

Chain-of-Generation: Progressive Latent Diffusion for Text-Guided Molecular Design

Lingxiao Li
Michigan State University
lilingx1@msu.edu

Haobo Zhang
University of Michigan
haobozha@umich.edu

Bin Chen
Michigan State University
chenbi12@msu.edu

Jiayu Zhou
University of Michigan
jiayuz@umich.edu

Abstract

Text-conditioned molecular generation aims to translate natural-language descriptions into chemical structures, enabling scientists to specify functional groups, scaffolds, and physicochemical constraints without handcrafted rules. Diffusion-based models, particularly latent diffusion models (LDMs), have recently shown promise by performing stochastic search in a continuous latent space that compactly captures molecular semantics. Yet existing methods rely on one-shot conditioning, where the entire prompt is encoded once and applied throughout diffusion, making it hard to satisfy all the requirements in the prompt. We discuss three outstanding challenges of one-shot conditioning generation, including the poor interpretability of the generated components, the failure to generate all substructures, and the overambition in considering all requirements simultaneously. We then propose three principles to address those challenges, motivated by which we propose **Chain-of-Generation (CoG)**, a training-free multi-stage latent diffusion framework. CoG decomposes each prompt into curriculum-ordered semantic segments and progressively incorporates them as intermediate goals, guiding the denoising trajectory toward molecules that satisfy increasingly rich linguistic constraints. To reinforce semantic guidance, we further introduce a post-alignment learning phase that strengthens the correspondence between textual and molecular latent spaces. Extensive experiments on benchmark and real-world tasks demonstrate that CoG yields higher semantic alignment, diversity, and controllability than one-shot baselines, producing molecules that more faithfully reflect complex, compositional prompts while offering transparent insight into the generation process.

1 Introduction

Text-conditioned molecular generation is an emerging and promising research direction at the intersection of natural language processing and molecular design. This task aims to generate molecular structures that match a given textual description, involve scientists directly in the design process, and enable intuitive and flexible control over molecular properties. This approach utilizes natural language as input, allowing users to specify high-level goals, such as desired functional groups, scaffolds, or physicochemical properties, thereby eliminating the need for expert-crafted constraints or domain-specific encoding schemes. Early work in this field primarily focused on retrieval-based approaches [7, 28], followed by direct generation methods such as text-to-SMILES sequence translation [8], where molecules are represented using the Simplified Molecular Input Line

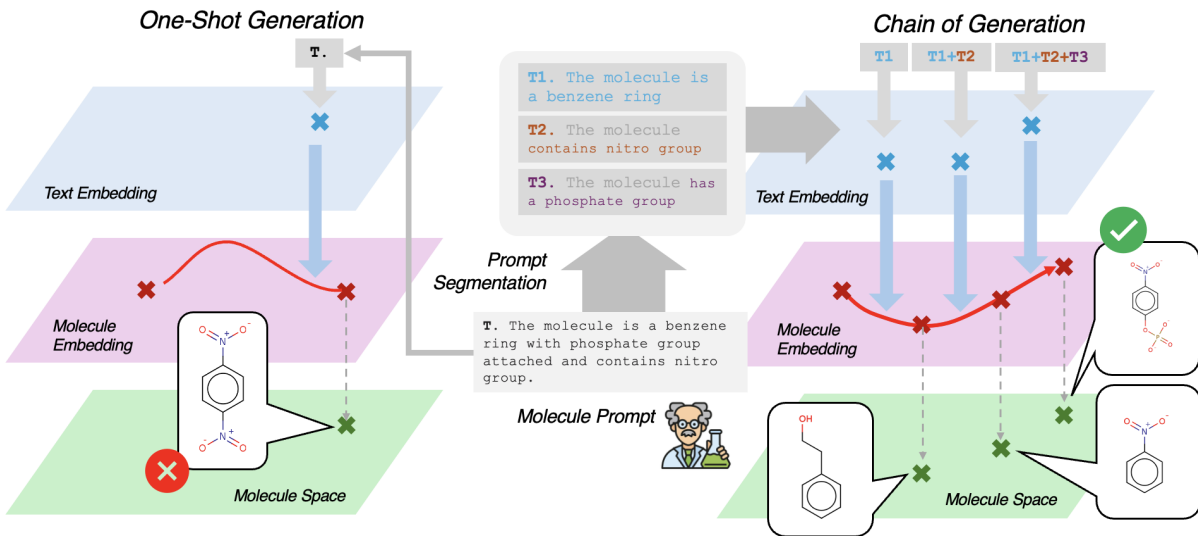


Figure 1: Overview of the proposed chain-of-generation (CoG). **Left:** Existing diffusion approaches (e.g., [29]) where a text prompt T is projected to a vector in the text-embedding space used for guiding the diffusion in the molecule embedding space for generation. **Right:** The proposed CoG approach firstly segments the text prompt into a set of components (T_1, T_2, T_3) and progressively includes the components into smaller prompts to guide multi-staged diffusion for molecule generation.

Entry System (SMILES) [24]. These approaches commonly employ a sequence-to-sequence modeling paradigm, directly mapping natural language prompts to SMILES strings.

Recent years have witnessed the impressive imaging generation capability of diffusion-based models [11], which model data through iterative denoising steps. This success has inspired their application to molecular generation tasks [9, 4], where their stochastic sampling process facilitates the creation of structurally diverse and novel molecules. Among these, latent diffusion models (LDMs) [20, 29] stand out for their efficiency and scalability. Instead of generating molecules directly in molecule space, LDMs operate over a learned continuous latent space, where generative modeling becomes more tractable, and the structural semantics of molecules can be more compactly captured. This decoupling of representation and generation allows LDMs to produce chemically valid and diverse molecules while remaining computationally efficient. LDMs typically follow a two-stage framework: (1) learning bidirectional projections that map molecular structures into and out of a rich latent space that captures molecular information; (2) guiding a generative search process within this latent space using natural language signals, then decoding the resulting latent vectors back into valid molecules.

While recent works focus on the first stage by improving molecular encoding and pretraining strategies [4], the second stage remains underdeveloped. In particular, without robust mechanisms to effectively relate the textual prompt space with the molecular latent space, the model may fail to identify the correct generation trajectory. This weak cross-modal correspondence can lead to semantically misaligned or structurally incoherent molecules, limiting both generation quality and controllability. Additionally, evaluating the semantic quality of generated molecules remains a critical

open challenge. Existing evaluation metrics are mostly adapted from NLP, such as BLEU [18] or Levenshtein distance [15], and do not fully capture whether a molecule truly reflects the intended semantics of a prompt, especially at the graph level. This makes human-in-the-loop assessment increasingly important to understand model behavior. Finally, most existing methods, such as 3M-Diffusion [29], rely on a one-shot conditioning paradigm, where the entire textual prompt is encoded once and used to guide the generation process. However, as shown in Fig. 1, we observe that this approach often fails when prompts are semantically complex or composed of multiple components. There are three main challenges of one-shot conditioning that limit the capacity of generation. First, as one-shot conditioning only uses the prompt once at the beginning of the generation, it is hard to attribute the generated components back to the prompt. Second, when there are multiple requirements in the prompt, the generated molecules may omit key substructures or components. Third, one-shot conditioning takes all the requirements at once, making it challenging to resolve and plan the generation procedure.

We then propose three principles to address these challenges. First, the generation process should be *guided* rather than end-to-end. Second, early-stage prompts should be utilized as a *regularization* in the later stage of generation. Third, the generation should consider *granularity*, ranging from coarse structures to fine-grained components. Building on this idea, we introduce a *training-free* Chain-of-Generation (CoG) framework for molecule generation. To further enhance semantic guidance, we introduce a post-alignment learning stage that strengthens the correspondence between textual signals and molecular latents. Finally, we perform extensive experiments on both synthetic and real-world datasets, showing that CoG yields more faithful, interpretable, and controllable generation.

2 Related work

Text-conditioned molecular generation is an emerging task at the intersection of computational chemistry and multimodal learning. It aims to generate molecular structures from natural language prompts, enabling intuitive, goal-directed molecular design. Early works approached this as a *retrieval* problem, learning joint embeddings to align text and molecular representations [7, 28]. While effective, these methods are constrained to pre-existing molecules and cannot generate novel structures. To overcome this, generation-based models emerged, typically translating text to SMILES strings [24]. Models such as MolT5 [8] and ChemT5 [5] use encoder-decoder transformers to map prompts to molecules, benefiting from large pretrained language models. However, prior sequence-to-sequence approaches exhibit severe *mode collapse*: despite the large solution space implied by flexible prompts, they repeatedly converge to a single rigid output. A second limitation is *chemical invalidity*. Because SMILES impose an artificially defined string grammar, models must implicitly relearn these rules, often producing syntactically or chemically invalid molecules. An illustrative example is provided in Section E. Graph-based diffusion methods alleviate the issue of invalid SMILES generation by operating directly on molecular graphs. However, performing diffusion in the original discrete graph space remains problematic: permutation invariance introduces many isomorphic orderings, and training often relies on auxiliary reconstruction losses [22, 21]. These losses are poorly suited for molecular graphs, where even small inconsistencies can distort chemical properties. To avoid these limitations, LDMs [29] encode graphs into a continuous latent space and perform text-conditioned diffusion there, preserving validity while enabling stochastic, conditional exploration for more diverse and novel molecules. Our work builds on LDMs, with a specific focus on improving the language-conditioned searching process. Rather than modifying molecular representations or encoders, we propose a multi-step planning framework that decomposes complex prompts into semantic subgoals, guiding the diffusion process to search within an aligned subspace in a more

faithful and interpretable manner.

3 Preliminary and Motivation

3.1 Preliminary

LDMs first learn a bidirectional mapping between molecular graphs and a continuous latent space, and then apply a conditional diffusion process to perform guided sampling in this space using natural language prompts, followed by decoding the resulting latent vectors back into molecular structures [20]. Zhu et al. [29] achieve this by training a variational graph auto-encoder [14], comprising a graph encoder and a graph decoder. The encoder maps input molecular graphs into latent Gaussian distributions using a pre-aligned Graph Isomorphism Network (GIN) [26]. The decoder reconstructs molecular structures from the latent vectors by a hierarchical variational autoencoder (HierVAE) [13]. The model is trained by minimizing the Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\text{elbo}} = -\mathbb{E}_{q(g|G)}[\log p(G | \hat{G})] + D_{\text{KL}} [q(g | G) \| p(g)], \quad (1)$$

where $\hat{G} = D(g)$ denotes the reconstructed molecular graph generated by the decoder D . The latent representation g is sampled from the approximate posterior $q(g | G)$, where G is the input molecular graph. Specifically, the encoder first processes G using a GIN, and then two separate multilayer perceptrons (MLPs) are applied to predict the mean and standard deviation of the latent Gaussian distribution. Using the reparameterization trick [14], a continuous latent vector g is sampled from this distribution. The Kullback–Leibler divergence term encourages the learned posterior $q(g | G)$ to stay close to the prior $p(g) = \mathcal{N}(0, I)$, thereby regularizing the latent space.

The text-conditioned generation is achieved by latent diffusion in the learned latent space. Starting from a clean latent representation g_0 , Gaussian noise is added according to a predefined schedule to obtain a noisy latent vector: $z_t = \sqrt{\bar{\alpha}_t}g_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$, where $\bar{\alpha}_t$ denotes the noise level at timestep t , and $\epsilon \sim \mathcal{N}(0, I)$. A denoising network θ is then trained to recover g_0 from z_t , conditioned on the text embedding c , by minimizing the following:

$$\mathbb{E}_{t,c,g_0,\epsilon} \left[\left\| \epsilon - \theta \left(\sqrt{\bar{\alpha}_t}g_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t, c \right) \right\|^2 \right],$$

which learns a probabilistic mapping from noisy latent representations to clean molecular embeddings, enabling the search for target molecules through the latent space based on natural language signals.

3.2 Motivation

Recent advances in latent diffusion models have shown promise in text-conditioned molecular generation, but complex prompts often remain challenging with one-shot conditioning: models sometimes fail to construct molecular structures that fully reflect all required semantics. To probe this issue, we investigate how diffusion models, as black-box generators, parse natural language descriptions and incrementally assemble molecular structures during denoising. We conduct a trajectory analysis by decoding partially denoised latents and visualizing intermediate outputs, enabling us to examine how structural features emerge and evolve throughout the generation process. We begin with simple prompts (e.g., "This molecule is made of a pyridine ring") and gradually increase complexity by adding compositional elements. We see that when the prompt includes three or more structural keywords, model performance begins to diverge significantly. Fig. 2 shows an example. The prompt is constructed from compositional vocabulary in our training set. In the visualization, colored regions correspond to substructures correctly generated in accordance with the

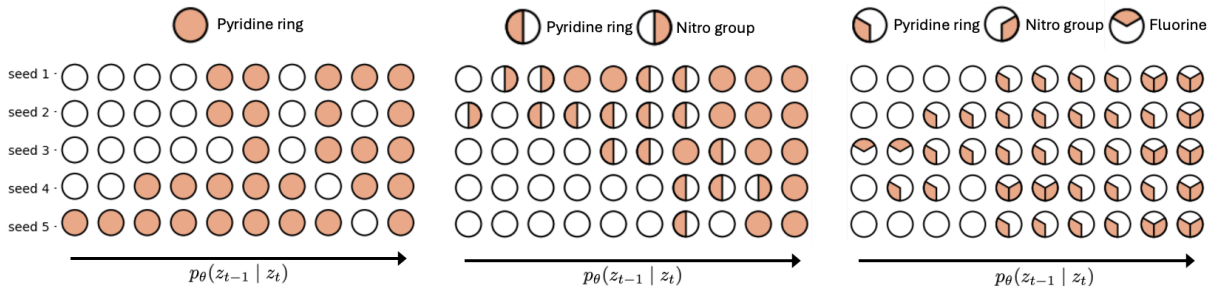


Figure 2: Generated molecules for three textual prompts. **Left:** "The molecule is made of a pyridine ring." **Middle:** "The molecule is a pyridine ring with a nitro substituent." **Right:** "The molecule is made of a pyridine ring substituted with both a nitro group and fluorine atoms." Note that as the complexity of the prompt increases, one-shot conditioning may not be able to satisfy all the requirements. Moreover, latent diffusion ensembles each component progressively during generation.

prompt, while uncolored parts indicate incomplete alignment. Notably, as we increase the number of components in the prompt, latent diffusion cannot simultaneously handle all the requirements, indicating the limitation of one-shot conditioning. Additionally, the model tends to resolve larger, high-level scaffolds first, for example, the pyridine ring, which is designated as the core structure in the prompt. Then, as denoising proceeds, it decodes peripheral substructures piece by piece (e.g., attaching a nitro group) that serve as modifiers or substituents. This illustrates that the model progressively integrates semantic elements from the prompt, aligning more closely with the intended meaning over the course of denoising, a behavior rarely observed in sequence-to-sequence models. Motivated by this observation, we propose our Chain-of-Generation framework in [Section 4.3](#).

4 Molecule Chain-of-Generation

Motivated by the aforementioned observation of one-shot conditioning in [Section 3.2](#), we propose a novel **Chain-of-Generation (CoG)** framework in this section, which decomposes each prompt into semantically meaningful segments and progressively guides the generation through these stages. Importantly, CoG operates entirely at sampling time without requiring retraining. We first discuss three challenges of one-shot conditioning. We then propose three principles of latent diffusion to address those challenges. Equipped with that understanding, we introduce our CoG framework. Finally, a post-alignment strategy is proposed to enhance the generation.

4.1 Challenge of one-shot conditioning

Although recent LDMs have demonstrated promising capabilities in text-conditioned molecular generation, a key challenge remains in how language guidance is incorporated into the latent generation process. Existing methods encode the entire textual prompt once and use it to condition the entire generation process. However, this one-shot conditioning strategy struggles with complex or compositional prompts. In [Section 3.2](#), empirical results are shown to illustrate the failure of one-shot conditioning. In this section, we further analyze the challenges of one-shot conditioning from three perspectives. **C1: Poor interpretability.** Since the prompt is used only once at the beginning, it is difficult to attribute parts of the generated molecule to specific components of the text, limiting transparency and controllability. **C2: Missing substructures.** When prompts describe multiple functional groups or hierarchical structures, the model often fails to assemble all

parts correctly, omitting key molecular subcomponents. As in Fig. 2, a prompt specifying (1) a pyridine ring, (2) a nitro group, and (3) fluorine yields a generated molecule that includes only the first two, omitting the fluorine atoms with one-shot conditioning. **C3: Ambitious conditioning.** One-shot strategies treat all prompt information simultaneously throughout the generation, ignoring the relationships among components and making it difficult to position components correctly within the overall structure. To address these limitations, we propose three principles of latent diffusion in the next section.

4.2 The Art of Latent Diffusion

In this section, we propose three principles to address the above three challenges, respectively.

Guided Search Process. To address **C1**, each attribute of the molecule is supposed to be traced back to the corresponding component in the prompt. Thus, rather than jumping blindly from Gaussian noise to the final manifold, the latent walks along a continuous path of feasible solutions where each stage acts as a bridging distribution that keeps the sample within a high-density overlapping region.

Early-Stage Prompts as Regularization. To help eliminate missing substructures in generation in **C2**, it is critical to consider both the current focus and historical information properly. The generation process should preserve the existing structure by leveraging historical information, while accurately producing the next components in accordance with the current stage.

Curriculum-Style Generation. **C3** highlights the difficulty of focusing on all the prompt components with one-shot conditioning. An alternative solution is to adopt a coarse-to-fine generation schedule. In generation, early stages establish the global structure, and later stages add details on top of it. Viewed heuristically, this resembles a continuation/curriculum procedure that makes small updates around the previous solution, which we find to stabilize sampling and improve adherence to later prompt components without overwriting earlier ones.

4.3 CoG: The Wisdom of Decomposing and Composing

Chain-of-Thought (CoT) reasoning in LLMs has shown how complex problems can be solved more reliably by decomposing them into sequential sub-tasks rather than tackling them all at once [23, 27]. Inspired by this philosophy, we design Chain-of-Generation (CoG) for molecular diffusion models. Unlike CoT, which operates in natural-language reasoning, CoG introduces a novel curriculum within the latent diffusion process. It begins by *decomposing* a molecular prompt into semantically meaningful components that reflect chemical granularity—such as scaffolds, functional groups, and fine-grained modifiers, and then organizes these components into a coarse-to-fine hierarchy. Each stage of diffusion is explicitly conditioned on the cumulative sub-prompts, allowing the model to progressively *assemble* molecular structures while preserving previously generated substructures. This structured planning and progressive denoising transform CoG into more than a direct adaptation of CoT: it is a chemically grounded mechanism that enables diffusion to generate molecules with higher semantic fidelity, interpretability, and controllability.

Prompt Decomposition and Planning. The first step of CoG is to decompose a text prompt T into a set of N prompt segments $\mathcal{S}(T) = [T_1, T_2, \dots, T_N]$. A planning process $\mathcal{P}$ then converts the segments into a set of sub-prompts:

$$\mathcal{P} \odot \mathcal{S}(T) = [T_1, T_1 + T_2, \dots, T_1 + T_2 + \dots + T_N \equiv T], \quad (2)$$

which describes a sequence of sub-goals that align with our hierarchical planning objective. In the illustrating example in Fig. 1, T is "The molecule is a benzene ring with a phosphate

group attached and contains a nitro group." We decompose it into: T_1 : "The molecule is a benzene ring.", T_2 : "The molecule contains a nitro group.", and T_3 : "The molecule has a phosphate group attached."

Researchers with a chemical background can segment the prompts manually, but this can be inefficient given a massive generation task. We propose an automated *hierarchical planning strategy* for prompt segmentation using a large-language model, inspired by how chemists typically approach molecule construction: The first sub-prompt T_1 is expected to capture the molecule’s primary scaffold or core structure that defines the molecular identity. Subsequent sub-prompts $T_2, T_3, \dots$ incrementally introduce additional structural details, such as functional groups (e.g., nitro, methyl, hydroxyl) that attach to the core. In later stages, the prompts typically encode finer-grained constraints, including specific atom types, substitution positions, or stereochemistry (e.g., "a chlorine atom at position 3" or "meta-substituted amine"). This planning strategy enables the model to prioritize the generation of key substructures early on and refine them step-by-step in a chemically interpretable and semantically grounded manner. Our strategy allows the model to construct the most chemically meaningful substructures early in the diffusion process, leading to more stable and interpretable intermediate representations. It promotes structural consistency by grounding the initial steps on high-confidence semantic anchors. In [Section B](#), we provide the instruction prompt we provided to LLM for segmentation and compare different LLM planning performances.

Progressive Latent Diffusion. In [Eq. \(2\)](#), we obtain a sequence of sub-prompts that incrementally refine the semantic constraints guiding generation while maintaining the inherited parts in the text space. At each stage k , we condition the model on $T_1 + \dots + T_k$ and use the latent vector generated in the previous stage z_{k-1} as the initialization for the next denoising process. Formally, we iterate:

$$z_k \sim \text{Denoising}(\text{init} = z_{k-1}, \text{cond} = E_t(T_{1:k})).$$

This chain-of-generation procedure enables the model to more faithfully follow the semantic progression of complex prompts. Earlier steps in the chain establish the coarse molecular backbone, while later stages introduce finer-grained modifications, such as specific functional groups or spatial arrangements. Importantly, the model should exhibit high confidence in retaining structures from previous steps. To achieve this, during later stages of the chain, we apply only the partial denoising trajectory determined by a step hyperparameter $s_{\text{start}} \in [0, S]$. In practice, we start the stochastic denoising process from an intermediate noise level, typically within the last 20% of the diffusion steps. This ensures that critical elements from z_{k-1} are largely preserved, while still allowing for prompt-conditioned stochastic exploration guided by T_k .

4.4 Post-alignment

The central representation for diffusion-based generation in our framework is the resampled latent vector g , obtained via the reparameterization trick in the VAE. This latent vector serves as the reconstruction target during diffusion training and defines the subspace where molecules are generated. To ensure that generation is effectively guided by language, it is crucial that g remains well aligned with the corresponding text embedding. Such cross-modal alignment enhances the semantic correspondence between molecular graphs and their textual descriptions, allowing the diffusion process to utilize prompt signals more effectively. In particular, by aligning g with the text representation, the model can perform more faithful and controllable search in the molecular embedding subspace, progressively assembling structures that match the intended semantics.

To achieve this, we introduce an additional post-alignment stage, where contrastive learning is performed directly between the resampled latent g and the text embedding extracted from SciBERT

[1]. During this stage, the graph encoder is frozen to fix the latent representation g , which serves as input to the diffusion model from Eq. (1). We conduct a contrastive learning *after* the latent vector g is sampled, optimizing the following:

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathcal{B}|} \sum_{(G_i, T_i) \in \mathcal{B}} \log \frac{\exp(\cos(\mathbf{g}_i, \mathbf{c}_i)/\tau)}{\sum_{j=1}^{|\mathcal{B}|} \exp(\cos(\mathbf{g}_i, \mathbf{c}_j)/\tau)},$$

where $\mathbf{g}_i = E_g(G_i)$ and $\mathbf{c}_i = E_t(T_i)$ denote the representations of the molecular graph G_i and its corresponding text prompt T_i , $\mathcal{B}$ represents a batch of molecule–text pairs sampled from the training set, $\cos(\cdot)$ denotes cosine similarity, and temperature τ controls the sharpness of the distribution.

By applying the post-alignment strategy, we establish a robust latent diffusion backbone, denoted as **GraphLDM**. This backbone serves as the foundation for all subsequent experiments. In Sections 5.2 and 5.3, we show that augmenting GraphLDM with our proposed Chain-of-Generation strategy further improves semantic alignment and enhances controllability throughout the generation process.

5 Experiments and results

We conduct extensive experiments to evaluate the proposed framework. In Section 5.1, we first examine the shortcomings of existing NLP-oriented metrics for text-to-molecule generation and motivate our adoption of MACCS fingerprints together with more chemically faithful evaluation measures. In Section 5.2, we then assess performance on benchmark datasets, showing that **GraphLDM** + **CoG** consistently outperforms both SMILES-based methods and prior graph-based baselines. Finally, in Section 5.3, we analyze the role of prompt planning, comparing different large language models (LLMs) for text segmentation and interpretation, and further study curriculum versus anti-curriculum strategies as well as the impact of prompt regularization in a synthetic setting.

5.1 Metrics

Our evaluation relies on graph similarity computed from MACCS fingerprints using the RDKit [2]. The MACCS fingerprint is a 166-bit, predefined keyset where each bit indicates the presence or absence of a specific substructure pattern, such as carbonyls or ring systems. The public 166-key subset is widely implemented in RDKit and it is commonly used throughout drug discovery [6, 17]. Because these bits are computed from the *molecular graph* instead of the SMILES string, MACCS is naturally invariant to SMILES enumeration and canonicalization, making it representation-agnostic for our text-to-graph task.

Previous sequence-based works have often adopted evaluation metrics such as BLEU, Fréchet ChemNet Distance (FCD) [3], and Text2Mol [7]. However, many of these metrics are either NLP-oriented or designed for tasks with different objectives, making them unsuitable for our text-to-graph molecular generation setting. They reward token-level alignment and are highly sensitive to SMILES permutations even when the underlying graphs are identical. Text2Mol measures text–SMILES alignment for *retrieval* rather than pairwise, graph-level generation fidelity, and FCD compares *set-level* unconditional distributions learning instead of one-to-one prompt-molecule pairs. For these reasons, MACCS plus Tanimoto similarity provides a chemically meaningful, permutation-invariant basis for assessing performance in our setting. As illustrated in Fig. 3, two molecules with nearly identical graphs (scaffold and functional groups) can have drastically different SMILES strings. String-based metrics (BLEU/Exact/Levenshtein) therefore underrate the match, whereas MACCS Tanimoto correctly reflects the high structural similarity.

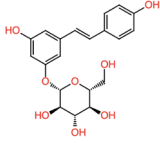
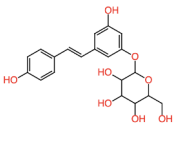
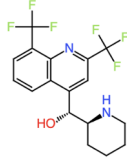
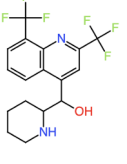
Text Description	Ground Truth	Distance Metrics	Output
The molecule is a stilbenoid that is trans-resveratrol substituted at position 3 by a beta-D-glucosyl residue... It is a stilbenoid, a polyphenol, a beta-D-glucoside and a monosaccharide derivative...	 <chem>C1=CC(=CC=C1/C=C/C2=CC(=CC(=C2)O[C@H]3[C@@H]([C@@H]([C@H]([C@H]([C@H]3O)O)O)O)O)O</chem>	MACCS SIM 1.0 Levenshtein distance: 61	 <chem>OCC1OC(Oc2cc(O)cc(C=Cc3ccc(O)cc3)c2)C(O)C(O)C1O</chem>
The molecule is an optically active form of [2,8-bis(trifluoromethyl)quinolin-4-yl]-(2-piperidyl)methanol having (+)-(11R,2'S)-erythro-configuration... It is an enantiomer of a (-)-(11S,2'R)-erythro-mefloquine.	 <chem>C1CCN[C@@H]([C1])C[C@@H]([C@H]2=CC(=NC3=C2C=C(C3C(C(F)(F)F)C(F)(F)F)O)N)C(F)(F)F</chem>	MACCS SIM 1.0 Levenshtein distance: 49	 <chem>OC(c1cc(C(F)(F)F)nc2c(C(F)(F)F)cccc12)C1CCCCN1</chem>

Figure 3: Two examples where nearly identical molecular graphs are represented by drastically different SMILES strings. In such cases, NLP-based string metrics (e.g., Levenshtein) often assign a high distance score, failing to reflect the true structural correspondence. In contrast, our graph-based evaluation correctly captures scaffold and functional group similarity.

We report four evaluation metrics to assess model performance based on the MACCS. (1) **Validity** measures the proportion of chemically valid molecules among all generated outputs, ensuring that the generation process respects fundamental chemical rules. (2) **Qualified Novelty(Q-Nov)** evaluates the extent to which generated molecules, once they satisfy the textual prompt, are further novel and diverse, highlighting the model’s ability to avoid redundancy and capture a broad range of structural patterns. (3) **Qualified Coverage(Q-Cov)** reflects how well molecules that satisfy the textual prompt explore new chemical space relative to known valid structures, highlighting the model’s potential to generate practically useful candidates beyond the training distribution. (4) **Balanced Quality Index(BQI)** integrates multiple perspectives into a single score, providing a holistic view of generation quality by balancing both the fidelity and the creativity of the generation. The detailed formulations are provided in [Section C](#).

5.2 Benchmark Evaluation

We evaluate the effectiveness of our Chain-of-Generation strategy on benchmark datasets. Specifically, we conduct experiments on PubChem [16] and ChEBI-20 [7], restricting all training, validation, and test molecules to fewer than 30 atoms to control molecular complexity. Following the progressive diffusion strategy described in [Section 4.3](#), we divide each textual prompt into three stages: core structure, functional groups and modifiers, and stereochemical configurations. We employ Grok v3 for segmentation and apply our CoG strategy for curriculum-based prompting.

[Table 1](#) reports the performance comparison on ChEBI-20 and PubChem. We denote our improved diffusion backbone as GraphLDM, which integrates a post-alignment strategy to establish a strong starting point. Building on this, we apply the CoG for progressive conditional generation, which incorporates our proposed segmentation and curriculum prompting strategy. As shown in the table, GraphLDM + CoG achieves consistent improvements over the baselines [19, 5, 8, 29] across both

Table 1: Performance comparison on ChEBI-20 and PubChem datasets. All values are reported in percentages.

Method	ChEBI-20				PubChem			
	BQI	Q-Cov	Q-Nov	Validity	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods								
MolT5 small	51.45 $\pm$ 0.00	24.25 $\pm$ 0.00	4.96 $\pm$ 0.00	78.45 $\pm$ 0.00	44.23 $\pm$ 0.00	10.88 $\pm$ 0.00	1.04 $\pm$ 0.00	72.93 $\pm$ 0.00
MolT5 large	55.95 $\pm$ 0.00	20.35 $\pm$ 0.00	2.23 $\pm$ 0.00	98.08 $\pm$ 0.00	51.44 $\pm$ 0.00	16.27 $\pm$ 0.00	1.45 $\pm$ 0.00	94.15 $\pm$ 0.00
ChemT5 small	57.62 $\pm$ 0.01	23.89 $\pm$ 0.01	3.27 $\pm$ 0.02	96.98 $\pm$ 0.01	52.75 $\pm$ 0.06	20.05 $\pm$ 0.06	2.51 $\pm$ 0.04	92.01 $\pm$ 0.14
ChemT5 base	57.89 $\pm$ 0.02	25.08 $\pm$ 0.02	3.48 $\pm$ 0.03	97.22 $\pm$ 0.04	53.01 $\pm$ 0.11	20.8 $\pm$ 0.18	2.79 $\pm$ 0.05	90.68 $\pm$ 0.14
BioT5 base	58.92 $\pm$ 0.17	20.58 $\pm$ 0.27	4.47 $\pm$ 0.09	99.99 $\pm$ 0.01	54.89 $\pm$ 0.15	17.03 $\pm$ 0.23	2.98 $\pm$ 0.08	99.97 $\pm$ 0.03
BioT5 plus	56.71 $\pm$ 0.12	15.31 $\pm$ 0.12	2.46 $\pm$ 0.01	100.00 $\pm$ 0.00	55.05 $\pm$ 0.05	17.14 $\pm$ 0.17	3.06 $\pm$ 0.02	100.00 $\pm$ 0.00
Graph-based methods								
3M diffusion	56.21 $\pm$ 0.14	40.20 $\pm$ 0.22	9.81 $\pm$ 0.06	100.00 $\pm$ 0.00	55.29 $\pm$ 0.01	40.10 $\pm$ 0.05	10.75 $\pm$ 0.03	100.00 $\pm$ 0.00
GraphLDM	59.54 $\pm$ 0.11	43.19 $\pm$ 0.12	11.12 $\pm$ 0.11	100.00 $\pm$ 0.00	57.88 $\pm$ 0.09	41.06 $\pm$ 0.22	12.59 $\pm$ 0.04	100.00 $\pm$ 0.00
GraphLDM + CoG	60.47 $\pm$ 0.03	45.58 $\pm$ 0.08	11.10 $\pm$ 0.01	100.00 $\pm$ 0.00	58.07 $\pm$ 0.08	43.13 $\pm$ 0.21	11.85 $\pm$ 0.10	100.00 $\pm$ 0.00

Table 2: Comparison of prompting strategies across different LLMs. All values are reported in percentages. The *validity* metric is excluded, as perfect chemical validity (100%) is guaranteed by our motif-based graph decoder for all combinations of LLM and prompting strategies.

LLM	Prompting Strategy	BQI	Q-Cov	Q-Nov
GPT-4o	One-shot (T)	49.64 $\pm$ 0.12	31.90 $\pm$ 0.35	9.98 $\pm$ 0.05
	One-shot (T_s) + CoG ($T_{s:m}$) + CoG (T)	46.93 $\pm$ 0.32	28.62 $\pm$ 0.62	7.84 $\pm$ 0.19
	One-shot (T_l) + CoG ($T_{l:m}$) + CoG (T)	48.66 $\pm$ 0.19	31.38 $\pm$ 0.31	8.55 $\pm$ 0.13
	One-shot (T_l) + CoG (T_m) + CoG (T_s)	38.18 $\pm$ 0.16	10.77 $\pm$ 0.00	3.19 $\pm$ 0.07
Gemini 2.5 Pro	One-shot (T_s) + CoG ($T_{s:m}$) + CoG (T)	45.86 $\pm$ 0.24	26.15 $\pm$ 0.31	7.34 $\pm$ 0.18
	One-shot (T_l) + CoG ($T_{l:m}$) + CoG (T)	50.69 $\pm$ 0.27	35.48 $\pm$ 0.18	10.19 $\pm$ 0.32
	One-shot (T_l) + CoG (T_m) + CoG (T_s)	34.32 $\pm$ 0.00	4.62 $\pm$ 0.00	1.22 $\pm$ 0.00
Grok v3	One-shot (T_s) + CoG ($T_{s:m}$) + CoG (T)	47.06 $\pm$ 0.06	29.02 $\pm$ 0.18	7.86 $\pm$ 0.02
	One-shot (T_l) + CoG ($T_{l:m}$) + CoG (T)	51.02 $\pm$ 0.12	36.41 $\pm$ 0.18	10.16 $\pm$ 0.14
	One-shot (T_l) + CoG (T_m) + CoG (T_s)	35.29 $\pm$ 0.00	6.15 $\pm$ 0.00	1.78 $\pm$ 0.00

datasets, particularly in BQI and Q-Cov, while maintaining perfect validity. For instance, Compared with GraphLDM, GraphLDM + CoG achieves clear gains: BQI increases from 59.54 ± 0.11 to 60.47 ± 0.03 , Q-Cov from 43.19 ± 0.12 to 45.58 ± 0.08 while maintaining perfect validity on ChEBI-20 dataset. These results highlight that CoG brings measurable benefits beyond baseline refinements, strengthening semantic alignment throughout the generation process. We conducted each experiment with three independent runs and report the mean $\pm$ standard deviation to ensure statistical robustness. The improvements are stable across runs, demonstrating the reliability of CoG’s performance gain. While Q-Nov occasionally shows a slight decrease, this is largely due to reduced variability among generated molecules, which is an expected trade-off for improved determinism, as CoG more reliably parses information from the text. We note that the performance gain on the PubChem dataset is comparatively smaller, which can be attributed to its less curated nature. As reported by Zhu et al. [30], nearly 50% of the corpus consists of uninformative or loosely aligned textual descriptions. For example, the entry for PubChem CID 5369129 reads: “*The molecule is https://www.chemicalbook.com/ProductChemicalPropertiesCB9738874_EN.htm. It has a role as a glycine receptor antagonist.*” Such noisy and unstructured text limits the effectiveness of our staged prompt strategy. In contrast, the more curated and semantically rich ChEBI-20 dataset enables CoG to better exploit structural–textual correlations, resulting in more pronounced improvements. A detailed analysis by prompt segmentation is provided in Section D.

5.3 Prompt Planning and Ablations

To evaluate the effectiveness of prompt segmentation, planning strategy, and prompt regularization on our proposed Chain-of-Generation. We construct a synthetic dataset with controlled semantic structure consisting of zero-shot molecule-prompt pairs. Each prompt T in this dataset is a compositional description of a molecular structure, designed such that it can be decomposed into three sequential subgoals: T_1 , T_2 , and T_3 . To test whether different LLMs can understand and segment prompts according to our proposed coarse-to-fine planning strategy, we compare three representative models: GPT-4o [12], Gemini 2.5 Pro [10], and Grok v3 [25]. We define three levels of prompt components. T_l : "Large" components representing the primary scaffold of the molecule; T_m : "Medium" components representing functional groups or substituents; T_s : "Small" components containing fine-grained modifications such as specific atoms. We also evaluate three prompting strategies to guide the generation process: one-shot, fine-to-coarse, and coarse-to-fine.

From Table 2, we see that coarse-to-fine prompting consistently outperforms the one-shot baseline, whereas the fine-to-coarse strategy performs worse across all settings. This shows the importance of planning design in the CoG framework. When the model is forced to focus prematurely on atomic-level details, it later struggles to assemble a coherent molecule around a stable scaffold. By contrast, starting from the core structure and progressively introducing functional groups and modifications allows the model to anchor on the main skeleton and refine the molecule step by step, closely mirroring how human chemists approach molecular design.

We further conduct an ablation study to examine the effect of early-stage prompt regularization, as discussed in Section 4.2. Following the coarse-to-fine prompting strategy, for example, we generate an initial molecule from the one-shot prompt T_l , and then continue the chain-of-generation using only T_m instead of the combined prompt $T_{l:m}$. The results show that when guided solely by a new intermediate prompt, the model tends to forget the previously established structure, even under structured planning. This leads to degraded final performance compared to the one-shot baseline. Turning to prompt segmentation quality across different LLMs, we find that Grok v3 generally provides the most reliable results, correctly identifying the primary scaffold, functional groups, and fine-grained details in complex prompts. In contrast, GPT-4o occasionally mixes core and peripheral elements, sometimes overemphasizing atomic-level information too early or revisiting the scaffold at inappropriate stages (see Section A for comparison).

6 Conclusion

We proposed CHAIN-OF-GENERATION, a novel framework that brings structured planning into text-to-molecule synthesis. By decomposing prompts into semantically meaningful stages and guiding generation through progressive latent diffusion, CoG transforms one-shot generation into a transparent and controllable process. This design enhances semantic fidelity and substructure alignment, providing interpretability by linking prompt components to molecular features. Coupled with a post-alignment backbone, CoG demonstrates significant gains over existing baselines and introduces a principled approach that can inspire more interpretable and controllable molecular generation in future work.

References

- [1] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

- [2] A Patrícia Bento, Anne Hersey, Eloy Félix, Greg Landrum, Anna Gaulton, Francis Atkinson, Louisa J Bellis, Marleen De Veij, and Andrew R Leach. An open source chemical structure curation pipeline using rdkit. *Journal of Cheminformatics*, 12(1):51, 2020.
- [3] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.
- [4] Jinho Chang and Jong Chul Ye. Ldmol: Text-to-molecule diffusion model with structurally informative latent space. *arXiv preprint arXiv:2405.17829*, 2024.
- [5] Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. Unifying molecular and textual representations via multi-task language modelling. In *International Conference on Machine Learning*, pages 6140–6157. PMLR, 2023.
- [6] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.
- [7] Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, 2021.
- [8] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.
- [9] Haisong Gong, Qiang Liu, Shu Wu, and Liang Wang. Text-guided molecule generation with diffusion language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 109–117, 2024.
- [10] Google DeepMind. Gemini 2.5 Pro, 2024. <https://deepmind.google/technologies/gemini/>.
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [12] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [13] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, pages 4839–4848. PMLR, 2020.
- [14] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.
- [15] VI Lcvenshtcin. Binary coors capable or ‘correcting deletions, insertions, and reversals. In *Soviet physics-doklady*, volume 10, 1966.
- [16] Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*, 2023.

- [17] Steve O’Hagan and Douglas B Kell. Understanding the foundations of the structural similarities between marketed drugs and endogenous human metabolites. *Frontiers in pharmacology*, 6:105, 2015.
- [18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [19] Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. *arXiv preprint arXiv:2310.07276*, 2023.
- [20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [21] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part I 27*, pages 412–422. Springer, 2018.
- [22] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.
- [23] 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.
- [24] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.
- [25] xAI. Grok v3, 2024. <https://x.ai/blog/grok-3>.
- [26] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [27] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.
- [28] Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1):862, 2022.
- [29] Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. 3m-diffusion: Latent multi-modal diffusion for language-guided molecular structure generation. In *First Conference on Language Modeling*, 2024.
- [30] Yihan Zhu, Gang Liu, Eric Inae, and Meng Jiang. Moltxttnet: A two-million molecule-text dataset for multimodal molecular learning. *arXiv preprint arXiv:2506.00009*, 2025.

A Visualization and segmentation across LLMs

As illustrated in Fig. 4, GPT-4o occasionally fails to follow the correct order, resulting in the CoG process prematurely focusing on small decorative components during the early coarse stage, and subsequently struggling to reorganize the intermediate results to accommodate the core scaffold in the finer stages. In contrast, Grok v3 shows a more stable and semantically aligned understanding of hierarchical prompts.

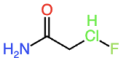
Primary structure?	GPT 4o	Gemini 2.5 Pro	Grok v3
	P		
	I		
	F	F	

Figure 4: Segmentation results across different LLMs

Table 3: Comparison of system prompt designs across different LLMs. All values are reported in percentages. The **validity** metric is excluded, as perfect chemical validity is guaranteed by our motif-based graph decoder.

LLM	Prompting Strategy	BQI	Q-Cov	Q-Nov
	One-shot (T)	49.64 ± 0.12	31.90 ± 0.35	9.98 ± 0.05
One piece	GPT-4o One-shot (T_l)	36.20 ± 0.00	9.23 ± 0.00	2.43 ± 0.00
	GPT-4o One-shot ⁻ (T_l)	35.96 ± 0.00	9.23 ± 0.00	2.34 ± 0.00
	Gemini 2.5 Pro One-shot (T_l)	37.61 ± 0.00	12.31 ± 0.00	3.18 ± 0.00
	Gemini 2.5 Pro One-shot ⁻ (T_l)	36.20 ± 0.00	9.23 ± 0.00	2.43 ± 0.00
	Grok v3 One-shot (T_l)	37.84 ± 0.00	12.31 ± 0.00	3.29 ± 0.00
	Grok v3 One-shot ⁻ (T_l)	39.70 ± 0.23	15.38 ± 0.00	4.31 ± 0.14
Two pieces	GPT-4o One-shot (T_l) + CoG ($T_{l:m}$)	45.08 ± 0.27	24.21 ± 0.47	6.98 ± 0.19
	GPT-4o One-shot ⁻ (T_l) + CoG ⁻ ($T_{l:m}$)	43.88 ± 0.16	23.28 ± 0.35	6.38 ± 0.09
	Gemini 2.5 Pro One-shot (T_l) + CoG ($T_{l:m}$)	46.52 ± 0.33	29.64 ± 0.47	7.94 ± 0.24
	Gemini 2.5 Pro One-shot ⁻ (T_l) + CoG ⁻ ($T_{l:m}$)	42.51 ± 0.41	20.62 ± 0.62	5.62 ± 0.25
	Grok v3 One-shot (T_l) + CoG ($T_{l:m}$)	48.54 ± 0.22	30.97 ± 0.35	8.06 ± 0.15
	Grok v3 One-shot ⁻ (T_l) + CoG ⁻ ($T_{l:m}$)	41.57 ± 0.30	18.87 ± 0.47	5.09 ± 0.18
Three pieces	GPT-4o One-shot (T_l) + CoG ($T_{l:m}$) + CoG (T)	48.66 ± 0.19	31.38 ± 0.31	8.55 ± 0.13
	GPT-4o One-shot ⁻ (T_l) + CoG ⁻ ($T_{l:m}$) + CoG (T)	49.82 ± 0.20	33.75 ± 0.18	9.69 ± 0.22
	Gemini 2.5 Pro One-shot (T_l) + CoG ($T_{l:m}$) + CoG (T)	50.69 ± 0.27	35.48 ± 0.18	10.19 ± 0.32
	Gemini 2.5 Pro One-shot ⁻ (T_l) + CoG ⁻ ($T_{l:m}$) + CoG (T)	49.68 ± 0.06	33.64 ± 0.47	9.43 ± 0.16
	Grok v3 One-shot (T_l) + CoG ($T_{l:m}$) + CoG (T)	51.02 ± 0.12	36.41 ± 0.18	10.16 ± 0.14
	Grok v3 One-shot ⁻ (T_l) + CoG ⁻ ($T_{l:m}$) + CoG (T)	50.50 ± 0.09	35.28 ± 0.64	9.90 ± 0.11

B System prompts design

Prompt segmentation plays a critical role in our CoG framework. The effectiveness of chain-of-generation heavily depends on whether LLMs can correctly understand a molecule’s natural language description and decompose it into components following a hierarchical order. When LLMs successfully perform this segmentation, identifying and separating structural elements in the correct sequence leads to better overall performance. As such, the design of the system prompt provided to the LLM is crucial.

In this section, we compare two designs of system prompts: one with detailed component definitions and one without them (–). As shown in Table 3, Grok v3 with the detailed component definitions shows a stable performance. The performance gap across different LLMs can be attributed to their varying abilities to understand the text description and accurately segment it into a coarse-to-fine hierarchy.

System Prompt without detailed instruction:

You are a highly capable AI tasked with generating four text outputs (A only, AB, C only, BC) by segmenting a single molecule’s text description provided as {DESCRIPTION}. The description includes three components (A, B, C) in a syntactic form such as "The molecule contains..." or "The molecule is made of...". Your task is to analyze the description, identify components A, B, and C based on their roles, and generate four outputs with modified text descriptions containing the specified component subsets.

Output Generation Rules:

A only: Include only the A component in the text description.
AB: Include both A and B components in the text description.
C only: Include only the C component in the text description.
BC: Include both B and C components in the text description.

Instructions:

Read the text description {DESCRIPTION} to identify A, B, and C based on their significance in the molecule.
Write the new description in a consistent format, e.g., "The molecule is made of [component(s)]", ensuring grammatical correctness.
If components are ambiguous, prioritize the most significant part as A, the next as B, and the least significant as C.

System Prompt with detailed instruction:

You are a highly capable AI tasked with generating four text outputs (A only, AB, C only, BC) by segmenting a single molecule’s text description provided as {DESCRIPTION}. The description includes three components (A, B, C) in a syntactic form such as "The molecule contains..." or "The molecule is made of...". Your task is to analyze the description, extract the components A, B, and C based on their semantic roles, and generate four outputs with modified text descriptions containing the specified component subsets. The output should include only the modified text descriptions.

Component Definitions:

A: The main structural core or primary ring system (e.g., benzene ring, naphthalene ring, pyrimidine ring) or the most central atom in smaller molecules (e.g., carbon, nitrogen,

phosphorus).

B: Medium-sized functional groups or secondary ring systems directly attached to the core (e.g., amide group, phosphate group, nitro group, cyclohexane ring).

C: Small atomic or functional group modifiers attached to A or B (e.g., fluorine, chlorine, bromine, iodine, hydroxyl group).

Output Generation Rules:

From heavy to light:

A only: Include only the A component in the text description.

AB: Include both A and B components in the text description.

From light to heavy:

C only: Include only the C component in the text description.

BC: Include both B and C components in the text description.

For the input description:

Read the text description {DESCRIPTION} to identify A, B, and C based on their structural significance, regardless of the syntactic structure.

Write the new description in a consistent format, e.g., "The molecule is made of [component(s)]", ensuring grammatical correctness.

If a component appears ambiguous, prioritize the largest ring system or core as A, the next significant group as B, and the smallest modifier as C.

System Prompt on real datasets:

You are a highly capable AI tasked with analyzing a molecule's text description provided as {DESCRIPTION} and outputting a single number representing the count of distinct components, along with increment of textual segmentations. The description may include phrases like "The molecule is made of..." or similar syntactic forms describing molecular components. Your task is to segment the description into meaningful components based on their semantic roles and provide the reasoning for your segmentation.

Component Identification Guidelines

To determine the number of components, consider the following candidate definitions as a flexible framework. Use your reasoning to adapt these definitions or create new ones based on the specific description provided:

Core Structure: The primary scaffold or backbone (e.g., glutamyl-alanine, tyrosine, glycerone) counts as 1 component.

Functional Groups: Each distinct functional group (e.g., hydroxy, methyl, oxo, amino) is counted separately, even if multiple instances exist (e.g., three hydroxy groups = three components).

Modifiers: Each distinct modifier (e.g., ionic forms, complex substituents like methylthioethyl, stereochemical configurations) is counted separately.

Stereochemistry: Enantiomeric or stereochemical configurations (e.g., L-enantiomer, (2S,4S)-configuration) are counted as modifiers if explicitly described as part of the molecule's structure.

Chemical Properties: Descriptive attributes of the molecule, such as acidity, basicity, or physical state (e.g., solid, liquid, gas).

You are free to refine or reinterpret these categories based on the description's context.

Output Format:

For a molecule with n components, the output line includes:
 The integer n (total component count).
 A tab.
 A fluent sentence for part 1 (core).
 A tab.
 A fluent sentence for parts 1+2 (core + first modifier/functional group).
 A tab.
 A fluent sentence for parts 1+2+3, and so on, up to all n parts.
 Sentences are concise, chemically accurate, and cumulative, incorporating all prior components in each subsequent sentence.

Component Counting: Each core, functional group, and modifier was counted separately, without grouping similar substituents.

Example:

Input: "The molecule is a 1,8-naphthyridine derivative that is 4-oxo-1,4-dihydro-1,8-naphthyridine-3-carboxylic acid bearing additional 2,4-difluorophenyl, fluoro and 3-aminopyrrolidin-1-yl substituents at positions 1, 6 and 7 respectively. It is a 1,8-naphthyridine derivative, an amino acid, a monocarboxylic acid, an organofluorine compound, an aminopyrrolidine, a tertiary amino compound, a primary amino compound and a quinolone antibiotic. It is a conjugate base of a 1-[6-carboxy-8-(2,4-difluorophenyl)-3-fluoro-5-oxo-5,8-dihydro-1,8-naphthyridin-2-yl]pyrrolidin-3-aminium."

Output:

Number: 5

Segmentations: The molecule is a heterocycle with a naphthyridine core.

The molecule is a heterocycle with a naphthyridine core, substituted by a 2,4-difluorophenyl group at position 1.

The molecule is a heterocycle with a naphthyridine core, substituted by a 2,4-difluorophenyl group at position 1 and a fluoro group at position 6.

The molecule is a heterocycle with a naphthyridine core, substituted by a 2,4-difluorophenyl group at position 1, a fluoro group at position 6, and a 3-aminopyrrolidin-1-yl group at position 7.

The molecule is a heterocycle with a naphthyridine core, substituted by a 2,4-difluorophenyl group at position 1, a fluoro group at position 6, a 3-aminopyrrolidin-1-yl group at position 7, and a carboxy group at position 3 to form a quinolone antibiotic.

Constraints:

Ensure the segmentation is concise, chemically informed, fluent, and directly tied to the description. If the description is ambiguous, make reasonable assumptions.

C Evaluation metrics

We derive four final metrics from basic proportions of generated molecules G' that satisfy different levels of MACCS Tanimoto similarity with the ground truth graph G , as well as their pairwise distinctness. Specifically, we define the following intermediate proportions:

- (1) p_{base} : proportion of generated molecules with $f(G', G) > 0.5$, where f denotes the MACCS Tanimoto similarity, representing molecules that satisfy the basic textual fidelity requirements.
- (2) p_{qual} : preferred proportion of generated molecules if $f(G', G)$ in $(0.5, 0.8)$, which indicates that the qualified molecules meet textual requirements while still retaining freedom and creativity.
- (3) p_{dist} : evaluates the structural variability among generated outputs. It is computed as the average

pairwise distance $1 - f(G'_i, G'_j)$ between all molecules that broadly satisfy the textual fidelity. (4) p_{val} : proportion of chemically valid molecules among all generated outputs, determined by standard chemistry rules.

Based on these quantities, we define three composite evaluation metrics. Qualified Novelty(Q-Nov) is computed as $p_{\text{qual}} \times p_{\text{dist}}$, capturing the extent to which the generated molecules simultaneously exhibit high fidelity, novelty, and mutual distinctness. Qualified Coverage(Q-Cov) is defined as $p_{\text{val}} \times p_{\text{qual}}$, reflecting how well valid molecules extend into new chemical space. Finally, the Balanced Quality Index (BQI) integrates all dimensions into a single score through a equally weighted combination of p_{val} , p_{base} , p_{dist} , and p_{qual} .

D Performance on benchmark subsets

To apply our CoG strategy on benchmark datasets, we first define segmentation rules with three stages: core structure, functional groups and modifiers, and stereochemical configurations. To fully capture the chemical information contained in the often complex and unnormalized textual prompts, we leverage Grok v3 and its reasoning capabilities to parse and segment each prompt, as illustrated in Section B. As shown in Fig. 5 and Fig. 6, most prompts can be segmented into two or three pieces, successfully extracting the core structure, functional groups and modifiers, and stereochemical configurations. Results across different segmentation subsets are summarized in Tables 4 to 10.

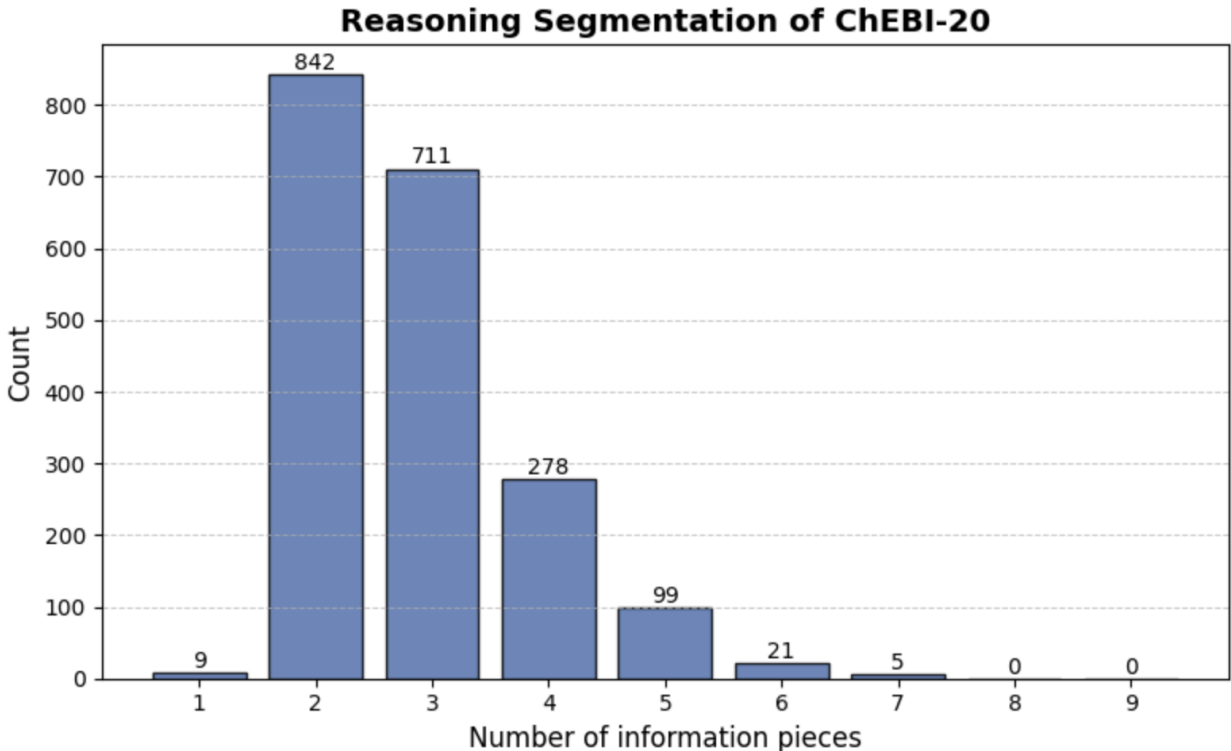


Figure 5: Distribution of number of information pieces after reasoning segmentation on ChEBI-20

E Visualization of generated samples

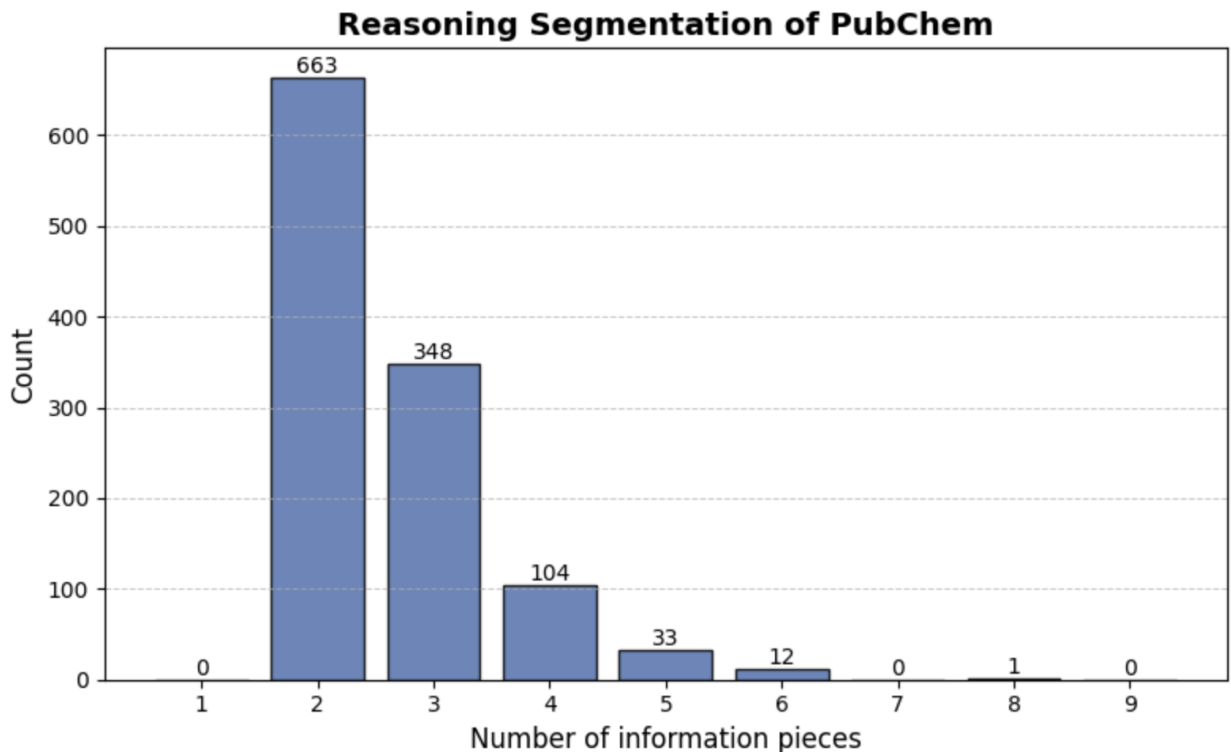


Figure 6: Distribution of number of information pieces after reasoning segmentation on PubChem

Table 4: Performance on the ChEBI-20 subset where each textual prompt contains five pieces of information. All values are reported as percentages.

Method	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods				
MolT5 small	51.37 $\pm$ 0.00	24.49 $\pm$ 0.00	7.07 $\pm$ 0.00	66.33 $\pm$ 0.00
MolT5 large	57.56 $\pm$ 0.00	20.41 $\pm$ 0.00	3.04 $\pm$ 0.00	97.96 $\pm$ 0.00
ChemT5 small	61.07 $\pm$ 0.00	30.54 $\pm$ 0.10	5.88 $\pm$ 0.00	94.90 $\pm$ 0.00
ChemT5 base	58.87 $\pm$ 0.10	22.59 $\pm$ 0.24	4.49 $\pm$ 0.00	96.07 $\pm$ 0.24
BioT5 base	67.39 $\pm$ 0.30	34.08 $\pm$ 1.74	13.27 $\pm$ 0.47	100.00 $\pm$ 0.00
BioT5 plus	63.20 $\pm$ 0.20	26.26 $\pm$ 0.42	7.15 $\pm$ 0.18	100.00 $\pm$ 0.00
Graph-based methods				
3M diffusion	59.70 $\pm$ 0.33	43.95 $\pm$ 0.59	12.72 $\pm$ 0.28	100.00 $\pm$ 0.00
GraphLDM	67.99 $\pm$ 0.20	58.16 $\pm$ 0.10	17.82 $\pm$ 0.37	100.00 $\pm$ 0.00
GraphLDM + CoG	68.37 $\pm$ 0.20	59.52 $\pm$ 0.62	16.94 $\pm$ 0.22	100.00 $\pm$ 0.00

Table 5: Performance on the ChEBI-20 subset where each textual prompt contains four pieces of information. All values are reported as percentages.

Method	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods				
MolT5 small	53.25 $\pm$ 0.00	27.79 $\pm$ 0.00	7.84 $\pm$ 0.00	68.95 $\pm$ 0.00
MolT5 large	59.80 $\pm$ 0.00	29.24 $\pm$ 0.00	4.39 $\pm$ 0.00	97.11 $\pm$ 0.00
ChemT5 small	59.68 $\pm$ 0.04	27.82 $\pm$ 0.04	4.84 $\pm$ 0.05	95.98 $\pm$ 0.09
ChemT5 base	60.61 $\pm$ 0.07	28.88 $\pm$ 0.08	5.27 $\pm$ 0.08	96.37 $\pm$ 0.04
BioT5 base	64.22 $\pm$ 0.64	30.42 $\pm$ 0.83	9.27 $\pm$ 0.30	100.00 $\pm$ 0.00
BioT5 plus	59.72 $\pm$ 0.69	19.54 $\pm$ 0.49	4.39 $\pm$ 0.13	100.00 $\pm$ 0.00
Graph-based methods				
3M diffusion	57.69 $\pm$ 0.62	42.96 $\pm$ 1.12	11.47 $\pm$ 0.14	100.00 $\pm$ 0.00
GraphLDM	63.27 $\pm$ 0.20	50.71 $\pm$ 0.42	14.82 $\pm$ 0.17	100.00 $\pm$ 0.00
GraphLDM + CoG	63.74 $\pm$ 0.04	51.58 $\pm$ 0.11	13.90 $\pm$ 0.09	100.00 $\pm$ 0.00

Table 6: Performance on the ChEBI-20 subset where each textual prompt contains three pieces of information. All values are reported as percentages.

Method	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods				
MolT5 small	52.61 $\pm$ 0.00	26.06 $\pm$ 0.00	5.47 $\pm$ 0.00	77.61 $\pm$ 0.00
MolT5 large	56.08 $\pm$ 0.00	20.28 $\pm$ 0.00	2.29 $\pm$ 0.00	98.03 $\pm$ 0.00
ChemT5 small	57.93 $\pm$ 0.03	23.70 $\pm$ 0.03	3.27 $\pm$ 0.02	97.04 $\pm$ 0.00
ChemT5 base	58.32 $\pm$ 0.04	25.93 $\pm$ 0.04	3.63 $\pm$ 0.03	97.15 $\pm$ 0.05
BioT5 base	59.16 $\pm$ 0.16	20.62 $\pm$ 0.26	4.12 $\pm$ 0.08	100.00 $\pm$ 0.00
BioT5 plus	57.08 $\pm$ 0.14	15.35 $\pm$ 0.32	2.39 $\pm$ 0.07	100.00 $\pm$ 0.00
Graph-based methods				
3M diffusion	56.29 $\pm$ 0.04	40.17 $\pm$ 0.05	10.00 $\pm$ 0.06	100.00 $\pm$ 0.00
GraphLDM	59.91 $\pm$ 0.15	43.57 $\pm$ 0.13	11.12 $\pm$ 0.19	100.00 $\pm$ 0.00
GraphLDM + CoG	60.61 $\pm$ 0.12	45.20 $\pm$ 0.24	11.35 $\pm$ 0.10	100.00 $\pm$ 0.00

Table 7: Performance on the ChEBI-20 subset where each textual prompt contains two pieces of information. All values are reported as percentages.

Method	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods				
MolT5 small	49.88 $\pm$ 0.00	21.52 $\pm$ 0.00	3.33 $\pm$ 0.00	83.71 $\pm$ 0.00
MolT5 large	54.39 $\pm$ 0.00	17.47 $\pm$ 0.00	1.38 $\pm$ 0.00	98.45 $\pm$ 0.00
ChemT5 small	56.27 $\pm$ 0.00	21.98 $\pm$ 0.03	2.44 $\pm$ 0.00	97.50 $\pm$ 0.00
ChemT5 base	56.53 $\pm$ 0.02	23.40 $\pm$ 0.04	2.65 $\pm$ 0.01	97.69 $\pm$ 0.05
BioT5 base	55.97 $\pm$ 0.13	15.74 $\pm$ 0.28	2.15 $\pm$ 0.05	99.99 $\pm$ 0.01
BioT5 plus	54.64 $\pm$ 0.07	12.61 $\pm$ 0.15	1.35 $\pm$ 0.04	100.00 $\pm$ 0.00
Graph-based methods				
3M diffusion	55.24 $\pm$ 0.04	38.88 $\pm$ 0.01	8.78 $\pm$ 0.04	100.00 $\pm$ 0.00
GraphLDM	57.01 $\pm$ 0.06	38.64 $\pm$ 0.12	9.13 $\pm$ 0.03	100.00 $\pm$ 0.00
GraphLDM + CoG	58.34 $\pm$ 0.03	42.30 $\pm$ 0.03	9.28 $\pm$ 0.04	100.00 $\pm$ 0.00

Table 8: Performance on the PubChem subset where each textual prompt contains four pieces of information. All values are reported as percentages.

Method	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods				
MolT5 small	48.59 $\pm$ 0.00	14.56 $\pm$ 0.00	1.01 $\pm$ 0.00	81.55 $\pm$ 0.00
MolT5 large	53.52 $\pm$ 0.00	13.60 $\pm$ 0.00	0.85 $\pm$ 0.00	99.03 $\pm$ 0.00
ChemT5 small	54.81 $\pm$ 0.05	17.48 $\pm$ 0.00	1.97 $\pm$ 0.03	94.17 $\pm$ 0.00
ChemT5 base	56.88 $\pm$ 0.01	20.38 $\pm$ 0.00	2.82 $\pm$ 0.01	97.48 $\pm$ 0.33
BioT5 base	56.27 $\pm$ 0.12	12.56 $\pm$ 0.45	1.93 $\pm$ 0.06	100.00 $\pm$ 0.00
BioT5 plus	54.80 $\pm$ 0.19	9.38 $\pm$ 0.22	1.23 $\pm$ 0.04	100.00 $\pm$ 0.00
Graph-based methods				
3M diffusion	60.08 $\pm$ 0.10	49.58 $\pm$ 0.12	13.18 $\pm$ 0.13	100.00 $\pm$ 0.00
GraphLDM	59.18 $\pm$ 0.40	41.75 $\pm$ 0.84	13.23 $\pm$ 0.23	100.00 $\pm$ 0.00
GraphLDM + CoG	60.89 $\pm$ 0.06	47.90 $\pm$ 0.30	14.14 $\pm$ 0.50	100.00 $\pm$ 0.00

Table 9: Performance on the PubChem subset where each textual prompt contains three pieces of information. All values are reported as percentages.

Method	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods				
MolT5 small	47.98 $\pm$ 0.00	12.10 $\pm$ 0.00	0.98 $\pm$ 0.00	80.40 $\pm$ 0.00
MolT5 large	52.31 $\pm$ 0.00	11.23 $\pm$ 0.00	0.65 $\pm$ 0.00	97.69 $\pm$ 0.00
ChemT5 small	54.91 $\pm$ 0.04	17.60 $\pm$ 0.03	1.82 $\pm$ 0.03	97.14 $\pm$ 0.03
ChemT5 base	55.87 $\pm$ 0.11	20.54 $\pm$ 0.13	2.24 $\pm$ 0.05	95.50 $\pm$ 0.20
BioT5 base	55.38 $\pm$ 0.36	13.41 $\pm$ 0.65	1.76 $\pm$ 0.17	100.00 $\pm$ 0.00
BioT5 plus	55.04 $\pm$ 0.11	11.83 $\pm$ 0.31	1.44 $\pm$ 0.05	100.00 $\pm$ 0.00
Graph-based methods				
3M diffusion	56.49 $\pm$ 0.00	41.73 $\pm$ 0.10	11.28 $\pm$ 0.07	100.00 $\pm$ 0.00
GraphLDM	59.00 $\pm$ 0.10	43.21 $\pm$ 0.24	13.66 $\pm$ 0.15	100.00 $\pm$ 0.00
GraphLDM + CoG	58.97 $\pm$ 0.10	44.71 $\pm$ 0.15	12.39 $\pm$ 0.20	100.00 $\pm$ 0.00

Table 10: Performance on the PubChem subset where each textual prompt contains two pieces of information. All values are reported as percentages.

Method	BQI	Q-Cov	Q-Nov	Validity
SMILES-based methods				
MolT5 small	41.58 $\pm$ 0.00	9.67 $\pm$ 0.00	1.08 $\pm$ 0.00	67.67 $\pm$ 0.00
MolT5 large	50.66 $\pm$ 0.00	19.33 $\pm$ 0.00	1.96 $\pm$ 0.00	91.54 $\pm$ 0.00
ChemT5 small	51.30 $\pm$ 0.08	21.73 $\pm$ 0.10	2.96 $\pm$ 0.06	88.99 $\pm$ 0.24
ChemT5 base	50.91 $\pm$ 0.13	21.02 $\pm$ 0.21	3.07 $\pm$ 0.06	87.10 $\pm$ 0.13
BioT5 base	54.42 $\pm$ 0.18	19.63 $\pm$ 0.16	3.78 $\pm$ 0.13	99.95 $\pm$ 0.05
BioT5 plus	55.11 $\pm$ 0.08	21.13 $\pm$ 0.17	4.19 $\pm$ 0.06	100.00 $\pm$ 0.00
Graph-based methods				
3M diffusion	53.92 $\pm$ 0.02	37.77 $\pm$ 0.02	10.10 $\pm$ 0.05	100.00 $\pm$ 0.00
GraphLDM	57.08 $\pm$ 0.13	39.83 $\pm$ 0.37	11.93 $\pm$ 0.03	100.00 $\pm$ 0.00
GraphLDM + CoG	57.16 $\pm$ 0.17	41.56 $\pm$ 0.28	11.22 $\pm$ 0.13	100.00 $\pm$ 0.00

	ChemT5-base augment	MolT5-large	GraphLDM
seed 1	<chem>C1=CC=C2C=C1</chem> 	 	 
seed 2	<chem>C1=CC=C2C=C1</chem> 	 	 
seed 3	<chem>C1=CC=C2C=C1</chem> 	 	 

Figure 7: Sequence-to-sequence methods struggle to produce valid and diverse outputs when given prompts such as “The molecule is a simple structure that contains a benzene ring.” In particular, ChemT5 and MolT5 exhibit severe mode collapse: despite the large solution space implied by flexible prompts, they repeatedly converge to a single rigid output. A second limitation lies in chemical invalidity, as exemplified by the ChemT5-base augment model.