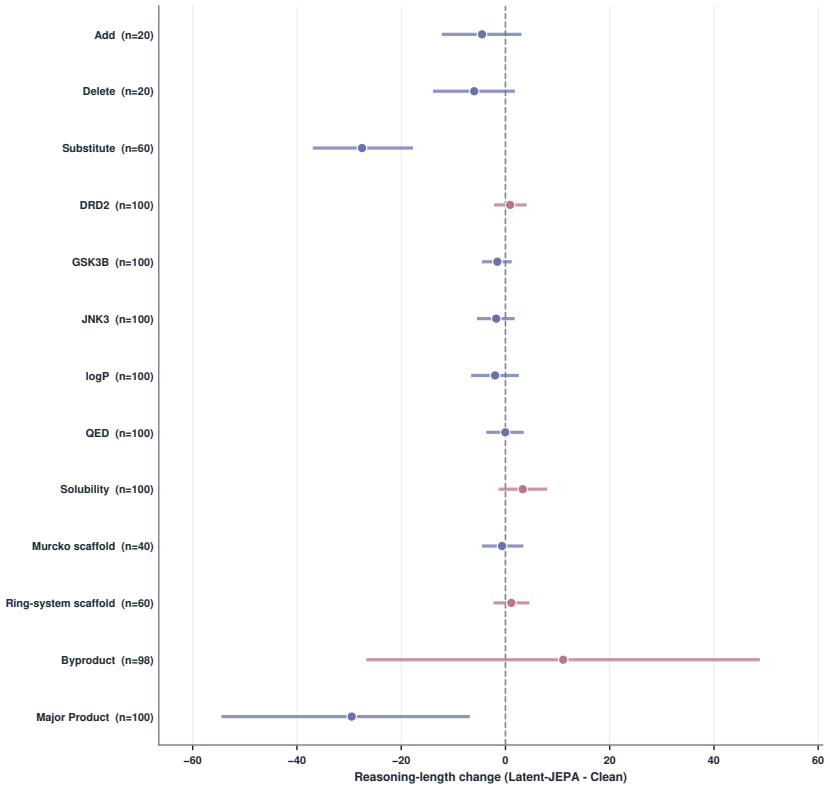


Figure 13: Change in explicit reasoning length by task. Points show the difference in mean pre-answer token count between LATENT-JEPA and Coconut-Chem, with bootstrap intervals. Negative values indicate shorter visible reasoning; token exclusions are specified in the text.

J EXPLICIT REASONING AFTER LATENT COMPUTATION

The preceding analyses concern latent states before answer generation. We also examine whether predictive supervision changes how much reasoning the model subsequently expresses in text. We compare the mean number of pre-answer tokens, excluding final SMILES and answer-declaration lines. Figure 13 reports the task-wise difference between LATENT-JEPA and Coconut-Chem with bootstrap intervals.

The clearest reductions occur in substitution and major-product prediction, whereas optimization tasks remain close to the baseline and byproduct prediction has a wider interval. These changes provide a behavioral counterpart to the representation analyses: predictive supervision can alter how a solution is expressed, with no uniform reduction across tasks.

Table 10: Cohorts used for representation analysis, with counts in examples.

Analysis	Cohort
Main linear probing and Hit@5	1,224 training examples across 13 tasks for probe fitting; 321 test examples across six tasks for evaluation
SMI-TED RSA	321 molecular-target-eligible test examples across six tasks
Morgan RSA	294 test examples with valid molecular outcomes across the same six tasks
Latent PCA and task organization	16,943 training examples; the separate retrieval analysis holds out 3,390 from probe fitting
Target calibration and task affinity	16,949 molecular training targets

K Methods for Representation Analysis

This appendix specifies the representations, cohorts, and measurements used to analyze molecular-outcome recoverability, molecular relationships, and task organization. Probing and RSA use the matched-view Coconut-Chem configurations described in Appendix H; optimization and curriculum settings are given in Appendix B.

K.1 Representations and evaluation cohorts

We distinguish the final-layer hidden state at a continuous-thought position from the feedback embedding supplied to the next recurrent computation. The former is the supervised readout and is used for linear probing; the latter is used for the RSA curves. On our fixed-length evaluation route, the hidden state at step t supplies the feedback embedding at step $t+1$. Neither view receives the gold continuation during evaluation extraction.

Table 10 distinguishes training data used to fit probes or describe task organization from test data used to evaluate molecular outcomes. Molecular targets use the training-mean calibration in Equation (7).

K.2 Linear probing and molecular retrieval

For each model and reasoning step, we fit a separate ridge probe:

$$(A, b) = \arg\min_{A,b} \sum_{i \in \mathcal{D}_{\text{fit}}} \|Au_i + b - z_i\|_2^2 + \alpha \|A\|_F^2, \qquad \widehat{z}_i = \frac{Au_i + b}{\|Au_i + b\|_2}, \tag{14}$$

where z_i is the normalized molecular target. For the main probe figure, u_i is a 512-dimensional sparse random projection of the latent hidden state after featurewise standardization. Means and standard deviations are estimated only on the probe-fitting split. Models and steps share the random projection and fitting examples, but not fitted probe weights. We use ridge coefficient $\alpha = 10$, an unpenalized intercept, and the LSQR solver with tolerance 10^{-4}. Training examples are selected by task-stratified hashing of example keys. Test examples do not enter probe fitting or selection of probe hyperparameters. We report example-mean cosine $\widehat{z}_i^\top z_i$. These probes are fitted after model training, independently of the JEPA predictors.

For retrieval, let $\mathcal{C}_q$ contain one target per distinct stored future-SMILES string among evaluation examples of task q. Each evaluation example's target belongs to its candidate set. Candidates receive cosine scores $s_{ij} = \widehat{z}_i^\top z_j$. The reported rank and Hit@k are:

$$r_i = 1 + \sum_{j \in \mathcal{C}_{q(i)}} \mathbf{1}[s_{ij} > s_{i,i} + 10^{-8}], \qquad \text{Hit@}k = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{n_q} \sum_{i : q(i) = q} \mathbf{1}[r_i \le k]. \tag{15}$$

Here $s_{i,i}$ denotes the score of the example's corresponding candidate, including when several examples share that candidate. Scores tied within the numerical tolerance do not count as outranking it. Main-figure candidate sets have 20, 20, 60, 40, 100, and 61 entries for addition, deletion, substitution, Murcko-scaffold, major-product, and byproduct tasks, respectively. Uniform random ranking gives the task-average baseline $|\mathcal{Q}|^{-1} \sum_q \min(k, |\mathcal{C}_q|)/|\mathcal{C}_q|$, which is 0.1401 for $k = 5$. The

task-organization figures instead report example-averaged Top-1 retrieval on their separate probe holdout from the training cohort, with candidates drawn from that holdout.

For the shuffled-target panel, we hold the fitted probes and their predictions fixed and randomly reassign evaluation targets within each task. Both models use the same permutation. This preserves task membership while disrupting example-level correspondence; the ordinary permutation retains self-matches and repeated molecular outcomes.

K.3 Representational similarity analysis

RSA measures whether examples that are close in one representation space are also close in another. For task q and reasoning step t, form two vectors over all distinct unordered example pairs:

$$a_{q,t} = \big(d_h(h_{i,t}, h_{j,t})\big)_{i<j,\ i,j\in q}, \qquad b_q = \big(d_z(z_i, z_j)\big)_{i<j,\ i,j\in q}. \tag{16}$$

We calculate

$$\mathrm{RSA}_{q,t} = \mathrm{corr}\big(\mathrm{rank}(a_{q,t}), \mathrm{rank}(b_q)\big), \qquad \mathrm{RSA}_t = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \mathrm{RSA}_{q,t}, \tag{17}$$

using Spearman rank correlation with average ranks for ties. Higher RSA indicates stronger agreement in relative distances. We first compute RSA within each task, then average equally across tasks. The task-depth heatmap reports the difference between LATENT-JEPA and Coconut-Chem RSA in each cell.

The SMI-TED analysis uses Euclidean distances between feedback embeddings and between training-mean-centered, L2-normalized molecular targets. The Morgan analysis instead uses latent cosine distance and fingerprint Tanimoto distance, $1 - |f_i \cap f_j|/|f_i \cup f_j|$, for bit fingerprints of radius two and length 2,048 with chirality enabled. Morgan fingerprints encode local atom neighborhoods; f_i denotes the set of active bit positions for molecule i. Tanimoto similarity is the fraction of active bits shared by the two fingerprints relative to their union, ranging from zero (no shared bits) to one (identical fingerprints). The distance is one minus this similarity. Identical fingerprints do not necessarily imply identical molecules.

Using cosine distance for the SMI-TED comparison preserves the ordering between models. First- and final-update RSA are 0.176 and 0.410 for LATENT-JEPA, compared with 0.138 and 0.118 for Coconut-Chem.

K.4 Projections and geometry statistics

For task organization, we use final-layer hidden states at the sixth latent thought, subtract each model's cohort mean, and L2-normalize each row. We fit one PCA basis across the four variants within each architecture. Task centroids are means of projected samples, with common axis limits within a comparison. Dispersion and task-centroid agreement use the full-dimensional calibrated states.

For t-SNE, we sample up to 100 examples per task, reduce to 50 PCA dimensions, and fit each model's projection with perplexity 30.

For a family g of normalized states u_i, its RMS dispersion is $\sqrt{|g|^{-1}\sum_{i\in g}\|u_i - \bar{u}_g\|_2^2}$, where $\bar{u}_g$ is the family mean. We average this statistic equally over four families. Smaller values indicate less variation around the family mean. Task-centroid agreement correlates the cosine relations among the 13 latent task means with those among molecular task means, using the 78 unordered task pairs. Target affinity between two distinct tasks is the mean cosine over all cross-task target pairs.

For a positive-semidefinite matrix with eigenvalues λ_j, let $p_j = \lambda_j / \sum_l \lambda_l$. Its spectral effective rank is

$$r_{\text{eff}} = \exp\left(- \sum_{j:p_j>0} p_j \log p_j \right). \tag{18}$$

It summarizes how evenly spectral energy is distributed. Cumulative spectral energy at k is $\sum_{j=1}^{k} p_j$ after sorting eigenvalues in descending order. The calibration figure reports spectra of the raw second moment and the mean-centered covariance, both computed before rowwise ℓ_2 normalization.

K.5 Error bands

Cosine-probe bands show the sample mean plus or minus 1.96 standard errors. Hit@5 bands use 95% task-stratified example-bootstrap intervals from 5,000 resamples. These intervals quantify evaluation-example variation with the models and probes held fixed. RSA curves show task-average point estimates. For each task and depth in Figure 12, we compute the difference between the two models' Hit@5 indicators on each shared example. We resample these example-wise differences with replacement 5,000 times and take the 2.5th and 97.5th percentiles of their mean. Dots mark intervals excluding zero. These are pointwise intervals for individual task-depth comparisons.

L Datasets and Evaluation

L.1 Chemical tasks

Our evaluation uses the following tasks from ChemCoTBench (Hao et al., 2026).

Molecular optimization. Given a source molecule and a requested property, the model generates a modified molecule. Unlike instruction-specified editing, this task requires choosing structural changes that improve the property. We consider LogP, aqueous solubility, and quantitative estimation of drug-likeness (QED), along with activity-related scores for DRD2, JNK3, and GSK3-β. The evaluator uses TDC property oracles for LogP, QED, and the three activity scores, and an ESOL estimator for solubility. These are computational scores of the generated structures, rather than experimental measurements.

For property p, improvement is the difference $\Delta_p = f_p(m_{\text{generated}}) - f_p(m_{\text{source}})$. Following the benchmark implementation, missing, invalid, or unscorable outputs contribute zero improvement. Within each completion stream, the evaluator clips improvements to their 5th and 95th percentiles before computing the mean. Success rate is the fraction of outputs with $\Delta_p > 0$, computed before this clipping. We average the resulting metrics over the three completion streams. Thus optimization measures improvement relative to the source, not recovery of a unique reference molecule.

Molecule editing. The input specifies a source molecule and an addition, deletion, or substitution operation. The model must produce a molecule implementing that instruction. We report operation correctness separately for the three editing tasks.

Molecular understanding and scaffold abstraction. Functional-group and ring-counting tasks ask the model to identify structural features of an input molecule; we report mean absolute count error. The Murcko task asks for the molecule's scaffold, the core formed by its rings and connecting linkers. The evaluator extracts scaffolds from the prediction and reference: identical scaffolds score one; otherwise it uses Tanimoto similarity between radius-two, 1,024-bit Morgan fingerprints, with zero assigned to invalid outputs or pairs without a common substructure. The ring-system task asks whether a supplied ring structure occurs in the input molecule and requires a yes/no answer; we report label accuracy.

Forward-reaction prediction. Given the reaction inputs, the model predicts either the major product or the byproduct. We report fingerprint Tanimoto similarity (FTS) to the reference outcome and the fraction of chemically parseable outputs (validity), separately for the two tasks. FTS is the arithmetic mean of Tanimoto similarities computed using RDKit topological, MACCS-key, and radius-two Morgan count fingerprints. These encode paths, predefined structural features, and local atom neighborhoods, respectively. Higher FTS indicates greater structural similarity on a zero-to-one scale. This three-fingerprint score differs from the Morgan-only comparison used for RSA in Section K.3.

Table 11: Molecular optimization success rates (%). Dark-red bold values in the LATENT-JEPA row match or improve upon Coconut-Chem at the reported precision; regressions remain black.

Method	LogP	Solubility	QED	DRD2	JNK3	GSK3-β
LatentChem	77.7	73.0	78.3	63.7	45.7	69.3
Coconut-Chem	82.3	85.0	89.0	77.3	58.3	81.3
Coconut-Chem + LATENT-JEPA	**82.3**	**86.0**	86.3	**79.3**	**64.0**	**84.0**

Outputs that RDKit cannot parse as molecules receive zero similarity and remain in the denominator. Validity measures parsability rather than correctness of the reaction outcome.

L.2 EVALUATION SET AND SCORING

The evaluation contains 1,020 optimization, molecular understanding, scaffold, and reaction examples, together with 100 editing examples. The tables report six property-improvement scores, three editing accuracies, two count errors, two scaffold scores, and four reaction metrics. Optimization success rates appear in Table 11.

We associate predictions with their reference examples using dataset indices and example identifiers before scoring. We will release the evaluation examples, canonicalization rules, and example-level outputs. We use the standard ChemCoTBench test set and the decoding settings specified in Section B.

Sampling and aggregation. For each checkpoint, we sample three responses per problem and score each response separately. Let S_j be the task score computed using the jth response to every problem. We report $\frac{1}{3}\sum_{j=1}^{3} S_j$; we do not select the best response. For editing accuracy, this is the fraction of correct responses across all three samples, rather than the fraction of problems solved by at least one sample. Where reported, the sample standard deviation of S_1, S_2, S_3 describes decoding variability for the same checkpoint. Intervals in the representation analyses summarize variation across evaluation examples, as detailed in Appendix K.

M OPTIMIZATION SUCCESS RATES

Success rate is the fraction of scored generations with positive property improvement. We average over the three completion streams used in our evaluations.