CAMS-DIFF: COUPLED ADAPTIVE MIXED-SPACE DIFFUSION FOR MOLECULAR TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

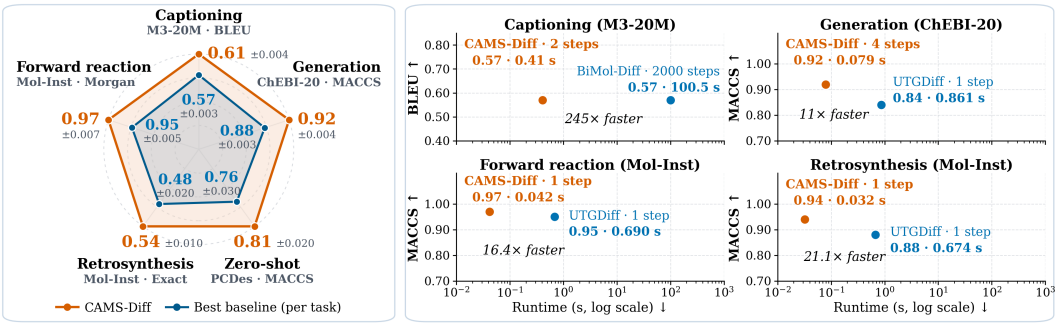
Diffusion models are increasingly used for molecular translation, including molecule captioning, text-guided generation, forward reaction prediction, and retrosynthesis. Continuous sequence diffusion provides a common formulation across these tasks, with molecules represented through SMILES. However, two limitations remain. (1) At low noise levels, continuous representations can retain substantial local token information, reducing reliance on broader sequence context and the input condition. (2) During inference, uniform timestep reduction can also skip informative regions of a non-uniform denoising trajectory. We introduce **C**oupled **A**daptive **M**ixed-**S**pace **Diff**usion (`CAMS-Diff`). During training, CAMS-Diff couples token-wise adaptive Gaussian noising with a discrete representation-masking process governed by the same learned schedule, selectively masking direct local token evidence. During inference, it profiles reconstruction changes in token-wise log-SNR space and uses the resulting trajectory importance to select denoising timesteps under a fixed sampling budget. Across five molecular translation and generalization settings, CAMS-Diff improves molecular and semantic recovery; on ChEBI-20, it reaches 0.916 MACCS compared with 0.883 for BiMol-Diff. Controlled ablations isolate both mechanisms: coupling outperforms adaptive noising, masking, and their uncoupled combination, while trajectory-informed timestep selection outperforms uniform subsampling at matched budgets. Two-step captioning, using two denoiser evaluations instead of 2000, achieves quality comparable to full-step BiMol-Diff with an approximately $245\times$ wall-clock speedup. CAMS-Diff also transfers zero-shot to PCDes, improving MACCS from 0.763 to 0.810.

1 INTRODUCTION

Molecular translation spans molecule captioning, text-guided molecule generation, forward reaction prediction, and retrosynthesis (Edwards et al., 2022; Pei et al., 2023; 2024). Existing approaches include autoregressive molecular language models (Edwards et al., 2022; Pei et al., 2023), continuous sequence diffusion (Gong et al., 2024; Shahane et al., 2026), and graph diffusion (Xiang et al., 2025). Continuous sequence diffusion provides a common conditional denoising formulation across tasks with textual or SMILES targets (Li et al., 2022; Gong et al., 2024), and has been extended to bidirectional molecular translation (Shahane et al., 2026). However two limitations of these molecular diffusion models remain underexplored: (1) how corruption should vary across token positions during training, and (2) where denoising steps should fall along the reverse trajectory at inference.

L1: Local token recoverability can reduce reliance on sequence context and conditioning. At low noise levels, continuous representations may retain substantial information about the underlying discrete token (Li et al., 2022; Chen et al., 2023), allowing the denoiser to reconstruct a position with limited reliance on the surrounding sequence or conditioning input. Such local recoverability undesirable for SMILES, where correct reconstruction can depend on non-local dependencies such as ring closures and nested branches (Özçelik et al., 2024; Jindal & Ju, 2026). Token-wise adaptive noising changes how strongly individual positions are corrupted, but does not explicitly remove this locally available evidence. Masking and absorbing-state corruption remove local evidence directly in text diffusion (Chen et al., 2023; Gong et al., 2023a), but the interaction between discrete masking and token-wise adaptive continuous corruption remains underexplored.

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



(a) Full-step performance across five settings **(b) Reduced-step inference: quality vs. runtime**

Figure 1: **Overview of CAMS-Diff.** (a) Task-representative full-step performance across five settings. (b) Reduced-step quality-runtime trade-offs at the denoising budgets shown.

L2: Reduced-step sampling is sensitive to where denoising evaluations are retained. Diffusion inference requires many sequential denoiser evaluations, so practical deployment depends on reducing the reverse trajectory to a small number of steps. Uniform timestep subsampling is the common choice, and BiMol-Diff degrades under it (Shahane et al., 2026), suggesting that the loss comes from which steps are discarded rather than from using fewer steps alone. In image diffusion, Align Your Steps (Sabour et al., 2024) confirms this: step placement matters, not only step count. Molecular work has instead pursued fewer evaluations directly; FlashMol (Wei et al., 2026) reaches few-step generation for 3D conformations through distillation and timestep respacing. Trajectory-informed placement under a fixed budget therefore remains unexplored for molecular translation.

To address these limitations, we introduce `CAMS-Diff`, a **C**oupled **A**daptive **M**ixed-**S**pace **Diff**usion framework for molecular translation. During training, it couples token-wise adaptive Gaussian noising with representation masking governed by the same learned schedule, selectively masking direct local token evidence. During inference, reconstruction changes along the reverse trajectory are profiled in log-SNR space and used to select informative timesteps under a fixed budget of K denoiser steps. The two components target complementary aspects of diffusion: adaptive mixed-space corruption and reduced-step allocation guided by learned denoising dynamics.

We evaluate `CAMS-Diff` across molecule captioning, text-guided generation, forward reaction prediction, retrosynthesis, and zero-shot generalization. Controlled ablations isolate both mechanisms: coupled mixed-space corruption improves over adaptive noising, masking, and their uncoupled combination, while trajectory-informed selection improves over uniform subsampling at matched budgets. On ChEBI-20, `CAMS-Diff` reaches 0.916 MACCS compared with 0.883 for BiMol-Diff, and zero-shot transfer to PCDes improves MACCS from 0.763 to 0.810. Two-step captioning matches full 2000-step BiMol-Diff quality using two rather than 2000 denoiser evaluations, an approximately 245× wall-clock speedup. Fig 1 summarizes experimental breadth and quality-runtime trade-offs.

Contributions. (1) *Coupled adaptive mixed-space diffusion*: representation masking follows the same learned token-wise schedule as Gaussian corruption rather than being allocated independently. (2) *Trajectory-informed reduced-step decoding*: post-training reconstruction dynamics in log-SNR space select which reverse timesteps to retain under a fixed budget. (3) The same formulation instantiated across molecule-to-text, text-to-molecule, and molecule-to-molecule translation, evaluated against autoregressive, sequence-diffusion, and graph-diffusion baselines.

2 Related Work

Prior molecular translation methods span autoregressive (**AR**), sequence-diffusion (**SD**), and graph-diffusion (**GD**) formulations. Table 1 compares diffusion space, adaptive noising (**AN**), explicit graph structural bias (**GSB**), few-step inference (**FS**), adaptive step allocation (**SA**), denoiser evaluations (**NFE**), and experimental breadth. Mixed continuous/discrete corruption has precedents in text diffusion (Chen et al., 2023; Gong et al., 2023a), but is not coupled to a learned token-wise adaptive schedule for molecular translation. Existing molecular methods otherwise address these properties largely separately: BiMol-Diff adapts continuous noising, UTGDiff provides one-step graph diffusion, and FlashMol targets accelerated 3D molecular generation. `CAMS-Diff` couples adaptive Gaussian noising with representation masking and combines it with trajectory-informed step allocation within the same sequence-diffusion formulation. A detailed discussion is in App. A.

Table 1: Comparison of molecular translation and molecular diffusion methods. NFE is the number of denoiser evaluations in each method's reported low-step setting; full-step CAMS-Diff uses 2000.

			Methodology / Inference					Experimental Breadth				
Type	**Method**	**Space**	**AN**	**GSB**	**FS**	**SA**	**NFE**	**Cap.**	**T2M**	**Fwd.**	**Retro.**	**ZS**
R	MolT5 (Edwards et al., 2022)	Tok. seq.	✗	✗	✗	✗	AR	✓	✓	✗	✗	✗
	BioT5+ (Pei et al., 2024)	Tok. seq.	✗	✗	✗	✗	AR	✓	✓	✓	✓	✗
D	TGM-DLM (Gong et al., 2024)	Cont.	✗	✗	✗	✗	2000	✗	✓	✗	✗	✗
	BiMol-Diff (Shahane et al., 2026)	Cont.	✓	✗	✗	✗	2000	✓	✓	✗	✗	✗
D	TDGD (Yao et al., 2026)	Disc. graph	✗	✓	✗	✗	500	✗	✓	✗	✗	✗
	UTGDiff (Xiang et al., 2025)	Disc. graph	✗	✓	✓	✗	1	✗	✓	✓	✓	✓
3D	FlashMol (Wei et al., 2026)	3D latent	✗	✗	✓	✗	4	✗	✗	✗	✗	✗
D	**CAMS-Diff**	**Mixed seq.**	✓	✗	✓	✓	**1–4**	✓	✓	✓	✓	✓

3 Background and Problem Formulation

Problem formulation. Let $\mathcal{G}$ denote a molecular structure and $\mathbf{s} = \mathrm{SMILES}(\mathcal{G}) = (s_1, \ldots, s_M)$ its SMILES sequence. Let $X = (x_1, \ldots, x_N)$ denote the corresponding natural-language description. Molecule captioning predicts X from $\mathbf{s}$, while text-guided molecule generation predicts $\mathbf{s}$ from X. The two directions are trained on paired data $(\mathbf{s}, X) \sim \mathcal{D}$ using separate parameter sets. For a common formulation, let C denote the conditioning sequence and $Y = (y_1, \ldots, y_L)$ the target sequence, with $(C, Y) = (\mathbf{s}, X)$ for captioning and $(C, Y) = (X, \mathbf{s})$ for generation. For forward reaction prediction, C is the reactant-side SMILES and Y the product SMILES; retrosynthesis reverses these roles. The condition is encoded as $H_C = \mathrm{Enc}(C)$, and the target is mapped by the trainable embedding map g_Φ to $Z_0 = g_\Phi(Y) \in \mathbb{R}^{L \times d}$. Across tasks, only C and Y change; the sequence-diffusion formulation is shared.

Token-wise continuous diffusion. We build on prior token-wise adaptive continuous diffusion (Shahane et al., 2026; Yuan et al., 2024). Let $\mathcal{S} = \{\bar{\alpha}_t^i\}_{i=1,t=0}^{L,T}$ denote a cumulative signal-retention schedule over T diffusion steps, with $1 = \bar{\alpha}_0^i > \bar{\alpha}_1^i \geq \cdots \geq \bar{\alpha}_T^i > 0$. Boundary noise levels are shared across positions, while the interior schedule is position-dependent. A noisy latent is sampled as

$$z_t^i = \sqrt{\bar{\alpha}_t^i}\, z_0^i + \sqrt{1 - \bar{\alpha}_t^i}\, \epsilon^i, \qquad \epsilon^i \sim \mathcal{N}(0, I), \tag{1}$$

where $\sqrt{1 - \bar{\alpha}_t^i}$ is the cumulative Gaussian noise scale at position i and timestep t. Although corruption is position-wise, the denoiser jointly processes $Z_t = (z_t^1, \ldots, z_t^L)$. The interior schedule is periodically recalibrated from an EMA-smoothed estimate of position-wise clean-latent prediction error, then interpolated and projected to remain monotone while preserving shared boundary noise levels. Full calibration details are in App. B.3.

Token-wise log-SNR and denoising. We parameterize each position by $\mathrm{SNR}_t^i = \bar{\alpha}_t^i / (1 - \bar{\alpha}_t^i)$ and $\lambda_t^i = \log \mathrm{SNR}_t^i$, with $\boldsymbol{\lambda}_t = (\lambda_t^1, \ldots, \lambda_t^L)$. Because positions may occupy different noise levels at the same timestep, the clean latent is predicted as $\widehat{Z}_{0,\theta} = f_\theta(Z_t, H_C, t, \boldsymbol{\lambda}_t)$. The clean-state objective uses $w_t^i = \frac{1}{2}(\mathrm{SNR}_{t-1}^i - \mathrm{SNR}_t^i)$. Since the boundary noise levels are shared, $\sum_{t=2}^T w_t^i$ is identical across positions; the token-wise schedule redistributes denoising weight along the trajectory rather than changing its total amount. The Gaussian objective and SNR weighting are derived in Appendices B.1 and B.2. This adaptive Gaussian process is our starting point; CAMS-Diff couples its learned corruption dynamics to representation masking and uses reverse-trajectory reconstruction dynamics for reduced-step decoding.

4 Methodology

`CAMS-Diff` addresses **L1** and **L2** through two complementary components: coupled adaptive mixed-space diffusion during training and trajectory-informed reduced-step decoding during inference. Both operate on the learned token-wise diffusion dynamics introduced in Sec. 3.

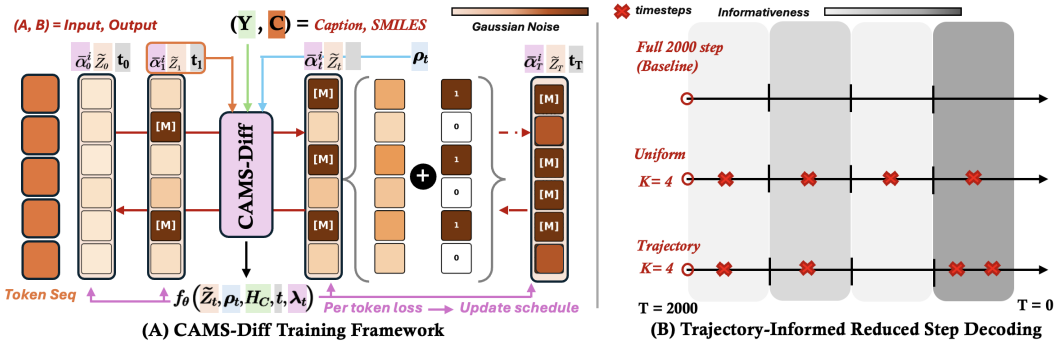


Figure 2: **Overview (A)** Training couples continuous diffusion with discrete masking through token-wise adaptive noise schedules. **(B)** At inference, trajectory informativeness is used to allocate a reduced budget of K reverse steps non-uniformly across the $T = 2000$ denoising trajectory.

4.1 Coupled Adaptive Mixed-Space Diffusion

To address **L1**, we augment the token-wise Gaussian process in Sec. 3 with discrete representation masking. Continuous representations can retain substantial local token evidence, particularly at low noise levels (Li et al., 2022). Masking removes this direct evidence and encourages reconstruction from the remaining sequence and conditioning input (Chen et al., 2023). This is particularly relevant for SMILES, where recovery can depend on non-local dependencies such as ring closures and nested branches (James et al., 2016). We term this *coupled adaptive mixed-space diffusion*: Gaussian diffusion evolves a continuous latent, while a discrete mask variable controls whether each position is observed or replaced by a learned mask representation.

Coupled corruption. Using the cumulative Gaussian noise scale $\sqrt{1-\bar{\alpha}_t^i}$ from Sec. 3, let $m \in \mathbb{R}^d$ denote a learned mask representation. We sample

$$\rho_t^i \sim \text{Bernoulli}(\varpi_t^i), \qquad \varpi_t^i = \eta\sqrt{1-\bar{\alpha}_t^i}, \qquad t = 1, \dots, T, \tag{2}$$

where η controls masking strength; we use $\eta = 0.5$ by default. Thus, masking is coupled directly to the learned token-wise Gaussian schedule: recalibrating $\bar{\alpha}_t^i$ also recalibrates ϖ_t^i. The mixed observation is

$$\widetilde{z}_t^i = (1-\rho_t^i)z_t^i + \rho_t^i m, \tag{3}$$

so a masked position replaces its Gaussian-noised representation with m. Let $\boldsymbol{\rho}_t = (\rho_t^1, \dots, \rho_t^L)$ and $\widetilde{Z}_t = (\widetilde{z}_t^1, \dots, \widetilde{z}_t^L)$. Conditioned on $\mathcal{S}$, the augmented forward process factorizes as $q_{\mathcal{S},\eta}(Z_{1:T}, \boldsymbol{\rho}_{1:T} \mid Z_0) = \prod_{t=1}^T q_{\mathcal{S}}(Z_t \mid Z_{t-1}) q_\eta(\boldsymbol{\rho}_t \mid t, \mathcal{S})$, where

$$q_\eta(\boldsymbol{\rho}_t \mid t, \mathcal{S}) = \prod_{i=1}^L (\varpi_t^i)^{\rho_t^i}(1-\varpi_t^i)^{1-\rho_t^i}. \tag{4}$$

Marginalizing $\boldsymbol{\rho}_{1:T}$ recovers the original Gaussian forward trajectory.

Denoising process. We retain Z_t as the Gaussian latent state and supply $\widetilde{Z}_t$ to the denoiser:

$$\widehat{Z}_{0,\theta}^{\text{mix},(t)} = f_\theta\left(\widetilde{Z}_t, \boldsymbol{\rho}_t, H_C, t, \boldsymbol{\lambda}_t\right). \tag{5}$$

The reverse parameterization is unchanged; only the clean-state prediction is formed from the mixed observation. During sampling, $\boldsymbol{\rho}_t$ is resampled at each denoiser evaluation using Eq. 2. Details are in App. B.1.

Training objective. Because the same masking distribution appears in the forward and generative factorizations, its probability terms cancel in the variational ratio, leaving the Gaussian posterior and token-wise weights w_t^i unchanged. The resulting objective is

$$\mathcal{L}_{\text{mix}} = \mathbb{E}_{q_{\mathcal{S},\eta}(Z_{1:T},\boldsymbol{\rho}_{1:T}|Z_0)}\left[\sum_{t=2}^T\sum_{i=1}^L w_t^i \left\|\widehat{z}_{0,\theta}^{i,\text{mix},(t)} - z_0^i\right\|_2^2 + \left\|Z_0 - \widehat{Z}_{0,\theta}^{\text{mix},(1)}\right\|_F^2\right] - \log p_\psi(Y \mid Z_0). \tag{6}$$

Thus, the learned token-wise schedule governs both Gaussian corruption and representation masking. Algo. 1 summarizes training; the full derivation is given in App B.1.

4.2 TRAJECTORY-INFORMED REDUCED-STEP DECODING

To address **L2**, we use reconstruction dynamics of the trained model to identify regions of the reverse trajectory where denoising changes most. Given a budget $K \ll T$, uniform timestep subsampling can skip such regions. We therefore profile the full reverse trajectory once on the validation set, construct a shared K-evaluation schedule, and reuse it unchanged at test time.

Measuring denoising progress. After training, we run the full T-step reverse process on the validation set. For position i, let $\mathcal{V}_i$ contain examples for which i is a non-padding target token. We measure mean clean-state reconstruction error as

$$E_{t,i} = \frac{1}{|\mathcal{V}_i|} \sum_{b \in \mathcal{V}_i} \frac{1}{d} \left\| \widehat{z}_{0,b,\theta}^{\,i,\mathrm{mix},(t)} - z_{0,b}^{i} \right\|_2^2, \tag{7}$$

where d is the latent dimension. We pair $E_{t,i}$ with its token-wise log-SNR λ_t^i and fit a smooth trajectory $E_i(\lambda)$. Unlike the training-time difficulty used to recalibrate the forward schedule (App. B.3), this quantity is measured along the trained model's reverse trajectory and is used only for timestep allocation. Profiling details are given in Appendix C.

Local trajectory importance. The curve $E_i(\lambda)$ describes how reconstruction error evolves along the reverse trajectory at position i. For two nearby retained SNR levels, skipping the interval gives the local approximation error $\frac{1}{2}(-\frac{dE_i}{d\,\mathrm{SNR}})(\Delta\,\mathrm{SNR})^2$. Since $\mathrm{SNR} = e^\lambda$ and $\Delta\,\mathrm{SNR} \approx e^\lambda \Delta\lambda$, this becomes $\frac{1}{2} a_i(\lambda)(\Delta\lambda)^2$, with

$$a_i(\lambda) = \left[-e^\lambda \frac{dE_i(\lambda)}{d\lambda} \right]_+ . \tag{8}$$

Thus, $a_i(\lambda)$ measures the local penalty associated with skipping intermediate log-SNR levels. The positive part suppresses small positive slopes introduced by smoothing.

Proposition 1 (Continuous step allocation)**.** *Under the approximation above, the position-wise allocation minimizing the reconstruction-based approximation error satisfies*

$$p_i^\star(\lambda) \propto \sqrt{a_i(\lambda)},$$

where $p_i^\star(\lambda)$ is the density of retained denoising levels around log-SNR λ for position i.

The derivation is given in Appendix C.

From token-wise profiles to a shared schedule. Proposition 1 gives a preferred allocation shape separately for each target position. Reverse diffusion, however, uses a shared timestep t: one denoiser evaluation updates the complete sequence even though positions follow different log-SNR trajectories λ_t^i. We therefore map the corresponding unnormalized contributions $\sqrt{a_i(\lambda_t^i)}\,\left|d\lambda_t^i/dt\right|$ to the common timestep axis and aggregate over retained positions $\mathcal{A}$:

$$p(t) = \left[\sum_{i \in \mathcal{A}} \sqrt{a_i(\lambda_t^i)} \left| \frac{d\lambda_t^i}{dt} \right| \right] \Big/ \left[\int_1^T \sum_{i \in \mathcal{A}} \sqrt{a_i(\lambda_s^i)} \left| \frac{d\lambda_s^i}{ds} \right| \, ds \right]. \tag{9}$$

This aggregation is a practical construction of one shared schedule; Proposition 1 establishes the position-wise continuous allocation, while the finite-K shared schedule is evaluated in Sec. 5.

Selecting K evaluations. We discretize $p(t)$ with a simple binning heuristic. For a budget of K denoiser evaluations, let $B_{\mathrm{bin}} = \max\{1, \mathrm{round}((T-1)/(2K))\}$, yielding approximately $2K$ contiguous interior bins. For each bin $\mathcal{B}_b$, define its importance mass $M_b = \int_{\mathcal{B}_b} p(t)\,dt$. We retain the $K-1$ interior bins with largest mass and represent each by its $p(t)$-weighted center, $\bar{t}_b = \frac{\int_{\mathcal{B}_b} t\,p(t)\,dt}{\int_{\mathcal{B}_b} p(t)\,dt}$. Together with the two boundaries, this gives $T = J_0 > J_1 > \cdots > J_{K-1} > J_K = 0$.

Reduced-step decoding. The retained timesteps $\{J_r\}$ are shared across positions, while position i retains its learned noise level $\lambda_{J_r}^i$. The scheduler changes only *where* the K denoiser evaluations occur: the trained model and reverse update rule remain unchanged. It can therefore be used with compatible reduced-step samplers. For the token-wise Gaussian process, the non-adjacent transition between retained levels is given in Appendix C.5. The final denoiser evaluation predicts $\widehat{Z}_0$, so the K evaluations connect the $K+1$ levels $J_0, \ldots, J_K$ without an additional model call at $J_K = 0$. Algorithm 2 summarizes the offline construction.

Algorithm 1 Coupled mixed-space training

Require: $\mathcal{D}$, schedule $\mathcal{S}$, mask strength η, update interval T_{up}
Ensure: model M_θ, calibrated schedule $\mathcal{S}$
for $n = 1, \dots, N$ **do**
  Sample $(C, Y) \sim \mathcal{D}$, $t \sim \text{Unif}\{1, \dots, T\}$
  Noise $Z_0 \to Z_t$ under $\mathcal{S}$
  Sample $\rho_t^i \sim \text{Bernoulli}(\varpi_t^i)$ Eq. 2
  Form $\widetilde{Z}_t$ Eq. 3
  Predict $\widehat{Z}_0^{\text{mix}}$ and update θ with $\mathcal{L}_{\text{mix}}(t)$
  if $n \bmod T_{\text{up}} = 0$ **then**
    Recalibrate $\mathcal{S}$ from denoising difficulty
  end if
end for
return $M_\theta, \mathcal{S}$

Algorithm 2 Trajectory-informed step allocation

Require: M_θ, schedule $\mathcal{S}$, validation set $\mathcal{V}$, budget K
Ensure: shared schedule $\mathcal{J}_K$
Run the full T-step reverse process on $\mathcal{V}$
Measure $E_{t,i}$ and fit $E_i(\lambda)$ Eq. 7
Compute trajectory importance $a_i(\lambda)$ Eq. 8
Map and aggregate importance into $p(t)$ Eq. 9
Split into $\approx 2K$ bins and score each bin by $M_b = \int_{\mathcal{B}_b} p(t)\, dt$
Keep the $K-1$ highest-mass interior bins
Represent each selected bin by its $p(t)$-weighted center $\bar{t}_b$
$\mathcal{J}_K \leftarrow \{T\} \cup \{\bar{t}_b\} \cup \{0\}$
return $\mathcal{J}_K$

5 Experiments

We evaluate CAMS-Diff across molecule-to-text, text-to-molecule, and molecule-to-molecule translation, together with zero-shot transfer. We first evaluate full-step performance across these settings, then isolate the two methodological questions: **L1**, whether coupling adaptive Gaussian noising with representation masking improves reconstruction, and **L2**, whether trajectory-informed timestep placement improves decoding under a fixed denoising budget. We further examine effect of masking strength, alternative step-allocation biases, and performance as generation difficulty increases.

Experimental Setup

Tasks and datasets. We evaluate **Captioning** on M3-20M, **Generation** on ChEBI-20, **Forward** and **Retrosynthesis** on Mol-Instructions, and **Zero-shot** generation on PCDes (Guo et al., 2025; Edwards et al., 2021; Fang et al., 2024; Zeng et al., 2022). For zero-shot transfer, the ChEBI-20-trained generation model is applied directly to PCDes without task-specific fine-tuning. Full task definitions, dataset statistics, and splits are given in App. D.

Baselines and evaluation. We compare against task-relevant autoregressive, continuous sequence-diffusion, and graph-diffusion baselines. Captioning uses Text+Chem T5, GraphT5, DiffuSeq, TGM-DLM, and BiMol-Diff; generation additionally includes MolT5, 3M-Diffusion, and UTGDiff (Christofidellis et al., 2023; Kim et al., 2025; Gong et al., 2023b; Zhu et al., 2024). Reaction and zero-shot settings use the corresponding available baselines. For captioning, **SM** denotes surface metrics and **EM** semantic/embedding metrics; for molecular outputs, **MM** denotes fingerprint similarity and **S** SMILES-level metrics. Reproduced baselines are marked in the tables; where multiple runs are available, we report mean $\pm$ std. Full baseline and metric details are in App. E.

Implementation. Captioning uses the 91M-parameter configuration; text-guided generation, forward reaction prediction, and retrosynthesis use the 180M-parameter molecular-output configuration. Unless stated otherwise, full-step inference uses $T = 2000$ diffusion steps with a square-root base schedule. The adaptive token-wise schedule is activated after a 20K-step warm-up and recalibrated every 20K steps thereafter. Architecture, optimization, schedule, mixed-space, and inference details are provided in App. F.

Overall Performance Across Molecular Translation Tasks

Molecule captioning. Table 2 shows strong performance across both surface and semantic metrics. Under the same evaluation protocol, CAMS-Diff improves BLEU from 0.557 for our reproduced BiMol-Diff baseline to 0.610, also exceeding the published BiMol-Diff score of 0.567. CAMS-Diff further achieves the strongest METEOR and BERTScore-F1 scores, reaching 0.740 and 0.861, respectively. MAUVE is 0.927 ± 0.016, comparable to the published BiMol-Diff score of 0.925, while chrF++ remains competitive at 0.705 versus the strongest score of 0.734. Relative to our reproduced BiMol-Diff baseline, CAMS-Diff improves four of the five captioning metrics.

Text-guided molecule generation. On ChEBI-20, CAMS-Diff achieves the strongest molecular similarity among the compared methods, reaching 0.916 MACCS, 0.811 RDK, and 0.790 Morgan similarity (Table 3), while also reaching 0.832 BLEU. Against BiMol-Diff*, CAMS-Diff improves

Table 2: Molecule captioning results on M3-20M. BiMol-Diff† denotes our reproduced baseline; reproduced results and CAMS-Diff are reported as mean ± std over three seeds.

		Autoregressive		Diffusion				Ours
	Metric	Text+Chem T5	GraphT5	DiffuSeq	TGM-DLM	BiMol-Diff	BiMol-Diff†	CAMS-Diff
	#P ↓	223M	272M	91M	125M	**63M**	**63M**	91M
SM	BLEU ↑	0.542	0.481	0.532	0.467	0.567	$0.557_{\pm 0.006}$	**0.610**$_{\pm 0.004}$
SM	chrF++ ↑	0.701	0.692	0.708	0.689	**0.734**	$0.710_{\pm 0.004}$	$0.705_{\pm 0.002}$
SM	METEOR ↑	0.648	0.545	0.601	0.589	0.626	0.632	**0.740**$_{\pm 0.001}$
EM	BERTScore-F1 ↑	0.728	0.810	0.812	0.779	0.843	$0.838_{\pm 0.002}$	**0.861**$_{\pm 0.003}$
EM	MAUVE ↑	0.866	0.913	0.887	0.856	0.925	$0.913_{\pm 0.011}$	**0.927**$_{\pm 0.016}$

Table 3: Text-guided molecule generation on ChEBI-20. TGM-DLM◇ denotes the variant without correction, BiMol-Diff⋆ our reproduced baseline, and UTGDiff† the non-pretrained reproduced setting. All results are reported as mean ± std over three seeds.

		Reference	Autoregressive			Diffusion				Ours
	Metric	GT	MolT5 Small	MolT5 Base	MolT5 Large	3M-Diffusion	UTGDiff †	BiMol-Diff⋆	TGM-DLM◇	CAMS-Diff
	#P ↓	-	77M	250M	800M	162M	125M	180M	125M	180M
MM	MACCS ↑	1.000	0.703	0.721	0.834	0.557	0.867	$0.883_{\pm 0.001}$	0.874	**0.916**$_{\pm 0.003}$
MM	RDK ↑	1.000	0.568	0.588	0.746	0.380	0.763	$0.785_{\pm 0.005}$	0.771	**0.811**$_{\pm 0.005}$
MM	Morgan ↑	1.000	0.517	0.529	0.684	0.302	0.695	$0.758_{\pm 0.004}$	0.722	**0.790**$_{\pm 0.005}$
S	BLEU ↑	1.000	0.755	0.769	**0.854**	0.507	0.817	$0.820_{\pm 0.003}$	0.828	$0.832_{\pm 0.003}$
S	Exact ↑	1.000	0.079	0.081	**0.311**	0.003	0.227	$0.242_{\pm 0.004}$	0.242	$0.232_{\pm 0.006}$
S	Valid ↑	1.000	0.721	0.772	**0.905**	0.595	0.812	$0.812_{\pm 0.004}$	0.789	$0.748_{\pm 0.005}$

all three molecular similarity metrics and BLEU while achieving comparable exact match (0.232 vs. 0.242). Its main limitation remains validity (0.748 vs. 0.812). The dominant invalid outputs involve paired SMILES constraints: most commonly unpaired ring labels (43.95%) and unclosed branches (41.35%), whereas invalid atomic valence is comparatively rare (4.87%); categories are not mutually exclusive. Thus, the validity gap is driven primarily by serialization failures rather than atom-level valence violations. CAMS-Diff also improves molecular similarity over UTGDiff†. Aggregate exact match further masks a strong difficulty-dependent pattern. As instruction length increases, CAMS-Diff separates increasingly from UTGDiff† (Fig. 4c), with gains of 13.7, 16.3, and 20.4 percentage points in the 65–96, 97–128, and 129–160 token regimes, respectively. Full statistics and difficulty analyses are in App. G.

Table 4: Retrosynthesis and forward reaction prediction performance across various baselines.

			Retrosynthesis					Forward Reaction				
			S		MM			S		MM		
Type	Model	#P	Exact ↑	Valid ↑	MACCS ↑	RDK ↑	Morgan ↑	Exact ↑	Valid ↑	MACCS ↑	RDK ↑	Morgan ↑
Auto-Regressive	GALACTICA	6.7B	0.000	0.986	0.274	0.167	0.134	0.000	0.995	0.127	0.036	0.051
Auto-Regressive	Mol-Instructions	7B	0.009	**1.000**	0.487	0.283	0.230	0.045	**1.000**	0.509	0.313	0.262
Auto-Regressive	Llama-7b (LoRA)	7B	0.000	**1.000**	0.294	0.136	0.106	0.012	**1.000**	0.649	0.499	0.407
Auto-Regressive	InstructMol-G	6.9B	0.114	**1.000**	0.523	0.422	0.285	0.114	**1.000**	0.717	0.519	0.457
Auto-Regressive	InstructMol-GS	6.9B	0.407	**1.000**	0.852	0.753	0.714	0.407	**1.000**	0.878	0.776	0.741
Auto-Regressive	TEXT+CHEM T5	223M	0.141	0.698	0.765	0.685	0.585	0.141	0.698	0.789	0.705	0.652
Auto-Regressive	BioT5	252M	0.480	**1.000**	0.904	0.843	0.810	0.684	**1.000**	0.954	0.907	0.890
Diffusion	UTGDiff*	**125M**	0.4620	0.8990	0.8911	0.8256	0.7889	0.8280	0.9420	0.9783	0.9571	0.9482
Diffusion	CAMS-Diff	180M	**0.5410**	0.8780	**0.9309**	**0.8944**	**0.8725**	**0.8300**	0.9210	**0.9850**	**0.9646**	**0.9673**

All results are reported as the mean over 3 runs; standard deviations are omitted due to space constraints.

Molecular reaction prediction. Table 4 extends CAMS-Diff to both directions of molecule-to-molecule translation. For retrosynthesis, CAMS-Diff improves exact match from 0.462 to 0.541 over our reproduced UTGDiff baseline, while also improving MACCS, RDK, and Morgan similarity. For forward reaction prediction, it achieves the strongest fingerprint similarities and exact-match score, while UTGDiff retains slightly higher validity. Difficulty-stratified results sharpen this pat-

Table 5: Component and coupling ablation across molecule captioning and molecule generation. Results are averaged over three seeds and reported as mean ± std.

Variant	Molecule Captioning		Molecule Generation			
	SM	EM	MM		S	
	BLEU ↑	BERT-F1 ↑	MACCS ↑	Morgan ↑	Exact ↑	Valid ↑
Uniform	$0.523_{\pm 0.003}$	$0.780_{\pm 0.007}$	$0.874_{\pm 0.004}$	$0.722_{\pm 0.006}$	$0.221_{\pm 0.001}$	$0.732_{\pm 0.003}$
+ Mixed Space	$0.557_{\pm 0.005}$	$0.821_{\pm 0.006}$	$0.870_{\pm 0.003}$	$0.748_{\pm 0.003}$	$0.220_{\pm 0.002}$	$0.743_{\pm 0.005}$
+ Adaptive Noising	$0.575_{\pm 0.004}$	$0.830_{\pm 0.005}$	$0.889_{\pm 0.002}$	$0.750_{\pm 0.004}$	$0.222_{\pm 0.003}$	$0.740_{\pm 0.003}$
Uncoupled	$0.597_{\pm 0.005}$	$0.841_{\pm 0.003}$	$0.909_{\pm 0.003}$	$0.768_{\pm 0.006}$	$0.226_{\pm 0.004}$	$0.745_{\pm 0.001}$
Coupled (CAMS-Diff)	$\mathbf{0.610}_{\pm 0.004}$	$\mathbf{0.861}_{\pm 0.005}$	$\mathbf{0.916}_{\pm 0.002}$	$\mathbf{0.790}_{\pm 0.004}$	$\mathbf{0.232}_{\pm 0.002}$	$\mathbf{0.748}_{\pm 0.004}$

tern: in the longest-sequence regime, exact match is 0.73 versus 0.56 for forward prediction and 0.51 versus 0.33 for retrosynthesis (Appendix I).

L1: Does Coupled Adaptive Mixed-Space Training Help?

To examine **L1**, we compare uniform noising, mixed-space masking, adaptive Gaussian noising, their uncoupled combination, and coupled CAMS-Diff. Table 5 reports controlled ablations for molecule captioning and text-guided generation. *Uncoupled* uses adaptive noising with a fixed masking probability matched to CAMS-Diff's overall masking rate, but independent of $\bar{\alpha}_t^i$; coupled CAMS-Diff instead lets the learned token-wise schedule determine the masking probability. For captioning, masking and adaptive noising individually improve BLEU from 0.523 to 0.557 and 0.575, while their uncoupled combination reaches 0.597. Coupling further raises BLEU to 0.610 and BERTScore-F1 from 0.841 to 0.861. The same benefit appears for generation: relative to the uncoupled variant, CAMS-Diff improves MACCS from 0.909 to 0.916 and Morgan similarity from 0.768 to 0.790. Figure 4a further varies the coupling strength η under full $T = 2000$ decoding. The common $\eta = 0.5$ setting lies in the strongest MACCS region across generation and both reaction directions, while performance changes smoothly as masking strength varies. Adaptive noising and masking are therefore complementary, and coupling them to a shared schedule adds further gains over applying them independently.

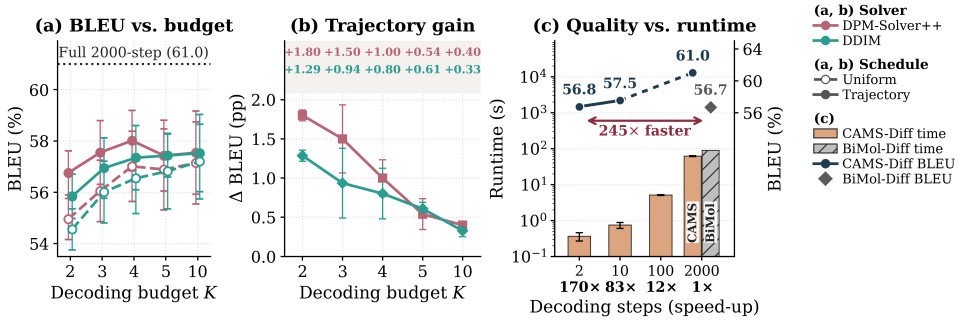


Figure 3: **Trajectory-informed reduced-step captioning.** (a) BLEU at matched budgets under uniform and trajectory-informed DPM-Solver++/DDIM. (b) Gain from trajectory-informed placement. (c) Quality-runtime trade-off.

L2: Does Trajectory-Informed Scheduling Improve Reduced-Step Decoding?

To examine **L2**, we compare uniform and trajectory-informed scheduling under matched reduced decoding budgets. Both use the same trained CAMS-Diff checkpoint, sampler, and number of denoiser evaluations K; they differ only in where those evaluations are placed along the reverse trajectory. We evaluate this comparison with DPM-Solver++ and DDIM across all four translation tasks. Figure 3 shows representative captioning results; the corresponding quality–runtime panels for generation, forward prediction, and retrosynthesis are given in Appendix C.3. Trajectory-informed placement improves BLEU over uniform scheduling at every evaluated budget, with the largest DPM-Solver++ gain of 1.8 points at $K = 2$. Two-step CAMS-Diff reaches 56.8 BLEU, exceeding our reproduced full-step BiMol-Diff result of 55.7 and remaining comparable to the published 56.7. Under matched single-A100 full-model timing, it requires 0.36 s for 50 samples: **170**× faster than full-step CAMS-Diff and approximately **245**× faster than reproduced full-step BiMol-Diff (89.0 s) (Table 14). The matched-budget advantage extends to molecular-output tasks: trajectory-informed scheduling improves generation MACCS by up to 1.81 points, retrosynthesis validity by up to 2.70 percentage

points, and forward prediction by up to 2.00 points, with per-task speedups reported alongside. At $K = 1$, only one denoiser evaluation is available, so there is no timestep-allocation freedom. Figure 4b tests whether the gain is explained by a fixed preference for the beginning, middle, or end of the reverse process. Trajectory-informed allocation improves over uniform generation by 0.20, 1.81, 0.67, and 1.13 MACCS points at $K = 2, 3, 4$, and 9, while no fixed-region allocation consistently matches it across budgets. Thus, reducing K provides the computational saving, whereas trajectory-informed placement improves the quality obtained from the same denoising budget.

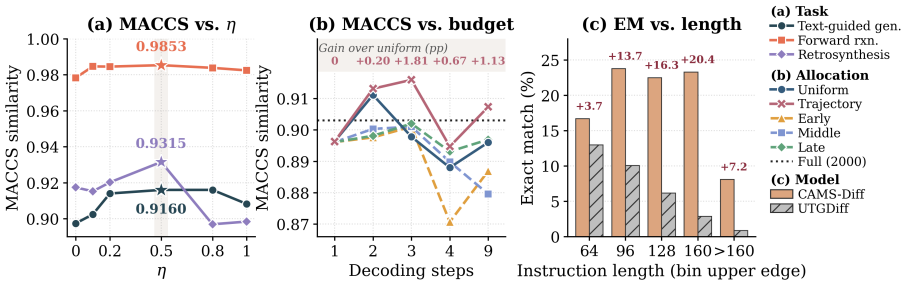


Figure 4: **Robustness diagnostics.** (a) MACCS sensitivity to η at $T = 2000$. (b) Fixed-region versus trajectory-informed step allocation. (c) Exact match across instruction-length regimes.

5.1 Zero-Shot Generalization

We finally test whether the learned molecular generation behavior transfers beyond the training distribution. The ChEBI-20-trained `CAMS-Diff` model is applied directly to PCDes without task-specific fine-tuning. As shown in Table 6, `CAMS-Diff` achieves the strongest molecular similarity among the compared methods, improving MACCS from 0.763 for UTGDiff to 0.810, RDK from 0.675 to 0.746, and Morgan similarity from 0.623 to 0.733. Exact match also improves from 0.386 to 0.427. Validity is lower than UTGDiff (0.783 vs. 0.847), indicating that the zero-shot gains are concentrated in structural similarity and exact molecular recovery rather than validity. These results show that the learned adaptive diffusion behavior transfers to an unseen molecular description distribution without additional fine-tuning.

Table 6: Zero-shot generalization on PCDes without task-specific fine-tuning. Subscripted values denote standard deviation over three runs for reproduced diffusion models.

		Sequence AR		Sequence Diffusion	Graph Diffusion		Ours
	Metric	**MolT5-base**	**BioT5-base**	**TGM-DLM**	**3M-Diffusion**	**UTGDiff**	**CAMS-Diff**
	#P ↓	250M	252M	180M	162M	**125M**	180M
MM	**MACCS ↑**	0.733	0.737	0.741	0.495	$0.763_{\pm 0.004}$	**0.810**$_{\pm 0.005}$
MM	**RDK ↑**	0.651	0.646	0.667	0.332	$0.675_{\pm 0.002}$	**0.746**$_{\pm 0.003}$
MM	**Morgan ↑**	0.621	0.595	0.612	0.242	$0.623_{\pm 0.006}$	**0.733**$_{\pm 0.005}$
S	**Exact ↑**	0.204	0.242	0.221	0.015	$0.386_{\pm 0.006}$	**0.427**$_{\pm 0.004}$
S	**Valid ↑**	0.761	**1.000**	0.765	**1.000**	$0.847_{\pm 0.002}$	$0.783_{\pm 0.003}$

6 Conclusion and Limitations

We introduced `CAMS-Diff` to address two limitations of molecular diffusion: (1) local token recoverability under continuous corruption can reduce reliance on sequence context and conditioning, and (2) uniform timestep reduction can skip informative regions of the reverse trajectory. `CAMS-Diff` couples token-wise adaptive Gaussian noising with discrete representation masking under the same learned schedule, selectively removing direct local token evidence. For reduced-step inference, denoising dynamics identify informative trajectory regions and guide timestep allocation under a fixed budget. Together, these mechanisms link adaptive corruption with trajectory-informed step allocation. Across molecular translation and zero-shot transfer, `CAMS-Diff` improves molecular and semantic recovery, while two-step captioning remains comparable to full-step BiMol-Diff with an approximately $245\times$ speedup. These gains come with few limitations. Our study is limited to sequence-based SMILES representations, with validity errors dominated by ring-closure and branch constraints, motivating corruption mechanisms that treat coupled SMILES constraints as structured units. Reduced-step schedules are validation-derived, and the allocation remains a position-wise local approximation rather than a globally optimal shared schedule.

AI USE STATEMENT

Generative AI tools were used solely to assist with drafting and polishing portions of the manuscript for clarity, readability, and presentation. They were not used to develop the research methodology, formulate hypotheses or mathematical claims, design or conduct experiments, implement the proposed methods, analyze data, or interpret experimental results. All AI-assisted text was reviewed and edited by the authors, who take full responsibility for the final content, claims, and conclusions of this work.

REPRODUCIBILITY STATEMENT

We provide an anonymized implementation of CAMS-Diff, including training, evaluation, and reduced-step inference code, at `https://anonymous.4open.science/r/cams-diff-32D7`. The main methodological formulation and algorithms are described in Sections 3-4, with complete derivations and implementation details provided in the appendices. Dataset construction, preprocessing, evaluation protocols, baseline settings, hyperparameters, and training configurations are documented in Appendices D-F. Additional ablations and analyses of masking strength, generation difficulty, denoising trajectories, and reduced-step allocation are reported in the remaining appendices. Together with the released code and configuration files, these details are intended to support reproduction of the reported results.

REFERENCES

Md Atik Ahamed, Qiang Ye, and Qiang Cheng. Mol-cadiff: text-conditional molecule generation via causality-aware autoregressive diffusion. *Nature Communications*, 17(1): 9190, 2026. doi: 10.1038/s41467-026-75702-5. URL `https://doi.org/10.1038/s41467-026-75702-5`.

Jinho Chang and Jong Chul Ye. LDMol: A text-to-molecule diffusion model with structurally informative latent space surpasses AR models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 7374–7392. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/chang25d.html`.

Jiaao Chen, Aston Zhang, Mu Li, Alex Smola, and Diyi Yang. A cheaper and better diffusion language model with soft-masked noise. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4765–4775, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.289. URL `https://aclanthology.org/2023.emnlp-main.289/`.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. Unifying molecular and textual representations via multi-task language modelling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6140–6157. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/christofidellis23a.html`.

Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2Mol: Cross-modal molecule retrieval with natural language queries. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 595–607, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.47. URL `https://aclanthology.org/2021.emnlp-main.47/`.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. In Yoav Goldberg, Zornitsa Kozareva, and

Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 375–413, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.26. URL https://aclanthology.org/2022.emnlp-main.26/.

Yi Fang, Yuan Liu, Zeyu Duan, Yu-Xuan Chen, Xiaoyong Pan, and Hong-Bin Shen. Condition controllable generation of 3d molecules using textual prompts by multimodal equivariant diffusion model. *npj Digital Medicine*, 2026. doi: 10.1038/s41746-026-03161-9. URL https://doi.org/10.1038/s41746-026-03161-9.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Tlsdsb6l9n.

Haisong Gong, Qiang Liu, Shu Wu, and Liang Wang. 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, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i1.27761. URL https://doi.org/10.1609/aaai.v38i1.27761.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq-v2: Bridging discrete and continuous text spaces for accelerated Seq2Seq diffusion models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9868–9875, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.660. URL https://aclanthology.org/2023.findings-emnlp.660/.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=jQj-_rLVXsj.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Craig A. James, Richard Apodaca, Noel O'Boyle, Andrew Dalke, John van Drie, Peter Ertl, Geoffrey Hutchison, Greg Landrum, Chris Morley, Egon Willighagen, Hans De Winter, Tim Vandermeersch, and John May. Opensmiles specification. https://opensmiles.org/opensmiles.html, 2016. Version 1.0, 2016-05-15.

Akhil Jindal and Harang Ju. Smollm: Small language models learn small molecular grammar, 2026. URL https://arxiv.org/abs/2605.06322.

Sangyeup Kim, Nayeon Kim, Yinhua Piao, and Sun Kim. Grapht5: Unified molecular graph-language modeling via multi-modal cross-token attention, 2025. URL https://arxiv.org/abs/2503.07655.

Seonghwan Kim, Jeheon Woo, and Woo Youn Kim. Diffusion-based generative ai for exploring transition states from 2d molecular graphs. *Nature Communications*, 15:341, 2024. doi: 10.1038/s41467-023-44629-6. URL https://doi.org/10.1038/s41467-023-44629-6.

Lingxiao Li, Haobo Zhang, Bin Chen, and Jiayu Zhou. Chain-of-generation: Progressive latent diffusion for text-guided molecular design, 2025. URL https://arxiv.org/abs/2511.11894.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Yanchen Luo, Junfeng Fang, Sihang Li, Zhiyuan Liu, Jiancan Wu, An Zhang, Wenjie Du, and Xiang Wang. Text-guided small molecule generation via diffusion model. *iScience*, 27(11): 110992, 2024. doi: 10.1016/j.isci.2024.110992. URL https://doi.org/10.1016/j.isci.2024.110992.

Nguyen Doan Hieu Nguyen, Nhat Truong Pham, Duong Thanh Tran, and Balachandran Manavalan. Lang2Mol-diff: A diffusion-based generative model for language-to-molecule translation leveraging SELFIES representation. In Carl Edwards, Qingyun Wang, Manling Li, Lawrence Zhao, Tom Hope, and Heng Ji (eds.), *Proceedings of the 1st Workshop on Language + Molecules (L+M 2024)*, pp. 128–134, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.langmol-1.15. URL https://aclanthology.org/2024.langmol-1.15/.

Rıza Özçelik, Sarah de Ruiter, Emanuele Criscuolo, and Francesca Grisoni. Chemical language modeling with structured state space sequence models. *Nature Communications*, 15(1):6176, 2024. doi: 10.1038/s41467-024-50469-9. URL https://www.nature.com/articles/s41467-024-50469-9.

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. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1102–1123, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.70. URL https://aclanthology.org/2023.emnlp-main.70/.

Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. BioT5+: Towards generalized biological understanding with IUPAC integration and multi-task tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1216–1240, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.71. URL https://aclanthology.org/2024.findings-acl.71/.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42947–42975. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sabour24a.html.

Aditya Hemant Shahane, Anuj Kumar Sirohi, Devansh Arora, Nitin Kumar, Prathosh AP, and Sandeep Kumar. BiMol-diff: A unified diffusion framework for molecular generation and captioning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23104–23117, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1059. URL https://aclanthology.org/2026.acl-long.1059/.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Qian Tan, Xuanyu Zhu, Lei Jiang, Zhonghang Yuan, Chen Zhang, and Yuqiang Li. Molgvr: A chemistry-grounded framework for text-to-molecule generation, 2026. URL https://arxiv.org/abs/2607.29479.

Ping Tuo, Jiale Chen, and Ju Li. Flow matching for reaction pathway generation. *Nature Communications*, 17:8769, 2026. doi: 10.1038/s41467-026-75654-w. URL https://doi.org/10.1038/s41467-026-75654-w.

Zixu Wang, Yangyang Chen, Xiulan Guo, Yayang Li, Pengyong Li, Chunyan Li, Xiucai Ye, and Tetsuya Sakurai. DiffSeqMol: A non-autoregressive diffusion-based approach for molecular sequence generation and optimization. *Current Bioinformatics*, 20(1):46–58, 2025. doi: 10.2174/0115748936285493240307071916. URL https://doi.org/10.2174/0115748936285493240307071916.

Xinyuan Wei, Zian Li, Shaoheng Yan, Cai Zhou, and Muhan Zhang. Flashmol: High-quality molecule generation in as few as four steps, 2026. URL https://arxiv.org/abs/2605.07020.

Wenchao Weng, Hanyu Jiang, Xiangjie Kong, and Giovanni Pau. Text-guided diverse-expression diffusion model for molecule generation. *Chinese Physics B*, 34(5):050701, 2025. doi: 10.1088/1674-1056/adbedd. URL https://doi.org/10.1088/1674-1056/adbedd.

Yuran Xiang, Haiteng Zhao, Chang Ma, and Zhi-Hong Deng. Instruction-based molecular graph generation with unified text-graph diffusion model, 2025. URL https://arxiv.org/abs/2408.09896.

Yang Yao, Xin Wang, Yaofei Wu, Zeyang Zhang, Daixin Wang, Zhiqiang Zhang, Hong Mei, and Wenwu Zhu. Text-guided molecule generation with conditional discrete graph diffusion model. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 12667–12678, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3818981. URL https://doi.org/10.1145/3770855.3818981.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Text diffusion model with encoder-decoder transformers for sequence-to-sequence generation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 22–39, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.2. URL https://aclanthology.org/2024.naacl-long.2/.

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. doi: 10.1038/s41467-022-28494-3.

Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. 3m-diffusion: Latent multi-modal diffusion for language-guided molecular structure generation, 2024. URL https://arxiv.org/abs/2403.07179.

A Expanded Related Work and Positioning

This section expands the compact comparison in Table 1. We distinguish three lines of work relevant to CAMS-Diff: molecular translation and text-conditioned molecular generation, adaptive and mixed-space sequence diffusion, and accelerated diffusion inference. Table 7 summarizes the resulting methodological positioning.

Molecular translation and text-conditioned molecular generation. Molecular translation connects natural-language and molecular representations through conditional generation. MolT5 (Edwards et al., 2022) introduced molecule captioning and text-guided molecule generation as paired molecule–language translation tasks using pretrained sequence-to-sequence models. Subsequent molecular language models, including BioT5/BioT5+ (Pei et al., 2023; 2024), expanded this paradigm through chemistry-aware pretraining and broader biological and reaction-related tasks. These methods operate autoregressively and therefore do not expose an explicit diffusion corruption process or reverse denoising trajectory.

Diffusion provides a non-autoregressive alternative in which molecular outputs are reconstructed iteratively. DiffSeqMol (Wang et al., 2025) studies diffusion-based molecular sequence generation and optimization, generating the complete molecular sequence non-autoregressively rather than token by token. TGM-DLM (Gong et al., 2024) applies continuous diffusion to text-guided SMILES generation and uses a two-stage generation/correction procedure. BiMol-Diff (Shahane et al., 2026) extends continuous sequence diffusion bidirectionally to molecule captioning and text-guided generation. Importantly for our setting, BiMol-Diff introduces token-wise adaptive Gaussian noising based on recovery difficulty, allowing different target positions to follow different corruption trajectories. CAMS-Diff builds on this adaptive continuous formulation, but differs in two respects: it couples the learned Gaussian corruption dynamics to an additional representation-masking process during training, and it uses post-training reverse-trajectory dynamics to determine reduced-step inference schedules.

A complementary direction changes the molecular representation or the latent space on which diffusion operates. Lang2Mol-Diff (Nguyen et al., 2024) uses SELFIES for language-to-molecule diffusion, directly targeting the validity limitations associated with SMILES sequence generation. TGDD (Weng et al., 2025) jointly uses SMILES and SELFIES through a diverse-expression molecular representation and performs segmented diffusion over the resulting sequence. LDMol (Chang & Ye, 2025) instead performs text-conditioned diffusion in a learned structurally informative molecular latent space, using representation learning to preserve molecular information before diffusion. These approaches address structural consistency primarily through the choice or learning of the molecular representation. CAMS-Diff instead retains a SMILES-based sequence formulation and modifies how information is removed during corruption by coupling Gaussian noising with representation masking. These directions are complementary: alternative representations such as SELFIES or learned molecular latents may also reduce the serialization failures that remain a limitation of our SMILES-based setting.

Recent methods also modify the conditioning or generation architecture rather than the molecular representation alone. Mol-CADiff (Ahamed et al., 2026) introduces causality-aware text-conditional molecular generation and explicitly models dependencies within and across textual and molecular modalities. MolGVR (Tan et al., 2026) adopts a Generator-Verifier-Refiner formulation, extracting chemical constraints from the description and using explicit verification and feedback-guided refinement to improve text-to-molecule generation. Chain-of-Generation (Li et al., 2025) takes a different latent-diffusion approach, decomposing complex prompts into progressively introduced semantic conditions during generation. These methods illustrate that text–molecule alignment and chemical correctness can be improved through stronger conditioning, verification, or progressive guidance. CAMS-Diff instead studies whether the corruption process and reverse sampling trajectory themselves can be adapted while leaving the underlying conditional sequence-diffusion formulation unchanged.

Text-conditioned molecular diffusion has additionally been studied directly in three-dimensional molecular space. TextSMOG (Luo et al., 2024) combines language conditioning with an equivariant molecular diffusion model and uses a text-derived reference geometry to guide generation of 3D molecular structures. TDMol (Fang et al., 2026) similarly studies condition-controllable 3D molecular generation using textual prompts and multimodal equivariant diffusion. These works target geometrical molecular construction rather than SMILES-level molecular translation, and are therefore complementary rather than interchangeable baselines for our sequence-based tasks.

Graph-based molecular diffusion offers another complementary representation. TDGD (Yao et al., 2026) and UTGDiff (Xiang et al., 2025) operate directly over discrete molecular graphs and therefore encode molecular connectivity more explicitly than SMILES-based sequence diffusion. UTGDiff combines masked graph diffusion with a unified text–graph denoiser and supports text-guided molecule generation as well as molecular reaction tasks. CAMS-Diff does not introduce explicit graph structural bias; instead, it asks whether a common sequence-diffusion formulation can support molecule-to-text, text-to-molecule, and molecule-to-molecule translation while adapting both the training corruption process and the retained reverse timesteps.

Generative modeling has also been extended to reaction-specific geometric objectives. TSDiff (Kim et al., 2024) uses diffusion to generate transition-state geometries from two-dimensional molecular graphs, directly modeling geometrical reaction intermediates. MolGEN (Tuo et al., 2026) instead formulates transition-state and reaction-product generation through conditional flow matching, pro-

viding a deterministic continuous-time alternative to stochastic diffusion and emphasizing efficient reaction-pathway generation. Their output spaces and evaluation protocols differ from the SMILES-level forward reaction prediction and retrosynthesis considered here, so they are not direct experimental baselines, but they situate CAMS-Diff within the broader development of non-autoregressive generative modeling for chemical reactions.

Mixed and adaptive sequence diffusion. Continuous diffusion is attractive for sequence generation because standard Gaussian diffusion machinery can be applied to token embeddings (Li et al., 2022). Its continuous representations, however, may retain considerable local token information at low noise levels. Several text-diffusion methods therefore introduce discrete-like or masking-based corruption. Masked-Diffuse LM (Chen et al., 2023) uses soft-masked noise to better connect continuous diffusion with the discrete nature of language. DiffuSeq-v2 (Gong et al., 2023a) similarly introduces a soft absorbing state that bridges discrete mutations and an underlying Gaussian space, while also using continuous-space solvers for accelerated sampling. These works establish important precedents for combining continuous and masking-based corruption.

CAMS-Diff therefore does *not* claim that mixing Gaussian and masking corruption is itself new. The distinction is in how the two corruption modes are related. Rather than specifying the masking process independently, CAMS-Diff sets its token-wise masking probability from the same learned corruption coefficient that governs adaptive Gaussian noising. Consequently, when the token-wise Gaussian schedule is recalibrated, the discrete masking process is recalibrated with it. The controlled comparison between masking alone, adaptive noising alone, their uncoupled combination, and CAMS-Diff directly tests whether this coupling contributes beyond simply using both corruption types.

Adaptive noise scheduling is itself also established in sequence diffusion. SeqDiffuSeq (Yuan et al., 2024) introduces an adaptive noise schedule that balances token-level denoising difficulty across diffusion steps, while BiMol-Diff adapts position-wise molecular corruption according to token recovery difficulty. CAMS-Diff inherits the general idea of token-wise adaptive continuous noising rather than claiming it as a new contribution. Our training-time contribution is instead the *coupling* of that learned schedule to the representation-masking process.

Few-step diffusion and timestep selection. A separate line of work accelerates diffusion by reducing the number of denoiser evaluations. Such methods include faster numerical solvers, distillation, and timestep respacing. In molecular generation, FlashMol (Wei et al., 2026) demonstrates that high-quality 3D molecular conformations can be generated in as few as four steps using distribution-matching distillation together with timestep respacing. Its setting is 3D molecular generation rather than conditional molecular translation, but it demonstrates the importance of aggressive few-step inference in the molecular domain.

Reaction-oriented generative models provide another perspective on inference efficiency. MolGEN (Tuo et al., 2026) uses conditional flow matching for transition-state and reaction-product generation and reports sub-second sampling for transition-state generation. Although flow matching changes the underlying generative dynamics rather than selecting steps from a trained diffusion trajectory, it reinforces the broader motivation for reducing the cost of iterative molecular generation.

Reducing the number of denoising evaluations and choosing *where* those evaluations occur are distinct problems. Align Your Steps (Sabour et al., 2024) studies the latter directly, optimizing sampling schedules for trained diffusion models using a principled noise-level formulation and demonstrating large benefits in few-step image and video generation. CAMS-Diff is related in motivation but differs in both signal and setting. We derive timestep importance from the observed clean-state reconstruction trajectory of the trained molecular translation model in token-wise log-SNR space. The resulting position-wise allocation profiles are mapped to the shared timestep axis and aggregated into one schedule that is constructed once from validation trajectories and reused at test time.

Thus, CAMS-Diff does not claim that non-uniform sampling schedules are new in general. Its inference contribution is a molecular-translation-specific trajectory criterion derived from reconstruction dynamics, together with a token-wise formulation compatible with the adaptive schedules used during training. Controlled experiments compare this schedule against uniform subsampling under the same trained checkpoint, sampler, and denoising budget, while additional fixed-region allocations are reported in Appendix H.

Table 7: **Detailed positioning of CAMS-Diff relative to representative molecular and sequence-diffusion methods.** AN: adaptive noising; MC: mixed continuous/discrete corruption; FS: few-step inference; SA: adaptive step allocation. "Training schedule / signal" denotes the corruption schedule or quantity used to adapt it, while "Inference strategy" summarizes the decoding or sampling procedure used at test time.

Type	Method	Space	AN	MC	FS	SA	Training schedule / signal	Inference strategy	Scope
AR	MolT5 (Edwards et al., 2022)	Tok.	✗	✗	✗	✗	AR training	Autoregressive decoding	Cap., T2M
	BioT5+ (Pei et al., 2024)	Tok.	✗	✗	✗	✗	AR training	Autoregressive decoding	Cap., T2M, Fwd., Retro.
SD	DiffSeqMol (Wang et al., 2025)	Mol. seq.	✗	✗	✗	✗	Fixed Gaussian diffusion	Iterative sequence denoising	Mol. gen./opt.
	Lang2Mol-Diff (Nguyen et al., 2024)	SELFIES	✗	✗	✗	✗	Fixed diffusion	Iterative SELFIES denoising	T2M
	TGDD (Weng et al., 2025)	SMILES/SELFIES	✗	✗	✗	✗	Segmented diffusion	Iterative sequence denoising	T2M
	Masked-Diffuse LM (Chen et al., 2023)	Cont./Mask	✗	✓	✗	✗	Fixed soft-mask schedule	Iterative denoising	Text
	DiffuSeq-v2 (Gong et al., 2023a)	Cont./Abs.	✗	✓	✓	✗	Fixed absorbing-state schedule	Continuous-space solver	Seq2Seq text
	SeqDiffuSeq (Yuan et al., 2024)	Cont.	✓	✗	✗	✗	Token denoising difficulty	Iterative denoising	Seq2Seq text
	TGM-DLM (Gong et al., 2024)	Cont.	✗	✗	✗	✗	Uniform Gaussian schedule	Generation + correction	T2M
	BiMol-Diff (Shahane et al., 2026)	Cont.	✓	✗	✗	✗	Token recovery difficulty	Iterative denoising	Cap., T2M
LD	LDMol (Chang & Ye, 2025)	Mol. latent	✗	✗	✗	✗	Latent diffusion	Latent-space denoising	T2M / editing
	Chain-of-Generation (Li et al., 2025)	Mol. latent	✗	✗	✗	✗	Progressive conditioning	Progressive latent generation	T2M
GD	TDGD (Yao et al., 2026)	Disc. graph	✗	✗	✗	✗	Discrete graph corruption	Graph denoising	T2M
	UTGDiff (Xiang et al., 2025)	Disc. graph	✗	✗	✓	✗	Masked graph corruption	Low-step graph denoising	T2M, Fwd., Retro., ZS
	Mol-CADiff (Ahamed et al., 2026)	Graph latent	✗	✗	✗	✗	Causality-aware diffusion	Conditional graph denoising	T2M
3D	TextSMOG (Luo et al., 2024)	3D geom.	✗	✗	✗	✗	Equivariant diffusion	3D equivariant denoising	Text→3D
	TDMol (Fang et al., 2026)	2D/3D geom.	✗	✗	✗	✗	Multimodal equivariant diffusion	3D equivariant denoising	Text→3D
	FlashMol (Wei et al., 2026)	3D latent	✗	✗	✓	✗	Distillation	Respaced few-step sampling	3D generation
Reaction	TSDiff (Kim et al., 2024)	3D geom.	✗	✗	✗	✗	Geometric diffusion	Transition-state denoising	Transition states
	MolGEN (Tuo et al., 2026)	Cont. flow	✗	✗	✓	✗	Conditional flow matching	ODE-based flow integration	TS / products
Sched.	Align Your Steps (Sabour et al., 2024)	Cont.	✗	✗	✓	✓	Fixed trained diffusion	Optimized noise-level schedule	Image / video
D	**CAMS-Diff**	**Mixed seq.**	✓	✓	✓	✓	**Token recovery difficulty + coupled masking**	**Trajectory-informed step allocation**	**Cap., T2M, Fwd., Retro., ZS**

Positioning of CAMS-Diff.

The closest prior ideas therefore address individual parts of the framework: sequence and latent molecular diffusion provide non-autoregressive text-to-molecule generation; SELFIES and multi-representation methods alter the molecular serialization; graph and 3D diffusion incorporate explicit structural or geometric information; adaptive Gaussian noising changes the forward corruption schedule; mixed-space text diffusion introduces masking or absorbing states; and accelerated diffusion reduces or redistributes reverse-process evaluations. CAMS-Diff connects two adaptation problems within a single sequence-based molecular translation framework: learned token-wise corruption dynamics jointly govern Gaussian noising and representation masking during training, while post-training reconstruction dynamics govern which reverse timesteps are retained during reduced-step inference. The same formulation is instantiated across molecule captioning, text-guided molecule generation, forward reaction prediction, and retrosynthesis, with additional zero-shot evaluation under distribution shift.

B Diffusion Objective and Schedule Derivations

B.1 Derivation of the Token-Wise and Mixed-Space Objectives

We derive the Gaussian token-wise objective and its mixed-space extension from the conditional diffusion variational bound (Sohl-Dickstein et al., 2015; Ho et al., 2020). Let C denote the conditioning sequence, $H_C = \mathrm{Enc}(C)$ its encoded representation, and $Z_0 = g_\Phi(Y)$ the continuous embedding of the discrete target sequence Y. We condition throughout on the current position-dependent schedule $\mathcal{S}$ and omit the outer expectation over training pairs $(C, Y) \sim \mathcal{D}$ for notational clarity.

Gaussian forward and reverse processes. The token-wise Gaussian forward process factorizes across diffusion steps and target positions:

$$q_{\mathcal{S}}(Z_{1:T} \mid Z_0) = \prod_{t=1}^{T} \prod_{i=1}^{L} q_{\mathcal{S}}(z_t^i \mid z_{t-1}^i). \tag{10}$$

The corresponding conditional reverse process is

$$p_\theta(Z_{0:T} \mid C) = p(Z_T) \prod_{t=1}^{T} p_\theta\left(Z_{t-1} \mid Z_t, H_C, t, \boldsymbol{\lambda}_t\right), \tag{11}$$

where $p(Z_T) = \prod_{i=1}^{L} \mathcal{N}(z_T^i; 0, I)$. The discrete sequence is recovered from the clean latent through a separate rounding distribution $p_\psi(Y \mid Z_0)$, as commonly used in continuous diffusion language models (Li et al., 2022; Gong et al., 2023a).

Conditional variational bound. Marginalizing the forward trajectory gives

$$p_\theta(Z_0 \mid C) = \int p_\theta(Z_{0:T} \mid C)\, dZ_{1:T}. \tag{12}$$

Introducing $q_{\mathcal{S}}(Z_{1:T} \mid Z_0)$ and applying Jensen's inequality gives

$$-\log p_\theta(Z_0 \mid C) \le \mathcal{L}_{\mathrm{VLB}}^{\mathrm{cont}}, \tag{13}$$

where

$$\begin{aligned}\mathcal{L}_{\mathrm{VLB}}^{\mathrm{cont}} &= D_{\mathrm{KL}}\left(q_{\mathcal{S}}(Z_T \mid Z_0) \,\|\, p(Z_T)\right) \\ &\quad + \mathbb{E}_{q_{\mathcal{S}}(Z_{1:T}|Z_0)}\Bigg[\sum_{t=2}^{T} D_{\mathrm{KL}}\left(q_{\mathcal{S}}(Z_{t-1} \mid Z_t, Z_0) \,\|\, p_\theta\left(Z_{t-1} \mid Z_t, H_C, t, \boldsymbol{\lambda}_t\right)\right) \\ &\qquad - \log p_\theta\left(Z_0 \mid Z_1, H_C, 1, \boldsymbol{\lambda}_1\right)\Bigg].\end{aligned} \tag{14}$$

The discrete rounding likelihood is added separately because it depends only on the clean latent Z_0:

$$\mathcal{L}_{\mathrm{VLB}} = \mathcal{L}_{\mathrm{VLB}}^{\mathrm{cont}} - \log p_\psi(Y \mid Z_0). \tag{15}$$

Intermediate denoising terms. For $t \ge 2$, the exact forward posterior factorizes across positions:

$$q_{\mathcal{S}}(Z_{t-1} \mid Z_t, Z_0) = \prod_{i=1}^{L} \mathcal{N}\left(z_{t-1}^i; \widetilde{\mu}_t^i, \widetilde{\beta}_t^i I\right), \tag{16}$$

where $\widetilde{\mu}_t^i = c_t^i z_0^i + d_t^i z_t^i$, with

$$c_t^i = \frac{\sqrt{\bar{\alpha}_{t-1}^i}\,\beta_t^i}{1-\bar{\alpha}_t^i}, \qquad d_t^i = \frac{\sqrt{\alpha_t^i}(1-\bar{\alpha}_{t-1}^i)}{1-\bar{\alpha}_t^i}, \tag{17}$$

and

$$\widetilde{\beta}_t^i = \frac{1-\bar{\alpha}_{t-1}^i}{1-\bar{\alpha}_t^i}\beta_t^i. \tag{18}$$

We parameterize the learned transition with the same posterior covariance:

$$p_\theta\left(z_{t-1}^i \mid Z_t, H_C, t, \boldsymbol{\lambda}_t\right) = \mathcal{N}\left(z_{t-1}^i; \mu_{\theta,t}^i, \widetilde{\beta}_t^i I\right), \tag{19}$$

where

$$\mu_{\theta,t}^i = c_t^i \widehat{z}_{0,\theta}^{i,(t)} + d_t^i z_t^i, \qquad \widehat{Z}_{0,\theta}^{(t)} = f_\theta\left(Z_t, H_C, t, \boldsymbol{\lambda}_t\right). \tag{20}$$

Although the posterior is expressed position-wise, each prediction $\widehat{z}_{0,\theta}^{i,(t)}$ may depend on the complete noisy sequence, the conditioning representation, and the full token-wise log-SNR vector. For non-degenerate steps with $\beta_t^i > 0$, the Gaussian KL is

$$\begin{aligned}\mathcal{L}_{t,i} &= D_{\mathrm{KL}}\left(q_{\mathcal{S}}(z_{t-1}^i \mid z_t^i, z_0^i) \,\|\, p_\theta\left(z_{t-1}^i \mid Z_t, H_C, t, \boldsymbol{\lambda}_t\right)\right) \\ &= \frac{1}{2\widetilde{\beta}_t^i}\left\|\widetilde{\mu}_t^i - \mu_{\theta,t}^i\right\|_2^2 \\ &= \frac{(c_t^i)^2}{2\widetilde{\beta}_t^i}\left\|z_0^i - \widehat{z}_{0,\theta}^{i,(t)}\right\|_2^2.\end{aligned} \tag{21}$$

Defining

$$w_t^i = \frac{(c_t^i)^2}{2\widetilde{\beta}_t^i}, \tag{22}$$

the intermediate denoising contribution is

$$\mathcal{L}_{\mathrm{den}} = \mathbb{E}_{q_{\mathcal{S}}(Z_{1:T}|Z_0)}\left[\sum_{t=2}^{T}\sum_{i=1}^{L} w_t^i \left\|\widehat{z}_{0,\theta}^{i,(t)} - z_0^i\right\|_2^2\right]. \tag{23}$$

The equivalent SNR-drop form, including the zero-weight convention for schedule plateaus, is derived in Appendix B.2.

Boundary reconstruction and rounding. At $t = 1$, we parameterize the continuous reconstruction likelihood using a fixed variance σ_1^2:

$$p_\theta\left(Z_0 \mid Z_1, H_C, 1, \boldsymbol{\lambda}_1\right) = \mathcal{N}\left(Z_0; \widehat{Z}_{0,\theta}^{(1)}, \sigma_1^2 I\right), \tag{24}$$

where $\widehat{Z}_{0,\theta}^{(1)} = f_\theta(Z_1, H_C, 1, \boldsymbol{\lambda}_1)$. Absorbing the fixed coefficient $1/(2\sigma_1^2)$ into the loss scale gives

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{q_{\mathcal{S}}(Z_1 \mid Z_0)}\left[\left\|Z_0 - \widehat{Z}_{0,\theta}^{(1)}\right\|_F^2\right]. \tag{25}$$

The rounding distribution factorizes across target positions as

$$p_\psi(Y \mid Z_0) = \prod_{i=1}^{L} p_\psi(y_i \mid z_0^i), \qquad p_\psi(y_i \mid z_0^i) = \text{Softmax}\left(W_\psi z_0^i + b_\psi\right)_{y_i}, \tag{26}$$

with teacher-forced loss

$$\mathcal{L}_{\text{round}} = -\log p_\psi(Y \mid Z_0) = -\sum_{i=1}^{L} \log p_\psi(y_i \mid z_0^i). \tag{27}$$

During training this term is evaluated at $Z_0 = g_\Phi(Y)$; during inference the same rounding distribution is evaluated on the predicted clean latent.

For fixed terminal signal-retention values, the terminal prior-matching term contains no denoiser parameters. Following simplified objectives commonly used in continuous diffusion language models (Li et al., 2022; Gong et al., 2023a), we omit this term and additive Gaussian constants from the practical objective. When g_Φ is trainable, omitting the prior term is a modeling choice rather than an exact constant with respect to all parameters. The resulting Gaussian objective is

$$\mathcal{L}_{\text{base}} = \mathbb{E}_{q_{\mathcal{S}}(Z_{1:T} \mid Z_0)}\left[\sum_{t=2}^{T}\sum_{i=1}^{L} w_t^i \left\|\widehat{z}_{0,\theta}^{i,(t)} - z_0^i\right\|_2^2 + \left\|Z_0 - \widehat{Z}_{0,\theta}^{(1)}\right\|_F^2\right] - \log p_\psi(Y \mid Z_0). \tag{28}$$

Mixed-space extension and cancellation of the mask factors. The mixed-space process in Sec. 4.1 introduces auxiliary mask variables $\boldsymbol{\rho}_{1:T}$. Conditioned on $\mathcal{S}$, the augmented forward process is

$$q_{\mathcal{S},\eta}(Z_{1:T}, \boldsymbol{\rho}_{1:T} \mid Z_0) = q_{\mathcal{S}}(Z_{1:T} \mid Z_0) \prod_{t=1}^{T} q_\eta(\boldsymbol{\rho}_t \mid t, \mathcal{S}), \tag{29}$$

where q_η is defined in Eq. 4. For the auxiliary-variable generative factorization, we sample the mask variables from the same fixed distribution:

$$p_{\theta,\eta}(Z_{0:T}, \boldsymbol{\rho}_{1:T} \mid C) = p(Z_T) \prod_{t=1}^{T} \left[q_\eta(\boldsymbol{\rho}_t \mid t, \mathcal{S})\, p_\theta\left(Z_{t-1} \mid Z_t, \boldsymbol{\rho}_t, H_C, t, \boldsymbol{\lambda}_t\right)\right]. \tag{30}$$

Here $\widetilde{Z}_t$ is the deterministic mixed observation defined by Eq. 3; the reverse mean in Eq. 30 is obtained from $f_\theta(\widetilde{Z}_t, \boldsymbol{\rho}_t, H_C, t, \boldsymbol{\lambda}_t)$. The mask variables are auxiliary rather than prediction targets, so their probability law is shared by the forward and generative factorizations. Consequently, the variational ratio contains

$$\frac{q_{\mathcal{S}}(Z_{1:T} \mid Z_0) \prod_{t=1}^{T} q_\eta(\boldsymbol{\rho}_t \mid t, \mathcal{S})}{p(Z_T) \prod_{t=1}^{T} q_\eta(\boldsymbol{\rho}_t \mid t, \mathcal{S})\, p_\theta(Z_{t-1} \mid Z_t, \boldsymbol{\rho}_t, H_C, t, \boldsymbol{\lambda}_t)} = \frac{q_{\mathcal{S}}(Z_{1:T} \mid Z_0)}{p(Z_T) \prod_{t=1}^{T} p_\theta(Z_{t-1} \mid Z_t, \boldsymbol{\rho}_t, H_C, t, \boldsymbol{\lambda}_t)}. \tag{31}$$

Thus the q_η probability terms cancel exactly. The expectation still remains over $q_{\mathcal{S},\eta}$ because the sampled masks change the mixed observation supplied to the denoiser.

After cancellation, the continuous mixed-space variational bound has the same Gaussian posterior terms as Eq. 14, but the learned reverse transition is conditioned on $\boldsymbol{\rho}_t$:

$$\begin{aligned}\mathcal{L}_{\text{VLB}}^{\text{mix}} = {} & D_{\text{KL}}\left(q_{\mathcal{S}}(Z_T \mid Z_0) \,\|\, p(Z_T)\right) \\ & + \mathbb{E}_{q_{\mathcal{S},\eta}}\Bigg[\sum_{t=2}^{T} D_{\text{KL}}\left(q_{\mathcal{S}}(Z_{t-1} \mid Z_t, Z_0) \,\|\, p_\theta\left(Z_{t-1} \mid Z_t, \boldsymbol{\rho}_t, H_C, t, \boldsymbol{\lambda}_t\right)\right) \\ & \qquad - \log p_\theta\left(Z_0 \mid Z_1, \boldsymbol{\rho}_1, H_C, 1, \boldsymbol{\lambda}_1\right)\Bigg],\end{aligned} \tag{32}$$

where the expectation shorthand denotes $q_{\mathcal{S},\eta}(Z_{1:T}, \boldsymbol{\rho}_{1:T} \mid Z_0)$. Using the same posterior covariance gives the same weights w_t^i, with the clean-state estimate replaced by $\widehat{Z}_{0,\theta}^{\mathrm{mix},(t)}$ from Eq. 5. Dropping the same terminal term and constants as above, and adding the rounding likelihood, yields

$$\mathcal{L}_{\mathrm{mix}} = \mathbb{E}_{q_{\mathcal{S},\eta}(Z_{1:T},\boldsymbol{\rho}_{1:T}|Z_0)}\left[\sum_{t=2}^{T}\sum_{i=1}^{L} w_t^i \left\|\widehat{z}_{0,\theta}^{i,\mathrm{mix},(t)} - z_0^i\right\|_2^2 + \left\|Z_0 - \widehat{Z}_{0,\theta}^{\mathrm{mix},(1)}\right\|_F^2\right] - \log p_\psi(Y \mid Z_0), \tag{33}$$

which is Eq. 6.

B.2 Token-Wise SNR Weighting

For a non-degenerate step with $\beta_t^i > 0$, Eq. 22 gives

$$w_t^i = \frac{(c_t^i)^2}{2\widetilde{\beta}_t^i}. \tag{34}$$

Substituting

$$c_t^i = \frac{\sqrt{\bar{\alpha}_{t-1}^i}\,\beta_t^i}{1-\bar{\alpha}_t^i}, \qquad \widetilde{\beta}_t^i = \frac{1-\bar{\alpha}_{t-1}^i}{1-\bar{\alpha}_t^i}\beta_t^i \tag{35}$$

gives

$$\begin{aligned} w_t^i &= \frac{\bar{\alpha}_{t-1}^i(\beta_t^i)^2}{2(1-\bar{\alpha}_t^i)^2}\frac{1-\bar{\alpha}_t^i}{(1-\bar{\alpha}_{t-1}^i)\beta_t^i} \\ &= \frac{\bar{\alpha}_{t-1}^i\beta_t^i}{2(1-\bar{\alpha}_{t-1}^i)(1-\bar{\alpha}_t^i)}. \end{aligned} \tag{36}$$

Using $\bar{\alpha}_t^i = \alpha_t^i\bar{\alpha}_{t-1}^i$ and $\beta_t^i = 1-\alpha_t^i$, we obtain

$$\begin{aligned} \mathrm{SNR}_{t-1}^i - \mathrm{SNR}_t^i &= \frac{\bar{\alpha}_{t-1}^i}{1-\bar{\alpha}_{t-1}^i} - \frac{\bar{\alpha}_t^i}{1-\bar{\alpha}_t^i} \\ &= \frac{\bar{\alpha}_{t-1}^i - \bar{\alpha}_t^i}{(1-\bar{\alpha}_{t-1}^i)(1-\bar{\alpha}_t^i)} \\ &= \frac{\bar{\alpha}_{t-1}^i\beta_t^i}{(1-\bar{\alpha}_{t-1}^i)(1-\bar{\alpha}_t^i)}. \end{aligned} \tag{37}$$

Therefore,

$$w_t^i = \frac{1}{2}\left(\mathrm{SNR}_{t-1}^i - \mathrm{SNR}_t^i\right). \tag{38}$$

If the projected schedule contains a plateau, $\bar{\alpha}_t^i = \bar{\alpha}_{t-1}^i$, then $\beta_t^i = 0$ and the Gaussian transition is deterministic. Equation 38 defines the corresponding denoising weight by continuity as $w_t^i = 0$.

Summing over the intermediate denoising steps gives

$$\begin{aligned} \sum_{t=2}^{T} w_t^i &= \frac{1}{2}\sum_{t=2}^{T}\left(\mathrm{SNR}_{t-1}^i - \mathrm{SNR}_t^i\right) \\ &= \frac{1}{2}\left(\mathrm{SNR}_1^i - \mathrm{SNR}_T^i\right). \end{aligned} \tag{39}$$

The first and terminal signal-retention values are shared across positions, so $\mathrm{SNR}_1^i = \mathrm{SNR}_1$ and $\mathrm{SNR}_T^i = \mathrm{SNR}_T$. Hence

$$\sum_{t=2}^{T} w_t^i = \frac{1}{2}\left(\mathrm{SNR}_1 - \mathrm{SNR}_T\right). \tag{40}$$

Thus every position receives the same total SNR-drop weight; the token-wise schedule changes only where that weight is placed along the diffusion trajectory.

B.3 Difficulty-Based Signal-Retention Calibration

Following the loss-based adaptive calibration strategy of Shahane et al. (2026); Yuan et al. (2024), the cumulative signal-retention schedule is recalibrated using position-wise clean-latent prediction difficulty. The reference schedule is used during the initial training block; thereafter the token-wise schedule is updated every T_{up} iterations and held fixed between consecutive updates.

Difficulty estimation. At calibration update k, consider a minibatch of size B evaluated at timestep t. The mean clean-latent prediction error at target position i is

$$\ell_{t,k}^{i} = \frac{1}{B}\sum_{b=1}^{B}\left\| z_{0,b}^{i} - \widehat{z}_{0,\theta,b}^{i,(t)} \right\|_2^2, \tag{41}$$

and the EMA-smoothed estimate is

$$\widehat{D}_{t,k}^{i} = (1-\xi)\widehat{D}_{t,k-1}^{i} + \xi \ell_{t,k}^{i}. \tag{42}$$

This is the same training-time quantity. The first available observation initializes each $\widehat{D}_{t,k}^{i}$.

Monotonicization and interpolation. Empirical losses may fluctuate across timesteps, so we first compute the non-decreasing isotonic projection

$$\widetilde{\boldsymbol{D}}_{k}^{i} = \mathrm{Iso}_{\uparrow}\left(\widehat{D}_{1,k}^{i}, \ldots, \widehat{D}_{T,k}^{i}\right), \qquad \widetilde{D}_{1,k}^{i} \le \cdots \le \widetilde{D}_{T,k}^{i}. \tag{43}$$

Let $\{\bar{\alpha}_t^{\mathrm{ref}}\}_{t=1}^{T}$ denote the reference cumulative signal-retention schedule. Isotonic regression can create plateaus, so several consecutive timesteps may share the same projected difficulty. We merge each such plateau into one difficulty knot. If $\mathcal{I}_r$ is the set of consecutive timesteps in plateau r, define

$$\delta_{r,k}^{i} = \widetilde{D}_{t,k}^{i} \quad (t \in \mathcal{I}_r), \qquad \gamma_r = \frac{1}{|\mathcal{I}_r|}\sum_{t\in\mathcal{I}_r} \bar{\alpha}_t^{\mathrm{ref}}. \tag{44}$$

The unique knots satisfy $\delta_{1,k}^{i} < \cdots < \delta_{R_i,k}^{i}$, while $\gamma_1 \ge \cdots \ge \gamma_{R_i}$. We define a piecewise-linear, non-increasing map $\Psi_{i,k}$ through the pairs $\{(\delta_{r,k}^{i}, \gamma_r)\}_{r=1}^{R_i}$. If fewer than two unique knots remain, the current schedule for that position is retained for the next training block.

For $R_i \ge 2$, we uniformly sample the observed difficulty range:

$$D_{t,k}^{i,\mathrm{ramp}} = \delta_{1,k}^{i} + \frac{t-1}{T-1}\left(\delta_{R_i,k}^{i} - \delta_{1,k}^{i}\right), \tag{45}$$

and obtain tentative signal-retention values

$$\widetilde{\alpha}_{t,k}^{i} = \Psi_{i,k}\left(D_{t,k}^{i,\mathrm{ramp}}\right). \tag{46}$$

The ramp redistributes the reference schedule over the observed difficulty range; it does not assume equal denoising errors across timesteps.

Constrained schedule projection. The tentative values are clipped to $[\varepsilon_{\mathrm{clip}}, 1-\varepsilon_{\mathrm{clip}}]$ and projected onto the set of non-increasing schedules with fixed first and terminal noise levels. Equivalently, for each position we solve

$$\begin{aligned}(\bar{\alpha}_{1,k}^{i}, \ldots, \bar{\alpha}_{T,k}^{i}) = \arg\min_{a_1,\ldots,a_T} & \sum_{t=1}^{T}\left(a_t - \widetilde{\alpha}_{t,k}^{i}\right)^2 \\ \text{s.t.} \quad & a_1 = \bar{\alpha}_1^{\mathrm{ref}}, \qquad a_T = \bar{\alpha}_T^{\mathrm{ref}}, \\ & 1 > a_1 \ge a_2 \ge \cdots \ge a_T > 0,\end{aligned} \tag{47}$$

with $\bar{\alpha}_{0,k}^{i} = 1$. This gives

$$1 = \bar{\alpha}_{0,k}^{i} > \bar{\alpha}_{1,k}^{i} \ge \cdots \ge \bar{\alpha}_{T,k}^{i} > 0. \tag{48}$$

After recalibration, all derived quantities are recomputed:

$$\alpha_t^{i} = \frac{\bar{\alpha}_t^{i}}{\bar{\alpha}_{t-1}^{i}}, \qquad \beta_t^{i} = 1 - \alpha_t^{i}, \qquad \lambda_t^{i} = \log\frac{\bar{\alpha}_t^{i}}{1-\bar{\alpha}_t^{i}}, \tag{49}$$

together with c_t^{i}, d_t^{i}, $\widetilde{\beta}_t^{i}$, and w_t^{i}. Plateaus are allowed by the constrained projection; as noted above, they give $\beta_t^{i} = w_t^{i} = 0$ at the corresponding step.

C TRAJECTORY-INFORMED REDUCED-STEP DECODING DETAILS

C.1 VALIDATION PROFILING AND ERROR-CURVE FITTING

We construct the reduced schedule once after training from the validation set $\mathcal{V}$. Each validation example is run through one full T-step reverse trajectory, with mixed-space masking active and $\boldsymbol{\rho}_t$ resampled at every denoiser evaluation exactly as in test-time sampling. No trajectory subsampling is used: all examples in $\mathcal{V}$ are profiled once. For position i, $\mathcal{V}_i$ contains only examples for which position i belongs to the target region and holds a non-padding, non-mask target token. Equation 7 then averages the clean-state prediction error over $|\mathcal{V}_i|$ trajectories.

A position is retained for schedule construction only when it has at least $n_{\min}$ validation observations; we use $n_{\min} = 5$. We additionally require at least four distinct log-SNR values after removing unavailable observations. These filters define the active set $\mathcal{A}$ used in Eq. 9. After filtering, each retained position enters the aggregation once, independent of its number of validation observations; its contribution is then weighted according to Eq. 9. For each retained position, we form the pairs

$$\left\{(\lambda_t^i, E_{t,i})\right\}_{t=1}^{T}. \tag{50}$$

A projected token-wise schedule can contain plateaus, in which case repeated log-SNR values occur. Before fitting, repeated λ values are collapsed by averaging their corresponding validation errors. Let n_i denote the number of remaining unique log-SNR knots. We fit a cubic smoothing spline $E_i(\lambda)$ with smoothing parameter

$$s_i = \gamma_{\mathrm{sm}}\, n_i\, \max\left\{\mathrm{Var}(E_i), 10^{-12}\right\}, \qquad \gamma_{\mathrm{sm}} = 0.005, \tag{51}$$

using spline degree $\min\{3, n_i - 1\}$. The theoretical derivation below assumes a non-increasing error curve. The smoothing spline itself is not constrained to be monotone; therefore, in implementation, small positive slopes introduced by validation noise or spline smoothing are removed when computing the importance through

$$a_i^{\mathrm{impl}}(\lambda) = \left[-e^{\lambda}\frac{dE_i(\lambda)}{d\lambda}\right]_+, \tag{52}$$

which is identical to Eq. 8 wherever the fitted curve is non-increasing. For numerical stability, only the exponential evaluation uses λ clipped to $[-60, 60]$; the spline is fitted at the original log-SNR coordinates. The training-time difficulty $\widehat{D}_{t,k}^i$ and the post-training validation error $E_{t,i}$ therefore serve different roles: the former calibrates $\mathcal{S}$ during training, while the latter is measured along the completed reverse process and is used only for reduced-step allocation.

C.2 DERIVATION OF THE STEP-DENSITY ALLOCATION

We derive Proposition 1. For clarity, we consider one target position i and omit its subscript within the derivation. Let $\nu = e^{\lambda}$ denote SNR. For this position, the noisy evaluation range is $\nu_{\min} = e^{\lambda_T^i}$ to $\nu_{\max} = e^{\lambda_1^i}$, with $\nu_{\min} < \nu_{\max}$. Accordingly, $\lambda_{\min} = \lambda_T^i$ and $\lambda_{\max} = \lambda_1^i$. Consider two nearby retained SNR levels ν and $\nu + \Delta\nu$. We define the following local discretization surrogate for skipping the intermediate SNR region:

$$\mathcal{R}_{\mathrm{loc}} = \Delta\nu\, E(\nu) - \int_{\nu}^{\nu+\Delta\nu} E(u)\, du. \tag{53}$$

For a differentiable non-increasing error curve,

$$E(u) = E(\nu) + \frac{dE(\nu)}{d\nu}(u - \nu) + \mathcal{O}((u-\nu)^2). \tag{54}$$

Integrating the remainder over the interval gives

$$\mathcal{R}_{\mathrm{loc}} = \frac{1}{2}\left(-\frac{dE(\nu)}{d\nu}\right)(\Delta\nu)^2 + \mathcal{O}((\Delta\nu)^3). \tag{55}$$

Thus the surrogate assigns a larger penalty to skipping a region where the reconstruction-error curve changes more rapidly. Let $p_\nu(\nu)$ be a normalized density of the K noisy denoiser evaluation levels,

$$\int_{\nu_{\min}}^{\nu_{\max}} p_\nu(\nu)\, d\nu = 1. \tag{56}$$

The clean readout at $t = 0$ is not an additional denoiser evaluation. Hence the K noisy evaluation levels contain $K-1$ inter-level spacings, consistent with the sampler: non-adjacent latent transitions are applied for the first $K-1$ calls, and the final call directly predicts $\widehat{Z}_0$. Under the large-K continuous approximation, the local spacing is therefore

$$\Delta\nu \simeq \frac{1}{(K-1)p_\nu(\nu)}. \tag{57}$$

An interval $d\nu$ contains approximately $(K-1)p_\nu(\nu)d\nu$ such spacings. Summing Eq. 55 over the trajectory gives

$$\mathcal{R} \simeq \frac{1}{2(K-1)} \int_{\nu_{\min}}^{\nu_{\max}} \frac{-\frac{dE(\nu)}{d\nu}}{p_\nu(\nu)}\, d\nu. \tag{58}$$

The factor $1/[2(K-1)]$ is constant under a fixed denoiser budget and does not affect the minimizing density. We next express the surrogate in log-SNR coordinates. Since $\nu = e^\lambda$,

$$d\nu = e^\lambda d\lambda, \qquad p_\nu(\nu) = e^{-\lambda} p(\lambda), \qquad \frac{dE}{d\nu} = e^{-\lambda}\frac{dE}{d\lambda}. \tag{59}$$

Substituting these relations into Eq. 58 yields

$$\mathcal{R} \simeq \frac{1}{2(K-1)} \int_{\lambda_{\min}}^{\lambda_{\max}} \frac{-e^\lambda \frac{dE(\lambda)}{d\lambda}}{p(\lambda)}\, d\lambda = \frac{1}{2(K-1)} \int_{\lambda_{\min}}^{\lambda_{\max}} \frac{a(\lambda)}{p(\lambda)}\, d\lambda, \tag{60}$$

where $a(\lambda) = -e^\lambda dE(\lambda)/d\lambda$, matching Eq. 8. We minimize Eq. 60 over normalized densities $p(\lambda)$ satisfying

$$\int_{\lambda_{\min}}^{\lambda_{\max}} p(\lambda)\, d\lambda = 1. \tag{61}$$

By Cauchy–Schwarz,

$$\left(\int_{\lambda_{\min}}^{\lambda_{\max}} \sqrt{a(\lambda)}\, d\lambda\right)^2 = \left(\int_{\lambda_{\min}}^{\lambda_{\max}} \sqrt{\frac{a(\lambda)}{p(\lambda)}}\sqrt{p(\lambda)}\, d\lambda\right)^2 \tag{62}$$

$$\leq \left(\int_{\lambda_{\min}}^{\lambda_{\max}} \frac{a(\lambda)}{p(\lambda)}\, d\lambda\right)\left(\int_{\lambda_{\min}}^{\lambda_{\max}} p(\lambda)\, d\lambda\right). \tag{63}$$

Because the second factor is one,

$$\int_{\lambda_{\min}}^{\lambda_{\max}} \frac{a(\lambda)}{p(\lambda)}\, d\lambda \geq \left(\int_{\lambda_{\min}}^{\lambda_{\max}} \sqrt{a(\lambda)}\, d\lambda\right)^2. \tag{64}$$

Equality holds when $\sqrt{a(\lambda)/p(\lambda)} \propto \sqrt{p(\lambda)}$, giving

$$p^*(\lambda) = \frac{\sqrt{a(\lambda)}}{\int_{\lambda_{\min}}^{\lambda_{\max}} \sqrt{a(u)}\, du}. \tag{65}$$

Therefore $p^*(\lambda) \propto \sqrt{a(\lambda)}$, proving Proposition 1. If $a(\lambda) = 0$ over the complete trajectory, the surrogate does not prefer any region and we use uniform spacing in log-SNR.

C.3 Additional Reduced-Step Results

The main paper reports molecule captioning as the representative trajectory-allocation experiment. Here, we provide the complete matched-budget results for text-guided molecule generation, forward reaction prediction, and retrosynthesis. In every comparison, the trained `CAMS-Diff` checkpoint, sampler, and denoising budget K are unchanged; uniform and trajectory-informed variants differ only in the placement of the retained reverse steps. We report both DPM-Solver++ and DDIM to test whether the effect persists across samplers.

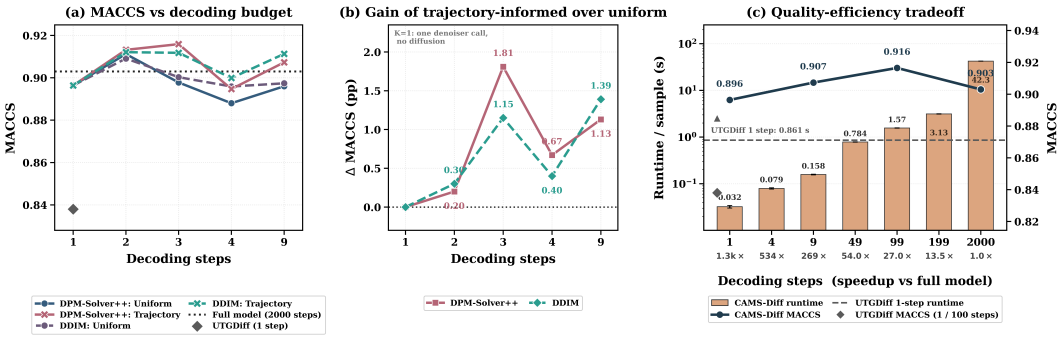


Figure 5: **Reduced-step decoding for text-guided molecule generation.** (a) MACCS under uniform and trajectory-informed DPM-Solver++ and DDIM across matched denoising budgets, with full-step `CAMS-Diff` and one-step UTGDiff shown as references. (b) MACCS improvement from trajectory-informed over uniform timestep placement. (c) Quality–efficiency trade-off across reduced-step operating points.

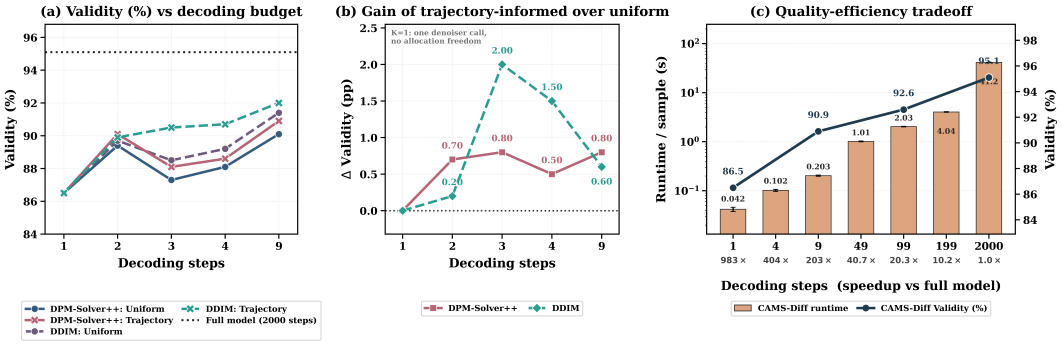


Figure 6: **Reduced-step decoding for forward reaction prediction.** (a) Validity under uniform and trajectory-informed DPM-Solver++ and DDIM across matched denoising budgets. (b) Validity improvement from trajectory-informed over uniform timestep placement. At $K = 1$, there is no allocation freedom. (c) Quality–efficiency trade-off across reduced-step operating points.

Text-guided molecule generation. Figure 5 reports MACCS across reduced decoding budgets. Trajectory-informed scheduling improves over uniform placement by up to 1.81 MACCS points with DPM-Solver++ and 1.39 points with DDIM. The advantage is most pronounced at small and intermediate budgets and narrows as more reverse evaluations are retained, consistent with timestep placement becoming less restrictive as K increases.

Forward reaction prediction. Figure 6 shows the corresponding results for forward reaction prediction. At matched budgets, trajectory-informed placement improves validity by up to 0.80 percentage points with DPM-Solver++ and 2.00 points with DDIM. The location and magnitude of the gain vary with K, indicating that the informative regions of the reverse trajectory need not coincide across samplers. At $K = 1$, both schedules necessarily coincide because only one denoiser evaluation is available; this point therefore reflects low-step model performance rather than timestep allocation.

Retrosynthesis. Trajectory-informed scheduling also improves reduced-step retrosynthesis (Fig. 7). Relative to uniform scheduling at the same K, validity increases by up to 1.90 percentage points with DPM-Solver++ and 2.70 points with DDIM. As in forward prediction, the gains are budget- and sampler-dependent rather than concentrated at a single fixed reverse region. The $K = 1$ operating point again contains no timestep-allocation choice.

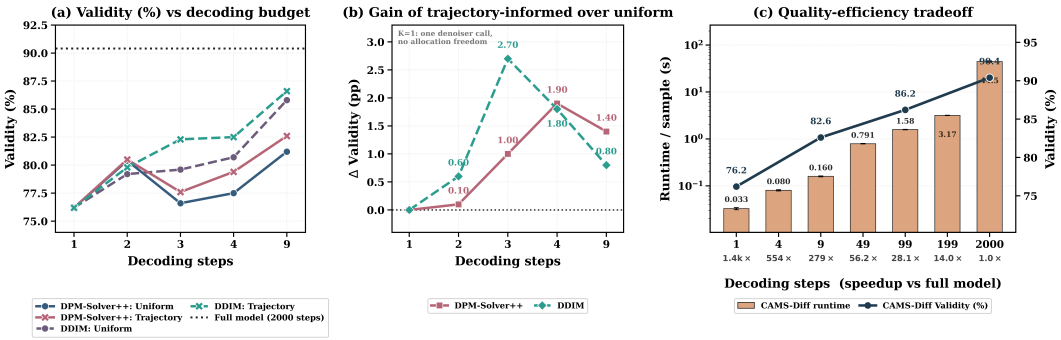


Figure 7: **Reduced-step decoding for retrosynthesis.** (a) Validity under uniform and trajectory-informed DPM-Solver++ and DDIM across matched denoising budgets. (b) Validity improvement from trajectory-informed over uniform timestep placement. At $K = 1$, there is no allocation freedom. (c) Quality–efficiency trade-off across reduced-step operating points.

C.4 Shared Timestep Aggregation and Discretization

Proposition 1 gives a position-wise density in log-SNR space, whereas decoding requires one set of model-evaluation timesteps shared by all positions. We therefore use the practical aggregation defined in Sec. 4.2. For each retained target position $i \in \mathcal{A}$, its unnormalized contribution on the common timestep axis is

$$\widetilde{p}_i(t) = \sqrt{a_i(\lambda_t^i)} \left| \frac{d\lambda_t^i}{dt} \right|. \tag{66}$$

We sum these contributions before normalization, so positions with larger total surrogate mass contribute proportionally more to the shared profile. This cross-position sum is a practical aggregation; Proposition 1 does not claim that it is the jointly optimal density across positions. The normalized profile is exactly Eq. 9:

$$p(t) = \frac{\sum_{i \in \mathcal{A}} \widetilde{p}_i(t)}{\int_1^T \sum_{i \in \mathcal{A}} \widetilde{p}_i(s)\, ds}. \tag{67}$$

In implementation, the derivative $d\lambda_t^i/dt$ is evaluated by finite differences on the learned schedule and the integral normalization is replaced by the corresponding sum over unit-spaced noisy timesteps.

The highest-noise level T is always retained as the first denoiser evaluation and the clean level 0 is reserved for the final readout. The binning heuristic therefore tiles only the interior noisy levels $t \in \{1, \dots, T-1\}$. For a budget of K denoiser evaluations, we use

$$B_{\text{bin}} = \max\left\{1, \text{round}\left(\frac{T-1}{2K}\right)\right\}, \tag{68}$$

which gives approximately $2K$ contiguous interior bins. For bin $\mathcal{B}_b$, its importance mass is

$$M_b = \int_{\mathcal{B}_b} p(t)\, dt. \tag{69}$$

We retain the $K-1$ interior bins with the largest M_b. If a selected bin has positive mass, its continuous representative is

$$\bar{t}_b = \frac{\int_{\mathcal{B}_b} t\, p(t)\, dt}{\int_{\mathcal{B}_b} p(t)\, dt}; \tag{70}$$

for a zero-mass bin we use its midpoint. The continuous representative is then rounded to the nearest unused integer timestep inside the same bin. Sorting the selected representatives in decreasing order gives

$$T = J_0 > J_1 > \cdots > J_{K-1} > J_K = 0. \tag{71}$$

The denoiser is evaluated at $J_0, \ldots, J_{K-1}$, giving exactly K model calls. Non-adjacent latent transitions are applied only for $J_r \to J_{r+1}$ with $r = 0, \ldots, K-2$, i.e. $K-1$ transitions. The final call at J_{K-1} directly produces $\widehat{Z}_0$, so no additional model evaluation or stochastic transition is applied at $J_K = 0$. The schedule is constructed once from the validation set and reused unchanged for all test examples.

C.5 Non-Adjacent Reverse Transition

For the posterior-form Gaussian update used in our implementation, consider two retained noisy levels $s < t$. Define the non-adjacent signal-retention coefficient

$$\alpha^i_{t|s} := \prod_{r=s+1}^{t} \alpha^i_r = \frac{\bar{\alpha}^i_t}{\bar{\alpha}^i_s}. \tag{72}$$

The corresponding forward transition is

$$q(z^i_t \mid z^i_s) = \mathcal{N}\left(\sqrt{\alpha^i_{t|s}}\, z^i_s, (1-\alpha^i_{t|s})I\right). \tag{73}$$

Together with

$$q(z^i_s \mid z^i_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}^i_s}\, z^i_0, (1-\bar{\alpha}^i_s)I\right), \tag{74}$$

the Gaussian posterior is

$$q(z^i_s \mid z^i_t, z^i_0) = \mathcal{N}\left(z^i_s; c^i_{t,s} z^i_0 + d^i_{t,s} z^i_t, \widetilde{\beta}^i_{t,s} I\right), \tag{75}$$

where

$$c^i_{t,s} = \frac{\sqrt{\bar{\alpha}^i_s}\,(1-\alpha^i_{t|s})}{1-\bar{\alpha}^i_t}, \tag{76}$$

$$d^i_{t,s} = \frac{\sqrt{\alpha^i_{t|s}}\,(1-\bar{\alpha}^i_s)}{1-\bar{\alpha}^i_t}, \tag{77}$$

$$\widetilde{\beta}^i_{t,s} = \frac{1-\bar{\alpha}^i_s}{1-\bar{\alpha}^i_t}(1-\alpha^i_{t|s}). \tag{78}$$

When $s = t-1$, we have $\alpha^i_{t|t-1} = \alpha^i_t$, so these expressions reduce directly to the one-step coefficients in Sec. 3. During reduced-step generation, z^i_0 is replaced by the mixed-observation clean-state prediction $\widehat{z}^{i,\mathrm{mix}}_{0,\theta}$. This posterior-form transition is used only for the $K-1$ inter-level updates before the final clean prediction; the last denoiser call directly returns $\widehat{Z}_0$. The retained schedule itself is independent of this posterior-form update and can instead be paired with DDIM, DPM-Solver++, or another compatible reverse solver.

D Experimental Tasks and Datasets

`CAMS-Diff` is evaluated across structure-to-text, text-to-structure, and structure-to-structure molecular translation. Figure 8 provides a visual overview of the five evaluation settings and shows how the conditioning input and prediction target change across tasks. The same conditional diffusion formulation is used throughout; only the task direction and the definitions of the condition and target change. Table 8 summarizes the corresponding datasets and evaluation roles.

D.1 Dataset Statistics and Split Protocol

For forward prediction and retrosynthesis, the 10K validation examples are held out from the original training partition and are used for validation and trajectory-based schedule construction; they are excluded from model optimization, and the 1K test sets are not used during schedule construction.

Table 9 summarizes the datasets and splits used in our experiments. For ChEBI-20, forward reaction prediction, and retrosynthesis, we follow the split protocol used by UTGDiff so that the molecular generation and reaction comparisons use the same benchmark setup. PCDes is used only for zero-shot evaluation and is never used to update `CAMS-Diff` parameters.

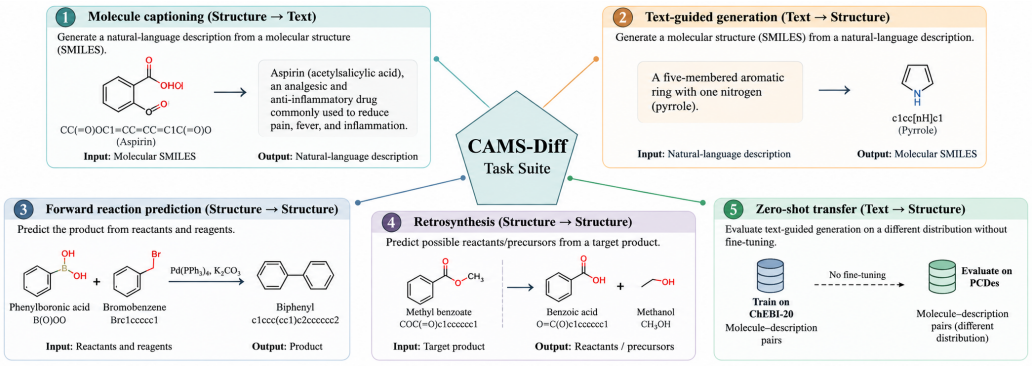


Figure 8: **Overview of the `CAMS-Diff` evaluation tasks.** Molecule captioning maps a molecular SMILES representation to a natural-language description, while text-guided generation performs the reverse mapping from text to molecular structure. Forward reaction prediction maps reactants and reagents to the reaction product, whereas retrosynthesis predicts possible reactants or precursors from a target product. Zero-shot transfer evaluates the ChEBI-20-trained generation model directly on PCDes without task-specific fine-tuning.

Table 8: Overview of the evaluation settings. C and Y denote the conditioning input and prediction target, respectively.

Direction	Task	Dataset	Condition C	Target Y	What the task tests
Structure → Text	Molecule captioning	M3-20M	Molecular SMILES	Natural-language description	Molecular-to-language grounding and semantic description of molecular structure.
Text → Structure	Text-guided generation	ChEBI-20	Natural-language description	Molecular SMILES	Whether a textual specification can be translated into the intended molecular structure.
Structure → Structure	Forward reaction	Mol-Instructions	Reactants and reagents	Product	Forward chemical transformation from reaction inputs to the expected product.
Structure → Structure	Retrosynthesis	Mol-Instructions	Product	Reactants / precursors	Inverse molecular transformation and precursor recovery for synthesis planning.
Zero-shot transfer	Text-guided generation	PCDes	Natural-language description	Molecular SMILES	Transfer to a different molecule–description distribution without task-specific fine-tuning.

D.2 Molecule Captioning on M3-20M

Molecule captioning maps molecular structure to natural language. In this setting, the conditioning input C is the molecular SMILES sequence and the target Y is its textual description. We use M3-20M, which contains approximately 300K paired SMILES–description examples.

Captioning tests a different output space from the molecular generation and reaction tasks. The model must preserve molecular information while producing fluent text rather than another molecular sequence. This makes the task useful for testing whether the same conditional diffusion framework can transfer from structured molecular targets to natural-language targets.

The molecular sequence is encoded as the condition and the caption is used as the diffusion target. Padding positions are excluded from token-level losses and from the position-wise trajectory statistics used for reduced-step allocation. Exact tokenizer, truncation, and sequence-length settings are reported in Appendix F.

Table 9: Dataset statistics and evaluation splits. M3-20M uses an $80/10/10\%$ train/validation/test split. Generation and reaction-task splits follow the UTGDiff evaluation protocol where indicated. For the reaction tasks, 10K examples are reserved from the original training partition for validation and trajectory profiling.

Dataset	Task	Split source	Train	Valid.	Test	Total	Content
M3-20M	Captioning	Our split	80%	10%	10%	∼300K	Paired molecular SMILES and natural-language descriptions.
ChEBI-20	Generation	UTGDiff split	26,407	3,301	3,300	33,008	Curated molecule–description pairs for text-guided molecular generation.
Mol-Instructions	Forward reaction	UTGDiff + holdout	∼110K	10K	1,000	∼121K	Reactant/reagent-to-product reaction pairs.
Mol-Instructions	Retrosynthesis	UTGDiff + holdout	∼110K	10K	1,000	∼121K	Product-to-reactant/precursor reaction pairs.
PCDes	Zero-shot generation	Standard split	10,500	1,500	3,000	15,000	Molecule–description pairs from a distribution different from ChEBI-20.

D.3 Text-Guided Molecule Generation on ChEBI-20

Text-guided molecule generation reverses the molecule-captioning direction. Here, C is a natural-language molecular description and Y is the target SMILES sequence. We use the standard ChEBI-20 benchmark split adopted by UTGDiff: 26,407 training, 3,301 validation, and 3,300 test examples.

This task requires both semantic alignment and molecular well-formedness. A generated molecule should agree with the textual condition, but its decoded SMILES must also describe a valid molecular structure. We therefore report molecular fingerprint similarity separately from exact match and validity. This separation is important because high structural similarity does not by itself imply exact recovery or valid SMILES generation.

In addition to aggregate results, we analyze generation as a function of instruction length and molecular complexity in Appendix G.

D.4 Forward Reaction Prediction

Forward reaction prediction maps the reactant side of a reaction to the expected product. The condition C contains the reactants and reagents, while Y is the product SMILES. We follow the Mol-Instructions forward-reaction split used by UTGDiff, which contains approximately 120K training pairs and 1,000 test examples.

Unlike text-guided generation, the condition is already molecular. The model must infer the chemical transformation and produce the product structure. This makes forward prediction a direct test of whether the denoising process can preserve molecular constraints while performing a non-trivial structural transformation.

Exact match evaluates recovery of the reference product. MACCS, RDK, and Morgan similarities provide complementary structural comparisons when the predicted molecule differs from the reference sequence. Difficulty-stratified results are reported in Appendix I.

D.5 Retrosynthesis

Retrosynthesis reverses the forward reaction direction. The condition C is the target product and Y is the reactant or precursor sequence. We follow the Mol-Instructions retrosynthesis split used by UTGDiff, with approximately 120K training pairs and 1,000 test examples.

Retrosynthesis is directly related to synthesis planning and is a challenging conditional generation problem because a target product can admit more than one chemically plausible precursor set. Exact

Table 10: Baselines used across the evaluation tasks.

Model	Representation	Used for
Autoregressive / encoder-decoder baselines		
Text+Chem T5	Token sequence	Captioning, forward reaction, retrosynthesis
GraphT5	Graph / text	Captioning
MolT5	Text / SMILES	Generation, PCDes
BioT5 / BioT5+	Text / SMILES	Forward reaction, retrosynthesis, PCDes
GALACTICA	Token sequence	Forward reaction, retrosynthesis
Mol-Instructions	Instruction / sequence	Forward reaction, retrosynthesis
LLaMA-7B (LoRA)	Instruction / sequence	Forward reaction, retrosynthesis
InstructMol-G / GS	Text / graph	Forward reaction, retrosynthesis
Continuous sequence diffusion baselines		
DiffuSeq	Continuous sequence	Captioning
TGM-DLM	Continuous sequence	Captioning, generation, PCDes
BiMol-Diff	Continuous sequence	Captioning, generation
Graph / discrete diffusion baselines		
3M-Diffusion	Molecular graph	Generation, PCDes
UTGDiff	Discrete molecular graph	Generation, forward reaction, retrosynthesis, PCDes
TDGD	Discrete molecular graph	Generation

match is therefore a strict metric. We report it together with validity and molecular fingerprint similarities to distinguish exact sequence recovery from broader structural agreement.

Forward prediction and retrosynthesis together evaluate both directions of molecule-to-molecule translation. We additionally stratify both tasks by sequence difficulty in Appendix I.

D.6 Zero-Shot Transfer to PCDes

PCDes is used to test whether a model trained for text-guided molecular generation transfers to a different molecule-description distribution. It contains 10,500 training, 1,500 validation, and 3,000 test pairs. `CAMS-Diff` is not fine-tuned on PCDes; the ChEBI-20-trained generation model is applied directly to the PCDes test set.

The same molecular metrics used for ChEBI-20 are retained so that the zero-shot comparison measures how molecular similarity, exact recovery, and validity transfer under dataset shift. Full PCDes results are reported in Table 6.

E Baselines and Evaluation Protocol

We compare `CAMS-Diff` with representative autoregressive, continuous sequence diffusion, and graph diffusion baselines. The baseline set varies by task depending on the available comparisons.

E.1 Baseline Models

Autoregressive baselines. Text+Chem T5, GraphT5, MolT5, and BioT5 provide sequence and molecule-language baselines. For the reaction tasks, we also compare with GALACTICA, Mol-Instructions, LLaMA-7B with LoRA, and InstructMol.

Continuous sequence diffusion baselines. DiffuSeq and TGM-DLM provide sequence diffusion baselines. BiMol-Diff is the closest continuous molecular diffusion baseline and is used for both molecule captioning and text-guided generation.

Graph diffusion baselines. We compare with 3M-Diffusion, UTGDiff, and TDGD for molecular generation. UTGDiff is also used for forward reaction prediction, retrosynthesis, zero-shot transfer,

Table 11: Baseline coverage across the evaluation tasks.

Task	Text+Chem T5	GraphT5	MolT5	BioT5 / BioT5+	DiffuSeq	TGM-DLM	BiMol-Diff	3M-Diffusion	UTGDiff	TDGD	Reaction AR	CAMS-Diff
Captioning	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✓
Generation	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓
Forward reaction	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓
Retrosynthesis	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓
PCDes	✗	✗	✓	✓	✗	✓	✗	✓	✓	✗	✗	✓

and the reduced-step efficiency comparison. For forward reaction prediction and retrosynthesis, the autoregressive baseline numbers are taken from the UTGDiff evaluation (Xiang et al., 2025).

E.2 Task-Specific Baseline Coverage

Table 11 summarizes the baselines used for each evaluation task.

E.3 Evaluation Metrics

We use different metric families for text and molecular outputs because surface agreement, semantic agreement, molecular similarity, and molecular validity measure different properties of a prediction.

Text-generation metrics. For molecule captioning, BLEU, chrF++, and METEOR measure surface overlap with the reference description. BERTScore-F1 measures contextual semantic similarity. MAUVE measures distributional agreement between generated and reference text. Higher values are better for all reported captioning metrics.

Exact match. Exact match is the fraction of predictions that exactly match the reference under the benchmark normalization used by the evaluation pipeline. For molecular outputs, this is intentionally strict. In retrosynthesis, for example, a chemically plausible precursor set can differ from the single reference sequence and still receive zero exact-match credit.

Validity. Validity is the fraction of decoded molecular strings that can be parsed into valid molecular structures by the molecular evaluation pipeline. We report it separately from similarity because a structurally close prediction is not useful if the decoded molecular representation is invalid.

Fingerprint similarity. For molecular outputs, we report Tanimoto similarity using three standard fingerprint families: **MACCS** structural keys, the **RDK** topological fingerprint, and circular **Morgan** fingerprints. These metrics emphasize different aspects of molecular structure and therefore provide a broader view than a single fingerprint alone. Higher values denote greater similarity to the reference molecule.

Why both similarity and validity are required. No single molecular metric captures the complete generation problem. Exact match measures exact recovery but can penalize structurally close outputs. Fingerprint metrics measure structural similarity but do not guarantee that the generated SMILES is valid. We therefore report exact match, validity, and molecular similarities together. This distinction is particularly important for text-guided generation, where `CAMS-Diff` can improve molecular similarity without uniformly improving exact match or validity.

E.4 Reported and Reproduced Results

Unless marked otherwise, baseline values are taken from the corresponding published benchmark results. Reproduced values are explicitly marked in the main result tables and are evaluated using the same task split and metric pipeline as `CAMS-Diff`.

For molecule captioning, BiMol-Diff† denotes reproduced baseline. Reproduced BiMol-Diff and `CAMS-Diff` captioning results are averaged over three random seeds and reported as mean $\pm$ standard deviation. For text-guided molecule generation, BiMol-Diff* denotes the reproduced base-

Table 12: Architecture and tokenization used across the evaluation tasks.

Tasks	#P	Condition encoder	Denoiser	Hidden	Heads	Diff. dim.	Max len.	Tokenization
Captioning	91M	–	12-layer decoder	128	8	128	128	SMILES / BERT uncased
Generation, forward, retro	180M	Frozen SciBERT	12-layer Transformer	1024	16	32	256	BERT uncased / SMILES

line under our molecular evaluation pipeline. For forward reaction prediction and retrosynthesis, UTGDiff* denotes the reproduced baseline used for direct comparison with `CAMS-Diff`.

TDGD is reported only as a supplementary published reference because an implementation is not available for independent reproduction. Its values are therefore excluded from best/second-best highlighting.

E.5 Evaluation Consistency

All reproduced models are evaluated on the same task split and with the same metric implementation used for `CAMS-Diff`. Molecular outputs pass through the same parsing, canonicalization, and fingerprint pipeline before computing exact match, validity, MACCS, RDK, and Morgan similarity. Reduced-step comparisons use the same trained `CAMS-Diff` checkpoint; only the retained reverse timesteps and reverse solver change. This isolates the effect of the inference schedule from differences in training.

Model architecture, optimization settings, training hyperparameters, and the runtime measurement procedure are provided in Appendix F.

F Implementation and Training Details

This section summarizes the model architecture, tokenization, optimization settings, adaptive mixed-space configuration, and inference procedure used in the experiments.

F.1 Architecture and Tokenization

We use two model configurations corresponding to the target modality. Molecule captioning uses a 91M-parameter encoder-decoder model with six encoder layers, nine decoder layers, and a 128-dimensional diffusion latent space. Text-guided molecule generation, forward reaction prediction, and retrosynthesis use the same 180M-parameter molecular-output architecture, with a frozen SciBERT condition encoder and a 12-layer Transformer denoiser. The three molecular-output tasks are trained as separate task-specific checkpoints using the same architecture. Molecular sequences are represented as SMILES. Natural-language inputs use an uncased BERT tokenizer. Table 12 summarizes the two configurations.

For molecule captioning, the molecular sequence forms the conditioning input and the text description is the diffusion target. We use a maximum sequence length of 128, with molecular conditioning occupying positions 0-86 and the caption target beginning at position 87; unused positions are padded. For the molecular-output tasks, the target SMILES sequence is truncated or padded to a maximum length of 256. Padding positions are excluded from position-wise loss statistics and trajectory analysis.

PCDes does not introduce an additional model configuration: the ChEBI-20-trained text-to-molecule model is evaluated directly on PCDes without further training.

F.2 Optimization and Diffusion Configuration

All models use a continuous diffusion process with $T = 2000$ timesteps and a square-root base noise schedule. The denoiser predicts the clean state x_0. Training is performed in FP16 precision. We use a batch size of 1024 for molecule captioning and 256 for the molecular-output tasks.

The main optimization and diffusion settings are summarized in Table 13.

Table 13: Optimization and diffusion settings used in CAMS-Diff.

Tasks	Batch	T	Base	Pred.	Optimizer	LR	Drop.	EMA	Clip	FP16
Captioning	1024	2000	sqrt	x_0	AdamW	5×10^{-5}	0.1	0.9999	2.4	✓
Generation, forward, retro	256	2000	sqrt	x_0	AdamW	5×10^{-5}	0.1	0.9999	2.4	✓

Table 14: Compute and runtime protocol. Runtime measurements use full-model inference; trajectory profiling and reduced-schedule construction are one-time offline costs and are excluded from inference latency.

Setting	Hardware	Batch	Time	Protocol
Captioning training	2×A100	1024	48–50 h	64-token target; 20K warm-up and 20K schedule updates.
Generation training	1×A100	256	∼50 h	20K warm-up and 20K schedule updates.
Retrosynthesis training	1×A100	256	48–50 h	20K warm-up and 20K schedule updates.
Inference timing	1×A100	50	10 batches	Full-model inference; wall-clock time averaged over 10 batches.

We use zero weight decay and linearly anneal the learning rate over 200K optimization steps. Model checkpoints are maintained using an exponential moving average with decay 0.9999.

The 20K-step adaptive-schedule warm-up described below is separate from the optimizer schedule. During this initial block, training uses the square-root reference schedule. Adaptive token-wise noising is introduced only after this initial training period.

Compute and runtime. All experiments use FP16 precision. Captioning is trained on two A100 GPUs with batch size 1024 and a 64-token target length, while molecular-output tasks use a single A100 GPU with batch size 256. The adaptive schedule is activated after a 20K-step warm-up and updated every 20K steps; this interval was selected from preliminary ablations. Reported experimental results are averaged over three runs. Reproduced strongest baselines are run on the same A100 hardware using their reported configurations. Table 14 summarizes the compute and runtime protocol.

F.3 Adaptive Mixed-Space Configuration

CAMS-Diff starts from the square-root cumulative signal-retention schedule and periodically recalibrates its interior values from position-wise clean-state prediction difficulty. The resulting schedule is token-wise: each target position can follow a different signal-retention trajectory, while the first and terminal noise levels remain shared across positions.

The reference schedule is used for the first 20K optimization steps. Thereafter, the adaptive schedule is updated every 20K steps and remains fixed between consecutive updates. This blockwise update avoids changing the corruption process at every minibatch. Representation masking is coupled to the corresponding cumulative Gaussian noise scale:

$$\varpi_t^i = \eta\sqrt{1-\bar{\alpha}_t^i}, \qquad t = 1, \ldots, T, \tag{79}$$

where $\eta = 0.5$ in the main experiments. Thus, positions at a higher cumulative Gaussian noise level have a larger probability of representation masking.

Only valid target positions contribute to the statistics used for schedule recalibration. Padding positions therefore do not accumulate denoising difficulty. After each update, the recalibrated cumulative schedule is projected to remain monotone and the one-step coefficients α_t^i and β_t^i are recomputed.

During training, the Bernoulli masking variables are sampled from the current position-dependent masking probabilities. During inference, masking variables are resampled at each denoiser evaluation using the masking probability of the corresponding retained reverse step. Full schedule-calibration details are given in Appendix B.3.

Table 15: Adaptive noising and coupled mixed-space settings used in the main experiments.

Warm-up	Update	Granularity	Base schedule	η	Mask rate	Between updates
20K	20K	Position-wise	sqrt	0.5	$\varpi_t^i = \eta\sqrt{1-\bar{\alpha}_t^i}$	Fixed

Table 16: Full-step and reduced-step inference configurations.

Setting	Trajectory	Step selection	Solver	Test-time use
Full-step	$T = 2000$	All diffusion levels	Ancestral reverse process	Complete reverse trajectory
Reduced-step	$K \ll T$	Validation-trajectory allocation	DDIM / DPM-Solver++	Fixed retained schedule for all test examples

F.4 Full-Step and Reduced-Step Inference

Full-step inference follows the complete $T = 2000$ diffusion trajectory using the learned token-wise schedule. All target positions share the same global timestep index, but their cumulative signal-retention and log-SNR values can differ because the learned schedule is position-dependent.

For reduced-step decoding, we first run the full reverse process on the validation set and record the position-wise reconstruction error along the trajectory. These validation trajectories are used to construct the shared reduced schedule described in Appendix C. Schedule construction is performed once after training. The resulting retained timesteps are then fixed and used for all examples in the corresponding test set.

We evaluate reduced-step decoding with DDIM and DPM-Solver++ at the reported budgets K. The trained checkpoint, token-wise schedule, and conditioning model remain unchanged; only the retained reverse-process evaluations and reverse solver are changed.

For two consecutive retained levels, the reverse transition is recomputed from their corresponding cumulative signal-retention values rather than by directly subsampling precomputed one-step coefficients. This preserves the non-adjacent transition induced by the learned token-wise schedule. The final denoiser evaluation directly predicts the clean latent, so the terminal clean state does not require an additional denoiser call.

The validation trajectory profiling used to construct the reduced schedule is an offline, one-time procedure and is not repeated for individual test examples.

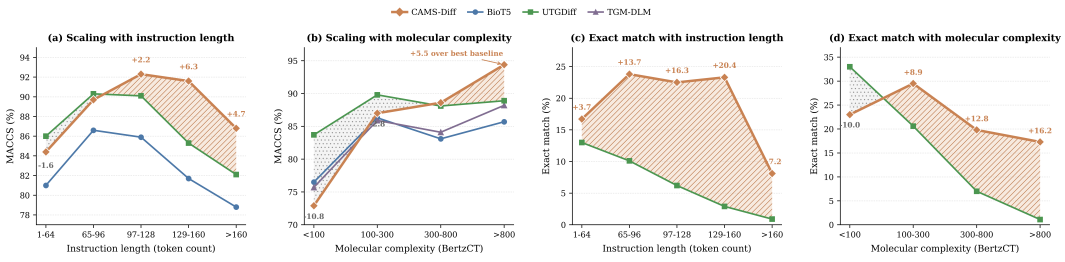


Figure 9: **Generation robustness across instruction length and molecular complexity.** Performance is stratified by conditioning-instruction length and target molecular complexity (BertzCT). `CAMS-Diff` becomes increasingly competitive in harder regimes, with particularly large gains in exact match for longer instructions and more complex molecular targets. Annotated values denote the improvement over the strongest competing baseline within each bin.

G Generation Difficulty Analysis

We examine how `CAMS-Diff` behaves as text-guided molecule generation becomes more difficult. We stratify the test set along two complementary axes: conditioning-instruction length and target

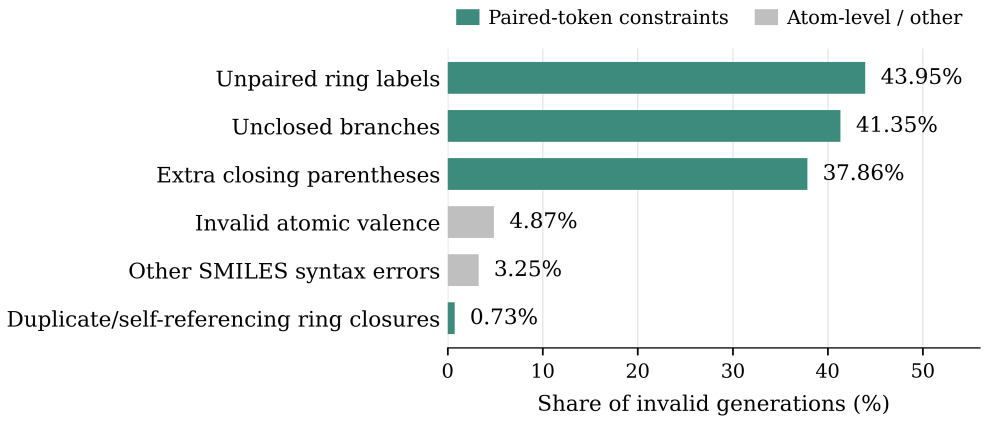


Figure 10: **Failure taxonomy of invalid text-guided molecular generations.** Percentage of invalid CAMS-Diff outputs exhibiting each failure type on ChEBI-20. Paired-token constraints-including unpaired ring labels, unclosed branches, and unmatched parentheses-account for the dominant observed failure modes, whereas invalid atomic valence is comparatively uncommon. Categories are non-exclusive and may co-occur within an output.

molecular complexity, measured by BertzCT. Figure 9 reports MACCS similarity and exact match across the resulting regimes.

For instruction length, CAMS-Diff is comparable to the strongest baselines in the shorter regimes, but its advantage emerges for longer instructions. For MACCS, the gain over the strongest baseline reaches 2.2, 6.3, and 4.7 percentage points in the 97–128, 129–160, and $>$ 160 token bins, respectively. The effect is stronger for exact recovery: the gain rises from 3.7 points for the shortest instructions to 13.7, 16.3, and 20.4 points as instruction length increases, before remaining 7.2 points higher in the longest bin.

A similar pattern appears with molecular complexity. CAMS-Diff is not uniformly better in the simplest regimes, but the gap grows as the target molecules become more complex. In the highest-complexity bin, CAMS-Diff improves MACCS by 5.5 points over the strongest baseline. The exact-match advantage is more pronounced, increasing from 8.9 points for moderately complex molecules to 12.8 and 16.2 points in the two harder regimes.

Overall, CAMS-Diff provides its clearest gains when generation requires recovering longer or more structurally complex molecular sequences. The trend is particularly strong for exact match, suggesting that adaptive mixed-space corruption helps preserve coordinated sequence-level structure as generation difficulty increases.

Failure taxonomy of invalid generations. We further characterize the lower generation validity observed in the aggregate results by categorizing invalid CAMS-Diff outputs on ChEBI-20. Figure 10 shows that the dominant failure modes involve paired-token and SMILES-serialization constraints: 43.95% of invalid generations contain unpaired ring labels, 41.35% contain unclosed branches, and 37.86% contain extra closing parentheses. In contrast, invalid atomic valence occurs in only 4.87% of invalid generations. Thus, the remaining validity errors are dominated by sequence-level serialization failures rather than atom-level valence violations. These categories are non-exclusive and may co-occur within the same invalid generation.

H Reverse-Step Allocation Ablation

The main experiments show that trajectory-informed scheduling improves over uniform step selection under the same decoding budget. We further test whether this gain can be explained by a fixed preference for a particular region of the reverse trajectory.

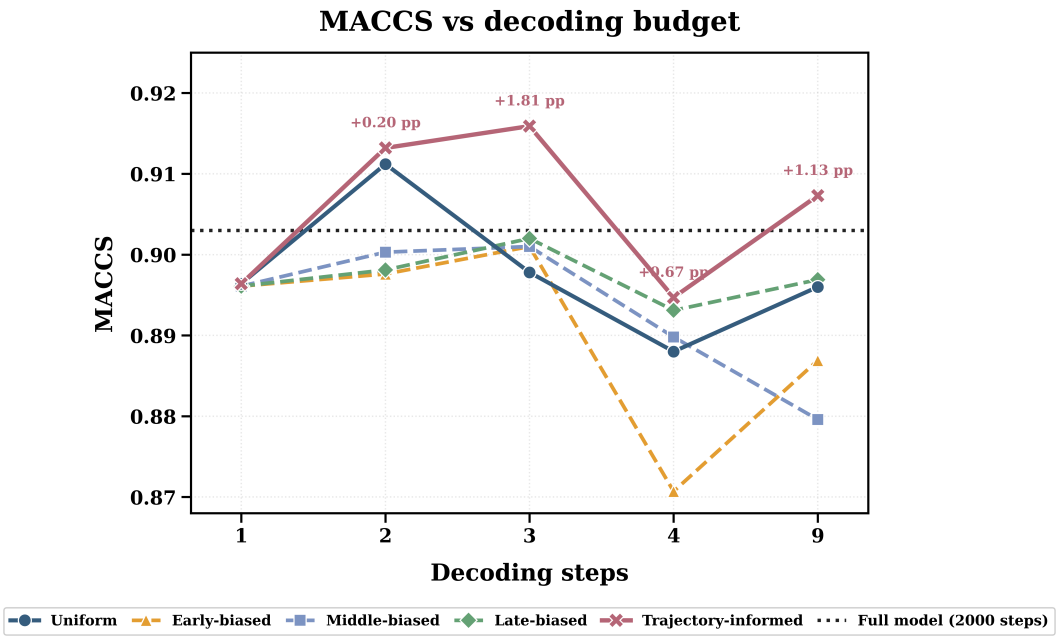


Figure 11: **Ablation of reverse-step allocation for text-guided molecule generation.** MACCS similarity under matched decoding budgets using DPM-Solver++. Early-, middle-, and late-biased controls sample retained denoising levels from fixed distributions concentrated in the corresponding regions of the reverse trajectory, whereas trajectory-informed allocation uses measured denoising dynamics to determine step placement. No single fixed-region allocation consistently reproduces the trajectory-informed schedule.

We construct three matched-budget controls by sampling retained denoising levels from fixed biased distributions over the reverse process. The *early-biased*, *middle-biased*, and *late-biased* variants concentrate probability mass toward the high-noise, intermediate, and low-noise portions of the trajectory, respectively. We use predefined Gaussian/heavy-tailed sampling profiles to obtain these allocations, while keeping the trained `CAMS-Diff` model, DPM-Solver++, and number of denoiser evaluations unchanged. In contrast, the trajectory-informed schedule derives its allocation from the measured reconstruction dynamics rather than a pre-specified trajectory region.

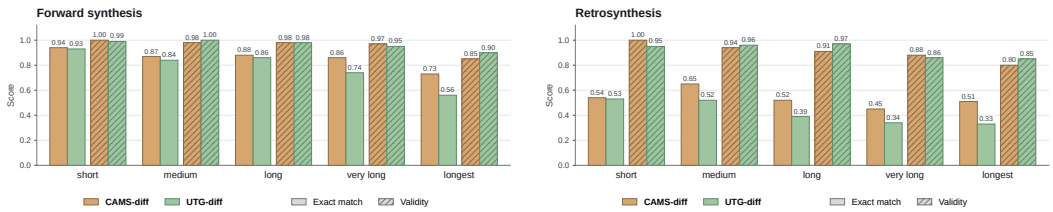
Figure 11 compares these strategies for text-guided molecule generation. Fixed concentration in any single region does not consistently reproduce the trajectory-informed schedule. The latter provides gains of 0.20, 1.81, 0.67, and 1.13 MACCS percentage points over uniform decoding at the corresponding 2-, 3-, 4-, and 9-step operating points. Although individual biased schedules can be competitive at specific budgets, their relative performance changes substantially with the available number of evaluations.

Overall, the results indicate that the reduced-step gains are not explained simply by allocating more evaluations to the beginning, middle, or end of the reverse process. Instead, the useful trajectory region depends on the decoding budget, motivating allocation from the observed denoising dynamics rather than from a fixed positional bias.

I Denoising Trajectory and Failure Analysis

We further analyze how reaction-prediction behaviour changes with sequence difficulty and how the corresponding molecular structures evolve during the reverse diffusion process. We consider both forward synthesis and retrosynthesis, stratifying examples into short, medium, long, very-long, and longest sequence regimes.

Figure 12 shows that the clearest difference between the two methods appears in exact molecular recovery. For forward synthesis, performance is similar in the short regime, with exact match



(a) Forward synthesis **(b) Retrosynthesis**

Figure 12: **Reaction prediction across increasing sequence length.** Exact match and validity are reported for CAMS-Diff and UTGDiff across short, medium, long, very-long, and longest sequence regimes. CAMS-Diff shows an increasingly clear advantage in exact recovery for harder sequences, while validity remains broadly comparable and does not follow the same uniform trend.

of 0.94 for CAMS-Diff and 0.93 for UTGDiff. The gap becomes larger for harder sequences: CAMS-Diff reaches 0.86 exact match compared with 0.74 for UTGDiff in the very-long regime, and 0.73 compared with 0.56 in the longest regime. Validity remains high across most forward-synthesis bins, but does not show the same consistent advantage; in the longest regime, for example, CAMS-Diff obtains 0.85 validity compared with 0.90 for UTGDiff.

Retrosynthesis shows an even clearer difference in exact recovery. The two methods are nearly identical for short sequences, with exact match of 0.54 and 0.53, respectively. CAMS-Diff then reaches 0.65 compared with 0.52 for medium sequences, 0.52 compared with 0.39 for long sequences, 0.45 compared with 0.34 for very-long sequences, and 0.51 compared with 0.33 for the longest sequences. As in forward synthesis, validity does not follow the same monotonic trend. The more consistent advantage is therefore in exact structural recovery rather than uniformly improved validity.

I.1 Evolution Along the Reverse Trajectory

The length-stratified results motivate examining how the corresponding predictions evolve during denoising. The full trajectory grids provided at the end of this appendix show CAMS-Diff and UTGDiff across all five sequence-length regimes for both reaction directions.

Each trajectory should be read from left to right, following the reverse process from the high-noise state at step 1999, through intermediate states at steps 1499, 999, and 499, to the step-0 prediction, final output, and target molecule. The intermediate states show molecular similarity together with structural diagnostics such as incomplete rings, unresolved branches, and unparsable fragments.

The trajectories illustrate two recurring qualitative behaviours. First, CAMS-Diff often establishes a target-consistent molecular scaffold earlier in the reverse process and subsequently preserves or locally refines that structure. In the shorter and medium examples, the predicted molecule can already become highly similar to the target while the corresponding UTGDiff trajectory remains structurally different.

Second, the contrast becomes more visible as sequence length increases. In the long, very-long, and longest retrosynthesis examples, CAMS-Diff can pass through incomplete rings, branches, or locally incorrect fragments before resolving them into the target molecule. The corresponding UTGDiff trajectories often retain larger structural discrepancies deeper into the reverse process and, in some difficult examples, remain incomplete or unparsable at late checkpoints.

A similar behaviour is visible for forward synthesis. In the longer examples, CAMS-Diff progressively resolves incomplete molecular structure and reaches the target prediction, whereas the corresponding UTGDiff trajectories can preserve substantial structural error for a larger fraction of the reverse process.

The important distinction is therefore not that CAMS-Diff produces valid or exact molecules at every intermediate step. Several trajectories contain temporarily invalid or incomplete structures early in denoising. Rather, target-consistent structure is often formed and preserved earlier, while local sequence errors are progressively corrected toward the end of the reverse process. This be-

haviour is consistent with the larger exact-match advantage observed in the harder sequence regimes in Figure 12.

These examples are intended as qualitative illustrations rather than population-level evidence. The aggregate reaction results and the length-stratified analysis provide the quantitative comparison; the trajectory grids show how these differences can arise during the reverse process.

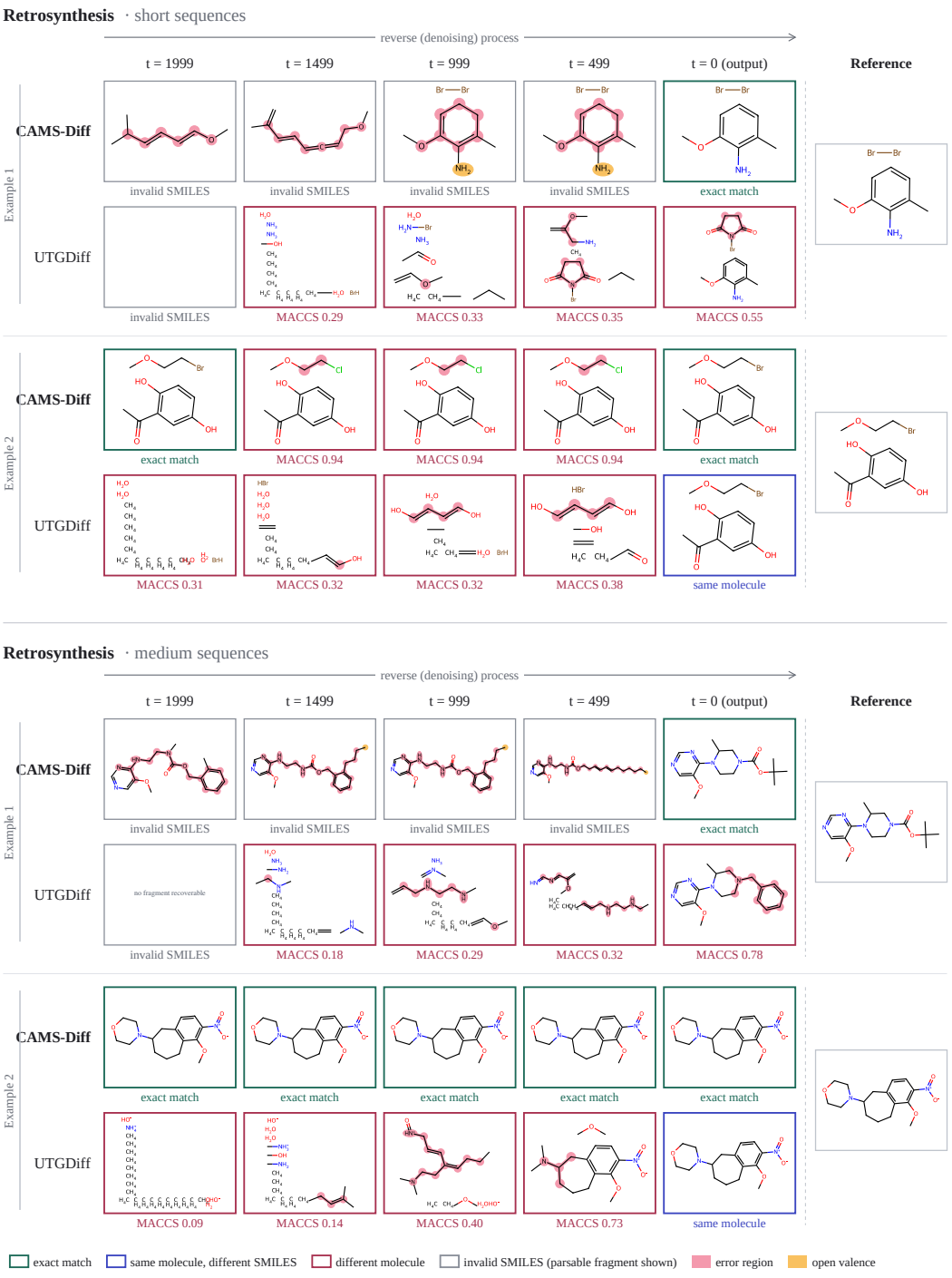


Figure 13: **Reverse-trajectory evolution for retrosynthesis (short and medium sequences).** Each cell decodes the clean-state readout $\widehat{Z}_0$ to SMILES at the indicated reverse step, and the right column shows the reference. Both methods use the full 2000-step reverse process (CAMS-Diff: ancestral; UTGDiff: DDIM). Frame colour gives the status of the decoded molecule; for different molecules, the label reports MACCS similarity to the reference. Pink marks atoms in the error region and orange marks open valences; for invalid SMILES, the largest parsable fragment is drawn. Examples are illustrative rather than population-level evidence.

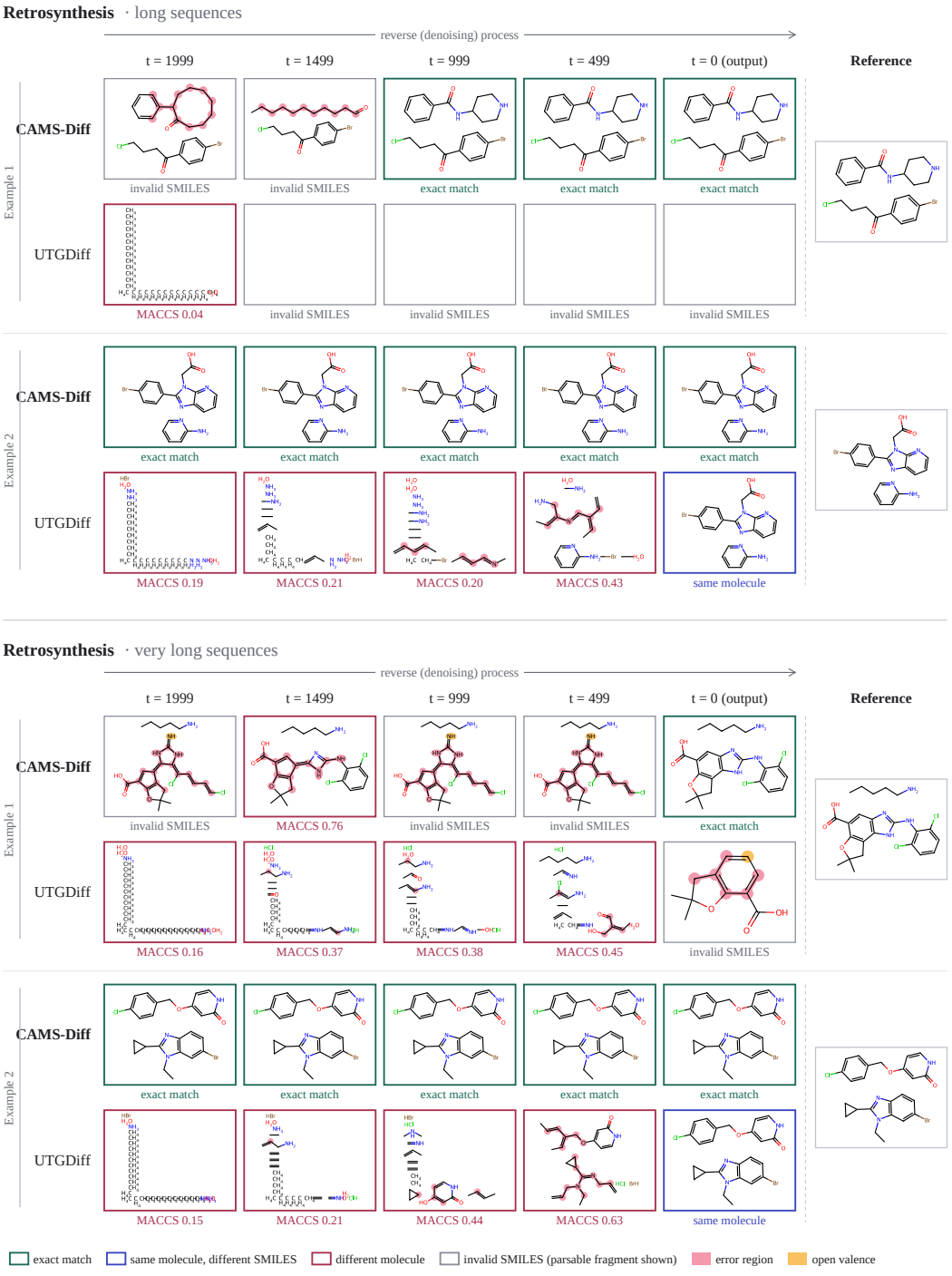


Figure 14: Retrosynthesis trajectories, long and very-long sequences. Notation as in Fig. 13.

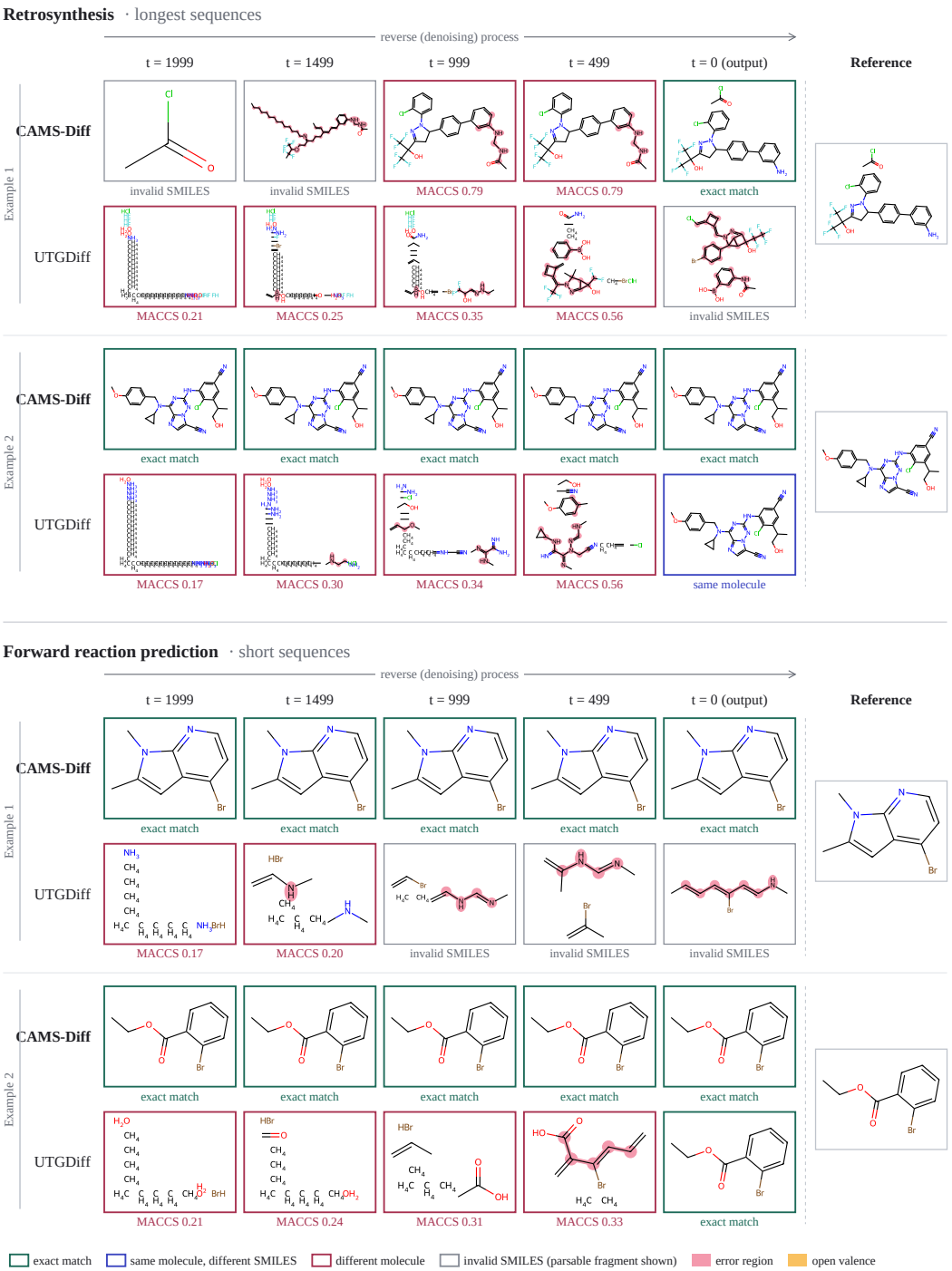


Figure 15: Retrosynthesis (longest) and forward reaction prediction (short) trajectories. Notation as in Fig. 13.

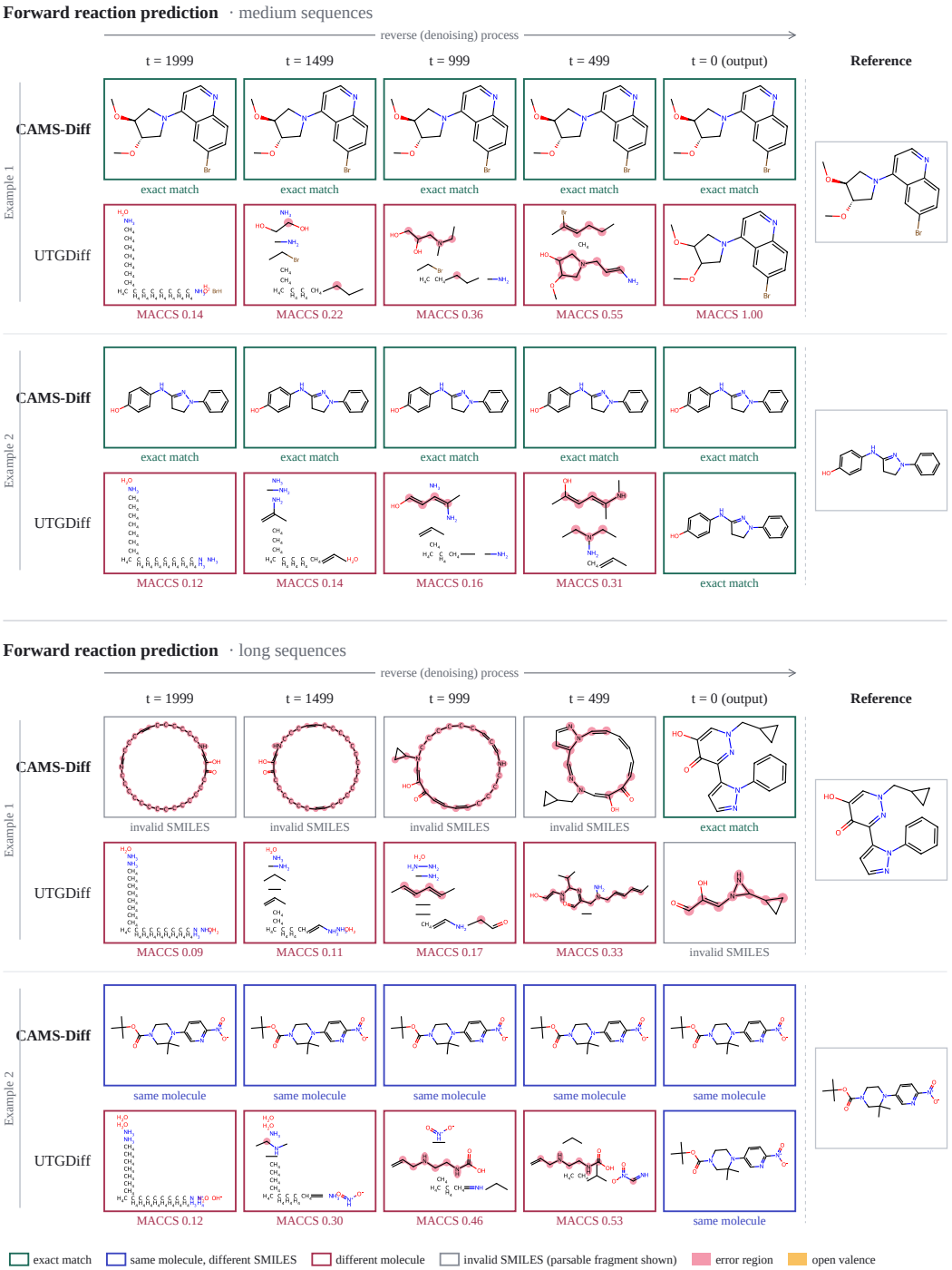


Figure 16: Forward reaction prediction trajectories, medium and long sequences. Notation as in Fig. 13.

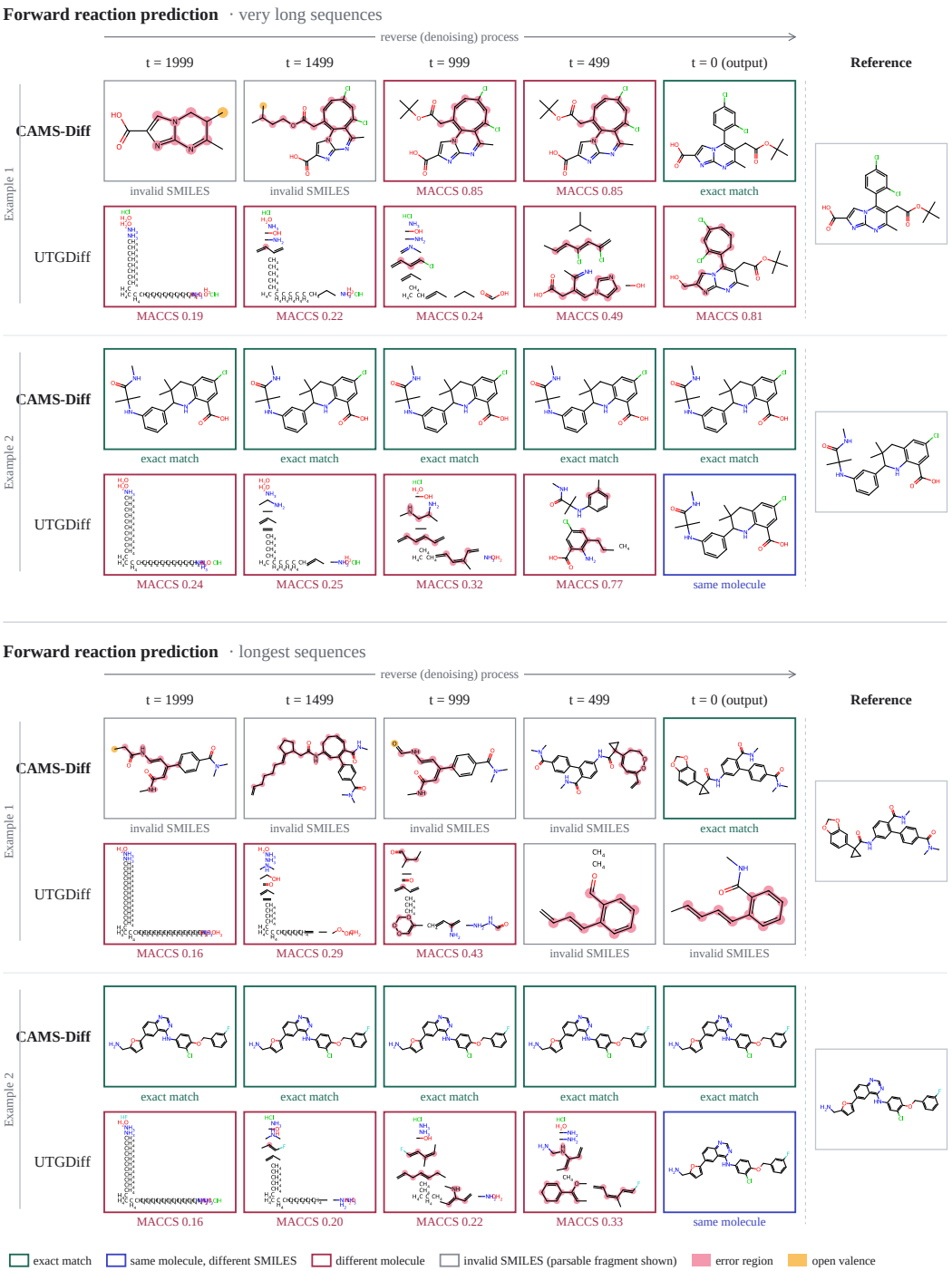


Figure 17: Forward reaction prediction trajectories, very-long and longest sequences. Notation as in Fig. 13.