BiMol-Diff: A Unified Diffusion Framework for Molecular Generation and Captioning

Anonymous ACL submission

Abstract

Bridging molecular structures and natural language is essential for controllable design. Autoregressive models struggle with long-range dependencies, while standard diffusion processes apply uniform corruption across positions, which can distort structurally informative tokens. We present BiMol-Diff, a unified diffusion framework for the paired tasks of text-conditioned molecule generation and molecule captioning. Our key component is a *token-aware noise schedule* that assigns position-dependent corruption based on token recovery difficulty, preserving harder-to-recover substructures during the forward process. On ChEBI-20 and M3-20M, `BiMol-Diff` improves molecule reconstruction with a 15.4% relative gain in Exact Match and achieves strong captioning results, attaining best BLEU and BERTScore among compared baselines. These results indicate token-aware noising improves fidelity in molecular structure–language modeling.

1 Introduction

Designing molecules requires balancing two distinct modalities: the molecular *structure* of atoms and bonds, and the linguistic descriptions of molecular properties (e.g., "a kinase inhibitor with good solubility"). Bridging this gap through cross-modal translation enables chemically grounded workflows where designs can be iteratively refined via natural language (Zheng et al., 2023; Fatemi et al., 2024). This motivates two complementary directions: (i) **molecule generation** (*text→molecule*) to materialize descriptions into structures, and (ii) **molecule captioning** (*molecule→text*) to explain molecular structure and properties via natural-language descriptions (Edwards et al., 2022a; Zeng et al., 2023). However, despite their symmetry, these tasks are typically developed in isolation with separate models and objectives, limiting consistency and reuse in iterative design loops.

A common bridge between molecular structure and sequence models is the Simplified Molecular-Input Line-Entry System (SMILES) (Weininger, 1988; Weininger et al., 1989), which linearizes a molecular graph into a string. This linearization has enabled a large family of sequence-based molecular methods, including (1) conditional SMILES generation from text prompts (Bagal et al., 2021; Frey et al., 2022; Edwards et al., 2022b; Irwin et al., 2022), and (2) SMILES or structure-conditioned captioning that maps molecules to natural-language descriptions (Edwards et al., 2022a; Zeng et al., 2023). Many of these systems rely on *autoregressive* (AR) pretrained language models (PLMs) such as GPT (Floridi and Chiriatti, 2020), T5 (Raffel et al., 2020a), and BART (Lewis et al., 2020), which generate tokens sequentially by conditioning each next token on the prefix (Bagal et al., 2021; Frey et al., 2022; Edwards et al., 2022b; Irwin et al., 2022). In this work, while we retain SMILES as the standard input/output interface, we explicitly project these molecules into a Knowledge Graph (KG) view represented as a serialized sequence of atom-bond-atom triplets to achieve better generalization and reduce dependency on specific SMILES syntax rules (e.g., ring number).

While effective, autoregressive models are not always well suited to SMILES-centric molecule–language modeling. First, decoding is inherently left-to-right, so early mistakes can propagate and are difficult to correct (Nie et al., 2025; Arriola et al., 2025). Second, SMILES encodes long-range syntactic and chemical dependencies (e.g., ring closures, branching, and valid substructures) that require coordinated decisions across distant positions; locally plausible token choices can therefore cascade into globally invalid molecules. Third, many objectives are inherently structure-level such as scaffold preservation that are hard to enforce under strict prefix conditioning. These limitations motivate non-autoregressive generation mech-

anisms that can revise all positions jointly, raising a broader question: *can we build models that generate and explain molecules while explicitly preserving global validity and structural controllability?* Diffusion language models offer a promising non-autoregressive alternative by iteratively denoising a corrupted representation, enabling holistic updates over all positions at each step and better coordination of long-range dependencies (Li et al., 2022; Nie et al., 2025). However, standard diffusion applies noise uniformly across tokens, which can corrupt chemically critical tokens (e.g., ring indices and branching markers) too early in the process and hinder recoverability.

To address this, we propose `BiMol-Diff`, a unified diffusion framework that addresses the paired tasks of *text→SMILES* generation and *SMILES→text* captioning under a common denoising formulation. Unlike standard diffusion, `BiMol-Diff` employs a token-aware noising strategy that assigns token-dependent noise levels using the per-token training loss as a proxy for recovery difficulty. Intuitively, tokens that are consistently harder to denoise receive a more conservative corruption schedule, improving recoverability. We apply this mechanism consistently across both directions: for *SMILES→text*, it helps preserve semantic tokens during denoising, and for *text→SMILES*, it helps preserve chemically salient SMILES tokens during generation.

Contributions. We make three primary contributions: (1) a token-aware noising strategy which preserves critical structural semantics during the diffusion process; (2) a unified framework that solves the bidirectional tasks of molecule generation and captioning, utilizing the same token-aware mechanism for both inverse problems; and (3) strong empirical performance across a wide range of metrics on both molecule generation and molecule captioning benchmarks.

2 Background and Preliminaries

2.1 Related Work

Diffusion for Molecular Graphs (Unconditional / Property-based): Diffusion models are widely used for molecular graph generation, spanning unconditional sampling and property-conditioned design. Recent variants improve expressivity via transformer-style graph denoisers (Liu et al., 2024a), explore hierarchical/latent formulations for scalability and structure preservation (Bian et al., 2024), and study control mechanisms for property prediction and guidance (Zhang et al., 2025).

Text-guided Molecule Generation: Text-to-molecule generation is typically formulated as learning $p(\tilde{\mathcal{G}} \mid \mathbf{S})$ (or $p(\text{SMILES} \mid \mathbf{S})$), where $\mathbf{S}$ describes desired structure or function. Recent work adapts PLMs to chemical strings, enabling text↔molecule translation with encoder–decoder pretraining (Edwards et al., 2022a) and improved multi-task / chemistry-aware extensions (Kim et al., 2025b). In parallel, diffusion-based text conditioning has emerged, including diffusion-LM style text-guided generation in discrete/embedded sequence spaces (Gong et al., 2024) and graph/latent diffusion variants that preserve molecular structure (Chang and Ye, 2025).

Molecule Captioning / Molecule-to-Text: Generating faithful descriptions from molecules (G2S) has been explored via translation-style baselines (Edwards et al., 2022a) and molecule captioning models that incorporate structural encoders to improve grounding (Liu et al., 2024d). More recent multimodal approaches further strengthen molecule–language alignment for captioning and related tasks.

Unifying / Bidirectional Molecule–Text Modeling: While text-guided generation and captioning have progressed independently, *fully bidirectional* modeling that supports both $p(\mathbf{S} \mid \tilde{\mathcal{G}})$ and $p(\tilde{\mathcal{G}} \mid \mathbf{S})$ within a single generation framework remains relatively limited. Translation-based PLM systems offer practical bidirectionality (Edwards et al., 2022a), and recent foundation-model efforts move toward broader cross-modal generalization (Liu et al., 2024d), but diffusion-based approaches are typically developed for one direction (most often S2G) and do not explicitly address token-level corruption heterogeneity. `BiMol-Diff` fills this gap by casting *both* directions under a unified conditional diffusion view and introducing a molecular graph-aware, token-dependent noising schedule to preserve chemically salient information during generation (see Appendix A.1).

2.2 Preliminaries

2.2.1 Diffusion Models: Forward and Reverse Process

Denoising diffusion probabilistic models (DDPMs) are generative models that learn a data distribution, often conditioned on some context $\mathbf{c}$, $p(\mathbf{z}_0 \mid \mathbf{c})$. They consist of a fixed forward process and a

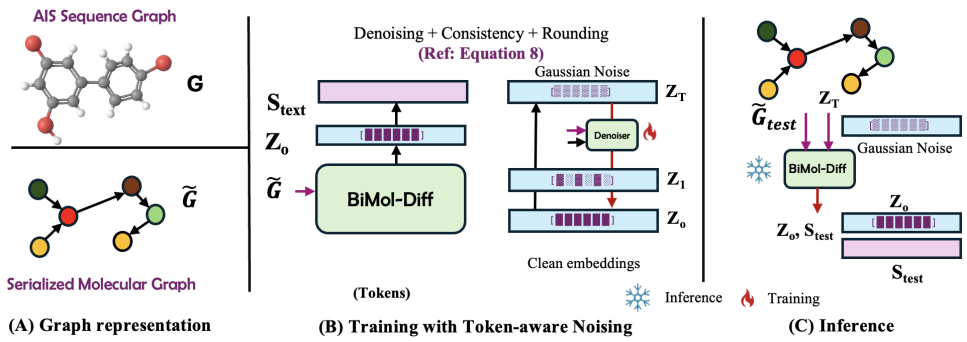

Figure 1: **The BiMol-Diff Framework (G2S).** (A) Molecules are serialized; (B) **Training:** The model is trained with token-aware noising that preserves chemically salient tokens, optimizing for denoising, consistency, and rounding. (C) **Inference:** starting from Gaussian noise, iterative denoising and rounding generate either a caption (G2S) or a molecule sequence (S2G), depending on the conditioning input.

learned reverse process.

Forward process: A standard DDPM forward process corrupts clean data $\mathbf{z}_0$ through a Markov chain with noise-schedule coefficients $\{\alpha_t\}_{t=1}^{T}$ controlling signal decay. This yields the standard closed-form for sampling a noised state $\mathbf{z}_t$ at timestep t:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \tag{1}$$

with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Standard diffusion models typically use a fixed, data-agnostic (isotropic) noise schedule.

Reverse process with Conditional Denoising: The reverse process learns to recover the clean data $\mathbf{z}_0$ from pure noise $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. It is defined as a Markov chain $p_\theta(\mathbf{z}_{0:T})$ where each reverse transition $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{c})$ is a Gaussian whose mean $\boldsymbol{\mu}_\theta$ and variance $\boldsymbol{\Sigma}_\theta$ are parameterized by a model $\mathcal{M}_\theta(\mathbf{z}_t, t, \mathbf{c})$. The model is trained to predict the mean of the true posterior $q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{z}_0)$. The model parameters θ are optimized by maximizing the variational lower bound (VLB) on the conditional log-likelihood:

$$\begin{aligned}\mathcal{L}_{\text{vlb}} = \mathbb{E}_q\Big[& \underbrace{-\log p_\theta(\mathbf{z}_0 \mid \mathbf{z}_1, \mathbf{c})}_{\text{Reconstruction } (L_0)} \\ & + \sum_{t=2}^{T} \underbrace{D_{\text{KL}}\big(q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{z}_0) \,\big\|\, p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{c})\big)}_{\text{Denoising Matching } (L_{t-1})} \\ & + \underbrace{D_{\text{KL}}\big(q(\mathbf{z}_T \mid \mathbf{z}_0) \,\big\|\, p(\mathbf{z}_T)\big)}_{\text{Prior Matching } (L_T)} \Big].\end{aligned} \tag{2}$$

While this formulation is tractable, direct optimization of the full VLB is often unstable.

2.2.2 Molecules as Graphs:

A molecule is represented as a graph $\mathcal{G} = (\mathbf{V}, \mathbf{X}, \mathbf{A}, \mathbf{P})$, where $\mathbf{V} = \{1, \ldots, n\}$ is the set of atoms (nodes) and $|\mathbf{V}| = n$. The atom feature matrix $\mathbf{X} \in \mathbb{R}^{n \times a}$ contains an a-dimensional feature vector for each atom. The adjacency tensor $\mathbf{A} \in \mathbb{R}^{n \times n \times b}$ encodes bond existence and bond types, where $\mathbf{A}_{ij:} \in \{0,1\}^b$ indicates the type of the bond between atoms i and j. The coordinate matrix $\mathbf{P} \in \mathbb{R}^{n \times 3}$ stores the 3D positions of atoms. Here, a is the atom feature dimension and b is the bond type. Depending on the task, subsets of these components may be used, e.g., $(\mathbf{V}, \mathbf{X}, \mathbf{A})$ for 2D graphs or $(\mathbf{V}, \mathbf{X}, \mathbf{P})$ for 3D conformations.

3 Methodology

3.1 Problem Statement

We represent each molecule as an Atoms-in-SMILES (AIS) (Ucak et al., 2023) token sequence $\mathcal{G} = \{m_1, \ldots, m_K\}$, which provides a chemistry-aware, graph-informed alternative to character-level SMILES. For molecule captioning (G2S), we describe the structure by serializing the molecular graph into an edge-list sequence (denoted by $\tilde{\mathcal{G}}$), using special tokens `[HEAD]`, `[REL]`, `[TAIL]`, `[SEP]`:

$$\tilde{\mathcal{G}} = \big\langle \texttt{[HEAD]}\ h_i\ \texttt{[REL]}\ r_{ij}\ \texttt{[TAIL]}\ t_j\ \texttt{[SEP]} \big\rangle_{(i,j)}$$

where (h_i, r_{ij}, t_j) denotes an atom-bond-atom triple. Let $\mathbf{S}_{\text{text}} = \{s_1, \ldots, s_N\}$ be a textual description. We study two conditional generation tasks: (i) **captioning** $\tilde{\mathcal{G}} \to \mathbf{S}_{\text{text}}$, and (ii) **molecule generation** $\mathbf{S}_{\text{text}} \to \tilde{\mathcal{G}}$. Accordingly, we learn $\mathcal{M}_\theta : \tilde{\mathcal{G}} \to \mathbf{S}_{\text{text}}$ and $\mathcal{M}_\phi : \mathbf{S}_{\text{text}} \to \tilde{\mathcal{G}}$ and maximize

$$\max_{\Theta}\ \log p(\mathbf{S}_{\text{text}} \mid \tilde{\mathcal{G}}; \theta) + \log p(\tilde{\mathcal{G}} \mid \mathbf{S}_{\text{text}}; \phi),$$

where $\Theta = \{\theta, \phi\}$. For evaluation, the gener-

ated $\tilde{\mathcal{G}}$ is deterministically mapped to a canonical SMILES string.

3.2 The BiMol-Diff Framework Overview

We introduce BiMol-Diff, a unified diffusion framework that addresses molecular *graph-to-sequence* (G2S) and *sequence-to-graph* (S2G) generation under a single conditional modeling view (Figure 1). Given paired data $(\mathcal{A}, \mathcal{B}) \in \{(\mathbf{S_{text}}, \tilde{\mathcal{G}}), (\tilde{\mathcal{G}}, \mathbf{S_{text}})\}$, BiMol-Diff learns the conditional distribution $p(\mathcal{A} \mid \mathcal{B})$, where $\mathcal{B}$ provides the conditioning context and $\mathcal{A}$ is the target modality.

Graph Encoding. We represent a molecule as triplets $\tilde{\mathcal{G}} = \{(h_i, r_{ij}, t_j)\}$ and serialize them into a single token sequence (with [HEAD]/[REL]/[TAIL] templates and [SEP] delimiters) to condition a transformer backbone in the G2S direction (Section 3.3); conversely, $\mathbf{S_{text}}$ conditions reconstruction of $\tilde{\mathcal{G}}$ in S2G.

Conditional Diffusion. For either direction, we embed the target $\mathcal{A}$ into continuous "clean" latents $\mathbf{z}_0 = g_\Phi(\mathcal{A})$ and apply a conditional DDPM to obtain $\mathbf{z}_t$. The reverse processes denoise conditioned on the source modality: $\mathcal{M}_\theta(\mathbf{z}_t, t, \tilde{\mathcal{G}})$ for G2S and $\mathcal{M}_\phi(\mathbf{z}_t, t, \mathbf{S_{text}})$ for S2G, enabling a unified objective across both directions.

Token-aware Noising. Unlike standard data-agnostic schedules that corrupt all tokens uniformly, BiMol-Diff uses a token-aware, token-dependent noising strategy that preserves chemically salient tokens more conservatively. We detail the construction and use of this schedule in Sec 3.4.

3.3 Graph Encoding for BiMol-Diff

In Section 3.1, a molecular graph is a set of relational triplets $\tilde{\mathcal{G}} = \{(h_i, r_{ij}, t_j)\}$. To integrate with transformer based encoder-decoder backbone, we serialize this set of triplets into a single token sequence. Following the PLM-based G2S convention, each triplet is mapped to a short template $\langle$[HEAD] h_i [REL] r_{ij} [TAIL] $t_j\rangle$, and all such segments are concatenated with a separator token [SEP]:

$$\tilde{\mathcal{G}} = \langle \texttt{[HEAD]}\ h_{i_1}\ \texttt{[REL]}\ r_{i_1 j_1}\ \texttt{[TAIL]}\ t_{j_1}\rangle\ \texttt{[SEP]} \cdots \texttt{[SEP]} \langle \texttt{[HEAD]}\ h_{i_M}\ \texttt{[REL]}\ r_{i_M j_M}\ \texttt{[TAIL]}\ t_{j_M}\rangle \tag{3}$$

where $\{(i_m, j_m)\}_{m=1}^{M}$ enumerates all bonds with $i_m < j_m$ to avoid duplicates. This linearized sequence $\tilde{\mathcal{G}}$ serves as the conditioning signal to BiMol-Diff in the graph-to-sequence task.

Example (molecule KG → caption). Consider the molecule with SMILES string CCO (ethanol). Its molecular graph has three atoms $h_1 = \mathtt{C}_1$, $h_2 = \mathtt{C}_2$, $h_3 = \mathtt{O}_3$ and two single bonds: $(1, 2)$ and $(2, 3)$. The corresponding KG triplets are $(\mathtt{C}_1,\ \mathtt{SINGLE},\ \mathtt{C}_2)$ and $(\mathtt{C}_2,\ \mathtt{SINGLE},\ \mathtt{O}_3)$. Our serialized KG becomes

$$\tilde{\mathcal{G}} = \langle \texttt{[HEAD]}\ \mathtt{C}_1\ \texttt{[REL]}\ \mathtt{SINGLE}\ \texttt{[TAIL]}\ \mathtt{C}_2\rangle\ \texttt{[SEP]}\ \langle \texttt{[HEAD]}\ \mathtt{C}_2\ \texttt{[REL]}\ \mathtt{SINGLE}\ \texttt{[TAIL]}\ \mathtt{O}_3\rangle \tag{4}$$

In the G2S setting, BiMol-Diff takes $\tilde{\mathcal{G}}$ as input and generates a natural-language caption $\mathbf{S_{text}}$ describing this molecule. For the S2G task, BiMol-Diff learns to reconstruct the underlying canonical SMILES string representation of molecular graph given a textual description $\mathbf{S_{text}}$.

3.4 Token Aware Noising

Motivation & Rationale: Standard diffusion models rely on fixed, data-agnostic noising schedules that apply noise uniformly across all tokens.

We argue this is suboptimal for molecular tasks because chemically salient tokens (e.g., functional groups, stereochemistry) typically carry higher information density than structurally redundant or low-information tokens.

Under uniform noising, chemically salient tokens and simple syntactic tokens are corrupted at the same rate, which can weaken molecular fidelity. To address this, we introduce a *Token Aware* noising schedule. We hypothesize that the model's per-timestep reconstruction loss ℓ_t^i can serve as a useful proxy for token difficulty and importance. By learning a mapping from ℓ_t^i to the corresponding token's noise schedule, we preserve high information tokens while allowing information-redundant tokens to tolerate higher noise levels. This re-parametrizes the denoising path, aiming to improve generation quality in both G2S and S2G tasks.

Noising Schedule: Following the rationale discussed in Section 3.4, for the G2S setting, we propose a token-dependent noising schedule, parameterized by a vector $\bar{\alpha}_{t,\text{new}}^i \in \mathbb{R}^N$, unlike the uniform noising with baseline cumulative schedule $\bar{\alpha}_t$ (for e.g.: *sqrt*) used by standard diffusion models. This formulation has two stages summarized in Algo 1.

Stage 1: Estimating token-wise difficulty. For each token i and diffusion step $t = 1, \ldots, T$, we define the denoising difficulty as:

$$\ell_t^i = \mathbb{E}_{\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{z}_0)} \left\| \mathcal{M}_\theta(\mathbf{z}_t, t, \tilde{\mathcal{G}})^{(i)} - \mathbf{z}_0^{(i)} \right\|^2 \tag{5}$$

Averaging over the training set yields a difficulty profile $(\ell_1^i, \ldots, \ell_T^i)$ for each i. Empirically, ℓ_t^i tends to increase with t (later steps are noisier), but the estimated profile is not strictly monotone. We also compute $\ell_{\min}^i = \min_t \ell_t^i$ and $\ell_{\max}^i = \max_t \ell_t^i$ to define the difficulty range for token i. In Stage 2 these profiles and their ranges are used to construct a token-wise cumulative schedule, and to obtain a monotonic difficulty profile for each token i.

Stage 2: Token-aware schedule. Given $(\ell_t^i)_{t=1}^T$ and the baseline cumulative schedule $(\bar{\alpha}_t)_{t=1}^T$, we construct an adaptive schedule $(\bar{\alpha}_{t,\text{new}}^i)_{t=1}^T$ for each token i. Since this schedule controls noise applied at each step, we want to reallocate noise according to denoising difficulty. Hence, we define a piecewise-linear map $\Psi_i : [\ell_{\min}^i, \ell_{\max}^i] \to (0, 1)$ that interpolates the baseline schedule as a function of loss:

$$\Psi_i(x) = \bar{\alpha}_{t-1} + \frac{\bar{\alpha}_t - \bar{\alpha}_{t-1}}{\ell_t^i - \ell_{t-1}^i}\left(x - \ell_{t-1}^i\right), \quad (6)$$

with $x \in [\ell_{t-1}^i, \ell_t^i)$, $t = 2, \ldots, T$, $\Psi_i(\ell_1^i) = \bar{\alpha}_1$ and $\Psi_i(\ell_T^i) = \bar{\alpha}_T$. In case $\ell_t^i = \ell_{t-1}^i$, we add a tiny jitter ε to avoid division by zero. Empirically, $(\ell_t^i)_{t=1}^T$ is not strictly monotone in t, so instead of using its raw values we introduce a new linear ramp in difficulty space:

$$\ell_t^{i,\text{new}} = \ell_{\min}^i + \frac{t-1}{T-1}\left(\ell_{\max}^i - \ell_{\min}^i\right), \quad (7)$$

with $t = 1, \ldots, T$. Substituting this $\ell_t^{i,\text{new}}$ into $\Psi_i(x)$ we get a new cumulative schedule $\bar{\alpha}_{t,\text{new}}^i = \Psi_i(\ell_t^{i,\text{new}})$ for $t = 1, \ldots, T$. We clamp $\bar{\alpha}_{t,\text{new}}^i$ to $(0, 1)$ and apply a non-increasing isotonic projection (refer Appendix A.3) over t to obtain the final schedule $0 < \bar{\alpha}_{t+1,\text{new}}^i \leq \bar{\alpha}_{t,\text{new}}^i < 1$ for all t.

3.5 Model Training and Inference

Training: Our training objective is derived from the Variational Lower Bound (VLB) (Eq. 2) presented in the preliminaries. While the full VLB optimization can be unstable (Ho et al., 2020), a common simplification is to train the model $\mathcal{M}_\theta$ to predict the added noise $\boldsymbol{\epsilon}$. However, our framework adopts an alternative $\mathbf{z}_0$-prediction reparameterization, which trains the model to directly predict the clean data $\mathbf{z}_0$ at every timestep t. A critical component of this objective is the rounding term $L_0 = -\log \tilde{p}_\Phi(\mathbf{S}_{\mathbf{text}} \mid \mathbf{z}_0)$, which handles the final step of converting the continuous latent

Algorithm 1 Token Aware Noise Schedule

Require: Baseline cumulative schedule $\{\bar{\alpha}_t\}_{t=1}^T$, update interval K
Ensure: Schedules $\{\bar{\alpha}_{t,\text{new}}^i\}_{t=1}^T$ for all i
if `train_step` % K == 0 **then**
  for all i **do**
    Estimate $\{\ell_t^i\}_{t=1}^T$ via Eq. 5; compute $\{\ell_{\min}^i, \ell_{\max}^i\}$.
    Define piecewise-linear map Ψ_i as in Eq. (6).
    Construct difficulty ramp $\{\ell_t^{i,\text{new}}\}_{t=1}^T$ via Eq. (7).
    Compute new schedule $\tilde{\alpha}_t^i = \Psi_i(\ell_t^{i,\text{new}})$.
    Clamp $\tilde{\alpha}_t^i$ to $(0, 1)$, apply a non-increasing isotonic projection to obtain $\{\bar{\alpha}_{t,\text{new}}^i\}_{t=1}^T$.
  end for
end if
return $\{\bar{\alpha}_{t,\text{new}}^i\}$ for all tokens i and steps t.

variable $\mathbf{z}_0$ back into discrete tokens $\mathbf{S}_{\mathbf{text}}$. We define this as a trainable rounding distribution: $\tilde{p}_\Phi(\mathbf{S}_{\mathbf{text}} \mid \mathbf{z}_0) = \prod_{i=1}^N \tilde{p}_\Phi(s_i \mid \mathbf{z}_{0,i})$, where each token s_i is sampled from a softmax distribution over the vocabulary, using logits derived from the corresponding output embedding $\mathbf{z}_{0,i}$. By combining this rounding term (for $t = 0$) with the denoising matching terms (for $t > 1$) using our $\mathbf{z}_0$-prediction reparameterization, we arrive at our final, composite objective. (The full derivation from the VLB is in App A.2).

$$\mathcal{L}_{\text{e2e-simple}}(\mathbf{S}_{\mathbf{text}}) = \mathbb{E}_q\Big[\sum_{t=2}^{T} \underbrace{\|\mathcal{M}_\theta(\mathbf{z}_t, t, \tilde{\mathcal{G}}) - \mathbf{z}_0\|^2}_{\text{Denoising}} + \underbrace{\|\, g_\Phi(\mathbf{S}_{\mathbf{text}}) - \mathcal{M}_\theta(\mathbf{z}_1, 1, \tilde{\mathcal{G}})\|^2}_{\text{Consistency}} - \underbrace{\log \tilde{p}_\Phi(\mathbf{S}_{\mathbf{text}} \mid \mathbf{z}_0)}_{\text{Rounding}}\Big] \quad (8)$$

This objective directly optimizes the most critical parts of the process: the denoising accuracy across all steps (Denoising), the consistency of the first denoising step with the true data embedding (Consistency), and the quality of the final conversion to discrete tokens (Rounding).

Inference-time schedule: The $\mathbf{z}_0$-prediction objective (Eq: 8) trains the denoiser to explicitly predict the clean latent at each step. At inference, given $\mathbf{z}_t$ and conditioning context $\mathbf{c}$, we compute $\hat{\mathbf{z}}_0 = \mathcal{M}_\theta(\mathbf{z}_t, t, \mathbf{c})$ (with $\mathbf{c} = \tilde{\mathcal{G}}$ for G2S and $\mathbf{c} = \mathbf{S}_{\mathbf{text}}$ for S2G). We then use our learned

Method	#P	B	CrF++	M	B-F1	MVE
Autoregressive Baselines						
MolT5-Base (Edwards et al., 2022a)	220M	0.452	0.651	0.510	0.681	0.852
Text+Chem T5 (Christofidellis et al., 2023)	223M	0.542	0.701	0.648	0.728	0.866
MolCA (Liu et al., 2024d)	110M	0.531	0.665	**0.651**	0.709	0.815
GitMol (Liu et al., 2024c)	700M	0.475	0.680	0.532	0.751	0.875
GraphT5 (Kim et al., 2025a)	272M	0.481	0.692	0.545	0.810	0.913
Diffusion Baselines						
Diffusion-LM (Li et al., 2022)	91M	0.512	0.702	0.602	0.783	0.861
DiffuSeq (Gong et al., 2023)	91M	0.532	0.708	0.601	0.812	0.887
TGM-DLM (Gong et al., 2024)	125M	0.467	0.689	0.589	0.779	0.856
`BiMol-Diff` (ours)	63M	**0.567**	**0.734**	0.626	**0.843**	**0.925**
%Gain (vs. Best AR)	**x3.5↓**	**+4.6%**	**+4.7%**	-3.8%	**+4.1%**	**+1.3%**
%Gain (vs. Best Diff)	**x1.4↓**	**+6.6%**	**+3.7%**	**+4.2%**	**+3.8%**	**+4.3%**

Table 1: Molecule captioning performance on the M3-20M dataset. Benchmarking `BiMol-Diff` against SoTA Autoregressive (AR) and Diffusion models. `BiMol-Diff` achieves state-of-the-art results, surpassing the best diffusion and AR baselines by substantial margins across all metrics while using fewer parameters.

token-wise cumulative schedule $\bar{\alpha}^{i}_{t,\text{new}}$ in the reverse sampling update, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. To mitigate rounding errors (refer section 3.5), we apply the clamping trick, which forces the predicted vector to commit to a word for intermediate diffusion steps:

$$\mathbf{z}^{(i)}_{t-1} = \sqrt{\bar{\alpha}^{i}_{t-1,\text{new}}}\,\text{Clamp}\Big(\mathcal{M}_\theta(\mathbf{z}_t, t, \mathbf{c})^{(i)}\Big) + \sqrt{1-\bar{\alpha}^{i}_{t-1,\text{new}}}\,\boldsymbol{\epsilon}^{(i)}, \tag{9}$$

where $\text{Clamp}(\cdot)$ denotes nearest-neighbor projection onto the embedding table. This strategy ensures that the diffusion trajectory remains grounded in valid token embeddings during sampling.

4 Experiments

4.1 Experimental Setup

Model Architecture. We implement `BiMol-Diff` using two distinct encoder-decoder Transformer models tailored for each task utilizing GeLU activations. Crucially, both tasks are trained for 200,000 steps using our proposed *token-aware noise schedule* (refer §3.4) over $T = 2000$ diffusion steps. We adapt the architecture for each task as follows: for Molecule Generation (S2G), we employ a 6-encoder/12-decoder configuration ($d_{model} = 1024$; $\approx$ 180M parameters). The encoder processes frozen SciBERT embeddings, enabling the decoder to generate SMILES sequence as output. This variant is trained with a batch size of 64 and a learning rate of 5×10^{-5}. Conversely, for Molecule Captioning (G2S), we utilize a 6-encoder/9-decoder configuration ($d_{model} = 512$; $\approx$ 63M parameters) with a peak learning rate of 10^{-4}. This model encodes serialized molecular structure using the AIS-based SMILES vocabulary augmented with learnable special tokens (`[HEAD]`, `[REL]`, `[TAIL]`, `[SEP]`), while the decoder generates captions utilizing the `bert-base-uncased` vocabulary (Devlin et al., 2019).

Datasets. We evaluate our framework on two distinct datasets tailored to each task. For Molecule Generation (S2G), we utilize the ChEBI-20 (Edwards et al., 2022a) benchmark, comprising 33,010 molecules with SMILES strings restricted to a maximum length of 256. This dataset is partitioned into an 80%/10%/10% split for training, validation, and testing, respectively. For Molecule Captioning (G2S), we employ a filtered subset of the M3-20M dataset (Guo et al., 2025b), containing approximately 360,000 SMILES–description pairs. This subset, also divided into an 80%/10%/10% split, pairs molecular graphs with diverse textual descriptions derived from curated scientific literature or synthesized via GPT-3.5.

Baselines. We compare `BiMol-Diff` against state-of-the-art (SoTA) autoregressive and diffusion-based methods for each task. For Molecule Generation (S2G), we benchmark against MolXPT (Liu et al., 2023) as a competitive autoregressive SMILES generator, and leading diffusion baselines including 3M-Diffusion (Zhu et al., 2024) and UT-GDiff (Xiang et al., 2025). For Molecule Captioning (G2S), we compare against MolT5-Base (Edwards et al., 2022a) as a standard SMILES→text sequence-to-sequence baseline, GraphT5 (Kim et al., 2025a) as a recent graph–text model, and DiffuSeq (Gong et al., 2023) as a representative diffusion-based sequence generator.

Evaluation Metrics. For Molecule Captioning

Method	Type	#P	Exact	MACCS	RDK	Morgan	Valid
Text-Based Autoregressive (SMILES)							
T5-Base (Raffel et al., 2020b)	AR	248M	0.069	0.731	0.605	0.545	0.660
MolT5-Base (Edwards et al., 2022a)	AR	248M	0.081	0.721	0.588	0.529	0.772
MolXPT (Liu et al., 2023)	AR	350M	0.215	**0.859**	**0.757**	**0.667**	**0.983**
T5 Enc + MolXPT MLP (Deng et al., 2025)	Adapter	111M	**0.227**	0.820	0.713	0.636	0.980
Text-Based Autoregressive (SELFIES)							
SciBERT + MolGen (Deng et al., 2025)	Adapter	317M	0.083	0.680	0.526	0.422	0.995
Galactica-1.3B + MolGen (Fang et al., 2024)	Adapter	1.5B	0.091	0.706	0.560	0.454	0.995
T5 Enc + MolGen (Deng et al., 2025)	Adapter	317M	0.165	0.758	0.616	0.527	0.995
Graph & Diffusion Baselines							
DiGress (sim. guidance) (Vignac et al., 2023)	Diff	289M	0.014	0.577	0.389	0.288	0.854
3M-Diffusion (Zhu et al., 2024)	Diff	162M	0.005	0.548	0.370	0.273	**1.000**
Graph-DiT (Liu et al., 2024b)	Diff	162M	0.000	0.374	0.269	0.159	0.909
UTGDiff (w/o pretrain) (Xiang et al., 2025)	Diff	125M	0.227	0.867	0.763	0.695	0.856
`BiMol-Diff` (ours)	Diff	180M	**0.262**	**0.894**	**0.791**	**0.762**	0.901
%Gain (vs. Best AR)	–	**x1.9↓**	**+15.4%**	**+4.1%**	**+4.5%**	**+14.2%**	-8.3%
%Gain (vs. Best Diff)	–	**x1↓**	**+15.4%**	**+3.1%**	**+3.7%**	**+9.6%**	**+5.3%**

Table 2: Molecule generation performance on the ChEBI-20 test set. `BiMol-Diff` outperforms state-of-the-art Autoregressive and Diffusion baselines in structure reconstruction (Exact Match) and fingerprint similarity, validating the effectiveness of token-aware denoising.

(G2S), we evaluate the quality of generated text using both surface-level and semantic measures. We report standard n-gram overlap metrics including BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and ChrF++ (Popović, 2015), alongside embedding-based metrics such as BERTScore-F1 (Zhang et al., 2020) and MAUVE (Pillutla et al., 2023) to assess semantic alignment and distributional divergence. For Molecule Generation (S2G), we focus on structural fidelity and chemical plausibility. We report Exact Match accuracy to measure precise reconstruction of the ground-truth molecule. To assess the capture of chemical substructures, we compute structural overlap using Tanimoto similarity on MACCS, RDKit, and Morgan fingerprints. Finally, we report chemical validity to ensure the generated SMILES strings correspond to valid molecular graphs.

4.2 Experimental Results

Molecule Captioning (G2S): Table 1 reports results on M3-20M. `BiMol-Diff` achieves the best overall performance on BLEU and ChrF++, reaching **0.567** BLEU and **0.734** ChrF++, which improves over the best performing autoregressive baseline (Text+Chem T5) by **4.6%** and **4.7%**, respectively. On semantic metrics, our model attains **0.843** BERTScore-F1 and **0.925** MAUVE, outperforming graph-aware baselines such as GraphT5, and consistently improving over diffusion models (e.g., DiffuSeq) across all reported metrics. While MolCA achieves the highest METEOR score, `BiMol-Diff` remains competitive (**0.626**) and offers a strong overall trade-off across surface and embedding-based measures, supporting the benefit of token-aware noising.

Molecule Generation (S2G): Table 2 summarizes results on ChEBI-20. `BiMol-Diff` attains an Exact Match of **0.262**, improving over both the best autoregressive adapter baseline (T5 Enc + MolXPT MLP) and the strongest diffusion baseline (UTGDiff), each at 0.227 (**+15.4%** relative). This gain is reflected in fingerprint similarities, achieving **0.894** (MACCS), **0.791** (RDKit), and **0.762** (Morgan), including a **14.2%** improvement in Morgan similarity over the best autoregressive baseline. Finally, `BiMol-Diff` maintains high chemical validity (**0.901**) while prioritizing reconstruction accuracy, whereas some baselines attain higher validity with weaker structural recovery.

4.3 Ablation

We analyze three aspects of `BiMol-Diff`: (i) the proposed token-aware noising strategy and its mapping function, (ii) the effect of molecular tokenization, and (iii) decoding efficiency of `BiMol-Diff`.

(i) Impact of noise schedule and choice of mapping function. Table 3 (top) shows that replacing a uniform corruption schedule (*sqrt*) with our token-aware schedule yields a substantial improvement (0.495→0.567 BLEU; 0.682→0.734 ChrF++; 0.531→0.626 METEOR). Among mapping choices, the linear mapping performs best, indicating that explicitly allocating lower corruption to harder-to-recover tokens is beneficial for caption recovery. Figure 2 visualizes how token-aware schedules differ from a global *sqrt* schedule. Instead of applying the same noise profile to all po-

Noise Schedule	Mapping / Tokenizer	B	CrF++	M
Impact of Noise Schedule & Mapping Function				
Uniform	Sqrt Schedule	0.495	0.682	0.531
Token-Aware	Cosine Mapping	0.548	0.720	0.595
Token-Aware (Ours)	Linear Mapping	**0.567**	**0.734**	**0.626**
Impact of Molecular Tokenization				
Token-Aware (Linear)	Regex-based	0.558	0.714	0.606
Token-Aware (Linear)	Atom-level	0.563	0.722	0.613
Token-Aware (Linear)	Atoms-in-SMILES (AIS)	**0.567**	**0.734**	**0.626**

Table 3: Ablation study on the M3-20M dataset. We analyze the impact of different (**Top**) Noise Schedules & Mapping Functions and (**Bottom**) Tokenization strategies on molecule captioning performance. `BiMol-Diff` (Token-Aware + Linear) with Atoms-in-SMILES (AIS) tokenization yields the best results.

sitions, we learn per-token schedules that modulate the corruption level according to token difficulty: positions with higher denoising loss receive more conservative corruption (lower noise at intermediate timesteps), while easier positions can be noised more aggressively. This selective corruption better preserves content during the forward process and improves recoverability in the reverse process, which aligns with the gains observed in the main results (Table 1, 2).

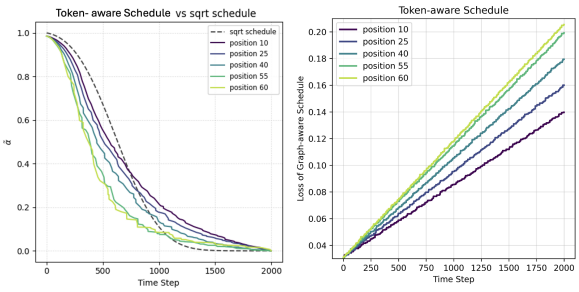

Figure 2: Token-aware noising vs. uniform sqrt schedule on captioning. *Left:* the learned token-wise schedules apply non-uniform corruption across positions compared to a global sqrt schedule. *Right:* per-token denoising loss increases linearly with diffusion steps which preserves the harder-to-recover tokens.

(ii) Impact of molecular tokenization. Table 3 (bottom) applies the best noising configuration (token-aware + linear) with different molecular tokenizers. Our current framework uses a AIS-based SMILES tokenizer for a stable, deterministic segmentation that preserves SMILES syntax (e.g., bracketed atoms and ring/branch symbols) and enables straightforward serialization, the tokenizer is plug-and-play and can be swapped without changing the denoising objective or architecture. We observe a clear monotonic trend: regex-based SMILES tokenization underperforms atom-level tokens, while Atoms-in-SMILES (AIS) yields the strongest performance across all reported metrics.

(iii) Decoding Efficiency. As diffusion models inherently require iterative refinement, we analyze the quality–latency trade-off in Appendix A.4 (Table 5). We observe that reducing the number of reverse steps yields substantial speedups at the cost of generation fidelity. To prioritize modeling performance over speed in our main analysis, we utilize T=2000 for all reported experiments, deferring efficiency optimizations to future work.

5 Conclusion

We introduced `BiMol-Diff`, a unified diffusion framework addressing the bidirectional tasks of molecular graph generation and textual captioning. By moving beyond standard data-agnostic corruption to a *token-aware noise schedule*, our approach explicitly preserves chemically salient substructures during the diffusion process. Empirical validation on M3-20M and ChEBI-20 confirms that this strategy yields significant gains over state-of-the-art autoregressive and diffusion baselines, particularly in structural fidelity (S2G) and semantic alignment (G2S). These findings demonstrate that token-aware denoising is essential for high-fidelity, controllable molecular language modeling, offering a robust alternative to autoregressive approaches in scientific domains.

6 Limitations

First, `BiMol-Diff` relies on linearized sequence representations and standard attention, lacking the explicit structural inductive bias inherent to graph-native architectures, which may limit data efficiency for complex topologies. Second, as with most diffusion models, our approach incurs higher computational costs during inference compared to autoregressive baselines due to the iterative denoising process, highlighting the need for future work on accelerated sampling.

References

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. *Preprint*, arXiv:2503.09573.

Utkarsh Bagal, Rishabh Aggarwal, PK Vinod, and Udai Priyakumar. 2021. Molgpt: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 61(6):2777–2790.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, MI. Association for Computational Linguistics.

Tian Bian, Yifan Niu, Heng Chang, Divin Yan, Junzhou Huang, Yu Rong, and Hong Cheng. 2024. Hierarchical graph latent diffusion model for conditional molecule generation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 130–140. ACM.

Jinho Chang and Jong Chul Ye. 2025. Ldmol: A text-to-molecule diffusion model with structurally informative latent space surpasses ar models. *Preprint*, arXiv:2405.17829.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. 2023. Unifying molecular and textual representations via multi-task language modelling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 6140–6157. PMLR.

Yifan Deng, Spencer S. Ericksen, and Anthony Gitter. 2025. Chemical language model linker: Blending text and molecules with modular adapters. *Journal of Chemical Information and Modeling*, 65(17):8944–8956.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. 2022a. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Michael Edwards, R Gokul, Shengyu Wang, Mark Tuckerman, Jascha Sohl-Dickstein, Kieron Burke, Philippe Schwaller, and Alpha A Lee. 2022b. Transmol: Repurposing language model for molecular generation. *Nature Machine Intelligence*, 4(5):392–400.

Yin Fang, Ningyu Zhang, Zhuo Chen, Lingbing Guo, Xiaohui Fan, and Huajun Chen. 2024. Domain-agnostic molecular generation with chemical feedback. *Preprint*, arXiv:2301.11259.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a graph: Encoding graphs for large language models. In *The Twelfth International Conference on Learning Representations*.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds Mach.*, 30(4):681–694.

Nolan Frey, David Kreutter, Jannis Born, Matteo Manica, Ola Engkvist, and Rafael Rodríguez-Pérez. 2022. Synthesizer: Expanding molecular diversity by iterative chemical language model sampling. *Frontiers in Chemistry*, 10:907363.

Haisong Gong, Qiang Liu, Shu Wu, and Liang Wang. 2024. Text-guided molecule generation with diffusion language model. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. Diffuseq: Sequence to sequence text generation with diffusion models. *Preprint*, arXiv:2210.08933.

Shuhan Guo, Yatao Bian, Ruibing Wang, Nan Yin, Zhen Wang, and Quanming Yao. 2025a. Unimot: Unified molecule-text language model with discrete token representation. *Preprint*, arXiv:2408.00863.

Siyuan Guo, Lexuan Wang, Chang Jin, Jinxian Wang, Han Peng, Huayang Shi, Wengen Li, Jihong Guan, and Shuigeng Zhou. 2025b. M^3-20m: A large-scale multi-modal molecule dataset for ai-driven drug design and discovery. *Preprint*, arXiv:2412.06847.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Preprint*, arXiv:2006.11239.

Ryan Irwin, Stefan Dimitrov, Roel van Deursen, and Djork-Arné Clevert. 2022. Chemformer: A pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(2):025015.

Sangyeup Kim, Nayeon Kim, Yinhua Piao, and Sun Kim. 2025a. Grapht5: Unified molecular graph-language modeling via multi-modal cross-token attention. *Preprint*, arXiv:2503.07655.

Seojin Kim, Hyeontae Song, Jaehyun Nam, and Jinwoo Shin. 2025b. Training text-to-molecule models with context-aware tokenization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22442–22460, Suzhou, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Preprint*, arXiv:2205.14217.

Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. 2024a. Graph diffusion transformers for multi-conditional molecular generation. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. 2024b. Graph diffusion transformers for multi-conditional molecular generation. *Preprint*, arXiv:2401.13858.

Pengfei Liu, Yiming Ren, Jun Tao, and Zhixiang Ren. 2024c. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *Computers in Biology and Medicine*, 171:108073.

Zequn Liu, Wei Zhang, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Ming Zhang, and Tie-Yan Liu. 2023. MolXPT: Wrapping molecules with text for generative pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1606–1616, Toronto, Canada. Association for Computational Linguistics.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2024d. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *Preprint*, arXiv:2310.12798.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *Preprint*, arXiv:2502.09992.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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, Philadelphia, PA. Association for Computational Linguistics.

Krishna Pillutla, Lang Liu, John Thickstun, Sean Welleck, Swabha Swayamdipta, Rowan Zellers, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. 2023. Mauve scores for generative models: Theory and practice. *Preprint*, arXiv:2212.14578.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.

Duong Thanh Tran, Nhat Truong Pham, Nguyen Doan Hieu Nguyen, and Balachandran Manavalan. 2024. Mol2Lang-VLM: Vision- and text-guided generative pre-trained language models for advancing molecule captioning through multimodal fusion. In *Proceedings of the 1st Workshop on Language + Molecules (L+M 2024)*, pages 97–102, Bangkok, Thailand. Association for Computational Linguistics.

Umit Ucak, Islambek Ashyrmamatov, and Juyong Lee. 2023. Improving the quality of chemical language model outcomes with atom-in-smiles tokenization. *Journal of Cheminformatics*, 15.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. 2023. Digress: Discrete denoising diffusion for graph generation. *Preprint*, arXiv:2209.14734.

David Weininger. 1988. 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.

David Weininger, Arthur Weininger, and Joseph L. Weininger. 1989. Smiles. 2. algorithm for generation of unique smiles notation. *Journal of Chemical Information and Computer Sciences*, 29(2):97–101.

Yuran Xiang, Haiteng Zhao, Chang Ma, and Zhi-Hong Deng. 2025. Instruction-based molecular graph generation with unified text-graph diffusion model. *Preprint*, arXiv:2408.09896.

Zheni Zeng, Bangchen Yin, Shipeng Wang, Jiarui Liu, Cheng Yang, Haishen Yao, Xingzhi Sun, Maosong Sun, Guotong Xie, and Zhiyuan Liu. 2023. Interactive molecular discovery with natural language. *Preprint*, arXiv:2306.11976.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Xiaochen Zhang, Shuangxi Wang, Ying Fang, and Qiankun Zhang. 2025. Mg-diff: A novel molecular graph diffusion model for molecular generation and optimization. *PLOS ONE*, 20(10):e0331450.

Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei YE, and Quanquan Gu. 2023. Structure-informed language models are protein designers. *Preprint*, arXiv:2302.01649.

Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. 2024. 3m-diffusion: Latent multi-modal diffusion for language-guided molecular structure generation. *Preprint*, arXiv:2403.07179.

A Appendix

A.1 Related Works

Table 4 positions `BiMol-Diff` relative to recent state-of-the-art molecule–text generation frameworks. While autoregressive (AR) baselines, such as MolT5 and UniMoT, have successfully demonstrated bidirectional capabilities, they often suffer from the standard limitations of sequential decoding (e.g., exposure bias). Conversely, existing diffusion-based approaches like TGM-DLM and UTGDiff offer non-autoregressive benefits but are predominantly unidirectional, focusing almost exclusively on the Text→Graph modality. `BiMol-Diff` bridges this gap as a unified diffusion framework capable of both Graph→Text and Text→Graph generation. Furthermore, we distinguish our approach by integrating a *token-aware mechanism*, a feature notably absent in prior diffusion baselines—which adapts the noising schedule to prioritize chemically significant tokens, similar to the strategy employed by the AR-based ChemT5.

A.2 Derivation

`BiMol-Diff` builds on the standard diffusion framework, which trades the flexibility of expressive generative models (e.g., GANs, VAEs, flow models) for the tractability of likelihood-based training in a continuous latent space $\mathbf{z}$. The overall goal is to minimize the negative log-likelihood

$$\mathbb{E}_{\mathbf{z}_0,\mathbf{c}}\big[-\log p_\theta(\mathbf{z}_0 \mid \mathbf{c})\big], \tag{10}$$

which is upper-bounded by the Variational Lower Bound (VLB).

A.2.1 Forward and Reverse Processes

The forward Markov chain is defined as $q(\mathbf{z}_{1:T} \mid \mathbf{z}_0) = \prod_{t=1}^{T} q(\mathbf{z}_t \mid \mathbf{z}_{t-1})$, where each transition is Gaussian:

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}\Big(\mathbf{z}_t \,\Big|\, \sqrt{1-\beta_t}\,\mathbf{z}_{t-1},\ \beta_t\,\mathbf{I}\Big). \tag{11}$$

Let $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. By induction, the marginal at time t satisfies:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I}), \tag{12}$$

so that $q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_t}\,\mathbf{z}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\Big)$. We used the *sqrt* schedule as the baseline schedule used in DiffusionLM (Li et al., 2022), namely $\bar{\alpha}_t = 1-\sqrt{t/T+s}$ with small $s>0$. The reverse denoising process then learns

$$\begin{aligned} p_\theta(\mathbf{z}_{0:T}) &= p(\mathbf{z}_T)\prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t), \\ p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) &= \mathcal{N}\Big(\boldsymbol{\mu}_\theta(\mathbf{z}_t,t),\ \boldsymbol{\sigma}_\theta^2(\mathbf{z}_t,t)\Big). \end{aligned} \tag{13}$$

Applying Bayes' rule to the forward transitions yields the exact posterior mean

$$\boldsymbol{\mu}_t(\mathbf{z}_t,\mathbf{z}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,\mathbf{z}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\,\mathbf{z}_0, \tag{14}$$

whose coefficients we denote by $\mathcal{U}$ and $\mathcal{E}$. `BiMol-Diff`'s training objective is then to match the network's predicted $\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta$ to these posterior quantities via a simple noise-prediction loss. We optimize the negative log-likelihood by upper-bounding it with the variational lower bound

$$\mathbb{E}\big[-\log p_\theta(x_0)\big] \;\le\; \mathcal{L}_{\text{vlb}} \;=\; \sum_{t=0}^{T} \mathcal{L}_t. \tag{15}$$

A.2.2 Variational Lower Bound (VLB)

Following Sohl-Dickstein et al.(Sohl-Dickstein et al., 2015), for conditional generation the VLB decomposes into:

$$\begin{aligned} \mathcal{L}_{\text{vlb}} &= \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{z}_0)}\Big[\mathcal{L}_T + \sum_{t=2}^{T}\mathcal{L}_t - \mathcal{L}_0\Big], \\ \mathcal{L}_T &:= \log\frac{q(\mathbf{z}_T \mid \mathbf{z}_0)}{p(\mathbf{z}_T)}, \\ \mathcal{L}_t &:= \log\frac{q(\mathbf{z}_{t-1} \mid \mathbf{z}_t,\mathbf{z}_0)}{p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t,\mathbf{c})}, \\ \mathcal{L}_0 &:= \log p_\theta(\mathbf{z}_0 \mid \mathbf{z}_1,\mathbf{c}). \end{aligned} \tag{16}$$

Method	Generation family	Text-conditioned	Graph→Text	Text→Graph	Bidirectional	Token-aware training
Autoregressive (AR)						
MolT5 (Edwards et al., 2022a)	Seq2Seq PLM (T5)	✓	✓	✓	✓	—
Text+ChemT5 (Christofidellis et al., 2023)	Seq2Seq PLM (T5)	✓	✓	✓	✓	—
CAMT5 (Kim et al., 2025b)	Seq2Seq PLM (T5)	✓	—	✓	—	✓
UniMoT (Guo et al., 2025a)	LLM-based multitask	✓	✓	✓	✓	—
LDMol (Chang and Ye, 2025)	PLM-based generator	✓	—	✓	—	—
MolCA (Liu et al., 2024d)	Encoder–decoder (graph-aware)	—	✓	—	—	—
Mol2Lang-VLM (Tran et al., 2024)	Multimodal PLM	—	✓	—	—	—
Non-autoregressive (Diffusion)						
TGM-DLM (Gong et al., 2024)	Diffusion-LM	✓	—	✓	—	—
3M-Diffusion (Zhu et al., 2024)	Latent diffusion (graph/seq)	✓	—	✓	—	—
UTGDiff (Xiang et al., 2025)	Graph diffusion	✓	—	✓	—	—
Graph DiT (Liu et al., 2024a)	Graph diffusion transformer	—	—	—	—	—
`BiMol-Diff` (ours)	Conditional diffusion	✓	✓	✓	✓	✓

Table 4: Feature comparison of `BiMol-Diff` against representative autoregressive (AR) and diffusion baselines. (✓= supported; — = not supported/reported). Token-aware mechanism denotes chemistry-aware tokenization/weighting in AR models and token-wise noising schedules in diffusion models. Unlike prior diffusion works which are typically unidirectional, `BiMol-Diff` enables fully bidirectional generation.

where each $\mathcal{L}_t$ is a KL divergence between Gaussians. The true posterior mean (via Bayes' rule) is:

$$\boldsymbol{\mu}_t(\mathbf{z}_t, \mathbf{z}_0) = \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}_{\mathcal{U}} \mathbf{z}_t + \underbrace{\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}}_{\mathcal{E}} \mathbf{z}_0, \tag{17}$$

with covariance $\mathbf{\Sigma}_q = \tilde{\beta}_t \mathbf{I}$, $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. In the standard simplification, the model's covariance is fixed to match the true posterior covariance $\mathbf{\Sigma}_\theta = \mathbf{\Sigma}_q$, the KL collapses to a weighted MSE:

$$\mathcal{L}_t = \frac{1}{2}\left\|\boldsymbol{\mu}_t - \boldsymbol{\mu}_\theta\right\|^2_{\mathbf{\Sigma}_q^{-1}} \propto \mathbb{E}\left[\left\|\mathbf{z}_0 - \mathcal{M}_\theta(\mathbf{z}_t, t, \mathbf{c})\right\|^2\right]. \tag{18}$$

Thus, for $2 \le t \le T$, $\mathcal{L}_t \to \|\mathbf{z}_0 - \mathcal{M}_\theta(\mathbf{z}_t, t, \mathbf{c})\|^2$. The final KL encourages $\mathbf{z}_T$ to match the unit Gaussian prior:

$$\mathcal{L}_T = \mathrm{KL}\big(q(\mathbf{z}_T \mid \mathbf{z}_0) \,\|\, p(\mathbf{z}_T)\big) \propto \left\|\boldsymbol{\mu}(\mathbf{z}_T)\right\|^2, \tag{19}$$

a constant w.r.t. θ. The discrete target $\mathbf{S}$ (sequence) is encoded into a continuous embedding $g_\Phi(\mathbf{S})$. The final term in VLB is $\mathcal{L}_0 = -\log p_\theta(\mathbf{z}_0 \mid \mathbf{z}_1, \mathbf{c})$. We need to integrate the discrete data $\mathbf{S}$ into this continuous likelihood term. We use the law of total probability to express the continuous likelihood $p_\theta(\mathbf{z}_0 \mid \mathbf{z}_1, \mathbf{c})$ by marginalizing over all possible discrete tokens in the target sequence $\mathbf{S} = \{s_1, s_2 \cdots, s_N\}$:

$$p_\theta(\mathbf{z}_0 \mid \mathbf{z}_1, \mathbf{c}) = \sum_{\mathbf{S}} p_\theta(\mathbf{z}_0, \mathbf{S} \mid \mathbf{z}_1, \mathbf{c}) \tag{20}$$

We then apply the product rule to the joint probability:

$$p_\theta(\mathbf{z}_0, \mathbf{S} \mid \mathbf{z}_1, \mathbf{c}) = p_\theta(\mathbf{z}_0 \mid \mathbf{S}, \mathbf{z}_1, \mathbf{c}) \cdot p_\theta(\mathbf{S} \mid \mathbf{z}_1, \mathbf{c}) \tag{21}$$

For training, we are interested in the specific ground-truth sequence $\mathbf{S}$. When we evaluate $\mathcal{L}_0$ during training, we consider only the term where $\mathbf{S}$ is the ground-truth sequence:

$$\begin{aligned}\mathcal{L}_0 &\approx -\log p_\theta(\mathbf{z}_0, \mathbf{S} \mid \mathbf{z}_1, \mathbf{c}) \\ &= -\log\Big[p_\theta(\mathbf{z}_0 \mid \mathbf{S}, \mathbf{z}_1, \mathbf{c}) \cdot p_\theta(\mathbf{S} \mid \mathbf{z}_1, \mathbf{c})\Big].\end{aligned} \tag{22}$$

The core approximation simplifies the dependency graph by asserting that the discrete data $\mathbf{S}$ is generated only from the clean latent $\mathbf{z}_0$, and is independent of $\mathbf{z}_1$ and $\mathbf{c}$ given $\mathbf{z}_0$.

$$\mathbf{S} \perp (\mathbf{z}_1, \mathbf{c}) \mid \mathbf{z}_0 \tag{23}$$

This allows us to replace the discrete conditional likelihood with the separate rounding network $\tilde{p}_\Phi(\mathbf{S} \mid \mathbf{z}_0)$: $p_\theta(\mathbf{S} \mid \mathbf{z}_1, \mathbf{c}) \approx \tilde{p}_\Phi(\mathbf{S} \mid \mathbf{z}_0)$. Substituting this back into the likelihood decomposition:

$$p_\theta(\mathbf{z}_0, \mathbf{S} \mid \mathbf{z}_1, \mathbf{c}) \approx p_{\text{cont}}(\mathbf{z}_0 \mid \mathbf{S}, \mathbf{z}_1, \mathbf{c}) \cdot \tilde{p}_\Phi(\mathbf{S} \mid \mathbf{z}_0) \tag{24}$$

Taking the negative logarithm of the approximation gives the two desired terms:

$$\mathcal{L}_0 \approx -\log p_{\text{cont}}(\mathbf{z}_0 \mid \mathbf{S}, \mathbf{z}_1, \mathbf{c}) - \log \tilde{p}_\Phi(\mathbf{S} \mid \mathbf{z}_0)$$

This split yields the two components used in the final training objective:

1. Consistency Term ($\mathcal{L}_{\text{Cons}}$): The first term is the negative log-likelihood of the continuous latent, which is minimized via the MSE loss on the means: $-\log p_{\text{cont}}(\mathbf{z}_0 \mid \mathbf{S}, \mathbf{z}_1, \mathbf{c}) \to \mathcal{L}_{\text{Consistency}} = \left\|g_\Phi(\mathbf{S}) - \mathcal{M}_\theta(\mathbf{z}_1, 1, \mathbf{c})\right\|^2$.

2. Rounding Term ($\mathcal{L}_{\text{Round}}$): This second term is the dedicated loss for the discrete data likelihood: $\mathcal{L}_{\text{Round}} = -\log \tilde{p}_\Phi(\mathbf{S} \mid \mathbf{z}_0)$

A.2.3 Final End-to-End Objective

Combining all components:

$$\mathcal{L}_{\text{vlb}} \propto \sum_{t=2}^{T} \underbrace{\left\|\mathbf{z}_0 - \mathcal{M}_\theta(\mathbf{z}_t, t, \mathbf{c})\right\|^2}_{\text{Denoising}} \tag{25}$$

$$+ \underbrace{\left\|g_\Phi(\mathbf{S}) - \mathcal{M}_\theta(\mathbf{z}_1, 1, \mathbf{c})\right\|^2}_{\text{Consistency}} \tag{26}$$

$$- \underbrace{\log \tilde{p}_\Phi(\mathbf{S} \mid \mathbf{z}_0)}_{\text{Rounding}}. \tag{27}$$

Dropping constant terms, the simplified end-to-end training loss is:

$$\mathcal{L}_{\text{e2e-simple}}(\mathbf{S}) = \mathbb{E}_q\Bigg[\sum_{t=2}^{T} \underbrace{\left\|\mathcal{M}_\theta(\mathbf{z}_t, t, \tilde{\mathcal{G}}) - \mathbf{z}_0\right\|^2}_{\text{Denoising}} + \underbrace{\left\|g_\Phi(\mathbf{S}) - \mathcal{M}_\theta(\mathbf{z}_1, 1, \tilde{\mathcal{G}})\right\|^2}_{\text{Consistency}} - \underbrace{\log \tilde{p}_\Phi(\mathbf{S} \mid \mathbf{z}_0)}_{\text{Rounding}}\Bigg]. \tag{28}$$

A.3 Non-increasing Isotonic Projection

After constructing the per-token cumulative schedule $\tilde{\alpha}_t^i$, we project it onto the set of non-increasing sequences $\{\bar{\alpha}_t^i\}_{t=1}^T$ such that $\bar{\alpha}_1^i \geq \bar{\alpha}_2^i \geq \cdots \geq \bar{\alpha}_T^i$. Concretely, this is a 1D isotonic regression problem with squared loss, which we solve using the standard Pool-Adjacent-Violators Algorithm (PAVA). This algorithm finds the closest monotone non-increasing sequence (in the least-squares sense) to the input. Intuitively, it smooths out spurious "bumps" in the loss profile while guaranteeing that the cumulative signal strength strictly decays over time, fulfilling the monotonicity requirement of the diffusion process.

A.4 Decoding Efficiency

Table 5 analyzes the trade-off between inference speed and generation quality for Graph-to-Sequence tasks. First, comparing within the diffusion family, `BiMol-Diff` demonstrates significantly better efficiency than the standard DiffuSeq baseline. At 2000 steps, our model achieves a BLEU score of 0.567 while requiring only 89 seconds to process a batch, representing a 3.6× speedup over DiffuSeq (317 seconds) with better output generation.

Method	Params	Steps	Time (s)	Speedup	BLEU ↑
Autoregressive Baselines					
MolT5-Base (Edwards et al., 2022a)	220M	–	<5s	–	0.452
Text+Chem T5 (Christofidellis et al., 2023)	223M	–	<5s	–	0.542
Diffusion Baselines					
DiffuSeq (Gong et al., 2023)	91M	2000	317s	1.0× (Ref)	0.532
Ours					
BiMol-Diff	63M	2000	89s	3.6×	**0.567**
BiMol-Diff	63M	1000	45s	7.0×	0.551
BiMol-Diff	63M	500	23s	13.8×	0.365
BiMol-Diff	63M	100	5s	63.4×	0.312

Table 5: Inference efficiency comparison on the Graph-to-Sequence (G2S) task. We compare `BiMol-Diff` against the best Autoregressive (AR) and Diffusion baselines from Table 1. Time is measured as total inference time for a batch size of 50 on a single NVIDIA V100 GPU.

When compared to Autoregressive (AR) baselines, our model shows a clear operational boundary. In the high-quality output range (1000–2000 steps), `BiMol-Diff` outperforms the strongest AR baseline, Text+Chem T5 (0.542 BLEU), confirming that the extra computational time yields better text generation. However, the model struggles when pushed for extreme speed. As we reduce the inference steps below 1000, we observe a sharp drop in performance. At 500 steps, although the inference time improves to 23 seconds (13.8× speedup), the BLEU score falls to 0.365, which is lower than even the base MolT5 model (0.452). This indicates that `BiMol-Diff` requires a minimum threshold of denoising steps (approximately 1000) to maintain its advantage. Consequently, while our model is ideal for high-precision generation, simple AR models remain the better choice for real-time, low-latency applications.