CoRe-Gen: Robust Spectrum-to-Structure Generation under Imperfect Fingerprint Conditions

Anonymous authors
Paper under double-blind review

Abstract

Molecular structure elucidation from tandem mass spectra (MS/MS) remains challenging, particularly for de novo generation beyond database coverage. A common approach decomposes the task into spectrum-to-fingerprint prediction followed by fingerprint-to-structure decoding, enabling the use of large-scale molecular corpora. However, at deployment, the decoder relies on predicted rather than oracle fingerprints, introducing structured errors that propagate into generation. Moreover, common autoregressive decoders threshold fingerprint probabilities and serialize active bits as discrete input tokens, blocking molecular generation gradients from the spectrum encoder.

We present *CoRe-Gen*, which improves the intermediate condition through synthetic-spectrum pretraining, matches deployment-time noise through frequency-aware corruption, and introduces a differentiable fingerprint-to-memory bridge for joint encoder–decoder finetuning. The bridge preserves soft bit confidences and conditions a structure-aware autoregressive decoder without discrete fingerprint tokenization. CoRe-Gen achieves 21.49%/35.42% Top-1/Top-10 exact-match accuracy on NPLIB1 and, under encoder-aligned evaluation, 20.02%/22.45% on MassSpecGym, while retaining efficient autoregressive inference.

1 Introduction

Molecular structure elucidation from tandem mass spectrometry (MS/MS) is a fundamental problem in metabolomics, natural product discovery, and chemical analysis (Stein, 2012; Horai et al., 2010; Guijas et al., 2018). Despite its importance, reliable spectrum-to-structure generation remains challenging due to the scarcity and bias of high-quality spectrum–molecule pairs, the variability of spectra across instruments and acquisition settings, and the strict chemical validity constraints imposed on generated molecules (Scheubert et al., 2013; Stein, 2012; Bushuiev et al., 2024). These challenges are further amplified in realistic settings where models must generalize beyond curated datasets to noisy, heterogeneous spectra encountered in practice.

Existing approaches broadly fall into two paradigms: retrieval-based identification and de novo generation. Retrieval methods (Dührkop et al., 2015; Wang et al., 2016; Huber et al., 2021a; Wolf et al., 2010; Allen et al., 2014; Wang et al., 2022; Brouard et al., 2016; Schymanski et al., 2017; Horai et al., 2010; Guijas et al., 2018) map spectra into a searchable space and rank candidates from reference databases, achieving strong performance when the target molecule or its analogues are covered. However, they are fundamentally limited by database completeness and are sensitive to experimental mismatch. In contrast, de novo generation methods (Litsa et al., 2023; Liu et al., 2023; Stravs et al., 2022; Wang et al., 2025; Butler et al., 2023) aim to synthesize molecular structures directly from spectra, enabling discovery beyond existing databases. While recent neural models (Stravs et al., 2022; Litsa et al., 2023; Liu et al., 2023; Han et al., 2025; Wang et al., 2025; Bohde et al., 2025) have improved generation quality, they remain constrained by limited paired supervision and are often unstable under real-world spectral variability.

A practical compromise is fingerprint-conditioned generation, where a spectrum encoder predicts a molecular fingerprint that conditions downstream structure decoding (Goldman et al., 2024b; Le et al., 2020; Ucak et al., 2023; Stravs et al., 2022; Bohde et al., 2025; Han et al., 2025). While this

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

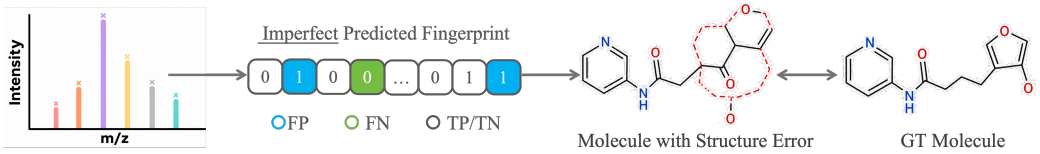


Figure 1: Imperfect Fingerprints Bottleneck Generation.

decomposition enables large-scale fingerprint–structure pretraining, many autoregressive implementations, including MolForge-style decoders and MS-BART, threshold the predicted fingerprint and serialize active bit indices as input tokens (Ucak et al., 2023; Han et al., 2025). This hard interface discards bit confidence and prevents the molecular generation loss from propagating through the thresholding/index-selection step to the spectrum encoder. Meanwhile, scarce and biased spectrum–molecule supervision makes fingerprint predictions noisy, especially for long-tail bits, producing structured errors that propagate into generation (Figure 1).

This exposes two coupled gaps: decoders are trained on clean fingerprints but deployed on noisy predictions, and their discrete condition interface prevents end-to-end correction of the upstream encoder. We therefore improve the predicted condition, match its corruption during training, preserve its soft confidence through a differentiable bridge, and exploit residual uncertainty during decoding.

To this end, we propose *CoRe-Gen* to explicitly address imperfect fingerprint conditions in spectrum-to-structure generation. Our framework is built on four complementary principles. (1) *Improve condition quality*: we pretrain the spectrum encoder on large-scale synthetic spectra from approximately 790K valid organic molecules, each paired with five types of in-silico MS/MS spectra, to enhance fingerprint prediction under limited real supervision. While synthetic data is less suitable for end-to-end de novo generation due to the domain gap between simulated and real spectra, fingerprint prediction provides a coarse and redundant intermediate target that is substantially more robust to such imperfections, making it a more effective and transferable pretraining objective. (2) *Match the condition to reduce the train–test gap*: we introduce frequency-aware fingerprint corruption during decoder pretraining to simulate realistic prediction noise, thereby aligning training conditions with deployment-time inputs. (3) *Enable joint finetuning*: a differentiable fingerprint-to-memory bridge converts soft bit probabilities directly into continuous decoder memory, allowing sequence loss to update the spectrum encoder. (4) *Mitigate imperfect conditions via structural regularization*: we design a structure-aware autoregressive decoder that incorporates compositional SELFIES representations (Krenn et al., 2020), auxiliary structural supervision, and lightweight chemical constraints to stabilize generation under uncertainty. By jointly addressing condition quality, condition mismatch, and robustness, CoRe-Gen transforms fingerprint-conditioned generation from an idealized assumption into a more practical and reliable pipeline.

Extensive experiments demonstrate the effectiveness of CoRe-Gen. On NPLIB1, it reaches 21.49%/35.42% Top-1/Top-10 exact-match accuracy and 0.57/0.67 Tanimoto similarity. On MassSpecGym, an encoder-aligned comparison yields 20.02%/22.45% exact-match accuracy and 0.40/0.47 Tanimoto similarity. CoRe-Gen also preserves efficient autoregressive decoding, requiring approximately 2 seconds per spectrum for 100 candidates on a single RTX 4090 GPU, compared with approximately 160 seconds for DiffMS.

Our contributions are summarized as follows:

- We identify imperfect predicted fingerprints as the central bottleneck in practical spectrum-to-structure generation and reformulate the problem as robust generation under condition mismatch.
- We propose a unified solution that improves fingerprint prediction via synthetic pretraining, matches deployment noise through corruption-aware training, and exploits imperfect conditions with structure-aware decoding.
- We replace hard fingerprint tokenization with a differentiable fingerprint-to-memory bridge, enabling joint encoder–decoder finetuning while retaining autoregressive efficiency.
- CoRe-Gen reaches 21.49%/35.42% exact-match accuracy on NPLIB1 and 20.02%/22.45% on MassSpecGym at Top-1/Top-10.

2 Related Work

Spectrum-to-Structure Elucidation. Spectrum-to-structure methods broadly follow retrieval-based identification or de novo generation. Retrieval systems search spectral libraries or rank database candidates using fragmentation and learned scoring, including GNPS, MassBank, METLIN, SIRIUS/CSI:FingerID, MetFrag, CFM-ID, IOKR, and molDiscovery (Stein, 2012; Wang et al., 2016; Horai et al., 2010; Guijas et al., 2018; Dührkop et al., 2015; 2019; Wolf et al., 2010; Allen et al., 2014; Wang et al., 2022; Brouard et al., 2016; Cao et al., 2021). They are effective when relevant candidates are covered, but remain limited by database coverage and experimental mismatch (Schymanski & Neumann, 2013; Schymanski et al., 2017). De novo methods instead generate beyond observed libraries using sequence, retrieval-guided, or graph-based models (Litsa et al., 2023; Liu et al., 2023; Butler et al., 2023; Wang et al., 2025; Bohde et al., 2025; Han et al., 2025), but remain sensitive to scarce paired supervision and spectral variability.

Learned spectral embeddings and molecular fingerprints bridge these paradigms by providing chemically meaningful intermediate targets for similarity learning and downstream prediction (Huber et al., 2021a;b; Li et al., 2021; de Jonge et al., 2023; Goldman et al., 2024b; Zhang et al., 2024; Wei et al., 2019; Ji et al., 2020; Young et al., 2024). We focus on fingerprint prediction not as an endpoint, but as an imperfect intermediate condition for molecular generation.

Fingerprint-Conditioned Generation under Imperfect Conditions. Fingerprint-conditioned generation is closest to our setting because it decouples spectrum interpretation from molecular decoding. Neural Decipher and MSNovelist invert ECFP-like fingerprints, while subsequent work combines fingerprint conditions with pretraining, retrieval guidance, and graph-based generation (Le et al., 2020; Stravs et al., 2022; Wang et al., 2025; Bohde et al., 2025; Han et al., 2025). MolForge represents a binary fingerprint as the sequence of active bit indices, and MS-BART similarly thresholds predicted probabilities before mapping active bits to fingerprint tokens (Ucak et al., 2023; Han et al., 2025). Although effective for autoregressive generation, this serialization discards confidence and blocks gradients through the discrete interface. Our bridge instead maps all soft bit probabilities to continuous decoder memory, enabling joint finetuning. We also address the clean-to-predicted mismatch caused by structured false-positive and false-negative errors. Graph generators can impose strong structural constraints but incur higher sampling cost (Bohde et al., 2025). Autoregressive sequence models remain attractive at scale because of their efficiency, but are more vulnerable to local error accumulation under imperfect conditions. Our approach therefore adds lightweight chemical priors to efficient autoregressive decoding while explicitly training for this mismatch.

3 Method

Given an MS/MS spectrum $\mathbf{x}$, our goal is to generate the target molecular sequence $\mathbf{y}$ through an intermediate molecular fingerprint $\mathbf{f} \in \{0,1\}^d$. Fingerprint-conditioned generation decomposes the problem as

$$p_\theta(\mathbf{y} \mid \mathbf{x}) \approx \sum_{\mathbf{f}} p_{\theta_e}(\mathbf{f} \mid \mathbf{x})\, p_{\theta_d}(\mathbf{y} \mid \mathbf{f}), \tag{1}$$

where θ_e and θ_d denote encoder and decoder parameters, respectively. At inference time, the decoder receives the predicted soft fingerprint $\hat{\mathbf{f}} \in [0,1]^d$ rather than the oracle fingerprint. Instead of thresholding it into a discrete index sequence, CoRe-Gen constructs continuous memory $\mathbf{M} = h_\phi(\hat{\mathbf{f}})$ and optimizes generation through this bridge:

$$\max_{\theta_e,\phi,\theta_d} \mathbb{E}_{(\mathbf{x},\mathbf{y})} \left[\log p_{\theta_d}\Big(\mathbf{y} \mid h_\phi(\hat{\mathbf{f}})\Big) \right], \qquad \hat{\mathbf{f}} = g_{\theta_e}(\mathbf{x}). \tag{2}$$

We propose CoRe-Gen, a condition-centric framework whose three robustness components are connected by a differentiable fingerprint interface. As illustrated in Figure 2, we first *improve the condition* through synthetic-spectrum encoder pretraining, then *match the condition* by exposing the decoder to structured fingerprint errors, and finally *exploit the imperfect condition* with structure-aware autoregressive decoding. The differentiable bridge below couples the encoder and decoder during final finetuning.

Differentiable fingerprint-to-memory bridge. Prior sequence decoders commonly retain only thresholded active bits. We instead keep all soft probabilities and fold the $d = 4096$ dimensions

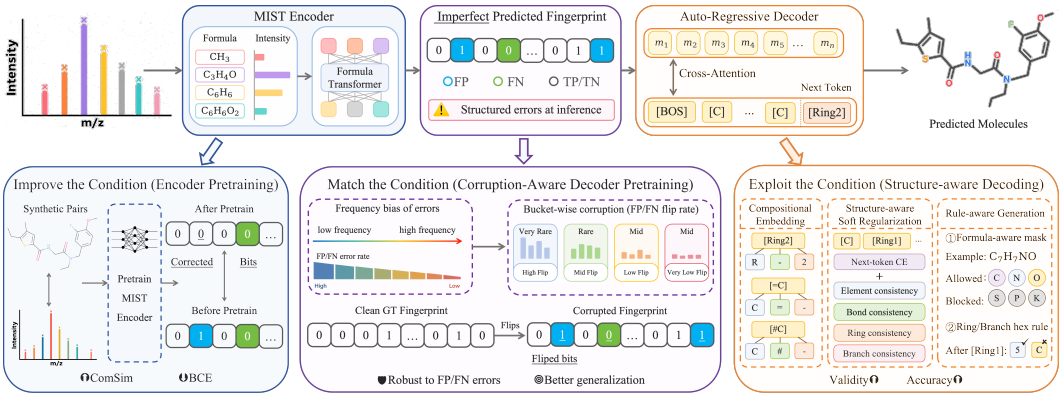


Figure 2: Overview of CoRe-Gen: improving, matching, and exploiting imperfect fingerprint conditions for robust spectrum-to-structure generation.

into $C = 64$ modulo groups. For group i, learned bit embeddings are confidence-weighted and normalized, $\mathbf{c}_i = \sum_k \hat{f}_{i+kC}\mathbf{e}_{i+kC}/(\sum_k \hat{f}_{i+kC} + \epsilon)$. The 64 chunk features are contextualized into fixed-length memory tokens and consumed by decoder cross-attention. Because the bridge contains no threshold or index-selection operation, the autoregressive loss backpropagates through the memory tokens into the spectrum encoder, enabling joint finetuning. Appendix E gives the full architecture and illustration.

3.1 Improve the Condition: Encoder Pretraining

We instantiate $p_{\theta_e}(\mathbf{f} \mid \mathbf{x})$ with the MIST spectrum encoder and a lightly adapted FP-Growing head that predicts Morgan fingerprints with $d = 4096$ bits. The main weakness of this spectrum-to-fingerprint interface is not architectural capacity alone, but insufficient supervision for most fingerprint dimensions. As shown in Figure 3, active bits in real paired datasets are highly long-tailed: many substructure bits appear in only a tiny fraction of training spectra, so the encoder receives few positive examples for learning them.

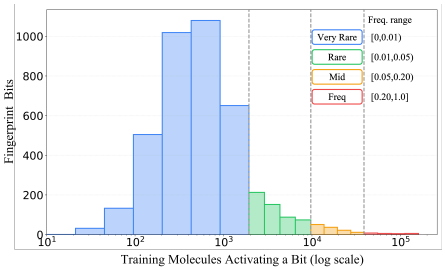


Figure 3: Long-tail distribution of active fingerprint bits in real paired data.

This long-tailed supervision directly degrades the predicted condition. Low-frequency bits are systematically under-trained, making the encoder less reliable for rare but chemically informative substructures. To alleviate this limitation, we introduce large-scale synthetic-spectrum pretraining. The synthetic pretraining data come from a large-scale simulated spectroscopic corpus containing approximately 790K valid organic molecules Alberts et al. (2024). Each molecule is associated with five types of in-silico MS/MS spectra: CFM-ID 4.0 spectra at collision energies of 10, 20, and 40 eV Allen et al. (2014); Dührkop et al. (2019), together with spectra generated by ICEBERG Goldman et al. (2024a) and SCARF Goldman et al. (2023). The combination of multiple simulators and collision-energy settings yields a substantially larger and more diverse spectrum–molecule corpus than typical experimental MS/MS libraries.

Compared with the MassSpecGym training set, this corpus substantially densifies the long-tail fingerprint space: fingerprint bits that occur at most 20 times in MassSpecGym are observed on average 1,164.8 times in the pretraining set, and all 199 such rare bits receive at least a 10× increase in occurrence count. Although simulated spectra are lower-fidelity than experimental spectra and may not fully capture real instrument noise, peak intensities, or acquisition-specific artifacts, their fragment peaks still provide useful chemical evidence for substructure-level fingerprint bits. Therefore, synthetic data are well suited for pretraining spectrum encoders to learn general spectrum-to-substructure associations, followed by fine-tuning on real spectra for adaptation to experimental noise and acquisition conditions.

After pretraining, the encoder is fine-tuned on real spectra with the standard BCE objective, with optional bit-frequency reweighting so that rare but informative bits are not overwhelmed by frequent fingerprint dimensions. Table 3 reports the effect of encoder pretraining on both benchmarks. In our framework, encoder pretraining matters because it improves the downstream usability of predicted fingerprints, especially on weakly supervised long-tail bits.

3.2 Match the Condition: Corruption-Aware Decoder Pretraining

Decoder pretraining on clean fingerprint-molecule pairs creates a condition mismatch: the decoder learns from oracle fingerprints but is deployed on predicted fingerprints. We reduce this gap by corrupting clean fingerprints with structured errors that reflect bit-frequency-dependent prediction behavior. Let c_j denote the occurrence frequency of bit j in the training corpus. Motivated by the long-tailed distribution in Figure 3, we partition the d bits into four frequency buckets $\{B_r\}_{r=1}^{4}$ by quantiles of $\{c_j\}_{j=1}^{d}$, corresponding to very-rare, rare, mid, and frequent bits, and let $b(j)$ denote the bucket index of bit j.

Table 1 shows this structure on MassSpecGym: very-rare bits dominate the fingerprint space but are predicted with much lower precision than frequent bits. We therefore model corruption at the bucket level instead of applying uniform random noise. For each bucket B_r, we estimate false-negative and false-positive tendencies from training data only (never from a downstream test set):

Table 1: MassSpecGym bit-frequency buckets for encoder prediction.

Bucket	# bits	Freq. range	Precision
Very rare	3422	$[0, 0.01)$	0.198
Rare	527	$[0.01, 0.05)$	0.314
Mid	122	$[0.05, 0.20)$	0.428
Frequent	25	$[0.20, 1.0]$	0.692

$$\eta_r^- = 1 - \mathrm{Recall}(B_r), \qquad \eta_r^+ = 1 - \mathrm{Precision}(B_r). \tag{3}$$

To avoid unstable per-bit estimates and degenerate zero-probability buckets, we convert these tendencies into bucket-level sampling weights:

$$w_r^{\pm} = \epsilon + \alpha \cdot \frac{\eta_r^{\pm} - \min_s \eta_s^{\pm}}{\max_s \eta_s^{\pm} - \min_s \eta_s^{\pm}}, \tag{4}$$

where ϵ is a small floor value and α controls the strength of the frequency-aware corruption pattern.

For a training sample with active set $A(\mathbf{f}) = \{j : f_j = 1\}$ and inactive set $\bar{A}(\mathbf{f}) = \{j : f_j = 0\}$, we first draw a corruption gate

$$z \sim \mathrm{Bernoulli}(p_{\mathrm{corr}}). \tag{5}$$

If $z = 0$, we keep the fingerprint unchanged. Otherwise, we draw a corruption budget

$$k \sim \mathrm{clip}(\mathrm{Poisson}(\lambda), k_{\min}, k_{\max}), \qquad k_{\mathrm{eff}} = \min\big(k,\ |A(\mathbf{f})|,\ |\bar{A}(\mathbf{f})|\big). \tag{6}$$

In our implementation, λ is set according to the maximum corruption budget, i.e., $\lambda = k_{\max}/2$.

We then sample a drop set $S^- \subseteq A(\mathbf{f})$ and an add set $S^+ \subseteq \bar{A}(\mathbf{f})$, both of size k_{eff}, using the normalized bucket-level weights:

$$p_j^- = \frac{w_{b(j)}^-}{\sum_{\ell \in A(\mathbf{f})} w_{b(\ell)}^-} \ (j \in A(\mathbf{f})), \qquad p_j^+ = \frac{w_{b(j)}^+}{\sum_{\ell \in \bar{A}(\mathbf{f})} w_{b(\ell)}^+} \ (j \in \bar{A}(\mathbf{f})). \tag{7}$$

The corrupted fingerprint $\tilde{\mathbf{f}}$ is constructed by

$$\tilde{f}_j = \big\{0 \text{ if } j \in S^-; \quad 1 \text{ if } j \in S^+; \quad f_j \text{ otherwise}\big\}. \tag{8}$$

This sparsity-preserving swap injects frequency-aware structured FP/FN errors. Then the decoder is pretrained to generate the target modified SELFIES sequence from $\tilde{\mathbf{f}}$, so its conditioning distribution more closely matches deployment-time predicted fingerprints.

3.3 Exploit the Imperfect Condition: Structure-Aware AR Decoding

Even after improving and matching the fingerprint condition, residual fingerprint errors can still be amplified by token-level autoregressive decoding. The role of structure-aware decoding is to stabilize local chemical recovery when the fingerprint condition remains imperfect. We therefore add three

lightweight structural priors to the decoder: compositional SELFIES embeddings, auxiliary structural supervision, and rule-aware logit masking.

Compositional SELFIES embedding. Instead of treating SELFIES tokens as independent identities, we parse each token t into a compositional tuple $(p_t, a_t, b_t, q_t, d_t)$ consisting of part type, element, bond prefix, ring/branch order, and optional hex digit. This factorization injects lightweight structural priors that are usually more explicit in graph-based models. For example, tokens such as `C` and `=C` should share their carbon-atom identity while preserving their different bond contexts. By sharing factor embeddings across such related tokens, the decoder can reuse chemical regularities across the vocabulary while still distinguishing the factors that matter for generation. This also reduces the effective token vocabulary and parameter cost, improving computational efficiency. Appendix F describes the detailed modified SELFIES interface. The token embedding is then built additively:

$$\begin{aligned}\mathbf{h}_t = \mathbf{e}_{\text{part}}(p_t) + \mathbf{e}_{\text{elem}}(a_t) + \mathbf{e}_{\text{bond}}(b_t) \\ + \mathbb{I}(q_t > 0)\,\mathbf{e}_{\text{rb}}(q_t) + \mathbb{I}(d_t \geq 0)\,\mathbf{e}_{\text{digit}}(d_t) \\ + \mathbb{I}(c(\tilde{y}_t) \geq \tau)\,\mathbf{e}_{\text{res}}(\tilde{y}_t),\end{aligned} \tag{9}$$

where $\tilde{y}_t$ is a canonicalized token form (removing isotope/chirality/H/charge markers while preserving bond prefix and element), $c(\cdot)$ is its corpus count, and τ is a frequency threshold. The residual term is enabled only for high-frequency canonical tokens, which preserves flexibility on common patterns without overfitting long-tail symbols.

Auxiliary structural supervision. The same molecule may admit multiple valid SELFIES strings, so pure token-level cross-entropy can over-penalize sequence variants while missing higher-level structural consistency. However, key molecular statistics are invariant across equivalent representations: the molecular formula is fixed, and so are coarse structural counts such as the number of branch controls, ring controls, and double or triple bonds. These invariants give us an opportunity to supervise generation at the topological-structure level rather than only at the next-token level. We therefore augment token cross-entropy with auxiliary losses over the corresponding chemical factors:

$$\mathcal{L}_{\text{dec}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{sent}} \left(w_{\text{elem}}\mathcal{L}_{\text{elem}} + w_{\text{bond}}\mathcal{L}_{\text{bond}} + w_{\text{ring}}\mathcal{L}_{\text{ring}} + w_{\text{branch}}\mathcal{L}_{\text{branch}} \right), \tag{10}$$

where auxiliary cross-entropy losses regularize predictions on the corresponding chemical factors. Detailed definitions and implementation of these auxiliary losses are provided in Appendix F.2.

Rule-aware logit masking. Following DiffMS, at inference time on both NPLIB1 and MassSpecGym we use the benchmark-provided ground-truth molecular formula, not a separately predicted formula. The same formula is used by a lightweight constraint controller during autoregressive decoding and by the post-generation reranker. Given decoder logits $\mathbf{z}_t \in \mathbb{R}^{|\mathcal{V}|}$ at time step t, we apply a rule-dependent mask $\mathbf{m}_t$ and decode from

$$\tilde{\mathbf{z}}_t = \mathbf{z}_t + \mathbf{m}_t, \qquad m_{t,j} \in \{0, -\infty\}, \tag{11}$$

where invalid tokens receive $-\infty$ and are therefore removed from the candidate set. In our structure-aware decoding design, we use two lightweight chemistry-aware constraints:

- disallow atom tokens whose element does not appear in the target molecular formula;
- after a ring or branch token is emitted, enforce the required hexadecimal continuation and forbid isolated hex tokens elsewhere.

These constraints directly target frequent failure modes of autoregressive SELFIES generation: producing chemically impossible elements and mismatching ring/branch control tokens with their numeric continuations.

4 Experiments

4.1 Experimental Setup

Datasets and baselines. The encoder is pretrained on the simulated multimodal spectroscopy dataset from Alberts et al. (2024). The decoder is pretrained on the same 2.8M fingerprint-molecule corpus organized by DiffMS, sampled from DSSTox, HMDB, COCONUT, and MOSES (Bohde et al., 2025;

Table 2: Main comparison on NPLIB1 and MassSpecGym. We report Top-1/Top-10 Accuracy (↑), MCES (↓), and Tanimoto (↑). Results marked with * are reproduced from benchmark or prior reports. † uses the publicly released DiffMS MassSpecGym encoder checkpoint. Bold indicates the best and underlined indicates the second-best.

Model	Top-1			Top-10		
	Accuracy↑	MCES↓	Tanimoto↑	Accuracy↑	MCES↓	Tanimoto↑
NPLIB1						
Spec2Mol*	0.00%	27.82	0.12	0.00%	23.13	0.16
MIST + NeuralDecipher*	2.32%	12.11	0.35	6.11%	9.91	0.43
MIST + MSNovelist*	5.40%	14.52	0.34	11.04%	10.23	0.44
MADGEN	2.10%	20.56	0.22	2.39%	12.69	0.27
DiffMS	8.34%	11.95	0.35	15.44%	9.23	0.47
MS-BART	7.45%	9.66	0.44	10.99%	8.31	0.51
CoRe-Gen	**21.49%**	**5.57**	**0.57**	**35.42%**	**4.03**	**0.67**
MassSpecGym						
SMILES Transformer*	0.00%	79.39	0.03	0.00%	52.13	0.10
SELFIES Transformer*	0.00%	38.88	0.08	0.00%	26.87	0.13
Random Generation*	0.00%	21.11	0.08	0.00%	18.26	0.11
Spec2Mol*	0.00%	37.76	0.12	0.00%	29.40	0.16
MIST + NeuralDecipher*	0.00%	33.19	0.14	0.00%	31.89	0.16
MIST + MSNovelist*	0.00%	45.55	0.06	0.00%	30.13	0.15
MADGEN	1.31%	27.47	0.20	1.54%	16.84	0.26
MS-BART	1.07%	16.47	0.23	1.11%	15.12	0.28
FOAM (DiffMS + PubChem)	1.50%	**12.21**	0.35	10.28%	**6.06**	**0.53**
DiffMS	2.30%	18.45	0.28	4.25%	14.73	0.39
CoRe-Gen†	**20.02%**	16.09	**0.40**	**22.45%**	13.90	0.47

Wishart et al., 2022; Sorokina et al., 2021; Polykovskiy et al., 2020). For finetuning and evaluation, we follow the official splits of NPLIB1 and MassSpecGym (Bushuiev et al., 2024). To prevent data leakage, we remove from both the encoder and decoder pretraining corpora any molecules that overlap with the test sets of NPLIB1 or MassSpecGym. We compare against three categories of baselines: (i) naïve autoregressive models (SMILES/SELFIES Transformers), (ii) standard fingerprint-conditioned methods (Spec2Mol, MIST+MSNovelist, and MIST+NeuralDecipher), and (iii) recent leading systems (MADGEN, DiffMS, MS-BART, and FOAM) (Goldman et al., 2024b; Stravs et al., 2022; Bohde et al., 2025; Han et al., 2025; Wang et al., 2025; Manjrekar et al., 2026).

Metrics. We report Top-k metrics with $k \in \{1, 10\}$. Our primary identification metric is Top-k exact match, computed by converting each generated molecule to a full InChIKey and checking whether the ground-truth molecule appears in the top-k set. As structural-similarity metrics, we report Top-k maximum Tanimoto similarity using RDKit Morgan fingerprints (radius $= 2$, length $= 4096$) and Top-k minimum MCES to quantify graph-level overlap between predictions and targets.

Formula information, decoding, and reranking. Following DiffMS, all main experiments on both benchmarks use the benchmark-provided ground-truth molecular formula rather than a separately predicted formula. CoRe-Gen uses it to mask atom tokens whose elements are absent from the formula, but does not fix elemental counts or guarantee an exact formula match. Following MS-BART, we generate 100 beam-search candidates per spectrum (beam size 100) and rerank them by formula distance $D(F_1, F_2) = \sum_{a \in \mathcal{A}} |n_a - m_a|$, breaking ties by model-estimated log-probability. DiffMS instead fixes the heavy-atom inventory before graph generation and ranks 100 sampled graphs by sampling frequency, so equivalent post-generation formula reranking is unnecessary. The candidate budget, formula information, and evaluation metrics are matched; generation-specific ranking is retained for each paradigm. Appendix C gives the full comparison and evaluates MSBuddy-predicted formulas.

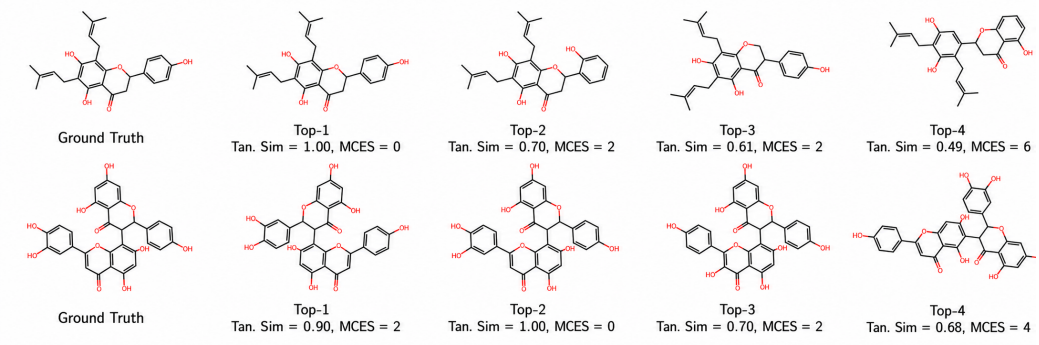


Figure 4: Representative successful generation examples of CoRe-Gen.

4.2 Main Results

Before evaluating molecular generation, we first verify that encoder pretraining substantially improves the predicted fingerprint condition. As shown in Table 3, pretraining reduces BCE by 29.03% on NPLIB1 and 23.81% on MassSpecGym, while also improving probability-level cosine similarity. This confirms that the downstream decoder receives a more usable condition before generation is evaluated.

Table 2 summarizes overall performance. On NPLIB1, CoRe-Gen raises Top-1 exact-match accuracy from 8.34% (DiffMS) to 21.49% and Top-10 accuracy from 15.44% to 35.42%. Its Top-1/Top-10 MCES reaches 5.57/4.03, while Tanimoto similarity reaches 0.57/0.67. Together with Table 5, these gains show that preserving soft fingerprint confidence and jointly adapting the encoder and decoder substantially improves both exact recovery and structural similarity.

Table 3: Encoder pretraining improves fingerprint prediction.

Metric	Before	CoRe-Gen	Impr. (%)
NPLIB1			
BCE↓	0.031	0.022	-29.03
CosSim↑	0.705	0.756	+7.23
MassSpecGym			
BCE↓	0.063	0.048	-23.81
CosSim↑	0.490	0.561	+14.49

On MassSpecGym, our independently trained MIST encoder did not reproduce the fingerprint quality of the publicly released DiffMS checkpoint despite following the released code, configurations, and reported hyperparameters. To align the encoder condition with the compared DiffMS-based methods, Table 2 therefore uses that public checkpoint; the CoRe-Gen decoder, data split, formula information, candidate budget, reranking, and metrics remain unchanged. Under this aligned setting, CoRe-Gen reaches 20.02%/22.45% Top-1/Top-10 exact match, 16.09/13.90 MCES, and 0.40/0.47 Tanimoto similarity. It achieves the best exact-match and Top-1 Tanimoto results, while FOAM remains strongest on MCES and Top-10 Tanimoto. Appendix B reports the reproduction details, independently trained encoder results, and fingerprint quality.

CoRe-Gen is also faster at inference, generating 100 candidates in approximately 2 seconds per spectrum on a single RTX 4090 GPU—about 80× faster than DiffMS in our setup. Figure 4 shows representative successful generations, with additional success and failure cases reported in the appendix.

Table 4: Inference time per spectrum for 100 candidates on a single RTX 4090 GPU.

Method	Time (s)	Rel. Speedup
DiffMS	≈160	1.00×
MADGEN	≈29	5.52×
CoRe-Gen	≈2	80.0×

4.3 Efficiency and Inference Cost

At inference time, CoRe-Gen uses beam-search decoding for structure generation. Thanks to structure-aware compositional embeddings, the effective decoding vocabulary is compact (43 tokens), which reduces branching complexity per step and improves practical decoding efficiency while preserving chemical validity constraints.

We systematically evaluate beam width and observe that increasing beam size yields clear Top-1/Top-10 gains on NPLIB1 while keeping latency practical. As summarized in Table 4, in a deployment-oriented setting that produces 100 candidates per spectrum on a single RTX 4090 GPU, CoRe-Gen requires approximately 2 seconds per spectrum, compared with approximately 29 seconds for MADGEN and 160 seconds for DiffMS. This corresponds to roughly a 15× speedup over MADGEN and an ∼80× speedup over DiffMS, showing that our structure-aware sequence decoder retains the practical efficiency advantage of autoregressive generation while remaining competitive in structural recovery.

4.4 Ablation Studies

We conduct ablation studies on NPLIB1. Table 5 tests both the condition-centric components and the differentiable interface. Removing joint finetuning reduces Top-1/Top-10 accuracy from 21.49%/35.42% to 19.54%/29.92%, with Top-1 Tanimoto decreasing from 0.57 to 0.56. The larger Top-10 gain indicates that end-to-end adaptation improves candidate-set coverage and ranking rather than only the first output.

Table 5: Core ablations of CoRe-Gen.

Variant	Top-1	Top-10	Top-1 Tan.
Full method	21.49	35.42	0.57
w/o joint finetune	19.54	29.92	0.56
w/o enc. pretrain	11.46	19.80	0.48
w/o corr. pretrain	14.77	23.93	0.53
w/o comp. SELFIES	18.19	26.86	0.55
w/o soft reg.	19.23	29.34	0.56
w/o rule-aware gen.	19.17	28.57	0.56
w/o exploit trio	17.95	26.98	0.55
GT fingerprint	82.91	87.30	0.94

The largest contribution comes from encoder pretraining. Removing it drops Top-1 accuracy from 21.49% to 11.46% and Top-10 accuracy from 35.42% to 19.80%, showing that synthetic spectrum-to-fingerprint supervision is critical for learning a reliable condition signal before adaptation on scarce real paired data. This large gain is consistent with the fingerprint-prediction improvements reported in Table 3, where pretraining substantially improves the predicted fingerprints used by the downstream generator.

Corruption-aware decoder pretraining is also important. Without it, Top-1 and Top-10 accuracy decrease to 14.77% and 23.93%, respectively, confirming that the decoder must be trained under noisy predicted-fingerprint conditions rather than only clean oracle fingerprints.

Compositional SELFIES embedding and its lightweight refinements provide smaller but consistent additional gains. Removing the compositional SELFIES embedding lowers Top-1/Top-10 accuracy to 18.19%/26.86%, while removing soft regularization or rule-aware generation produces more modest declines. These results indicate that compositional tokenization supplies the main local chemical inductive bias, with auxiliary supervision and decoding rules serving as useful refinements on top of improved and better-matched conditions. When we remove the full exploit module ('w/o exploit trio'), Top-1 accuracy further drops to 17.95%, confirming that the combined effect of compositional embedding, soft regularization, and rule-aware generation is meaningful under imperfect fingerprint conditioning.

Finally, the 'GT fingerprint' row (Top-1 82.91%, Top-10 87.30%) indicates substantial headroom beyond all predicted-fingerprint settings, suggesting that the dominant bottleneck is still the encoder's ability to predict highly accurate fingerprints.

5 Conclusion

We presented a condition-centric framework for fingerprint-conditioned spectrum-to-structure generation that treats imperfect predicted fingerprints as the central bottleneck of practical deployment. CoRe-Gen improves the predicted condition through synthetic-spectrum encoder pretraining, matches deployment-time corruption through corruption-aware decoder pretraining, and replaces the discrete fingerprint handoff with a differentiable memory bridge for joint encoder–decoder adaptation. It then exploits residual condition noise through structure-aware autoregressive decoding. Across standard benchmarks, this design surpasses leading compared methods on NPLIB1 and, under an encoder-aligned evaluation, on MassSpecGym exact-match accuracy, while preserving the practical efficiency advantage of autoregressive generation. More broadly, robust spectrum-to-structure generation should model intermediate-condition uncertainty through stronger encoders, condition-aligned pretraining, and deployment-robust decoding.

AI USE STATEMENT

In preparing this manuscript, we used generative AI tools to assist with language editing, document restructuring, LaTeX formatting, reference-format checking, and migration to the ICLR 2027 template. The authors reviewed the AI-assisted edits against the underlying text, numerical results, citations, tables, and figures. The authors take responsibility for the final content of this work, including text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

CoRe-Gen produces ranked structural hypotheses rather than definitive molecular identifications. Its outputs should be validated with reference spectra or orthogonal experimental evidence before being used in scientific, clinical, forensic, or safety- critical decisions. Like other structure-elucidation systems, it also has potential dual-use implications for identifying harmful or illicit compounds. We discuss these limitations and responsible-use considerations further in Appendix K.

REPRODUCIBILITY STATEMENT

The appendix documents the datasets and split rules, molecular and spectral representations, training schedule and hyperparameters, decoding and reranking protocols, encoder-alignment procedure, baseline-comparison settings, and variation across random seeds. It also reports robustness analyses for predicted formulas, training-set coverage, scaffold recovery, and library depletion. Together with the equations and algorithmic descriptions in the main paper, these details specify the reported experimental pipeline and evaluation procedure.

REFERENCES

Marvin Alberts, Oliver Schilter, Federico Zipoli, Nina Hartrampf, and Teodoro Laino. Unraveling molecular structure: A multimodal spectroscopic dataset for chemistry. *Advances in neural information processing systems*, 37:125780–125808, 2024.

Felicity Allen, Allison Pon, Michael Wilson, Russ Greiner, and David Wishart. CFM-ID: a web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra. *Nucleic acids research*, 42(W1):W94–W99, 2014.

Montgomery Bohde, Mrunali Manjrekar, Runzhong Wang, Shuiwang Ji, and Connor W Coley. DiffMS: Diffusion generation of molecules conditioned on mass spectra. *arXiv preprint arXiv:2502.09571*, 2025.

Céline Brouard, Huibin Shen, Kai Dührkop, Florence d'Alché Buc, Sebastian Böcker, and Juho Rousu. Fast metabolite identification with input output kernel regression. *Bioinformatics*, 32(12): i28–i36, 2016.

Roman Bushuiev, Anton Bushuiev, Niek F de Jonge, Adamo Young, Fleming Kretschmer, Raman Samusevich, Janne Heirman, Fei Wang, Luke Zhang, Kai Dührkop, et al. MassSpecGym: A benchmark for the discovery and identification of molecules. *Advances in Neural Information Processing Systems*, 37:110010–110027, 2024.

Thomas Butler, Abraham Frandsen, Rose Lightheart, Brian Bargh, Thomas Kerby, Kiana West, Joseph Davison, James Taylor, Christoph Krettler, TJ Bollerman, et al. MS2Mol: A transformer model for illuminating dark chemical space from mass spectra. *ChemRxiv*, 2023. doi: 10.26434/chemrxiv-2023-vsmpx-v4.

Liu Cao, Mustafa Guler, Azat Tagirdzhanov, Yi-Yuan Lee, Alexey Gurevich, and Hosein Mohimani. MolDiscovery: learning mass spectrometry fragmentation of small molecules. *Nature communications*, 12(1):3718, 2021.

Niek F de Jonge, Joris JR Louwen, Elena Chekmeneva, Stephane Camuzeaux, Femke J Vermeir, Robert S Jansen, Florian Huber, and Justin JJ van der Hooft. MS2Query: reliable and scalable MS2 mass spectra-based analogue search. *Nature Communications*, 14(1):1752, 2023.

Kai Dührkop, Huibin Shen, Marvin Meusel, Juho Rousu, and Sebastian Böcker. Searching molecular structure databases with tandem mass spectra using CSI: FingerID. *Proceedings of the National Academy of Sciences*, 112(41):12580–12585, 2015.

Kai Dührkop, Markus Fleischauer, Marcus Ludwig, Alexander A Aksenov, Alexey V Melnik, Marvin Meusel, Pieter C Dorrestein, Juho Rousu, and Sebastian Böcker. SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information. *Nature methods*, 16(4):299–302, 2019.

Samuel Goldman, John Bradshaw, Jiayi Xin, and Connor Coley. Prefix-tree decoding for predicting mass spectra from molecules. *Advances in neural information processing systems*, 36:48548–48572, 2023.

Samuel Goldman, Janet Li, and Connor W Coley. Generating molecular fragmentation graphs with autoregressive neural networks. *Analytical Chemistry*, 96(8):3419–3428, 2024a. doi: 10.1021/acs.analchem.3c04654.

Samuel Goldman, Jiayi Xin, Joules Provenzano, and Connor W Coley. MIST-CF: Chemical formula inference from tandem mass spectra. *Journal of chemical information and modeling*, 64(7):2421–2431, 2024b.

Carlos Guijas, J Rafael Montenegro-Burke, Xavier Domingo-Almenara, Amelia Palermo, Benedikt Warth, Gerrit Hermann, Gunda Koellensperger, Tao Huan, Winnie Uritboonthai, Aries E Aisporna, et al. METLIN: a technology platform for identifying knowns and unknowns. *Analytical chemistry*, 90(5):3156–3164, 2018.

Yang Han, Pengyu Wang, Kai Yu, Xin Chen, and Lu Chen. MS-BART: Unified Modeling of Mass Spectra and Molecules for Structure Elucidation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 111459–111481, 2025. doi: 10.52202/085713-3721.

Hisayuki Horai, Masanori Arita, Shigehiko Kanaya, Yoshito Nihei, Tasuku Ikeda, Kazuhiro Suwa, Yuya Ojima, Kenichi Tanaka, Satoshi Tanaka, Ken Aoshima, et al. MassBank: a public repository for sharing mass spectral data for life sciences. *Journal of mass spectrometry*, 45(7):703–714, 2010.

Florian Huber, Lars Ridder, Stefan Verhoeven, Jurriaan H Spaaks, Faruk Diblen, Simon Rogers, and Justin JJ Van Der Hooft. Spec2Vec: Improved mass spectral similarity scoring through learning of structural relationships. *PLoS computational biology*, 17(2):e1008724, 2021a.

Florian Huber, Sven van der Burg, Justin JJ van der Hooft, and Lars Ridder. MS2DeepScore: a novel deep learning similarity measure to compare tandem mass spectra. *Journal of cheminformatics*, 13(1):84, 2021b.

Hongchao Ji, Hanzi Deng, Hongmei Lu, and Zhimin Zhang. Predicting a molecular fingerprint from an electron ionization mass spectrum with deep neural networks. *Analytical chemistry*, 92(13):8649–8653, 2020.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947.

Tuan Le, Robin Winter, Frank Noé, and Djork-Arné Clevert. Neuraldecipher–reverse-engineering extended-connectivity fingerprints (ECFPs) to their molecular structures. *Chemical science*, 11(38):10378–10389, 2020.

Yuanyue Li, Tobias Kind, Jacob Folz, Arpana Vaniya, Sajjan Singh Mehta, and Oliver Fiehn. Spectral entropy outperforms MS/MS dot product similarity for small-molecule compound identification. *Nature Methods*, 18(12):1524–1531, 2021.

Eleni E Litsa, Vijil Chenthamarakshan, Payel Das, and Lydia E Kavraki. An end-to-end deep learning framework for translating mass spectra to de-novo molecules. *Communications Chemistry*, 6(1):132, 2023.

Yanmin Liu, Xuan Zhang, Wei Zhao, Daming Zhu, and Xuefeng Cui. De novo molecular structure generation from mass spectra. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 373–378. IEEE, 2023.

Mrunali Manjrekar, Runzhong Wang, Samuel Goldman, Jenna C Fromer, and Connor W Coley. Generative structural elucidation from mass spectra as an iterative optimization problem. *arXiv preprint arXiv:2602.07709*, 2026.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (MOSES): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

Christoph Ruttkies, Emma L Schymanski, Sebastian Wolf, Juliane Hollender, and Steffen Neumann. MetFrag relaunched: incorporating strategies beyond in silico fragmentation. *Journal of cheminformatics*, 8(1):3, 2016.

Kerstin Scheubert, Franziska Hufsky, and Sebastian Böcker. Computational mass spectrometry for small molecules. *Journal of cheminformatics*, 5(1):12, 2013.

Emma L Schymanski and Steffen Neumann. The critical assessment of small molecule identification (CASMI): challenges and solutions. *Metabolites*, 3(3):517–538, 2013.

Emma L Schymanski, Christoph Ruttkies, Martin Krauss, Céline Brouard, Tobias Kind, Kai Dührkop, Felicity Allen, Arpana Vaniya, Dries Verdegem, Sebastian Böcker, et al. Critical assessment of small molecule identification 2016: automated methods. *Journal of cheminformatics*, 9(1):22, 2017.

Maria Sorokina, Peter Merseburger, Kohulan Rajan, Mehmet Aziz Yirik, and Christoph Steinbeck. COCONUT online: collection of open natural products database. *Journal of Cheminformatics*, 13 (1):2, 2021.

Stephen Stein. Mass spectral reference libraries: an ever-expanding resource for chemical identification. *Analytical Chemistry*, 84(17):7274–7282, 2012. doi: 10.1021/ac301205z.

Michael A Stravs, Kai Dührkop, Sebastian Böcker, and Nicola Zamboni. MSNovelist: de novo structure generation from mass spectra. *Nature Methods*, 19(7):865–870, 2022.

Umit V Ucak, Islambek Ashyrmamatov, and Juyong Lee. Reconstruction of lossless molecular representations from fingerprints. *Journal of cheminformatics*, 15(1):26, 2023.

Fei Wang, Dana Allen, Siyang Tian, Eponine Oler, Vasuk Gautam, Russell Greiner, Thomas O Metz, and David S Wishart. CFM-ID 4.0–a web server for accurate MS-based metabolite identification. *Nucleic acids research*, 50(W1):W165–W174, 2022.

Mingxun Wang, Jeremy J Carver, Vanessa V Phelan, Laura M Sanchez, Neha Garg, Yao Peng, Don Duy Nguyen, Jeramie Watrous, Clifford A Kapono, Tal Luzzatto-Knaan, et al. Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. *Nature biotechnology*, 34(8):828–837, 2016.

Yinkai Wang, Xiaohui Chen, Liping Liu, and Soha Hassoun. MADGEN: Mass-spec attends to de novo molecular generation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=78tc3EiUrN.

Jennifer N Wei, David Belanger, Ryan P Adams, and D Sculley. Rapid prediction of electron–ionization mass spectrometry using neural networks. *ACS central science*, 5(4):700, 2019.

David S Wishart, AnChi Guo, Eponine Oler, Fei Wang, Afia Anjum, Harrison Peters, Raynard Dizon, Zinat Sayeeda, Siyang Tian, Brian L Lee, et al. HMDB 5.0: the human metabolome database for 2022. *Nucleic acids research*, 50(D1):D622–D631, 2022.

Sebastian Wolf, Stephan Schmidt, Matthias Müller-Hannemann, and Steffen Neumann. In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC bioinformatics*, 11(1):148, 2010.

Shipei Xing, Sam Shen, Banghua Xu, Xiaoxiao Li, and Tao Huan. BUDDY: molecular formula discovery via bottom-up MS/MS interrogation. *Nature Methods*, 20(6):881–890, 2023.

Adamo Young, Hannes Röst, and Bo Wang. Tandem mass spectrum prediction for small molecules using graph transformers. *Nature Machine Intelligence*, 6(4):404–416, 2024.

Hailiang Zhang, Qiong Yang, Ting Xie, Yue Wang, Zhimin Zhang, and Hongmei Lu. MSBERT: embedding tandem mass spectra into chemically rational space by mask learning and contrastive learning. *Analytical Chemistry*, 96(42):16599–16608, 2024.

A Experimental Details

A.1 Datasets, Pretraining Corpora, and Split Rules

The encoder is pretrained on large-scale simulated spectra with Morgan-4096 fingerprint targets. The decoder is pretrained on a large fingerprint–molecule corpus constructed independently of the downstream spectrum-conditioned benchmarks and kept aligned with the DiffMS setup. The final encoder and decoder adaptation uses experimental spectra and the corresponding encoder-predicted fingerprints.

The encoder-pretraining and decoder-pretraining splits follow the corresponding original data sources, and we strictly remove from the pretraining training sets any molecules that overlap with the test sets of the two finetuning benchmarks, NPLIB1 and MassSpecGym.

We use different pretraining corpora for two reasons. First, decoder pretraining follows the 2.8M fingerprint–molecule corpus organized by DiffMS, which keeps the decoder-side setup aligned with the strongest baseline. Second, the current bottleneck is the spectrum encoder rather than the fingerprint-to-structure decoder: the decoder corpus is already several times larger than the simulated spectroscopic corpus, whereas the additional data needed by the encoder must contain spectra. Since the DiffMS decoder-pretraining corpus has no spectra, encoder pretraining relies on the simulated spectrum–molecule corpus instead.

For downstream evaluation, we keep the dataset partitions aligned with DiffMS. In particular, NPLIB1 follows the NPLIB1 split used in DiffMS, and MassSpecGym follows the same split protocol as the DiffMS benchmark export. Thus, all reported results on NPLIB1 and MassSpecGym are obtained under the same train/validation/test partitions as DiffMS.

For final joint finetuning, the bridge receives the encoder's soft fingerprint probabilities together with the target modified SELFIES sequence. No thresholding or active-bit index selection is applied, so the sequence loss can update the spectrum encoder through the differentiable bridge.

A.2 Preprocessing and Representation Details

Across encoder training, the prediction target is the Morgan 4096-bit fingerprint (`morgan4096`). The encoder uses the standard MIST spectrum inputs together with subformula and MAGMa side information when available. During encoder pretraining and finetuning, spectra are augmented by peak removal and intensity perturbation. The simulated pretraining configuration uses augmentation probability 0.5, removal probability 0.5, intensity perturbation probability 0.1, and exponential removal weights. The NPLIB1 finetuning configuration uses the same augmentation strengths, while the MassSpecGym finetuning configuration uses a milder setup with augmentation probability 0.1 and removal probability 0.1.

For structure generation, target molecules are represented as modified SELFIES strings and tokenized into a hexadecimal-compatible modified SELFIES vocabulary. Unless otherwise noted, the decoder uses our structure-aware compositional embedding, which factorizes token identity into chemical attributes such as part type, element, bond, and ring/branch indicators, while omitting charge- and explicit-H-specific factors to avoid unnecessary sparsity. The maximum target length is set to 256 during training and increased only for generation evaluation when needed. In the main experiments, both benchmarks use the benchmark-provided ground-truth molecular formula. It masks only atom tokens whose elements are absent from the formula during decoding and is also used for post-generation formula-distance reranking; elemental counts are not fixed.

A.3 Training Configuration

The encoder backbone follows the MIST architecture with hidden size 512, 8 attention heads, 3 peak-attention layers, and 5 refinement layers. The decoder side uses a 6-layer autoregressive Transformer with hidden size 512, 8 heads, feed-forward dimension 2048, and our structure-aware compositional token embedding. The source interface is our fingerprint-to-memory bridge: it retains all 4096 soft bit values, folds them into 64 modulo chunks, and contextualizes the resulting memory sequence with 6 Transformer layers of hidden size 512, 8 heads, and RoPE positional encoding.

Encoder pretraining. The encoder is pretrained on simulated spectra with batch size 768 over 3 GPUs, learning rate 8×10^{-4}, weight decay 10^{-7}, and up to 100 epochs. We use spectra dropout 0.1, iterative fingerprint prediction with `iterative-loss-weight` $= 0.4$, cosine loss, and MAGMa auxiliary supervision with coefficient 8. Training uses a scheduler with decay fraction 0.9 and early stopping on validation loss with patience 20.

Encoder finetuning. Encoder finetuning initializes from the simulated-pretrained checkpoint and adapts it on experimental spectra. On the NPLIB1 pipeline, we use batch size 128 on 2 GPUs, learning rate 8×10^{-5}, weight decay 10^{-7}, cosine loss, and up to 300 epochs. On the MassSpecGym pipeline, we use batch size 192 on 3 GPUs, learning rate 3×10^{-5}, warmup 1000 steps, BCE loss, and up to 600 epochs. The NPLIB1 setting uses `frac-orig`=0.6 forward augmentation, while the MassSpecGym setting additionally uses bit-frequency reweighting for calibration.

Decoder pretraining. The fingerprint-to-structure decoder is pretrained with batch size 448 on 2 GPUs, learning rate 3×10^{-4}, weight decay 2×10^{-2}, warmup 1000, gradient clipping 1.0, dropout 0.15, and 40 training epochs. We optimize the autoregressive cross-entropy together with the structure-aware auxiliary sentence loss described in Section F.2, using total auxiliary weight 0.1, with factor weights 1.0 for element prediction and 0.2 each for bond, ring, and branch prediction. Corruption-aware pretraining uses a phased schedule consisting of 5 clean epochs, 5 ramp epochs, and a full-corruption phase thereafter. The corruption probability ramps from 0.3 to its final value of 0.6. Whenever corruption is applied, the bit-swap budget is sampled from the clipped-Poisson distribution in Eq. (6), with $\lambda = k_{\max}/2$ and final $k_{\max} = 24$, followed by the bucket-weighted selection in Eq. (7).

Joint encoder–decoder finetuning. The final stage initializes the pretrained spectrum encoder and corruption-aware decoder, then jointly updates the encoder, fingerprint-to-memory bridge, and decoder. We train with batch size 672 on 3 GPUs, learning rate 10^{-5}, weight decay 4×10^{-7}, dropout 0.1, and 20 epochs. The objective combines autoregressive cross-entropy, the fingerprint-prediction loss, and the auxiliary structural loss. Model selection uses decoder cross-entropy.

A.4 Decoding and Reranking Protocol

In all main experiments, CoRe-Gen uses the benchmark-provided ground-truth molecular formula, generates 100 candidates per spectrum with beam size 100, and reranks them by molecular-formula distance, following MS-BART. Soft fingerprint probabilities are fed directly to the bridge without thresholding; ties under formula-distance reranking are broken by model-estimated log-probability. The predicted-formula experiment in Appendix C.1 is the only reported setting that replaces the ground-truth formula with formulas inferred from the spectrum.

The decoder evaluation scripts support both greedy and beam decoding, but all reported top-k generation metrics use beam search. During decoder evaluation, the active finetuning script enables hexadecimal-token normalization, batched beam search, and the constraint controller during generation. The generation configuration uses `beam_size`$= 100$, `beam_num`$= 100$, `topk`$= 10$, and evaluates on the test split. This setup matches the protocol used for our comparisons against DiffMS and other baselines.

Table 6: Fingerprint prediction on MassSpecGym.

Encoder	BCE↓	Cosine↑	Tanimoto↑
CoRe-Gen before pretraining	0.063	0.490	0.251
CoRe-Gen	0.048	0.561	0.322
MS-BART	0.079	0.470	0.213
DiffMS released checkpoint	**0.044**	**0.623**	**0.464**

Table 7: Effect of encoder alignment on CoRe-Gen. Only the encoder checkpoint changes between rows.

	Top-1			Top-10		
Encoder	Accuracy↑	MCES↓	Tanimoto↑	Accuracy↑	MCES↓	Tanimoto↑
Independently trained	1.23%	15.19	0.26	2.67%	13.69	0.32
DiffMS released checkpoint	20.02%	16.09	0.40	22.45%	13.90	0.47

B MassSpecGym Encoder Alignment

CoRe-Gen is a two-stage framework, so generation quality depends directly on the fingerprint predicted by the MIST encoder. Synthetic-spectrum pretraining improves our independently trained MassSpecGym encoder on all three fingerprint metrics in Table 6, and it also outperforms the MS-BART encoder. However, it remains weaker than the publicly released DiffMS checkpoint.

We attempted to reproduce the released DiffMS checkpoint using the authors' publicly released code, configurations, and reported hyperparameters, but our independently trained encoder still did not reach the released checkpoint's fingerprint quality. We contacted the DiffMS author team for clarification but did not receive a response before submission. We therefore isolated this unresolved discrepancy in a controlled encoder-alignment experiment. Because the independently trained encoder gives CoRe-Gen and DiffMS conditions of different quality, all encoder-aligned results use the released DiffMS checkpoint; the data split, formula information, candidate budget, reranking, and metrics remain unchanged.

In this controlled comparison, changing only the encoder checkpoint raises Top-1/Top-10 exact match from 1.23%/2.67% to 20.02%/22.45%. Thus, the weak independently trained result is substantially affected by encoder quality and should not be interpreted solely as a decoder limitation. The released-checkpoint row is the encoder-aligned CoRe-Gen setting reported in Table 2.

C Formula Information and Baseline Fairness

For both NPLIB1 and MassSpecGym, the main results use the benchmark-provided ground-truth molecular formula, not a separately predicted formula. All formula-conditioned methods receive the same formula, but use it according to their original generation paradigm. DiffMS fixes the number and types of heavy atoms before graph generation and mainly predicts connectivity. CoRe-Gen only disallows atom tokens whose elements are absent from the formula; it neither fixes elemental counts nor guarantees an exact formula match. Following MS-BART, CoRe-Gen then reranks 100 beam-search candidates first by formula distance and then by model probability. It therefore receives no additional formula information relative to the principal formula-conditioned baselines.

DiffMS ranks duplicate structures by sampling frequency, whereas CoRe-Gen retains the MS-BART ranking rule. Imposing identical duplicate handling would change the methods' native candidate construction. We therefore standardize the candidate budget, formula information, and evaluation metrics while retaining each model's original generation-specific procedure.

C.1 Robustness to Predicted Molecular Formulas

This auxiliary experiment evaluates CoRe-Gen without ground-truth formulas. We use the Top-5 formulas predicted directly from each spectrum by MSBuddy (Xing et al., 2023) rather than

Table 8: Evaluation controls and paradigm-specific choices.

Setting	DiffMS	CoRe-Gen / MS-BART	Standardized?
Candidate budget	100 sampled graphs	100 beam-search candidates	Yes
Molecular formula	Benchmark-provided formula	Benchmark-provided formula	Yes
Evaluation metrics	Exact match, MCES, Tanimoto	Exact match, MCES, Tanimoto	Yes
Candidate generation	Graph-diffusion sampling	Autoregressive beam search	No; paradigm-specific
Duplicate handling / ranking	Sampling frequency	Formula distance, then model probability	No; paradigm-specific

Table 9: MSBuddy molecular-formula prediction accuracy.

Dataset	Top-1 accuracy	Top-5 accuracy
NPLIB1	82.42%	84.86%
MassSpecGym	49.93%	57.69%

manually perturbing the reference formula. MSBuddy includes the correct formula much less often on MassSpecGym than on NPLIB1 (Table 9). Candidates are reranked by their minimum weighted formula distance to these five predictions, with ties broken by decoder log-probability.

On NPLIB1, formula errors primarily reduce Top-1 ranking, while Top-20 exact match, Tanimoto, and scaffold recovery remain 30.58%, 0.6592, and 57.51%. MassSpecGym is more sensitive because the correct formula appears in MSBuddy's Top-5 for only 57.69% of spectra. Nevertheless, increasing the candidate budget raises exact match from 4.88% to 16.97% and scaffold recovery from 11.62% to 27.08%. Formula uncertainty therefore affects prioritization most strongly, although chemically relevant hypotheses often remain in the larger candidate set.

D Variation Across Random Seeds

We conducted three independent runs of CoRe-Gen on both benchmarks. Table 11 reports mean $\pm$ standard deviation, showing consistent performance across random seeds.

E Differentiable Fingerprint-to-Memory Bridge

Autoregressive fingerprint decoders commonly convert a binary fingerprint into a variable-length sequence containing only active bit indices. For a predicted fingerprint, this requires a hard threshold and index-selection operation. The resulting sequence is suitable for a language-model encoder, but discards uncertainty in inactive or borderline bits and blocks gradients from molecular token loss to the spectrum encoder. CoRe-Gen replaces this handoff with the bridge in Figure 5.

Let $\hat{\mathbf{f}} \in [0,1]^{4096}$ be the encoder output and let $C = 64$. Modulo chunk i contains indices $i + kC$ for $k \in \{0, \dots, 63\}$. With a learned embedding $\mathbf{e}_j \in \mathbb{R}^{512}$ for each fingerprint bit, its initial chunk feature is

$$\mathbf{c}_i = \frac{\sum_{k=0}^{63} \hat{f}_{i+kC} \mathbf{e}_{i+kC}}{\sum_{k=0}^{63} \hat{f}_{i+kC} + \epsilon}, \qquad i = 0, \dots, 63. \tag{12}$$

This modulo assignment follows fingerprint-folding practice: adjacent fingerprint bits are distributed across different chunks, while bits aggregated in the same chunk are well separated in the original index space, reducing local feature confusion. The weighted average then produces a fixed-length source sequence. Six 512-dimensional, 8-head Transformer layers with RoPE contextualize $\{\mathbf{c}_i\}_{i=0}^{63}$ into memory $\mathbf{M} \in \mathbb{R}^{64 \times 512}$, which conditions every autoregressive decoding layer through cross-attention.

During decoder pretraining, the same bridge accepts clean or corruption-augmented fingerprints. During final spectrum-conditioned training, it consumes soft predictions directly. We jointly optimize $\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{AR}} + \lambda_{\text{fp}} \mathcal{L}_{\text{fp}} + \lambda_{\text{sent}} \mathcal{L}_{\text{sent}}$. Because all bridge operations are differentiable, $\mathcal{L}_{\text{AR}}$ updates the spectrum encoder, bridge, and autoregressive decoder end to end; the fingerprint loss simultaneously preserves the quality of the intermediate chemical condition.

Table 10: CoRe-Gen using MSBuddy-predicted formulas instead of benchmark-provided formulas.

Dataset	Metric	Top-1	Top-10	Top-20
NPLIB1	Exact match↑	13.94%	28.45%	30.58%
	Tanimoto↑	0.5134	0.6404	0.6592
	Scaffold match↑	38.95%	56.04%	57.51%
MassSpecGym	Exact match↑	4.88%	14.53%	16.97%
	Tanimoto↑	0.3567	0.4702	0.4954
	Scaffold match↑	11.62%	24.66%	27.08%

Table 11: Performance of CoRe-Gen across three independent runs. The MassSpecGym row uses the released DiffMS encoder checkpoint.

	Top-1			Top-10		
Model	Accuracy↑	MCES↓	Tanimoto↑	Accuracy↑	MCES↓	Tanimoto↑
CoRe-Gen (NPLIB1)	$21.49 \pm 0.29\%$	5.57 ± 0.12	0.57 ± 0.03	$35.42 \pm 0.40\%$	4.03 ± 0.19	0.67 ± 0.04
CoRe-Gen† (MassSpecGym)	$20.02 \pm 0.26\%$	16.09 ± 0.45	0.40 ± 0.02	$22.45 \pm 0.30\%$	13.90 ± 0.33	0.47 ± 0.03

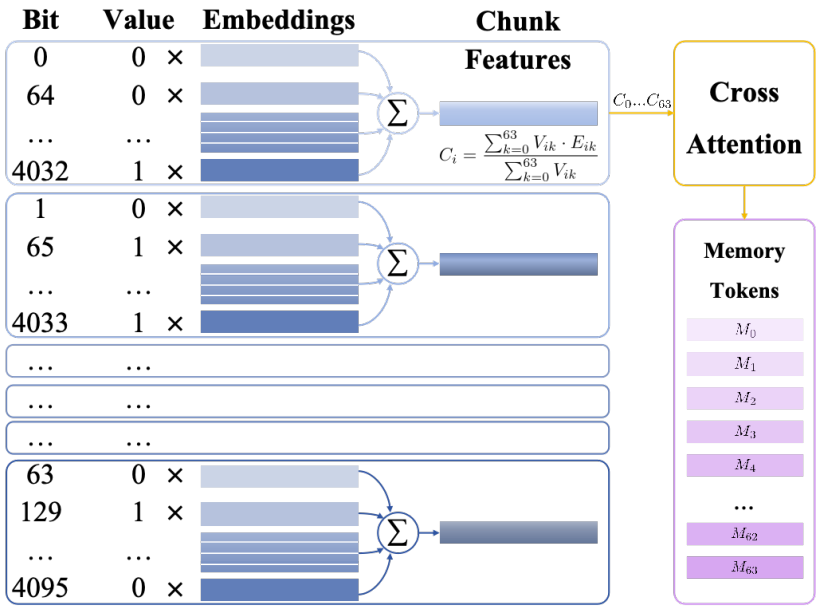


Figure 5: Differentiable fingerprint-to-memory bridge. The 4096 soft fingerprint values are assigned to 64 modulo chunks. Within each chunk, bit-specific embeddings are weighted by their predicted values and normalized to form a chunk feature. The contextualized chunk sequence provides 64 memory tokens for decoder cross-attention.

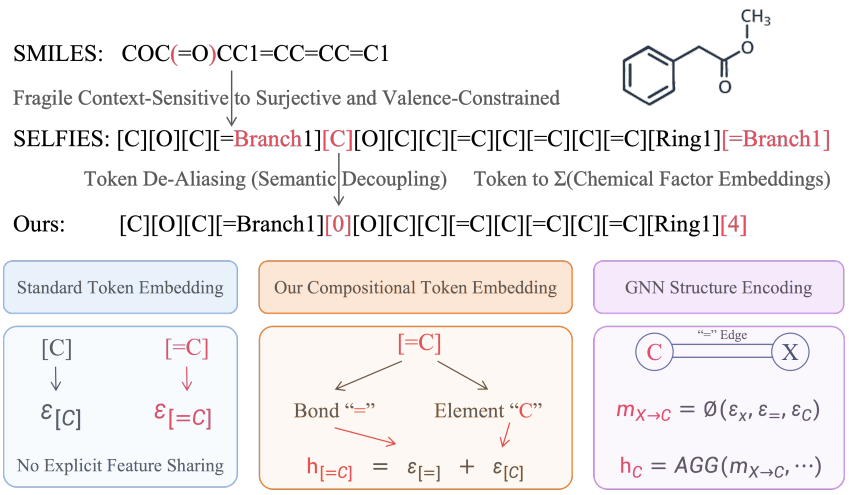


Figure 6: Illustration of the modified SELFIES token interface used for structure-aware decoding. We explicitly separate part type, element, bond prefix, ring/branch order, and hexadecimal continuation to reduce token-level ambiguity inherited from the original SELFIES logic.

F Modified SELFIES Interface

F.1 Token Factorization and Canonicalization Rules

Figure 6 illustrates how we modify the original SELFIES interface. Instead of treating each token as an atomic symbol with an unrelated embedding vector, we decompose a token into several interpretable factors: its part type (e.g., atom, ring, branch, or hexadecimal continuation), its chemical element when applicable, its bond prefix, and its ring/branch order or hexadecimal index when needed. In addition, we canonicalize the token representation before factorization so that superficial variants do not create unnecessary vocabulary fragmentation. In particular, we keep the core chemical identity needed for generation while avoiding a proliferation of low-frequency token variants tied to explicit hydrogens or charges. The interface also resolves a SELFIES-specific ambiguity: original SELFIES reuses post-ring/branch tokens to encode hexadecimal continuation rather than atom semantics, so the same surface token can act as either a structural-control symbol or a chemical-identity symbol depending on context. We make this role explicit through the part-type and digit factors.

This redesign serves two purposes. First, it reduces token-level ambiguity inherited from standard SELFIES, where symbols that are chemically related can still look independent to the decoder. By sharing factors across related tokens, the model can reuse statistical strength across common local motifs such as bond type, atom identity, and ring/branch behavior. Second, it improves robustness under limited supervision by making the representation more compositional: a rare token can still benefit from common factor embeddings even when its full token identity is infrequent in the finetuning data.

An additional advantage of this factorized interface is that it exposes structured subtargets on top of the autoregressive token prediction task. Because each target token can be mapped to element-, bond-, ring-, and branch-level attributes, we can inject lightweight structure supervision during decoder training without changing the decoding architecture. This is the basis of the auxiliary sentence-level loss used in our decoder pretraining and finetuning.

F.2 Auxiliary Structure Supervision

Given a target sequence $y_{1:T}$ and decoder logits over the modified SELFIES vocabulary, we first compute the standard autoregressive cross-entropy loss. We then derive additional factor-level supervision from the same logits by grouping vocabulary items according to their structural attributes. Concretely, each target token induces an element label, a bond label, a ring label, and a branch label whenever the corresponding attribute is defined. Let $p_{t,v}$ denote the softmax probability of vocabulary

item v at time step t, and let $\mathcal{V}_a$ denote the subset of tokens associated with a structural attribute a. For each attribute, we compare the expected count under the model distribution against the count implied by the target sequence. Let $\mathcal{L}_{\text{ce}}$ denote the standard token cross-entropy. The auxiliary sentence-level losses are defined as follows.

Element loss. For each heavy element e, we compute its predicted count by summing token probabilities over all vocabulary items whose atom identity is e, and its target count by counting how many target tokens belong to that element. Both predicted and target element-count vectors are then normalized by their total mass, and the loss is the mean squared error between the normalized vectors:

$$\mathcal{L}_{\text{elem}} = \text{MSE}\left(\frac{\hat{\mathbf{c}}^{\,\text{elem}}}{\|\hat{\mathbf{c}}^{\,\text{elem}}\|_1 + \varepsilon}, \frac{\mathbf{c}^{\text{elem}}}{\|\mathbf{c}^{\text{elem}}\|_1 + \varepsilon} \right).$$

This term encourages the decoder to preserve the overall elemental composition pattern even when the exact token sequence is uncertain.

Bond loss. We separately track the expected and target counts of double-bond and triple-bond prefixed tokens. The bond loss is the sum of two mean-squared errors:

$$\mathcal{L}_{\text{bond}} = \text{MSE}(\hat{c}_{=}, c_{=}) + \text{MSE}(\hat{c}_{\#}, c_{\#}).$$

This term encourages the decoder to match the amount of higher-order bonding indicated by the target sequence.

Ring and branch losses. Similarly, we define ring and branch losses by comparing the expected and target counts of ring tokens and branch tokens:

$$\mathcal{L}_{\text{ring}} = \text{MSE}(\hat{c}_{\text{ring}}, c_{\text{ring}}), \qquad \mathcal{L}_{\text{branch}} = \text{MSE}(\hat{c}_{\text{branch}}, c_{\text{branch}}).$$

These terms encourage consistency in the predicted amount of ring closure and branching behavior.

In our training setup, the active sentence-level rule set is `element,bond,ring,branch`. Therefore, the auxiliary term used in optimization is

$$\mathcal{L}_{\text{sent}} = w_{\text{elem}}\mathcal{L}_{\text{elem}} + w_{\text{bond}}\mathcal{L}_{\text{bond}} + w_{\text{ring}}\mathcal{L}_{\text{ring}} + w_{\text{branch}}\mathcal{L}_{\text{branch}}.$$

Our implementation also supports additional inequality and valence penalties, but these terms are not enabled in the experiments reported in this paper.

The final training objective used in decoder pretraining is

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda_{\text{sent}}\mathcal{L}_{\text{sent}},$$

where $\lambda_{\text{sent}} = 0.1$ in our experiments. We enable this auxiliary term in both decoder pretraining and decoder finetuning. The active factor set is `element,bond,ring,branch`, with weights $w_{\text{elem}} = 1.0$ and $w_{\text{bond}} = w_{\text{ring}} = w_{\text{branch}} = 0.2$.

Operationally, this loss is computed from the same decoder logits as the main autoregressive objective and therefore adds no extra decoding-time cost. Its role is not to replace token prediction, but to bias the model toward recovering chemically meaningful substructure attributes even when the exact full token is difficult to predict. This is especially useful under noisy fingerprint conditioning, where the exact token identity may be unstable but its structural factors remain partially recoverable.

G Evidence for the Predicted-Fingerprint Corruption Gap

G.1 Clean-versus-predicted Fingerprint Gap

We define the predicted-fingerprint corruption gap as the mismatch between the clean fingerprint extracted from the ground-truth molecule and the fingerprint predicted from the input spectrum by the encoder. This gap matters because the decoder is pretrained on fingerprint–molecule pairs, but at deployment time it receives predicted fingerprints rather than oracle ones. If predicted fingerprints differ substantially from the clean distribution, a decoder trained only on clean inputs will face a genuine train–test condition mismatch. The test-set statistics below are a post-hoc diagnostic only:

Table 12: Clean-versus-predicted fingerprint mismatch on the downstream test sets. $\mathrm{swap_}k = \min(\mathrm{FP}, \mathrm{FN})$ summarizes the effective corruption budget relevant to our swap-style decoder corruption process.

Dataset	n	Pred ones	GT ones	FP	FN	Wrong	swap_k mean	swap_k p90 / p95
NPLIB1	819	50.67	46.27	16.66	12.26	28.92	10.97	23 / 27
MassSpecGym	17,556	57.78	61.31	29.43	32.96	62.39	24.86	37 / 42

no downstream test-set statistic was used to design the method, choose corruption hyperparameters, or select a model. The motivating observation—that low-frequency bits have higher prediction error—was established from training data before the proposed components were developed.

We quantify this mismatch directly by comparing the number of active bits and the number of bit errors between predicted and ground-truth fingerprints. For NPLIB1 test samples ($n = 819$), predicted fingerprints are slightly denser than ground truth on average (50.67 active bits vs. 46.27), with mean false-positive and false-negative counts of 16.66 and 12.26, respectively. The mean total number of wrong bits is therefore 28.92. For our budget-preserving swap-style corruption process, the most relevant quantity is $\mathrm{swap_}k = \min(\mathrm{FP}, \mathrm{FN})$, which measures how many bit drops and additions can be paired. On NPLIB1, this quantity has mean 10.97, median 9, 90th percentile 23, 95th percentile 27, and is nonzero for 91.45% of test samples.

The mismatch is even stronger on MassSpecGym. After aligning by unique spectrum name, we obtain 17,556 unique test samples. Here predicted fingerprints are slightly sparser than ground truth on average (57.78 active bits vs. 61.31), but the corruption itself is substantially heavier: mean false-positive and false-negative counts are 29.43 and 32.96, giving a mean total error count of 62.39. The corresponding $\mathrm{swap_}k$ has mean 24.86, median 24, 90th percentile 37, 95th percentile 42, and is nonzero for 99.97% of test samples.

These statistics support two design choices in CoRe-Gen. First, because many bits are rare and receive limited direct supervision, improving fingerprint prediction itself is important; this motivates encoder pretraining on large-scale synthetic spectra. Second, even a strong encoder still produces fingerprints with substantial structured FP/FN errors at test time, especially on MassSpecGym. This motivates corruption-aware decoder pretraining, where the decoder is explicitly exposed to noisy fingerprint conditions rather than only clean ones.

These downstream statistics were computed only after training to illustrate the resulting condition mismatch. They did not determine the clipped-Poisson budget, its clipping bounds, the corruption probability, or the training schedule. The final hyperparameters are reported independently in Appendix A.3.

G.2 Bit-level Error Statistics

To characterize the fingerprint corruption pattern more precisely, we analyze errors at the level of individual fingerprint bits and stratify them by training-set frequency. The train-defined bit frequencies are computed on the downstream training split by counting, for each Morgan bit, in how many training spectra that bit is present. In our implementation, this frequency vector is also used to construct bit weights via $w_b \propto 1/\sqrt{f_b + \epsilon}$ with $\epsilon = 1.0$, followed by mean normalization and clipping to $[0.1, 10.0]$. These weights are used for calibration/reweighting, while the frequency statistics themselves provide a natural way to analyze which bits are easy or hard to predict.

For analysis, we convert each bit count into an empirical training frequency and divide bits into four train-defined buckets according to that frequency: *very rare* $[0, 0.01)$, *rare* $[0.01, 0.05)$, *mid* $[0.05, 0.20)$, and *frequent* $[0.20, 1.0]$. Intuitively, these buckets separate bits that are seen in almost none of the training spectra from those that are observed regularly. Since the learning signal for a bit is proportional to how often it appears in training, this analysis directly tests whether sparsely observed substructures are indeed harder for the encoder to recover.

Table 13 shows a clear monotonic trend on MassSpecGym: the less often a bit appears in the training set, the harder it is to predict reliably. The effect is especially strong for very rare bits, which account for most of the 4096-bit space (3422 bits) but achieve only 0.198 precision, 0.083 recall, and 0.117

Table 13: MassSpecGym example of train-defined bit-frequency bucket analysis on the validation split. Lower-frequency bits are substantially harder to predict than frequently observed bits.

Bucket	# bits	Freq. range	Precision	Recall	F1	TP / FP / FN
Very rare	3422	$[0, 0.01)$	0.198	0.083	0.117	69,350 / 280,274 / 767,242
Rare	527	$[0.01, 0.05)$	0.314	0.285	0.299	184,132 / 402,700 / 462,108
Mid	122	$[0.05, 0.20)$	0.428	0.581	0.493	279,174 / 373,700 / 200,920
Frequent	25	$[0.20, 1.0]$	0.692	0.932	0.794	477,280 / 212,432 / 34,644

Table 14: NPLIB1 Top-1 exact match stratified by maximum structural similarity to the training set.

Maximum similarity to training set	Top-1 exact match↑
< 0.4	2.3%
0.4–0.6	11.2%
0.6–0.8	21.9%
0.8–0.9	35.3%
> 0.9	36.4%

F1. By contrast, the frequent bucket contains only 25 bits but reaches 0.692 precision, 0.932 recall, and 0.794 F1. The false-negative count also drops sharply from 767,242 in the very-rare bucket to only 34,644 in the frequent bucket.

This pattern supports our central motivation. In a finite training set, rare bits are observed only a small number of times, so the encoder receives much weaker supervision for the corresponding substructures. As a result, predicted fingerprints are not corrupted uniformly: the corruption is concentrated on low-frequency, poorly trained bits. This observation motivates encoder pretraining, whose role is to improve prediction quality for weakly supervised bits by exposing the model to much broader spectrum–substructure coverage. It also explains why decoder-side corruption should not be modeled as uniform random noise: the dominant errors arise from structured, frequency-dependent failures of the upstream predictor.

H Reliability, Coverage, and Practical Utility

The following analyses characterize the reliability, coverage, and practical utility of CoRe-Gen.

H.1 Reliability versus Training-Set Coverage

CoRe-Gen outputs should be treated as ranked structural hypotheses rather than definitive identifications unless supported by reference spectra or orthogonal evidence such as retention time, ion mobility, or NMR. To quantify where predictions are more reliable, we group NPLIB1 test spectra by the maximum ECFP4 similarity of the reference molecule to the training set.

Accuracy is substantially higher in well-covered chemical regions; predictions far from the training chemistry require correspondingly greater caution.

H.2 Exact-Structure, Scaffold, and Family Recovery

We compare exact-structure recovery with canonical Bemis–Murcko scaffold recovery for CoRe-Gen on NPLIB1. The correct scaffold appears for 44.20% of spectra at Top-1 and 57.43% at Top-10, substantially more often than the exact molecule at the corresponding candidate budgets.

We also identify several natural-product families in the NPLIB1 test set using CoRe-Gen (Table 16); no tropane alkaloids meet our filtering criteria. Scaffold recovery is particularly strong for flavonoids and flavonoid glycosides. The smaller subgroups should be interpreted cautiously because they contain few examples.

Table 15: Exact and Murcko-scaffold recovery of CoRe-Gen on NPLIB1.

Top-K	Exact match	Scaffold match
Top-1	21.49%	44.20%
Top-10	35.42%	57.43%

Table 16: Top-20 recovery of CoRe-Gen for selected NPLIB1 compound families.

Compound family	Number	Exact@20	Scaffold@20
Flavonoid	38	60.53%	76.32%
Flavonoid glycoside	28	53.57%	71.43%
Benzylisoquinoline alkaloid	3	0.00%	66.67%
Steroid/triterpenoid saponin	5	0.00%	20.00%
Overall	74	51.35%	70.27%

H.3 Library-Depletion Experiment

We simulate an out-of-library setting on 261 unique test molecules from five NPClassifier superclass families. There is no exact molecule overlap between the CoRe-Gen training and test sets, and 30.27% of queries are scaffold-unseen relative to training. For each query, we remove from the reference library every molecule with the same InChIKey, the same canonical Bemis–Murcko scaffold, or ECFP4 Tanimoto similarity ≥ 0.6. The library baseline uses the same benchmark-provided formula as CoRe-Gen, exact-formula filtering, and modified-cosine ranking. CoRe-Gen uses its original checkpoint, 100 candidates, and formula reranking without retraining or experiment-specific tuning.

Exact and Murcko recovery are zero by construction for the depleted library, so Tanimoto, MCES, and no-hit rate are the informative comparisons. CoRe-Gen improves Tanimoto by 0.304 at Top-1 and 0.383 at Top-10 and reduces MCES by 4.70 and 5.37, respectively. Paired bootstrap confidence intervals over 10,000 resamples exclude zero for all four comparisons. An oracle over all remaining library structures reaches only 0.338 Tanimoto and 8.762 MCES, nearly matching actual library-search Top-10 performance; the principal limitation is therefore library coverage rather than spectral ranking.

H.4 Candidate-Screening Workload Proxy

We use scaffold recovery and runtime as proxies for candidate-screening workload. For 100 randomly selected NPLIB1 test molecules, MetFrag (Ruttkies et al., 2016) ranks PubChem isomers using each spectrum and its benchmark-provided ground-truth formula, whereas CoRe-Gen generates its Top-20 candidates.

MetFrag fails on five queries because of excessively large isomer sets. CoRe-Gen completes all queries, is approximately $45\times$ faster, and achieves higher Scaffold@1 and Scaffold@10, suggesting that correct structural cores are ranked earlier and require less screening. A formal chemist user study is still required to measure actual workload reduction.

H.5 Post-Hoc Fingerprint-to-Token Attribution

We examine two links in the pipeline. MIST uses peak-to-substructure supervision, which encourages peak evidence to enter chemically meaningful fingerprint features (Goldman et al., 2024b). For fingerprint-to-token attribution, at every decoding step we ablate each active fingerprint bit and measure the change in token log-probability. Influential shared ECFP4 bits are mapped to radius-2 atom environments, and we test whether the generated atom lies in its most influential environment; control tokens are excluded.

In the random case study in Figure 7, 22 of 50 atom tokens (44.0%) match the environment of their most influential bit, and the rate exceeds 60% among the highest-contributing bits. This is a post-hoc example rather than a comprehensive interpretability evaluation.

Table 17: Out-of-library comparison after removing the true molecule, same-scaffold structures, and close analogues.

Metric	Depleted library	CoRe-Gen
Exact@1↑	0	0.245
Exact@10↑	0	0.398
Tanimoto@1↑	0.312	0.616
Tanimoto@10↑	0.338	0.721
MCES@1↓	9.746	5.044
MCES@10↓	8.765	3.398
Murcko@1↑	0	0.464
Murcko@10↑	0	0.598
No-hit rate↓	0.502	0

Table 18: Scaffold recovery and runtime on 100 randomly selected NPLIB1 queries.

Method	Completed	Scaf.@1	Scaf.@10	Scaf.@20	Runtime
MetFrag	95/100	27.37%	51.58%	62.11%	∼2 h 30 min
CoRe-Gen	100/100	42.00%	59.00%	62.00%	∼200 s

H.6 Transfer beyond Simulator-Specific Artifacts

Synthetic pretraining is intended to improve coverage of rare, long-tailed substructures that receive little supervision from experimental datasets. To reduce dependence on any single simulator, each molecule contributes five spectra: CFM-ID at 10, 20, and 40 eV, ICEBERG, and SCARF. The encoder is then fine-tuned and evaluated only on experimental spectra. Improvement on both downstream benchmarks therefore shows that at least part of the learned signal transfers beyond the synthetic domain.

The current experiments do not isolate transferable chemical knowledge from simulator-specific effects. Doing so requires controlled single-simulator, size-matched, and leave-one-simulator-out ablations. We therefore treat simulator complementarity as a limitation and leave those controlled comparisons to future work.

I Additional Failure Cases

Figure 8 presents additional representative failure cases of CoRe-Gen.

J Limitations

The current method pipeline still has several limitations:

- **Encoder quality remains the main bottleneck.** The MIST-style spectrum encoder is still the dominant bottleneck of the pipeline. Our results show that synthetic data can partially alleviate the scarcity of real paired spectra, but the improvement is still not sufficient to fully close the predicted-fingerprint gap. Stronger spectrum encoders with better generalization, together with more high-quality real spectra, remain important directions.
- **Dependence on molecular formula information.** The main results use the benchmark-provided ground-truth formula for token masking and reranking. Appendix C.1 shows that MSBuddy-predicted formulas reduce performance, especially for Top-1 ranking and on MassSpecGym. This motivates both better formula predictors and generation methods less dependent on formula availability.
- **Hypotheses rather than definitive identifications.** Accuracy is lower in chemical regions distant from training data (Appendix H). Generated structures should therefore be validated with reference spectra or orthogonal evidence such as retention time, ion mobility, or NMR before being treated as identifications.

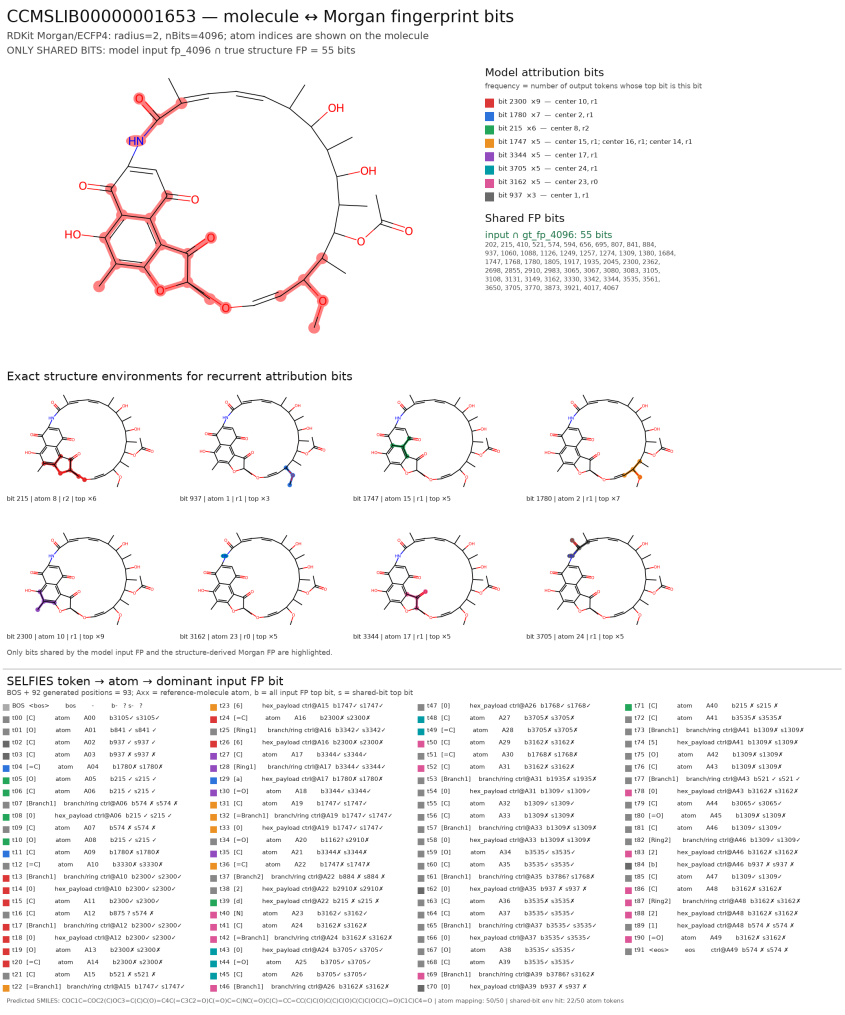


Figure 7: Fingerprint-to-token attribution case study. Recurrent influential shared ECFP4 bits are mapped to their radius-2 molecular environments, and each generated atom token is linked to its dominant input fingerprint bit.

- **Speed-oriented single-pass generation.** CoRe-Gen is designed as a fast single-pass generation pipeline. In settings where accuracy is more important than latency, reference-based search, iterative recycling, or hybrid refinement strategies may be preferable, shifting the speed–accuracy trade-off toward higher precision at higher computational cost.

K Broader Impacts

This work advances de novo molecular structure elucidation from tandem mass spectrometry (MS/MS), offering significant positive impacts for various scientific domains. By improving the robustness of spectrum-to-structure generation under imperfect conditions, our framework (CoRe-Gen) can substantially accelerate metabolomics profiling, natural product discovery, and environmental monitoring, potentially aiding in the faster development of life-saving therapeutics.

However, we must also acknowledge the dual-use potential of robust chemical generation models. Like other computational tools for structural elucidation, our method could theoretically be misused to reverse-engineer or identify illicit chemical compounds, such as novel psychoactive substances (NPS) or harmful toxins, intended to evade standard forensic detection. To mitigate these risks, we

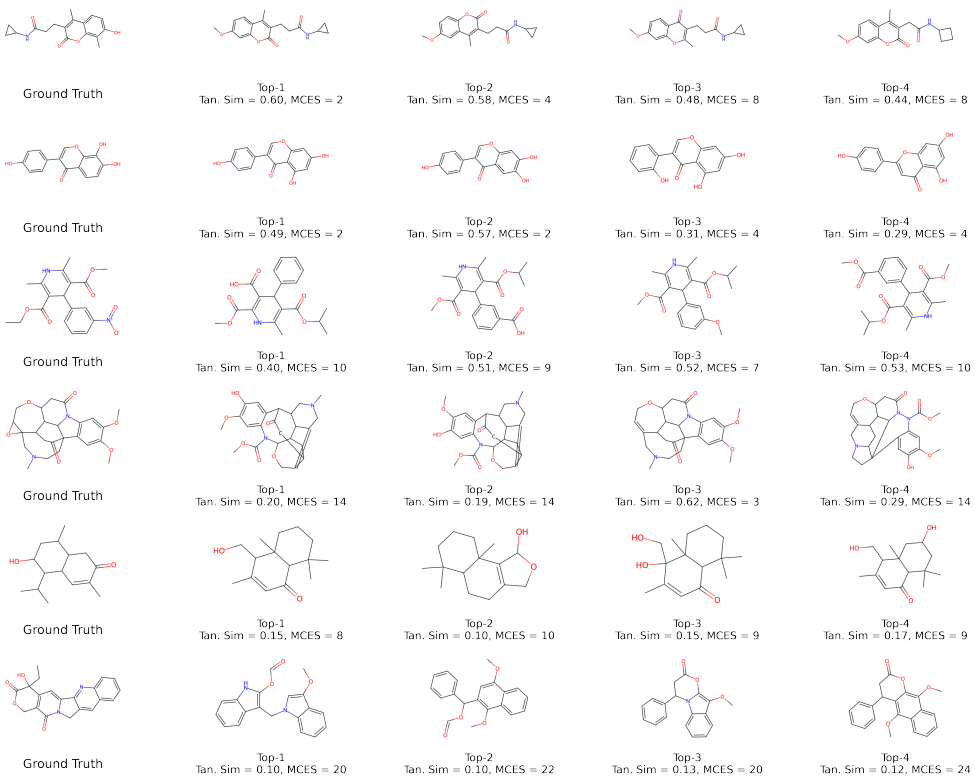


Figure 8: Additional representative failure cases. We show examples where generation errors arise from structurally similar analogues, isomeric ambiguity, or upstream fingerprint corruption.

emphasize the importance of responsible deployment and advocate for integrating such identification pipelines with established chemical safety and security screening protocols.