MassSpecGym in the Wild: Uncovering and Correcting Evaluation Pitfalls in AI-Driven Molecule Discovery

Hongxuan Liu [1 *] **Roman Bushuiev** [2 3 *] **Ivy Lightheart** [4 *] **Mrunali Manjrekar** [1] **Anton Bushuiev** [2 3] **Magdalena Lederbauer** [1] **Filip Jozefov** [2] **Yinkai Wang** [5] **Soha Hassoun** [5] **Josef Sivic** [3] **James Taylor** [4] **Runzhong Wang** [1] **David Healey** [4 †] **Tomáš Pluskal** [2 †] **Connor W. Coley** [1 †]

Abstract

Reliable benchmarking is critical for developing machine learning models for tandem mass spectrometry (MS/MS) based molecule discovery. Subtle issues in experimental design and model evaluation procedures can degrade the trustworthiness of such benchmarks and lead to erroneous conclusions. We conduct a thorough review of model evaluation issues in the recent MS/MS machine learning literature, using the standard MassSpecGym benchmark suite as a case study to illustrate the impact of these issues. We find evaluation issues in at least 17 of 26 papers reporting MassSpecGym benchmark results in the first year of its adoption. We isolate three classes of failures: (i) data leakage, (ii) shortcut learning, and (iii) implementation bugs and metric divergence. Through extensive experimentation and code replication, we quantify the impact of these issues and show how they corrupt the evaluation standards MassSpecGym was designed to enforce. We distill our findings into recommendations generalizable to MS/MS challenges, benchmarks, and custom evaluation setups. We also release MassSpecGym v1.5, an implementation of our recommendations in the MassSpecGym benchmarking suite which addresses the failure modes identified in this audit. MassSpecGym v1.5 is publicly available at `https://github.com/pluskal-lab/MassSpecGym`.

[*]Equal contribution. [†]Corresponding authors. [1]Massachusetts Institute of Technology [2]Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences [3]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague [4]Enveda Biosciences [5]Tufts University. Correspondence to: David Healey <david.healey@enveda.com>, Tomáš Pluskal <tomas.pluskal@uochb.cas.cz>, Connor W. Coley <ccoley@mit.edu>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

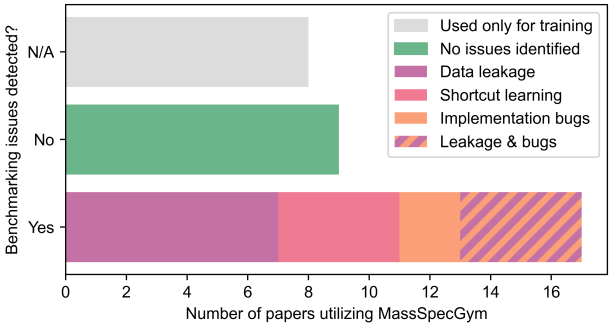

Figure 1. Distribution of 34 papers utilizing MassSpecGym, grouped by primary audit outcome. Benchmarking issues are pervasive throughout the first year of MassSpecGym adoption. Eight papers (gray bar) used MassSpecGym as a dataset, not as a benchmark.

1. Introduction

Machine learning is rapidly advancing the discovery and identification of small molecules given their tandem mass spectra (MS/MS) (Pollmann et al., 2026). Public benchmarks are central to this progress because they enable standardized and reproducible comparison across methods. This is especially important for molecule discovery, as confirming a novel predicted molecular structure is difficult without expensive and low-throughput experiments. (Atanasov et al., 2021; Breitmaier, 2002; Stienstra et al., 2025). As a result, *in silico* benchmarks serve not merely as convenient proxies, but as the main operational interface through which the field measures progress.

When benchmarks are misused, they silently reward shortcut learning, data leakage, and other causes of overfitting (Liao et al., 2021; Thomas & Uminsky, 2022; Raji et al., 2021). As benchmark-driven development accelerates, it can be unclear to what extent leaderboard gains reflect genuine progress on the underlying scientific task rather than artifacts of evaluation. The increasing automation of implementation, model reuse, and hyperparameter search further raises the standard required of benchmark design: a benchmark must be sufficiently transparent that optimiza-

tion against its blind spots does not go undetected (Liao et al., 2021).

Over the last year, MassSpecGym (Bushuiev et al., 2024b) has quickly emerged as the major benchmarking suite for machine learning in computational metabolomics. MassSpecGym formulates the problem of inferring molecular structures from tandem mass spectra as a set of standardized machine learning tasks, including: (i) the spectrum simulation challenge, in which models predict mass spectra from molecular structures; (ii) the molecule retrieval challenge, in which models identify the most compatible molecule from a candidate set given an observed mass spectrum; and (iii) the *de novo* generation challenge, in which models generate molecular structures directly from a given mass spectrum.

In this paper, we conduct a systematic audit of the MassSpecGym ecosystem. Across 26 papers reporting MassSpecGym results in the first year since its introduction, we identify evaluation issues in at least 17 of them that affect the validity of their conclusions. Our goal is not to critique individual papers, but to characterize recurring failure modes and quantify their impact under corrected evaluation conditions. We organize our findings into three categories:

1. Data leakage is more complex than a binary predicate. Data leakage in molecular machine learning admits a gradient of stringency. Splitting on exact structure matches can yield train/test splits where the local chemical space around the test set is densely represented in training. The choice of data splitting criterion encodes a different hypothesis about what generalization in chemical space means (Bushuiev et al., 2024b). Beyond training data, we identify leakage introduced by shared components that participate in training or inference—such as formula predictors or spectral simulators—as well as data leakage from synthetic data (§4).

2. Retrieval tasks admit strong shortcuts unrelated to spectral reasoning. Even with clean train–test splits, discrepancies in molecular candidate sets and composition, such as inconsistent SMILES canonicalization and ranking biases in candidate sets, can enable shortcut learning in retrieval tasks. This can inflate hit rates (recall) to near-perfect levels without the model ever learning meaningful spectrum–molecule relationships, or in some cases without any spectral input at all (§5).

3. Metric and implementation divergence propagates silently across codebases. Errors and implementation inconsistencies in shared encoders, inference scripts, and metric implementations are often reused unchecked and adopted as convention—collectively drifting from the fair evaluation standards MassSpecGym was designed to enforce (§6).

For each category, we present controlled experiments quantifying the impact on benchmark conclusions. We distill our findings into recommendations generalizable to mass spectrometry challenges, benchmarks, and custom evaluation setups. We consolidate the best practices into MassSpecGym v1.5[1], addressing the failure modes identified in this audit and providing canonical implementations of state-of-the-art models, standardized implementation of evaluation metrics, and an agentic workflow for automated auditing and continuous leaderboard maintenance (Table 3) for new methods.

2. Related Work

Benchmark-driven progress and evaluation pitfalls. Benchmark overfitting is a common problem in machine learning, related to Goodhart's law: optimization improves the measured objective without improving the intended one (Goodhart, 1984). This has been demonstrated through a broad literature exploring the misspecification and misuse of ML benchmarks (Liao et al., 2021; Thomas & Uminsky, 2022; Raji et al., 2021; MacDiarmid et al., 2025; Recht et al., 2019). In particular, common evaluation pitfalls have been characterized in genomics (Whalen et al., 2021), structural biology (Škrinjar et al., 2025; Bushuiev et al., 2024a), and small molecule machine learning (Özçelik & Grisoni, 2025; Kretschmer et al., 2025).

Benchmarking in MS/MS-based metabolomics. A number of benchmark frameworks have been proposed over the years for the evaluation of MS/MS based machine learning models. The Critical Assessment of Small Molecule Identification (CASMI) competition has run six times since 2012 (Schymanski et al., 2017), but its irregular schedule limits continuous method development. Goldman et al. (2023b) created the MIST-CANOPUS benchmarking set based on GNPS spectra (Wang et al., 2016), later formalizing it as NPLIB1 (Goldman et al., 2023a). MassSpecGym (Bushuiev et al., 2024b) advanced this direction with comprehensive public data coverage, fixed leakage-free splits, clearly defined tasks, and standardized metrics.

Several works in recent years have performed assessments of the state of MS/MS machine learning. These include efforts to summarize recent developments (Beck et al., 2024), unify baselines (Che et al., 2026), compare datasets and metrics (Schneider et al., 2026), and analyze task formulation (Waele et al., 2026). Other works have demonstrated the impact of problematic meta-scores (Hoffmann et al., 2023), made recommendations for model evaluation best practices (de Jonge et al., 2022), and introduced new evaluation frameworks (Strobel et al., 2025; Zhong et al., 2026). While these methods focus on individual model de-

[1] https://github.com/pluskal-lab/MassSpecGym

velopment and benchmarking concerns, we take a broader lens: thoroughly evaluating the MassSpecGym ecosystem, classifying error modes, and experimentally demonstrating problems. We identify recurring evaluation problems in papers utilizing MassSpecGym and release a more robust MassSpecGym v1.5 with concrete benchmarking recommendations.

3. Study Scope and Audit Protocol

Our audit covers the first year of MassSpecGym adoption, targeting papers that report results on the MassSpecGym benchmark before May 1, 2026. For each paper, we inspect the reported task setting, data provenance, use of external components, metric definitions, and any publicly available implementation details or checkpoints.

This paper combines three complementary forms of evidence. First, we conduct a structured literature and code audit to identify recurring evaluation assumptions and implementation patterns across the ecosystem. Second, we perform controlled reproductions and ablations to quantify the effect of design choices such as data splitting criteria, candidate canonicalization, and metric definitions. Third, we construct targeted stress tests to probe whether published task formulations are susceptible to adversarial or unintentional learning shortcuts. We focus on recurring failure modes that are both scientifically consequential and practically auditable from this first year of benchmark adoption.

4. Data Leakage from Pre-training or External Model Components

Data leakage occurs when information about test examples enters training and inflates evaluation metrics beyond what true generalization would produce. This is problematic because performance gains are driven by memorization rather than generalizable chemical reasoning. MassSpecGym mitigates data leakage by introducing a Maximum Common Edge Subgraph (MCES) (Kretschmer et al., 2025) distance-based train–validation–test split that ensures molecules across data partitions are not similar more than an MCES edit distance of 10. Moreover, it introduces two explicit tracks of competition, one where formula is assumed to be known (the "Bonus," formula-based challenge) and the second where it is not known (the mass-based challenge).

Data leakage can still occur when models use external pretraining data beyond MassSpecGym. Many recent *de novo* generative models follow an encoder-decoder procedure: an encoder model predicts a molecular fingerprint given an MS/MS spectrum, and a decoder model generates a molecule given the predicted fingerprint. Here, we demonstrate a common pattern of MassSpecGym adoption leading to limited model generalization: using decoders (§4.1) or encoders (§4.2) pretrained on data overlapping with the MassSpecGym test set, or on synthetic data from simulators that violate the test structure exclusion. Additionally, we show how models that use chemical-formula-based encoders leak ground-truth formulas into the mass-based challenges when trained improperly (§4.3).

4.1. Limited Generalization in *De Novo* Models Pre-trained to Decode Fingerprints

Many recent *de novo* generation methods adopt a pretrained MIST (Goldman et al., 2023b) spectrum-to-fingerprint encoder and focus their innovation on the fingerprint-to-molecule decoder. These decoders are typically trained on large unlabeled molecular datasets with test compounds excluded only by exact 2D InChIKey matches—a much weaker criterion than the MCES ≥ 10 separation that defines generalization in MassSpecGym. We hypothesize that reported gains are thus largely driven by memorization of fingerprint–molecule pairs structurally close to the test set.

Experiments. To test this hypothesis, we train MIST+MolForge decoders on 2.5M molecules sampled from a curated collection of specialized databases (Appendix A.1), and on 10M, and 50M molecule datasets sampled from a combined pool of approximately 1B molecules from ZINC20 (Irwin et al., 2020) and UniChem (Chambers et al., 2013). For each dataset, we train on 6 different filtering variants using progressively stricter dataset splitting criteria based on exact match exclusion and Tanimoto similarity. Each trained decoder is paired with a MIST encoder and evaluated on the formula-based MassSpecGym *de novo* generation benchmark. Full condition definitions, data composition, and numerical results (including fingerprint specifications used in Tanimoto similarity calculations) are in Appendix A.1.

Figure 2 demonstrates that benchmark performance is heavily influenced by having seen close test structure analogs (Tanimoto ≥ 0.7) in the pretraining set. In Figure 2a-b, we see that accuracy is substantially higher when close structure analogs to test compounds are seen during pretraining, indicating limited generalization. This is further confirmed in Figure 2c, where MolForge decoders trained on 2.5M, 10M, and 50M molecule datasets show high sensitivity to exact test compounds in train. The 2.5M model shows additional sensitivity to training on close structure analogs to test compounds, motivating the following recommendation.

Recommendation. *When using external pretraining data, we recommend excluding exact test matches (based on the 2D InChIKey) from all train splits. If making claims about generalization, any close test structure analogs (Tanimoto similarity*[2] ≥ 0.7 *or MCES* < 10*) should be excluded as*

[2]Using the fingerprint details mentioned in Appendix A.6

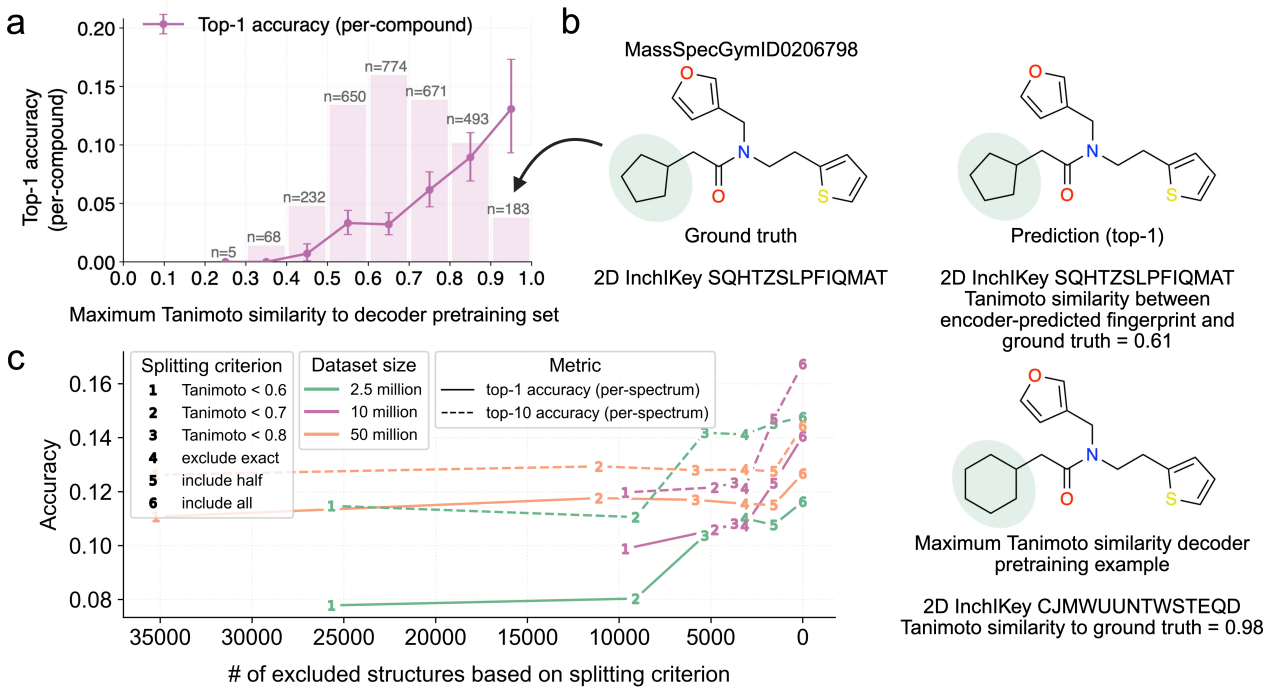

Figure 2. Decoder memorization of near-identical MassSpecGym test compounds. Tanimoto similarities were computed on 2048 bit ECFP4 Morgan radius 2 fingerprints. Further experimental setup details can be found in Appendix A.1. (**a**) *De novo* generation accuracy correlates with similarity to the pretraining set. Per-compound top-1 accuracy with 95% bootstrap confidence intervals is shown across bins of maximum Tanimoto similarity to the pretraining set; n is the per-bin count. This decoder was trained on the 50M dataset in the "exclude exact" setup. (**b**) Example of near-memorization. The ground truth (left) is reproduced perfectly by the model (top right) despite a poor encoder fingerprint prediction (Tanimoto 0.61). This is achieved by slightly modifying the closest pretraining molecule (bottom right; Tanimoto 0.98), which differs from the target at a single fingerprint bit that the encoder predicts correctly. (**c**) Benchmark sensitivity to data splitting criteria. Top-1 and top-10 accuracies for MolForge decoders trained on 2.5M, 10M, and 50M molecule datasets across six splitting criteria.

well. We also recommend reporting performance stratified by the same metric between the test set and any structure in the pretraining set (as in Figure 2a). To facilitate future model development, we release pretraining datasets of 2.5M and 50M molecules excluding close structure analogs (Tanimoto similarity ≥ 0.7*) to any MassSpecGym test molecule.*

4.2. Spectrum-to-Fingerprint Encoders Pretrained on External Data Memorize Test Examples

Spectrum-to-fingerprint models (Goldman et al., 2023b; Stravs et al., 2022) play an important role in many *de novo* generation approaches, and are susceptible to inflated performance due to data leakage or other errors. Spectrum encoders benefit from auxiliary components like formula-annotation tools, which can greatly aid formula-conditioned methods; and spectral simulators (Wang et al., 2025a; Alberts et al., 2024), which can be used for synthetic data augmentation. Possible issues in these auxiliary components, especially when reproduced from other sources, requires closer scrutiny. For example, we identify a commonly reproduced implementation error in the MIST encoder and discuss its impact in §6.1).

Table 1. **Leakage from synthetic spectrum augmentation.** "ICEBERG Split" denotes the MassSpecGym partition used to train ICEBERG, while "Augmented Data Source" denotes the molecules that ICEBERG generates synthetic spectra for. Setting 4 is the original MassSpecGym-reported MIST with no augmentation. Inflated results are marked in gray.

#	ICEBERG Split	Augmented Data Source	MIST spec2fp Tanimoto ↑	MIST Retrieval R@1 ↑
1	MSG Train + Test	PubChem + MSG Test	0.640	67.75%
2	MSG Train	PubChem + MSG Test	0.617	63.85%
3	MSG Train	PubChem	0.581	53.76%
4	MIST (no augmentation)		0.262	9.57%

We focus on the leakage risk of using synthetic spectra generated by forward models such as ICEBERG for spectrum-to-fingerprint training. The leakage risk is two-fold. First, the forward model may have been trained with a data split that is not aligned with the MassSpecGym split, and thus may have learned spectrum–structure associations for molecules that appear in the MassSpecGym test set. Second, generating synthetic spectra for test molecules can expose the encoder to test-molecule-specific representations during training, effectively allowing it to overfit to the test distribution. Data leakage thus traverses ICEBERG's generative process and spectrum-to-fingerprint encoder training before surfacing in metrics.

Table 2. **Examination of formula-prediction leakage impact on MIST's performance on the MassSpecGym mass-based retrieval challenge.** The first two rows indicate how ground-truth formula filtration (mirroring the distinct formula-known challenge) can boost random classification. The remaining rows highlight retrieval performance of MIST, which relies on chemical formula, when used with upstream MIST-CF-inferred formulas. Inflated results are marked in gray. The values in brackets indicate 99.9% confidence intervals upon bootstrapping (20,000 resamples).

Method	Hit rate @1 ↑	Hit rate @5 ↑	Hit rate @ 20 ↑	MCES @ 1 ↓
Random	0.43 (0.29-0.61)	2.01 (1.68-2.38)	7.99 (7.34-8.74)	30.91 (30.50-31.35)
Random w/ ground-truth formula	12.98 (12.15-13.83)	39.09 (37.94-40.31)	71.33 (70.21-72.44)	12.46 (12.28-12.65)
MIST (MIST-CF top-1 formula)	26.64 (25.53–27.75)	38.47 (37.21–39.68)	54.90 (53.69–56.06)	16.41 (16.05–16.79)
MIST (MIST-CF test-only top-1 formula)	31.39 (30.24–32.52)	43.32 (42.06–44.51)	57.63 (56.35–58.88)	14.44 (14.10–14.78)
MIST (MIST-CF train-test top-1 formula)	35.15 (34.01–36.30)	47.97 (46.75–49.21)	62.89 (61.62–64.17)	13.04 (12.71–13.38)
MIST (ground-truth formula)	51.45 (50.22–52.74)	66.25 (65.12–67.46)	80.58 (79.59–81.56)	6.98 (6.75–7.21)

Experiments. We train two ICEBERG models: one on the MassSpecGym training set only, and the other on the training and test sets combined. Each ICEBERG model is then used to generate synthetic spectra at a variable number of collision energies for structures from PubChem, optionally spiked with MassSpecGym test-set structures, yielding three synthetic datasets (Table 1). We train three MIST encoders *only* on the resulting augmented corpora and evaluate them on the MassSpecGym test set, measuring spectrum-to-fingerprint Tanimoto similarity and formula-based retrieval hit rate@1. Full setup and condition definitions are in Appendix A.2.

Among all settings that use synthetic data augmentation, only Setting 3—where a train-split ICEBERG generates augmentation for PubChem structures with MassSpecGym test-set molecules excluded—is leakage-free. According to Table 1, the other two settings exhibit varying degrees of performance inflation. In particular, Setting 1, where ICEBERG is trained on the test data and generates test-split augmentation, significantly exceeds both the leakage-free synthetically-augmented MIST performance (Setting 3) and the original MassSpecGym-reported MIST performance (Setting 4). Setting 1 further outperforms Setting 2 despite augmenting exactly the same molecule set, indicating that the test-trained ICEBERG's memorization of test spectra additionally drives the inflation.

Recommendation. *We encourage developers to leverage simulated data to further scale training only using simulators that respect the original training data split of the benchmark. Furthermore, the synthetic data corpora generated should continue to exclude test molecules.*

4.3. Implicit Formula Leakage in the Mass-Based Retrieval Challenge

Beyond trained spectral simulators for data augmentation, formula annotators can also be critical auxiliary components of elucidation pipelines, as formula knowledge greatly constrains the search space. While often treated as solved, formula prediction remains challenging in some cases such as halogenated molecules or molecules with masses above 600 Da. MassSpecGym differentiates along this axis: the formula-based evaluations provide ground-truth formula as input, and the mass-based evaluations withhold this information. In the retrieval challenge, this additionally imposes a constraint on candidate sets where the mass-based track includes candidates consistent with the observed precursor m/z; while the formula-based track only includes formula-consistent isomers.

Some recent methods assume known chemical formulas yet report evaluations on the mass-based tracks, leaking ground-truth formula information. Moreover, trained formula annotators that do not respect the train-test split can leak ground-truth formula information into mass-based evaluations, mirroring the transitive leakage observed with spectral simulation (§4.2). We evaluate two possible routes of leakage through a simple filtration experiment and by training MIST-CF (Goldman et al., 2024), a formula candidate ranker, under varying amounts of test spectrum leakage.

Experiments. We first show a lower bound on the impact of formula leakage in the mass-based challenge by pruning candidates to those whose chemical formula matches the ground truth and then selecting one at random. This modification alone improves the random baseline's Hit rate@1 from 0.43% to 12.98% (30× higher; Table 2).

To exemplify how a contaminated auxiliary formula annotator can influence downstream elucidation, we next train MIST-CF on the standard MassSpecGym split, a test-only contaminated split, and a train-and-test contaminated split. The formula accuracies of these models are recorded in Table 7. Given a set of candidate formulas (we use the SIRIUS mass decomposition algorithm (Dührkop et al., 2019)), MIST-CF ranks the candidate formulas by their agreement with the observed spectrum. We take the top 1 formula prediction on the MassSpecGym test spectra made by the three models to feed to MIST (which requires formula) to perform retrieval. Here, the predicted formulas are *not* used

Table 3. **Updated Retrieval Performance Leaderboard on the MassSpecGym Retrieval Challenge.** The top section presents the uncorrected leaderboard as reported in the literature, where methods affected by candidate-set artifacts dominate. The subsequent sections isolate these artifact-exploiting methods and report their corrected or spectrum-blind performance. Problematic results are marked in gray, including methods taking advantage of shortcut learning, models using the pretrained MIST from DiffMS with the batching issue fixed (see Table 4), and models leaking a few test structures during training. The baselines leveraging PubChem ranking biases are marked in *gray italics*. Models where an internal reimplementation was developed are marked with a †. The values in brackets indicate 99.9% BCa confidence intervals upon bootstrapping (20,000 resamples).

Method	Hit rate@1 ↑	Hit rate@5 ↑	Hit rate@20 ↑	MCES@1 ↓
Reported Performance in Retrieval Task w/ Bonus (i.e., Formula Known)				
Random	3.06 (2.64-3.52)%	11.35 (10.60-12.12)%	27.74 (26.52-28.84)%	13.87 (13.70-14.03)
DeepSets	4.42 (3.92-4.97)%	14.46 (13.58-15.36)%	30.76 (29.67-31.93)%	15.04 (14.89-15.19)
Fingerprint FFN	5.09 (4.57-5.66)%	14.69 (13.83-15.56)%	31.97 (30.86-33.10)%	14.94 (14.79-15.09)
DeepSets + Fourier	6.56 (5.95-7.16)%	16.46 (15.58-17.35)%	33.46 (32.39-34.59)%	14.14 (13.98-14.31)
MIST vMassSpecGym	9.57 (8.88-10.30)%	22.11 (21.10-23.13)%	41.12 (39.98-42.34)%	12.75 (12.59-12.91)
JESTR (Kalia et al., 2025)	11.82 (11.03-12.68)%	33.48 (32.33-34.68)%	61.46 (60.21-62.63)%	11.71 (11.54-11.87)
MVP (Chen & Hassoun, 2026)	13.96 (13.10-14.82)%	36.88 (35.71-38.02)%	68.12 (67.00-69.26)%	10.36 (10.19-10.53)
FLARE (Chen et al., 2026)	22.66 (21.63–23.74)%	50.00 (48.78–51.22)%	75.15 (74.10–76.22)%	9.00 (8.82–9.18)
Nearest Neighbor† (Khoo, 2025)	9.58 (8.84-10.30)%	22.26 (21.25-23.33)%	39.92 (38.75-41.09)%	13.82 (13.64-13.99)
MIST vDiffMS (Bohde et al., 2025)	46.42 (45.07-47.58)%	54.74 (53.43-55.96)%	66.27 (65.10-67.46)%	7.72 (7.51-7.95)
DiffMS (generative retrieval) (Bohde et al., 2025)	32.29 (31.13-33.45)%	54.36 (53.11-55.58)%	75.47 (74.38-76.55)%	6.67 (6.50-6.84)
MIST+MolForge (generative retrieval)† (Neo et al., 2025)	39.09 (37.86-40.26)%	49.25 (47.98-50.48)%	61.60 (60.38-62.76)%	8.73 (8.53-8.96)
ICEBERG (forward model retrieval)[3] (Wang et al., 2025a)	36.18 (34.96–37.32)%	60.70 (59.58–61.86)%	78.83 (77.74–79.78)%	6.61 (6.43–6.79)
MIST vFRIGID (Bohde et al., 2026)	53.76 (52.51-55.02)%	65.32 (64.09-66.45)%	74.71 (73.64-75.73)%	5.16 (4.99-5.33)
FRIGID (generative retrieval) (Bohde et al., 2026)	45.27 (44.06-46.43)%	58.30 (57.02-59.48)%	69.46 (68.30-70.56)%	6.28 (6.09-6.47)
Exploiting Canonicalization Artifacts (Not Valid Under Corrected Candidate Canonicalization)				
(1) DreaMS + ChemBERTa Alignment (inflated)	82.41 (81.46-83.33)%	92.57 (91.91-93.21)%	97.41 (97.00-97.78)%	2.86 (2.70-3.03)
(1) DreaMS + ChemBERTa Alignment (corrected)	6.70 (6.09-7.35)%	19.82 (18.89-20.83)%	40.62 (39.44-41.79)%	13.13 (12.97-13.28)
(2) ChemFormer Two-Stage Retrieval (inflated)	63.89 (62.71-65.03)%	70.52 (69.37-71.70)%	75.86 (74.79-76.89)%	5.39 (5.20-5.60)
(2) ChemFormer Two-Stage Retrieval (corrected)	7.60 (6.94-8.31)%	19.12 (18.18-20.11)%	39.31 (38.16-40.53)%	13.23 (13.06-13.39)
(3) ChemBERTa SMILES classifier (inflated)	99.66 (99.50-99.79)%	99.84 (99.72-99.92)%	99.99 (99.95-100.00)%	0.05 (0.03-0.07)
(3) ChemBERTa SMILES classifier (corrected)	8.2 (7.53-8.89)%	18.39 (17.38-19.36)%	38.14 (36.96-39.36)%	13.8 (13.64-13.96)
Exploiting Other Structure Biases (Spectrum-Blind Stress Test)				
(4) PubChem default ranking	*49.98 (48.72-51.20)%*	*54.13 (52.93-55.28)%*	*61.04 (59.83-62.22)%*	*7.43 (7.22-7.64)*
(5) Ranking by num. of chiral atoms	*4.88 (4.36-5.44)%*	*16.85 (15.92-17.79)%*	*37.86 (36.72-39.09)%*	*14.4 (14.23-14.56)*

to filter structure candidates. Full experiment details are in Appendix A.2.3.

Here, direct or implicit leakage of ground-truth formulas can confer unfair benefits on downstream models that compete in the mass-based track. Using only its top-1 formula prediction, a MIST-CF trained on test examples can enable a increase of up to 9% (35% relative improvement) in Hit rate@1. As an upper bound, access to ground-truth formulas almost doubles the accuracy of MIST retrieval compared to using the data-safe MIST-CF trained here. Though these experiments only highlight evaluations on the retrieval challenge, the same transitive leakage can also affect the *de novo* challenge. As part of MassSpecGym v1.5, we provide the data-safe version of MIST-CF for use with formula-aware architectures to facilitate comparison to formula-free models.

Recommendation. *If ground-truth formulas of* test *spectra are made available at training or inference, results should be reported under the formula-based track. MassSpecGym v1.5 provides a data-safe MIST-CF formula annotator for formula-based architectures to compete in the mass-based track, and welcomes development of improved formula annotation models and strategies.*

5. Shortcut Learning by Exploiting Retrieval Candidate Artifacts

The MassSpecGym retrieval task (Bushuiev et al., 2024b) requires a model to identify the correct ground-truth molecular structure from a candidate set of decoys, given an input MS/MS spectrum. We identify two distinct candidate-set artifacts that allow models to learn shortcuts and bypass spectral reasoning: SMILES canonicalization inconsistency (§5.1) and ranking biases (§5.2).

5.1. Shortcut Learning Based on Canonicalization of SMILES Strings

The ground-truth SMILES strings in MassSpecGym originate directly from disparate experimental repositories (e.g., GNPS (Wang et al., 2016)), which may retain repository-specific formatting conventions. In contrast, the decoy candidates are uniformly collected from PubChem but canon-

icalized using RDKit (Landrum et al., 2026). This discrepancy introduces a highly discriminative distributional signature. In effect, a model can succeed by recognizing the one SMILES string formatted differently from the rest, rather than by learning the intended relationship between a mass spectrum and a molecular structure.

Experiments. To quantify the magnitude of this shortcut effect, we evaluated several recent architectures and constructed an explicit artifact-exploiting baseline (see Appendix A.4 for complete formulations). Table 3 reports the results. (1) ChemBERTa + Spectral Alignment: a DreaMS spectral encoder (Bushuiev et al., 2026) is aligned with a frozen ChemBERTa molecular encoder (Ahmad et al., 2022) via a bidirectional InfoNCE loss; because the ground-truth SMILES strings retain non-canonical syntax while all decoys are RDKit-canonicalized, the model can achieve over 82% hit rate @1 by matching formatting cues rather than learning true structure–spectrum correspondence. (2) ChemFormer Two-Stage Retrieval: a frozen ChemFormer encoder (Irwin et al., 2022) is paired with a trained spectral encoder via dual-path InfoNCE for initial re-ranking (Stage 1), followed by a cross-fusion decoder that autoregressively generates SMILES conditioned on the top-K retrieved candidates and uses them to refine the ranking (Stage 2); the same canonicalization discrepancy biases from Stage 1 also score true molecules higher, an effect that cascades into Stage 2 to achieve over 63% hit rate@1. (3) ChemBERTa SMILES Classifier: a format classifier trained only on SMILES syntax to distinguish query molecules from their retrieval candidates achieves near-perfect hit rate@1.

Recommendation. *Uniform RDKit canonicalization should be enforced across all candidate and query SMILES before training and evaluation. MassSpecGym v1.5 applies this correction by default.*[4]

5.2. Shortcut Learning via PubChem Ranking Biases

The chemical space of structures with measured MS/MS spectra is inherently non-uniform and exhibits a distinct distribution from PubChem (or other potential sources of decoys). Ground-truth molecules are typically common metabolites, widely available standards, or easily synthesizable compounds. They tend to appear with higher deposition frequency, and may share structural features that distinguish them from arbitrary entries in candidate pools drawn from compound libraries such as PubChem. A model can exploit these spurious correlations to retrieve the correct candidate while ignoring the MS/MS spectrum entirely.

Experiments. To empirically demonstrate shortcut learning from candidate pools, we evaluate two spectrum-free baselines (Table 3; implementation details are available in Appendix A.4). (4) PubChem Default Ranking: ranking candidates by PubChem PUG-REST API's default retrieval order, ignoring the spectrum entirely, can exceed 49% Hit rate@1. (5) Ranking by num. of chiral atoms: a spectrum-blind baseline ranking candidate molecules by the number of chiral centers performs on par with simple learnable fingerprint prediction baselines. More generally, the distribution shift between query and candidate PubChem molecules is apparent from the performance of the corrected baseline (3), which achieves 8.2% Hit rate@1 by directly learning the distributional artifacts that distinguish them.

While the canonicalization artifact can be straightforwardly fixed by enforcing uniform RDKit canonicalization across all query and candidate molecules (which we implement in MassSpecGym v1.5), the ranking bias and spurious correlations between queries and candidates (e.g., num. of chiral atoms) represent a more profound challenge for the community: molecules with available experimental spectra for benchmarking are not uniformly distributed across the chemical space of compound libraries such as PubChem. Instead, they disproportionately fall into sub-regions comprising molecules that are more routinely synthesized, characterized, and analyzed. We therefore treat this as an open evaluation-design question: how should retrieval benchmarks disentangle a model's true spectral reasoning capability from shortcut learning over query and candidate molecules?

In structure-based drug design, the fact that property-matched decoys are harder to distinguish from true binders than random decoys has been discussed for decades (Huang et al., 2006; Mysinger et al., 2012). Analogously, in their work concurrent to this article, Gupta & Skinnider (2026) showed that using property-matched decoys for the MassSpecGym retrieval benchmark reduces the performance of a spectrum-blind baseline model from competitive to near random. This motivates future benchmarking design in computational metabolomics to alleviate distribution shifts with property-matched decoy structures.

Recommendation. *Authors of new methods should verify that their models do not rely on spurious correlations between query and candidate molecules—for example, by masking all input MS/MS spectra with a constant value and confirming that performance degrades accordingly.*

6. Implementation Bugs and Metric Divergence Undermine Reproducibility

Benchmark failures are not limited to data curation and task design. A subtler class of problems emerges at evaluation

[3] ICEBERG is evaluated on the full MassSpecGym test split, roughly half of which lack collision energy annotations. We imputed these missing collision energies; details of the annotation pipeline are given in Appendix A.5.

[4] https://huggingface.co/datasets/roman-bushuiev/MassSpecGym

Table 4. **Comparison of corrected and faulty MIST + MolForge performance on the MassSpecGym *de novo* bonus task.** † indicates that the method was reimplemented. Inflated results are marked in gray.

	Top-1			Top-10			Encoder
Models and results on *de novo* task	Accuracy ↑	MCES ↓	Tanimoto ↑	Accuracy ↑	MCES ↓	Tanimoto ↑	MIST Tanimoto ↑
MIST + MolForge† (corrected)	10.73%	22.15	0.37	14.48%	17.88	0.41	0.457
MIST + MolForge† (inflated, batch size=24)	31.75%	12.30	0.68	40.55%	9.80	0.74	0.812

time in the form of implementation bugs and configuration inconsistencies. These may only be apparent when auditing the code.

As MassSpecGym adoption has grown, its evaluation ecosystem has become increasingly modular: pre-trained encoders are reused as frozen feature extractors, inference scripts are copied across codebases, and configuration choices made in one project are inherited, often without examination, by the next. We show that excessive reuse can propagate silent errors. Undocumented default settings in shared components carry over to downstream methods and inflate their reported performance. Separately, we show that implementation-level divergence in metric computation—fingerprint type, radius, or bit length for Tanimoto similarity, stereochemistry handling in InChIKey matching, solver parameters for MCES distance—is sufficient to reorder leaderboard rankings on identical predictions.

6.1. The MIST Inference Batching Pitfall

MIST (Goldman et al., 2023b) has become a widely reused encoder in the MassSpecGym ecosystem, adopted as a feature extractor by several subsequent works (Bohde et al., 2025; Neo et al., 2025). Our audit found two distinct MIST-related failure modes: a conventional padding-mask error in batched settings, and a separate checkpoint-provenance problem in one specific commonly-reused MIST model.

6.1.1. Padding Mask Bug in Batched MIST

The original MIST encoder was validated in single-spectrum settings. When inputs are batched, its attention code as initially published in the MIST and DiffMS codebases mistakenly applied the raw boolean mask via `attn += attn_mask`. Padding tokens therefore receive a finite $+1$ attention offset rather than being excluded from the softmax, affecting both training and inference with padded batches.

6.1.2. Unreproducible MIST (vDiffMS) Checkpoint Affecting Downstream Decoder Development

The subsequent erroneous MIST (vDiffMS) checkpoint trained and used for inference as described in Section 6.1.1 produces fingerprint predictions, at least in batched mode, that are consistent with patterns of data leakage or contamination; however, we cannot reproduce this inflation from the documented training recipe. Moreover, this erroneous MIST checkpoint has been reused in several *de novo* generation pipelines, including MIST + MolForge (Neo et al., 2025), inflating spectrum-to-fingerprint performance far above what we obtain from a freshly trained, data-safe MIST encoder. As Table 4 shows, replacing batched inference settings with single-spectrum settings sharply reduces downstream MIST + MolForge performance back to an expected range.

Recommendation. *Practice careful review when adopting existing model components or other code, especially when used in different development settings than originally designed. Unintended bugs can occur during training, end-to-end finetuning, and/or inference. MassSpecGym v1.5 includes the corrected MIST implementation.*

6.2. Metric Implementation Divergence Hindering Reproducibility

A second reproducibility failure stems from metric implementations that vary across codebases. We identify four metrics—InChIKey hit rate, MCES distance, Tanimoto fingerprint similarity, and close/meaningful match rates—requiring implementation choices that are neither standardized across works nor consistently documented (complete formal definitions and the full variant table are given in Appendix A.6).

InChIKey: as spectral information often cannot disambiguate stereochemistry, deduplication and accuracy evaluation should only use the 14-character, 2D connectivity layer; using the full 27-character key treats stereoisomers as distinct molecules and under-counts true overlaps. **MCES**: the distance $|E(G_1)| + |E(G_2)| - 2|\text{MCES}(G_1, G_2)|$ is NP-hard and computed via heuristic solvers whose timeout and approximation parameters vary across implementations. **Tanimoto similarity**: computed on molecular fingerprints, the choice of fingerprint algorithm and parameters like radius and bit length can significantly deviate Tanimoto similaries. Appendix A.7 provides empirical results.

Recommendation. *Pinned, standardized metric implementations should be used: 14-character connectivity-layer matching for InChIKey; 2048-bit ECFP4 for Tanimoto similarity;* `threshold=15` *and* `always_stronger_bound=True` *for MCES. We fur-*

ther recommend reporting mean test-set metrics with 99.9% confidence intervals over 20,000 bootstrapping resamples. MassSpecGym v1.5 provides reference implementations for all metrics; metric code copied from third-party repositories should be independently verified against the benchmark specification.

7. Discussion and Release of MassSpecGym v1.5

The evaluation failures documented in the preceding sections have been highlighted in the context of MassSpecGym, but are broadly relevant to computational mass spectrometry. In several cases, these effects produce unexpectedly strong performance, underscoring the need for skepticism toward results that appear too good to be true. The failure modes we have identified have motivated specific recommendations that will promote more defensible evaluations for this rapidly growing field.

MassSpecGym v1.5 as a benchmark revision. The revised benchmark accompanying this paper is MassSpecGym v1.5, which operationalizes the recommendations and findings discussed above. **At the data level**, we perform uniform RDKit canonicalization across all candidate SMILES, eliminating the formatting shortcut. We also release our 2.5M and 50M decoder pretraining sets that exclude close structure analogs (Tanimoto similarity ≥ 0.7) to any test compound, providing a safer and more interpretable foundation for generative model training. **At the model level**, we release a zoo of models including officially retrained data-safe versions of MIST-CF (Goldman et al., 2024), ICEBERG (Wang et al., 2025a), and DreaMS (Bushuiev et al., 2026) to provide leakage-free components for formula prediction, spectral simulation, and MS/MS embeddings computation. We also update and unify baseline implementations under strict data-safe conditions. **At the evaluation level**, we package the pinned metric implementations, corrected inference settings, reference results used throughout this audit, and an LLM-augmented agentic workflow for code review outlined in section A.8, so that future work can be compared under the same benchmark assumptions. Together, these updates reduce ambiguity in how MassSpecGym results are produced, interpreted, and reproduced.

Limitations and broader implications. The three-category audit framework we apply—data leakage analysis, task validity stress-testing, and reproducibility verification—is likely to transfer to related benchmark ecosystems. At the same time, our analysis has important limitations. First, the audit covers only the first year of MassSpecGym adoption to date (34 papers), and evaluation practice may change as the community matures and more model implementations are made publicly available. Second, our experiments (particularly the pre-training data leakage spectrum in §4) illustrate trends in performance under example design choices, but the quantitative results and implications (e.g., recommended similarity thresholds) may shift as more models and corpora are analyzed. Third, although MassSpecGym v1.5 corrects several major sources of ambiguity, it does not resolve every evaluative challenge exposed by this audit—most notably, the ranking bias in candidate sets (§5) remains an open design problem inherent to the retrospective nature of this, or any, benchmark.

References

Ahmad, W., Simon, E., Chithrananda, S., Grand, G., and Ramsundar, B. Chemberta-2: Towards chemical foundation models. *CoRR*, abs/2209.01712, 2022. doi: 10.48550/ARXIV.2209.01712. URL https://doi.org/10.48550/arXiv.2209.01712.

Alberts, M., Schilter, O., Zipoli, F., Hartrampf, N., and Laino, T. Unraveling molecular structure: A multimodal spectroscopic dataset for chemistry. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 125780–125808. Curran Associates, Inc., 2024. doi: 10.52202/079017-3996. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e38e60b33bb2c6993e0865160cdb5cf1-Paper-Datasets_and_Benchmarks_Track.pdf.

Atanasov, A. G., Zotchev, S. B., Dirsch, V. M., Orhan, I. E., Banach, M., Rollinger, J. M., Barreca, D., Weckwerth, W., Bauer, R., Bayer, E. A., Majeed, M., Bishayee, A., Bochkov, V., Bonn, G. K., Braidy, N., Bucar, F., Cifuentes, A., D'Onofrio, G., Bodkin, M., Diederich, M., Dinkova-Kostova, A. T., Efferth, T., El Bairi, K., Arkells, N., Fan, T.-P., Fiebich, B. L., Freissmuth, M., Georgiev, M. I., Gibbons, S., Godfrey, K. M., Gruber, C. W., Heer, J., Huber, L. A., Ibanez, E., Kijjoa, A., Kiss, A. K., Lu, A., Macias, F. A., Miller, M. J. S., Mocan, A., Müller, R., Nicoletti, F., Perry, G., Pittalà, V., Rastrelli, L., Ristow, M., Russo, G. L., Silva, A. S., Schuster, D., Sheridan, H., Skalicka-Woźniak, K., Skaltsounis, L., Sobarzo-Sánchez, E., Bredt, D. S., Stuppner, H., Sureda, A., Tzvetkov, N. T., Vacca, R. A., Aggarwal, B. B., Battino, M., Giampieri, F., Wink, M., Wolfender, J.-L., Xiao, J., Yeung, A. W. K., Lizard, G., Popp, M. A., Heinrich, M., Berindan-Neagoe, I., Stadler, M., Daglia, M., Verpoorte, R., Supuran, C. T., and the International Natural Product Sciences Taskforce. Natural products in drug discovery: advances and opportunities. *Nature Reviews Drug Discovery*, 20(3):200–216, Mar 2021. ISSN 1474-1784. doi: 10.1038/s41573-020-00114-z. URL https://doi.org/10.1038/s41573-020-00114-z.

Beck, A. G., Muhoberac, M., Randolph, C. E., Beveridge, C. H., Wijewardhane, P. R., Kenttämaa, H. I., and Chopra, G. Recent developments in machine learning for mass spectrometry. *ACS Measurement Science Au*, 4(3):233–246, 2024. doi: 10.1021/acsmeasuresciau.3c00060. URL https://doi.org/10.1021/acsmeasuresciau.3c00060.

Bohde, M., Manjrekar, M., Wang, R., Ji, S., and Coley, C. W. Diffms: Diffusion generation of molecules conditioned on mass spectra. In *International Conference on Machine Learning*, pp. 4737–4756. PMLR, 2025.

Bohde, M., Liu, H., Manjrekar, M., Lederbauer, M., Ji, S., Wang, R., and Coley, C. W. Frigid: Scaling diffusion-based molecular generation from mass spectra at training and inference time. *arXiv preprint arXiv:2604.16648*, 2026.

Breitmaier, E. *Structure elucidation by NMR in organic chemistry: a practical guide*. John Wiley & Sons, 2002. doi: 10.1002/0470853069.

Bushuiev, A., Bushuiev, R., Sedlár, J., Pluskal, T., Damborský, J., Mazurenko, S., and Sivic, J. Revealing data leakage in protein interaction benchmarks. *CoRR*, abs/2404.10457, 2024a. doi: 10.48550/ARXIV.2404.10457. URL https://doi.org/10.48550/arXiv.2404.10457.

Bushuiev, R., Bushuiev, A., de Jonge, N. F., Young, A., Kretschmer, F., Samusevich, R., Heirman, J., Wang, F., Zhang, L., Dührkop, K., Ludwig, M., Haupt, N. A., Kalia, A., Brungs, C., Schmid, R., Greiner, R., Wang, B., Wishart, D. S., Liu, L., Rousu, J., Bittremieux, W., Rost, H., Mak, T. D., Hassoun, S., Huber, F., van der Hooft, J. J. J., Stravs, M. A., Böcker, S., Sivic, J., and Pluskal, T. MassSpecGym: A benchmark for the discovery and identification of molecules. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL http://papers.nips.cc/paper_files/paper/2024/hash/c6c31413d5c53b7d1c343c1498734b0f-Abstract-Datasets_and_Benchmarks_Track.html.

Bushuiev, R., Bushuiev, A., Samusevich, R., Brungs, C., Sivic, J., and Pluskal, T. Self-supervised learning of molecular representations from millions of tandem mass spectra using DreaMS. *Nature Biotechnology*, 44:630–640, April 2026. doi: 10.1038/s41587-025-02663-3. Published online May 23, 2025.

Butler, T., Frandsen, A., Lightheart, R., Bargh, B., Kerby, T., West, K., Davison, J., Taylor, J., Krettler, C., Bollerman, T. J., Voronov, G., Moon, K., Kind, T., Dorrestein, P., Allen, A., Colluru, V., and Healey, D. MS2Mol: A transformer model for illuminating dark chemical space from mass spectra. *ChemRxiv*, September 2023.

Chambers, J., Davies, M., Gaulton, A., Hersey, A., Velankar, S., Petryszak, R., Hastings, J., Overington, J. P., Steinbeck, C., and Leach, A. R. UniChem: a unified chemical structure cross-referencing and identifier tracking system. *Journal of Cheminformatics*, 5(1):3, 2013. doi: 10.1186/1758-2946-5-3.

Che, X., Du, X., and Xu, D. Comparative analysis of formula and structure prediction from tandem mass spectra, 2026. URL https://arxiv.org/abs/2601.00941.

Chen, Y. Z. and Hassoun, S. Learning from all views: A multiview contrastive framework for metabolite annotation. *Analytical Chemistry*, 2026. doi: 10.1021/acs.analchem.5c05675.

Chen, Y. Z., Rushing, B., and Hassoun, S. FLARE: Fine-grained learning for alignment of spectra-molecule REpresentation enhances metabolite annotation. *bioRxivorg*, pp. 2026.01.27.702086, January 2026.

de Jonge, N. F., Mildau, K., Meijer, D., Louwen, J. J. R., Bueschl, C., Huber, F., and van der Hooft, J. J. J. Good practices and recommendations for using and benchmarking computational metabolomics metabolite annotation tools. *Metabolomics*, 18(12):103, Dec 2022. ISSN 1573-3890. doi: 10.1007/s11306-022-01963-y. URL https://doi.org/10.1007/s11306-022-01963-y.

Dührkop, K., Fleischauer, M., Ludwig, M., Aksenov, A. A., Melnik, A. V., Meusel, M., Dorrestein, P. C., Rousu, J., and Böcker, S. SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information. *Nat. Methods*, 16(4):299–302, April 2019.

Goldman, S., Bradshaw, J., Xin, J., and Coley, C. W. Prefix-tree decoding for predicting mass spectra from molecules. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/97d596ca21d0751ba2c633bad696cf7f-Abstract-Conference.html.

Goldman, S., Wohlwend, J., Stražar, M., Haroush, G., Xavier, R. J., and Coley, C. W. Annotating metabolite mass spectra with domain-inspired chemical formula

transformers. *Nature Machine Intelligence*, 5(9):965–979, Sep 2023b. ISSN 2522-5839. doi: 10.1038/s42256-023-00708-3. URL https://doi.org/10.1038/s42256-023-00708-3.

Goldman, S., Xin, J., Provenzano, J., and Coley, C. W. MIST-CF: Chemical formula inference from tandem mass spectra. *J. Chem. Inf. Model.*, 64(7):2421–2431, April 2024.

Goodhart, C. A. E. *Problems of Monetary Management: The UK Experience*, pp. 91–121. Macmillan Education UK, London, 1984. ISBN 978-1-349-17295-5. doi: 10.1007/978-1-349-17295-5_4. URL https://doi.org/10.1007/978-1-349-17295-5_4.

Grulke, C. M., Williams, A. J., Thillanadarajah, I., and Richard, A. M. EPA's DSSTox database: History of development of a curated chemistry resource supporting computational toxicology research. *Computational Toxicology*, 12:100096, 2019. doi: 10.1016/j.comtox.2019.100096.

Gupta, V. and Skinnider, M. A. Confronting spurious evaluations of computational methods in small molecule mass spectrometry. *bioRxiv*, 2026. doi: 10.64898/2026.05.03.722532. URL https://www.biorxiv.org/content/early/2026/05/06/2026.05.03.722532.

Han, Y., Wang, P., Yu, K., Chen, X., and Chen, L. MSBART: Unified modeling of mass spectra and molecules for structure elucidation. In *Advances in Neural Information Processing Systems*, 2025.

Hoffmann, M. A., Kretschmer, F., Ludwig, M., and Böcker, S. Mad hatter correctly annotates 98% of small molecule tandem mass spectra searching in pubchem. *Metabolites*, 13(3), 2023. ISSN 2218-1989. doi: 10.3390/metabo13030314. URL https://www.mdpi.com/2218-1989/13/3/314.

Huang, N., Shoichet, B. K., and Irwin, J. J. Benchmarking sets for molecular docking. *Journal of medicinal chemistry*, 49(23):6789–6801, 2006. doi: 10.1021/jm0608356.

Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., Moroz, Y. S., Mayfield, J., and Sayle, R. A. ZINC20—a free ultralarge-scale chemical database for ligand discovery. *Journal of Chemical Information and Modeling*, 60(12):6065–6073, 2020. doi: 10.1021/acs.jcim.0c00675.

Irwin, R., Dimitriadis, S., He, J., and Bjerrum, E. J. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022. doi: 10.1088/2632-2153/ac3ffb.

Kalia, A., Chen, Y. Z., Krishnan, D., and Hassoun, S. Jestr: Joint embedding space technique for ranking candidate molecules for the annotation of untargeted metabolomics data. *Bioinformatics*, 41(7):btaf354, 2025. doi: 10.1093/bioinformatics/btaf354.

Khoo, L. M. S. A data attribution-based approach to model diagnosis in lc-ms/ms structure prediction. Master's thesis, Massachusetts Institute of Technology, 2025. URL https://dspace.mit.edu/handle/1721.1/164644.

Kretschmer, F., Seipp, J., Ludwig, M., Klau, G. W., and Böcker, S. Coverage bias in small molecule machine learning. *Nature Communications*, 16(1):554, Jan 2025. ISSN 2041-1723. doi: 10.1038/s41467-024-55462-w. URL https://doi.org/10.1038/s41467-024-55462-w.

Landrum, G., Tosco, P., Kelley, B., Rodriguez, R., Cosgrove, D., Vianello, R., sriniker, Gedeck, P., Jones, G., Kawashima, E., NadineSchneider, Nealschneider, D., tadhurst cdd, Dalke, A., Swain, M., Cole, B., Turk, S., Savelev, A., Maeder, N., Pechersky, Y., Vaucher, A., Wójcikowski, M., Walker, R., Faara, H., Take, I., Scalfani, V. F., Probst, D., Ujihara, K., Monat, J., and Lehtivarjo, J. rdkit/rdkit: 2026_03_2 (q1 2026) release, April 2026. URL https://doi.org/10.5281/zenodo.19922430.

Liao, T., Taori, R., Raji, I. D., and Schmidt, L. Are we learning yet? A meta review of evaluation failures across machine learning. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/757b505cfd34c64c85ca5b5690ee5293-Abstract-round2.html.

MacDiarmid, M., Wright, B., Uesato, J., Benton, J., Kutasov, J., Price, S., Bouscal, N., Bowman, S., Bricken, T., Cloud, A., Denison, C., Gasteiger, J., Greenblatt, R., Leike, J., Lindsey, J., Mikulik, V., Perez, E., Rodrigues, A., Thomas, D., Webson, A., Ziegler, D., and Hubinger, E. Natural emergent misalignment from reward hacking in production rl, 2025. URL https://arxiv.org/abs/2511.18397.

Mismetti, L., Alberts, M., Krause, A., and Graziani, M. Test-time tuned language models enable end-to-end de novo molecular structure generation from ms/ms spectra. *arXiv preprint arXiv:2510.23746*, 2025.

Mysinger, M. M., Carchia, M., Irwin, J. J., and Shoichet, B. K. Directory of useful decoys, enhanced (dud-e): Bet-

ter ligands and decoys for better benchmarking. *Journal of Medicinal Chemistry*, 55(14):6582–6594, Jul 2012. ISSN 0022-2623. doi: 10.1021/jm300687e. URL https://doi.org/10.1021/jm300687e.

Neo, N. K. N., Jing, L., Preston, N. Y. Z., Serene, K. X. T., and Shen, B. One small step with fingerprints, one giant leap for de novo molecule generation from mass spectra. *arXiv preprint arXiv:2508.04180*, 2025. doi: 10.48550/arXiv.2508.04180. Accepted at AI4Mat-NeurIPS-2025 Workshop.

Özçelik, R. and Grisoni, F. How evaluation choices distort the outcome of generative drug discovery. *Journal of Cheminformatics*, 17(1), November 2025. ISSN 1758-2946. doi: 10.1186/s13321-025-01108-y. URL http://dx.doi.org/10.1186/s13321-025-01108-y.

Pollmann, J., Bushuiev, R., Bushuiev, A., Pluskal, T., and Huber, F. Bridging ms2 spectra and chemical space: Advances in spectral similarity, molecular retrieval, and de novo structure discovery. *ChemRxiv*, 2026(0309), 2026. doi: 10.26434/chemrxiv.15000536/v2. URL https://chemrxiv.org/doi/abs/10.26434/chemrxiv.15000536/v2.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Johansson, S., Chen, H., Nikolenko, S., Aspuru-Guzik, A., and Zhavoronkov, A. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11:565644, 2020. doi: 10.3389/fphar.2020.565644.

Raji, I. D., Denton, E., Bender, E. M., Hanna, A., and Paullada, A. AI and the everything in the whole wide world benchmark. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/084b6fbb10729ed4da8c3d3f5a3ae7c9-Abstract-round2.html.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 2019. URL http://proceedings.mlr.press/v97/recht19a.html.

Schneider, M. Y., Kholmanskikh, D. D., Romanov, K. Y., Perekina, E. A., Nikolenko, S. A., Lukin, R. Y., and Golov, I. V. De novo structure prediction from tandem mass spectra: Algorithms, benchmarks, and limitations. *Molecules*, 31(5), 2026. ISSN 1420-3049. doi: 10.3390/molecules31050769. URL https://www.mdpi.com/1420-3049/31/5/769.

Schymanski, E. L., Ruttkies, C., Krauss, M., Brouard, C., Kind, T., Dührkop, K., Allen, F., Vaniya, A., Verdegem, D., Böcker, S., Rousu, J., Shen, H., Tsugawa, H., Sajed, T., Fiehn, O., Ghesquière, B., and Neumann, S. Critical assessment of small molecule identification 2016: automated methods. *Journal of Cheminformatics*, 9(1):22, Mar 2017. ISSN 1758-2946. doi: 10.1186/s13321-017-0207-1. URL https://doi.org/10.1186/s13321-017-0207-1.

Škrinjar, P., Eberhardt, J., Tauriello, G., Schwede, T., and Durairaj, J. Have protein-ligand cofolding methods moved beyond memorisation? *bioRxiv*, 2025. doi: 10.1101/2025.02.03.636309. URL https://www.biorxiv.org/content/early/2025/08/04/2025.02.03.636309.

Sorokina, M., Merseburger, P., Rajan, K., Yirik, M. A., and Steinbeck, C. COCONUT online: Collection of open natural products database. *Journal of Cheminformatics*, 13(1):2, 2021. doi: 10.1186/s13321-020-00478-9.

Stienstra, C. M. K., Song, J., Healey, D., Voronov, G., Gardner, E., Patel, A., Macherla, V., Krettler, C. A., Kind, T., Dorrestein, P. C., and Domingo-Fernández, D. Structure characterization with nmr molecular networking. *Communications Chemistry*, 9(1):28, Dec 2025. ISSN 2399-3669. doi: 10.1038/s42004-025-01839-x. URL https://doi.org/10.1038/s42004-025-01839-x.

Stravs, M. A., Dührkop, K., Böcker, S., and Zamboni, N. MSNovelist: de novo structure generation from mass spectra. *Nature Methods*, 19(7):865–870, 2022. doi: 10.1038/s41592-022-01486-3.

Strobel, M., Gil-de-la Fuente, A., Zare Shahneh, M. R., Abiead, Y. E., Bushuiev, R., Bushuiev, A., Pluskal, T., and Wang, M. An evaluation methodology for machine learning-based tandem mass spectra similarity prediction. *BMC Bioinformatics*, 26(1):174, Jul 2025. ISSN 1471-2105. doi: 10.1186/s12859-025-06194-1. URL https://doi.org/10.1186/s12859-025-06194-1.

Thomas, R. L. and Uminsky, D. Reliance on metrics is a fundamental challenge for ai. *Patterns*, 3(5):100476, 2022. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2022.100476. URL https://www.sciencedirect.com/science/article/pii/S2666389922000563.

Waele, G. D., Wydmuch, M., Dembczyński, K., Kotłowski, W., and Waegeman, W. Small molecule retrieval from tandem mass spectrometry: what are we optimizing for?, 2026. URL https://arxiv.org/abs/2602.16507.

Wang, M., Carver, J. J., Phelan, V. V., Sanchez, L. M., Garg, N., Peng, Y., Nguyen, D. D., Watrous, J., Kapono, C. A., Luzzatto-Knaan, T., Porto, C., Bouslimani, A., Melnik, A. V., Meehan, M. J., Liu, W.-T., Crüsemann, M., Boudreau, P. D., Esquenazi, E., Sandoval-Calderón, M., Kersten, R. D., Pace, L. A., Quinn, R. A., Duncan, K. R., Hsu, C.-C., Floros, D. J., Gavilan, R. G., Kleigrewe, K., Northen, T., Dutton, R. J., Parrot, D., Carlson, E. E., Aigle, B., Michelsen, C. F., Jelsbak, L., Sohlenkamp, C., Pevzner, P., Edlund, A., McLean, J., Piel, J., Murphy, B. T., Gerwick, L., Liaw, C.-C., Yang, Y.-L., Humpf, H.-U., Maansson, M., Keyzers, R. A., Sims, A. C., Johnson, A. R., Sidebottom, A. M., Sedio, B. E., Klitgaard, A., Larson, C. B., Boya P, C. A., Torres-Mendoza, D., Gonzalez, D. J., Silva, D. B., Marques, L. M., Demarque, D. P., Pociute, E., O'Neill, E. C., Briand, E., Helfrich, E. J. N., Granatosky, E. A., Glukhov, E., Ryffel, F., Houson, H., Mohimani, H., Kharbush, J. J., Zeng, Y., Vorholt, J. A., Kurita, K. L., Charusanti, P., McPhail, K. L., Nielsen, K. F., Vuong, L., Elfeki, M., Traxler, M. F., Engene, N., Koyama, N., Vining, O. B., Baric, R., Silva, R. R., Mascuch, S. J., Tomasi, S., Jenkins, S., Macherla, V., Hoffman, T., Agarwal, V., Williams, P. G., Dai, J., Neupane, R., Gurr, J., Rodríguez, A. M. C., Lamsa, A., Zhang, C., Dorrestein, K., Duggan, B. M., Almaliti, J., Allard, P.-M., Phapale, P., Nothias, L.-F., Alexandrov, T., Litaudon, M., Wolfender, J.-L., Kyle, J. E., Metz, T. O., Peryea, T., Nguyen, D.-T., VanLeer, D., Shinn, P., Jadhav, A., Müller, R., Waters, K. M., Shi, W., Liu, X., Zhang, L., Knight, R., Jensen, P. R., Palsson, B. Ø., Pogliano, K., Linington, R. G., Gutiérrez, M., Lopes, N. P., Gerwick, W. H., Moore, B. S., Dorrestein, P. C., and Bandeira, N. Sharing and community curation of mass spectrometry data with global natural products social molecular networking. *Nature Biotechnology*, 34(8):828–837, Aug 2016. ISSN 1546-1696. doi: 10.1038/nbt.3597. URL https://doi.org/10.1038/nbt.3597.

Wang, R., Manjrekar, M., Mahjour, B., Avila-Pacheco, J., Provenzano, J., Reynolds, E., Lederbauer, M., Mashin, E., Goldman, S., Wang, M., et al. Neural spectral prediction for structure elucidation with tandem mass spectrometry. *BioRxiv*, 2025a.

Wang, Y., Chen, X., Liu, L., and Hassoun, S. MADGEN: Mass-spec attends to de novo molecular generation. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=78tc3EiUrN.

Whalen, S., Schreiber, J., Noble, W. S., and Pollard, K. S. Navigating the pitfalls of applying machine learning in genomics. *Nature Reviews Genetics*, 23(3):169–181, November 2021. ISSN 1471-0064. doi: 10.1038/s41576-021-00434-9. URL http://dx.doi.org/10.1038/s41576-021-00434-9.

Wishart, D. S., Guo, A., Oler, E., Wang, F., Anjum, A., Peters, H., Diber, R., Liang, Z., Gautam, V., Wishart, D., et al. HMDB 5.0: the human metabolome database for 2022. *Nucleic Acids Research*, 50(D1):D622–D631, 2022. doi: 10.1093/nar/gkab1062.

Zhong, Y., Tang, Y., Li, Y., Yang, J., Liu, P., and Xia, J. Flexms is a flexible framework for benchmarking deep learning-based mass spectrum prediction tools in metabolomics, 2026. URL https://arxiv.org/abs/2602.22822.

A. Technical Appendices and Supplementary Material

A.1. Experimental Details and Full Results of MolForge Decoder Pretraining

A.1.1. Data contamination Condition Definitions

All structural comparisons in this section use ECFP4 Morgan fingerprints (radius 2, 2048 bits) and the MassSpecGym test set as the reference. We curate three decoder pretraining corpora at different scales—2.5M, 10M, and 50M molecules—with six variations of similarity-based exclusion criteria each, discussed in detail below. The 2.5M splits are drawn from the composite dataset of **DSSTox** (Grulke et al., 2019), **HMDB** (Wishart et al., 2022), **COCONUT** (Sorokina et al., 2021), and **MOSES** (Polykovskiy et al., 2020) commonly used in prior work (Neo et al., 2025; Bohde et al., 2025; Wang et al., 2025b; Han et al., 2025). The 10M and 50M corpora are randomly sampled from a combined pool of approximately 1B molecules from **ZINC20** (Irwin et al., 2020) and **UniChem** (Chambers et al., 2013). For each size, we curate six dataset variants according to the following criteria:

- **EXCLUDE_EXACT:** For each target size, we randomly draw molecules from the corresponding source corpus until obtaining 2.5M, 10M, or 50M molecules whose connectivity-layer InChIKey (first 14 characters, i.e., the 2D structure hash) does not match any MassSpecGym test-set molecule. Because the first InChIKey block ignores stereochemical layers, this criterion also excludes all stereoisomers of test molecules.
- **EXCLUDE_TANI080, EXCLUDE_TANI070, and EXCLUDE_TANI060:** Starting from the same source order, we replace all molecules whose maximum fingerprint Tanimoto similarity to the test set exceeds the specified threshold with additional molecules from the same source corpus that satisfy the threshold. We choose Tanimoto similarity for this experiment because MCES distance is expensive and slow to compute at scale. We expect that either metric should be sufficient for this experiment. For all pretraining molecules x and test molecules $\mathcal{T}_{\text{test}}$, the retained set satisfies $\max_{y \in \mathcal{T}_{\text{test}}} \text{Tan}(\text{ECFP4}(x), \text{ECFP4}(y)) < \theta$, with $\theta \in \{0.80, 0.70, 0.60\}$.
- **INCLUDE_HALF:** We construct a partially contaminated condition by starting from EXCLUDE_EXACT and replacing an equal number of randomly selected non-test molecules with 50% of the unique MassSpecGym test-set molecules. This isolates the effect of direct test-structure exposure while preserving the total number of decoder pretraining molecules.
- **INCLUDE_ALL:** We construct a fully contaminated condition analogously, replacing non-test molecules in EXCLUDE_EXACT with all unique MassSpecGym test-set molecules. This provides an upper-bound direct-leakage condition under otherwise identical corpus size and training settings.

Table 5. **Dataset construction audit for the six leakage control variants at each decoder pretraining scale.** Entries are the number of molecules removed relative to the size-matched EXCLUDE_EXACT construction for Tanimoto-filtered variants, or the number of MassSpecGym test-set molecules inserted into the INCLUDE variants. Note that the 2.5M dataset is drawn from a different source corpus from the 10M and 50M datasets, and has a much higher concentration of close structure analogs to MassSpecGym test structures. This may affect the interpretation of 2c and 6, giving more weight to the 2.5M model.

Target size	EXCLUDE_EXACT	EXCLUDE_TANI080	EXCLUDE_TANI070	EXCLUDE_TANI060	INCLUDE_HALF	INCLUDE_ALL
2.5M	0 removed	2,174 additionally removed	5,939 additionally removed	22,504 additionally removed	1,585 inserted	3,170 inserted
10M	0 removed	559 additionally removed	1,633 additionally removed	6,495 additionally removed	1,585 inserted	3,170 inserted
50M	0 removed	2,666 additionally removed	7,946 additionally removed	32,070 additionally removed	1,585 inserted	3,170 inserted

A.1.2. MolForge Training and Inference Configuration

We train a separate MolForge SMILES decoder (Neo et al., 2025) for each size–variant pair.[5] MolForge is well suited for this leakage audit because the decoder is simple, fast to retrain across many corpus variants, and modular: decoder pretraining can be varied while the spectrum encoder and evaluation protocol are held fixed. We use the 4096-bit Morgan-fingerprint-to-SMILES preset with a global batch size of 512, split across distributed data-parallel ranks. Models are trained with Adam, token-level negative log-likelihood loss, gradient clipping at norm 1.0, and bfloat16 automatic mixed precision. The learning-rate schedule follows MolForge's Noam-with-restarts schedule, using factor 5.0, 5,000 warmup optimizer steps, and a hard restart every 50,000 optimizer steps.

[5]We use the public MolForge implementation released by the authors: `https://github.com/knu-lcbc/molforge`.

For the 2.5M, 10M, and 50M corpus sizes, respectively, we train all dataset variants for 15, 8, and 3 epochs. These schedules were chosen to ensure that models at each scale reach convergence under the fixed training budget. For each trained model, we select the checkpoint with the best validation performance and use that checkpoint for test evaluation. Training is run in distributed mode on a node with 172 CPU cores, 2048 GB of memory, and 8 NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs; wall-clock training time ranges from approximately 1.5 to 6 hours per model, depending on the corpus size and filtering condition.

Inference uses our MIST–MolForge integration.[6] Each trained decoder is paired with the same MIST encoder checkpoint trained on MassSpecGym with peak-formula annotations to predict molecular fingerprints. No end-to-end joint finetuning is performed, as it was observed to degrade performance. For each spectrum, the MIST encoder predicts Morgan fingerprint probabilities, bits above the fixed threshold of 0.172 are passed to the MolForge decoder, and beam search generates top candidate token sequences that are decoded into SMILES with the SentencePiece tokenizer trained in the original MolForge repository. We evaluate on the standard MassSpecGym *de novo* test split after excluding 474 spectra whose ground-truth molecules contain atom types outside the MIST-supported set $\{\mathrm{C}, \mathrm{O}, \mathrm{P}, \mathrm{N}, \mathrm{S}, \mathrm{Cl}, \mathrm{F}, \mathrm{H}\}$, resulting in 17,082 spectra covering 2,905 unique molecular structures. MCES is not computed in these experiments; reported metrics are Acc@1 and Acc@10 (exact 2D-InChIKey match among the top generated SMILES candidates). Inference is also run in distributed mode on the same 172-core, 2048-GB, 8-GPU node; full 8-GPU inference over the benchmark takes approximately 1.5 hours per model.

A.1.3. Full Per-Condition Metric Table

Table 6 reports exact-match generation accuracy of the selected validation-best checkpoint for every corpus size and leakage variant. Each cell gives Acc@1/Acc@10, rows are ordered by decoder pretraining scale, and columns are ordered from direct test-set inclusion to progressively stricter exclusion. The 2.5M model shows the worst-case performance gradient with respect to splitting criterion, motivating our recommendation that training datasets exclude structures with Tanimoto similarity ≥ 0.7 to any test set compound when reporting results about generalization.

Table 6. **Exact-match accuracy on the MassSpecGym *de novo* challenge for MIST + MolForge models trained on all corpus sizes and dataset variants.** Each entry reports Acc@1/Acc@10 for the checkpoint with the best validation performance.

Target size	INCLUDE_ALL	INCLUDE_HALF	EXCLUDE_EXACT	EXCLUDE_TANI080	EXCLUDE_TANI070	EXCLUDE_TANI060
2.5M	11.6/14.8%	10.8/14.5%	11.0/14.1%	10.4/14.2%	8.0/11.1%	7.8/11.5%
10M	14.1/16.7%	12.3/14.7%	10.7/12.1%	10.8/12.3%	10.6/12.1%	9.9/12.0%
50M	12.7/14.4%	11.5/12.8%	11.6/12.8%	11.7/12.8%	11.8/12.9%	11.1/12.6%

A.2. Experimental Details for Shared Components and Synthetic Data Leakage Study

A.2.1. ICEBERG Training and Data Augmentation Configuration

ICEBERG (Wang et al., 2025a) is a spectral simulator that generates synthetic MS/MS spectra from molecular structures. It is also representative of a broader class of reused shared components in recent MS/MS work, including recent test-time tuning pipelines pretrained on simulated spectra that include ICEBERG generations (Mismetti et al., 2025). Our implementation is based on the `coleygroup/ms-pred` codebase.[7] We train two ICEBERG variants under controlled conditions:

- **Leakage-free ICEBERG:** Trained only on MassSpecGym training-set spectrum–structure pairs, with all test and validation molecules excluded. This model represents the data-safe simulator that respects the benchmark split.
- **Adversarial ICEBERG:** Trained on MassSpecGym training and test sets spectrum–structure pairs combined, solely to construct an adversarial leakage scenario. This model is not intended as a deployable simulator; it isolates the effect of a forward model that has directly learned test-set spectrum–structure associations.

ICEBERG training follows the `ms-pred` DAG-generation and intensity-prediction pipelines. The DAG generator uses the MassSpecGym ICEBERG configuration with GGNN message passing, batch size 32, 200 maximum epochs, learning rate 9.96×10^{-4}, learning-rate decay 0.7214, dropout 0.2, hidden size 512, six graph layers, and collision-energy, adduct, formula, element-group, and instrument embeddings. The intensity model uses the matching binned-intensity configuration

[6] https://github.com/harrylaucngd/MIST-MolForge
[7] https://github.com/coleygroup/ms-pred

with batch size 32, 200 maximum epochs, learning rate 7.36×10^{-4}, learning-rate decay 0.825, dropout 0.1, hidden size 256, cosine loss, 20 ppm tolerance, and the same MassSpecGym metadata embeddings. Training and synthetic data generation are performed on one NVIDIA RTX PRO 6000 Blackwell Server Edition GPU. ICEBERG is a relatively lightweight forward model in this setting; training each ICEBERG model takes approximately 2 hours.

To isolate the effect of leakage introduced by the simulator, we construct three synthetic augmentation corpora that vary two factors: the ICEBERG generator and whether MassSpecGym test-set molecules enter the augmentation pool. The generator is either *leakage-free*, trained on the MassSpecGym training split alone, or *adversarial*, additionally trained on the test split. We first build a data-safe augmentation pool of 435,072 PubChem structures, from which we exclude every molecule overlapping with the MassSpecGym test set. Simulating each structure at a variable number of collision energies (5, 10, 15, . . . , 30) yields 2,333,510 leakage-free spectrum–molecule pairs. The test-spiked pool extends this set with the 2,998 unique MassSpecGym test-set molecules, each simulated at 10, 20, 30, 40, and 50 eV, adding 14,990 pairs for a total of 2,348,500.

The three corpora then pair these pools with the two generators: (i) the leakage-free generator on the data-safe pool (clean simulator, no test molecules); (ii) the leakage-free generator on the test-spiked pool (clean simulator, test molecules present); and (iii) the adversarial generator on the test-spiked pool (test molecules present, and the simulator has itself seen the test spectra). Comparing (i) with (ii) isolates the effect of injecting test molecules, whereas comparing (ii) with (iii) isolates the effect of the generator's memorization of test spectra.

A.2.2. MIST Training Configuration with ICEBERG Augmentation Data

We train three MIST encoders, each on the MassSpecGym training set combined with one of the three ICEBERG augmentation corpora described above, using a modified version of the original `samgoldman97/mist` implementation[8] that supports forward-simulation data augmentation. Every MIST model is trained on the same 194,119 experimental spectrum–molecule pairs from the MassSpecGym training set, into which synthetic spectra are mixed on the fly. We keep the experimental fraction at $\texttt{frac_orig} = 0.08$, so that each training set consists of 8% experimental and 92% synthetic spectrum–molecule pairs; the synthetic examples are drawn with replacement from the corpus, yielding $\big((1-0.08)/0.08\big) \times 194{,}119 \approx 2{,}232{,}368$ synthetic pairs per epoch. The three models differ only in which synthetic corpus is mixed in; all other settings are identical.

In each case, the MIST encoder is trained to predict 4096-bit Morgan (ECFP4) fingerprints from peak-formula–annotated spectra (positional–cosine formula embedding, pairwise peak featurization, up to 10 peaks), using 2 peak-attention layers and 4 refinement layers with hidden size 640 and spectra dropout 0.1. Fingerprints are predicted iteratively (growing scheme, iterative-loss weight 0.4) under a cosine fingerprint loss; the MAGMa auxiliary loss of the original MIST is removed. Optimization uses a cosine learning-rate schedule (peak learning rate 7.7×10^{-4}, warmup fraction 0.1, decay fraction 0.9), weight decay 10^{-7}, batch size 256, on-the-fly spectral augmentation, and an exponential moving average of model weights (decay 0.995). Each model is trained for 150 epochs (early stopping is disabled under the cosine schedule), and the checkpoint with the lowest validation loss is retained. After training, each model is benchmarked on the MassSpecGym test set using spectrum-to-fingerprint Tanimoto similarity and formula-constrained retrieval hit rate. To compute Tanimoto similarity, the continuous fingerprint predictions are binarized with a per-model threshold, calibrated on the validation set so that the on-bit density of the predicted fingerprints matches the empirical on-bit frequency of the reference fingerprints. All training and benchmarking are performed on a single NVIDIA RTX PRO 6000 Blackwell Server Edition GPU, with training taking roughly 12 hours per model.

A.2.3. MIST-CF Formula Annotation Components

MIST-CF (Goldman et al., 2024) is a machine learning model that predicts which molecular formula best explains a MS/MS spectrum from a set of candidate formulas. Because it can provide automated inference of formulas, it can serve as a formula annotator to pair with formula-aware architectures for evaluation in the mass-based MassSpecGym challenge tracks. As it may be reused as a fixed auxiliary model, its training provenance matters for the same reason as ICEBERG's: a public checkpoint trained on overlapping data can transmit information into downstream evaluation even when the downstream split itself is clean.

We therefore retrain MIST-CF on the MassSpecGym dataset, as well as two contaminated splits, and report the Top 1 and Top 5 formula annotation accuracies in Table 7. MIST-CF is a formula transformer that ranks candidate chemical

[8] https://github.com/coleygroup/FRIGID/tree/MIST-FRIGID

Table 7. **Performance of MIST-CF trained and evaluated on the MassSpecGym benchmark.** Versions of MIST-CF with leakage of test spectra are colored in gray.

MIST-CF training setup	Accuracy ↑	
	Top-1	Top-5
Train samples	50.5%	73.2%
Both train and test samples	59.0%	80.5%
Test samples	66.5%	84%

formulas for an observed precursor mass using an energy-based approach: for each spectrum, a set of candidate formulas is enumerated via the SIRIUS decomposition algorithm, subformula assignments are precomputed for each candidate peak, and the model is trained to score the correct formula above a set of sampled decoys. The architecture uses 2 transformer layers with hidden size 128, sinusoidal formula embeddings, dropout of 0.1, and embeds adduct type, instrument type, and neutral-loss fragment formulas as additional covariates; up to 20 subpeaks and 32 decoy formulas are used per training example. Models are trained with Adam (learning rate 4.5×10^{-4}, no weight decay) and a negative log-likelihood loss over the candidate set. The learning rate schedule applies a 100-step linear warmup followed by exponential decay at rate 0.88 every 5,000 steps. Training runs for up to 200 epochs with early stopping (patience 5) on validation loss, and the best-validation checkpoint is selected for inference. Training uses a single NVIDIA H100 GPU with 16 CPU workers and up to 200 GB of memory; all models (batch size 4) trained to convergence in approximately 7 hours over a maximum of 10 epochs. As earlier stated, we generate formula candidates from SIRIUS's mass decomposition algorithm, which takes the precursor m/z and determines all plausible formulas. We use the SIRIUS default element tolerances, which includes a maximum tolerance of 1 for each halogen element (F, Cl, Br, I). Since this protocol can result in potentially thousands of formula candidates for a single mass, we adopt the fast-filter model also developed by MIST-CF to filter candidates to a maximum of 256 candidates that MIST-CF itself needs to consider. This filter model is trained on all formulas present in the original MassSpecGym pretraining set, with all MassSpecGym test and validation formulas excluded. We release both the data-safe MIST-CF and the complementary fast-filter model in MassSpecGym v1.5. The codebase also now supports evaluation of formula-aware models paired with formula annotators on the mass-based challenge.

Table 8. **Additional results of MIST retrieval with top-5 MIST-CF formula predictions.** Retrieval performance of MIST on the MassSpecGym mass-based retrieval challenge, using top-5 formula predictions from three MIST-CF variants (standard, test-only, and train-test), aggregated via average or element-wise max across the five formula-conditioned fingerprint predictions. Inflated results are marked in gray. The values in brackets indicate 99.9% confidence intervals upon bootstrapping (20,000 resamples).

Method	Hit rate @1 ↑	Hit rate @5 ↑	Hit rate @ 20 ↑	MCES @ 1 ↓
MIST (MIST-CF top-5 formula, element-wise max.)	28.23 (27.06–29.33)	40.77 (39.46–41.93)	56.22 (55.00–57.44)	15.51 (15.16–15.85)
MIST (MIST-CF test-only top-5 formula, average)	31.44 (30.77–32.14)	45.32 (44.59–46.03)	60.02 (59.32–60.78)	14.07 (13.87–14.26)
MIST (MIST-CF test-only top-5 formula, element-wise max.)	31.79 (30.62–32.90)	45.16 (43.87–46.37)	58.97 (57.77–60.20)	14.25 (13.91–14.60)
MIST (MIST-CF train-test top-5 formula, average)	32.34 (31.65–33.04)	46.39 (45.65–47.13)	62.04 (61.33–62.78)	13.93 (13.71–14.13)
MIST (MIST-CF train-test top-5 formula, element-wise max.)	32.99 (31.85–34.19)	46.32 (45.07–47.62)	61.06 (59.86–62.27)	13.88 (13.52–14.25)

We pair these models' predictions with MIST, a spectrum-to-fingerprint model, by either generating fingerprints based only on the top-1 formula prediction, or the five fingerprints each generated with the top 5 formula predictions and take an element-wise max across the five fingerprints. We then perform retrieval (as measured by lowest Tanimoto distance to the fingerprint) on the mass-based candidates. Prefiltration of the mass-based candidates to only candidates that match the formula predictions is not applied here. The main results are reported in Table 2; additional results based on MIST-CF's top-5 formula predictions are provided for comparison in Table 8.

A.3. Updated Performance Leaderboard on the MassSpecGym *De Novo* Bonus Task

Table 9 provides the full updated performance leaderboard for the *de novo* bonus task.

A.4. Detailed Formulations of Artifact-Exploiting Stress-Test Baselines

We provide complete technical formulations for the five artifact-exploiting or spectrum-blind baselines evaluated in Section 5, based on deep audits of the corresponding codebases.

Table 9. **Updated leaderboard on the MassSpecGym *de novo* generation challenge w/ bonus.** The best performing model for each metric is **bold** and the second best is underlined. † indicates our implementations of baseline approaches without public code. The inflated MIST + MolForge results (Neo et al., 2025), caused by the batched inference bug, are marked in gray to indicate they are not valid.

	Top-1			Top-10		
Model	Accuracy ↑	MCES ↓	Tanimoto ↑	Accuracy ↑	MCES ↓	Tanimoto ↑
MIST + MSNovelist	0.00%	39.84	0.06	0.00%	18.83	0.15
Spec2Mol	0.00%	36.78	0.12	0.00%	36.02	0.16
MIST + Neuraldecipher	0.00%	22.93	0.14	0.00%	21.76	0.16
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
DiffMS	2.30%	13.96	0.28	4.25%	11.68	0.39
FOAM (DiffMS + PubChem)	1.50%	**12.21**	0.35	10.28%	**6.06**	**0.53**
MIST + MolForge† (corrected)	10.73%	22.15	0.37	14.48%	17.88	0.41
MIST + MolForge† (inflated, batch size=24)	31.75%	12.30	0.68	40.55%	9.80	0.74
FRIGID	**18.29%**	13.49	**0.43**	**22.00%**	11.65	0.47

Method 1: DreaMS + ChemBERTa Alignment. This architecture bridges the spectral and molecular modalities via a lightweight trainable adapter between two frozen pre-trained encoders. The spectral encoder is DreaMS (Bushuiev et al., 2024b), a 7-layer Graphormer-style Transformer with 8 attention heads that encodes MS/MS peaks as (m/z, intensity) pairs with Fourier positional features, producing a pooled embedding $\mathbf{z}_s \in \mathbb{R}^{1024}$. The molecular encoder is ChemBERTa-2 (Ahmad et al., 2022) (`Derify/ChemBERTa_augmented_pubchem_13m`), a RoBERTa-style masked language model pretrained on 13M SMILES strings, whose CLS-token output $\mathbf{z}_m^{(0)} \in \mathbb{R}^{768}$ provides a SMILES-surface representation.

The adapter f_ϕ, a multi-block MLP with $N = 8$ residual blocks (hidden dim 2048), projects the DreaMS embedding into ChemBERTa's latent space: $\hat{\mathbf{z}}_m = f_\phi(\mathbf{z}_s)$. Training jointly optimizes a bidirectional InfoNCE objective with a learnable temperature τ:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{2B}\sum_{i=1}^{B}\left[\log\frac{e^{\hat{\mathbf{z}}_{m,i}\cdot\mathbf{z}_{m,i}/\tau}}{\sum_j e^{\hat{\mathbf{z}}_{m,i}\cdot\mathbf{z}_{m,j}/\tau}} + \log\frac{e^{\mathbf{z}_{m,i}\cdot\hat{\mathbf{z}}_{m,i}/\tau}}{\sum_j e^{\mathbf{z}_{m,i}\cdot\hat{\mathbf{z}}_{m,j}/\tau}}\right] \tag{1}$$

where the sum is over all B items in a batch, and the positive pair (i, i) is the spectrum–ground-truth-SMILES pair. A Supervised Contrastive (SupCon) loss with isomer-aware negatives is optionally added to handle structural ambiguity. Only the adapter f_ϕ and the last two DreaMS transformer layers are trained; ChemBERTa remains completely frozen. At retrieval, candidate molecules are embedded by ChemBERTa and ranked by cosine similarity to the adapter output $\hat{\mathbf{z}}_m$.

Critical failure mechanism: because ChemBERTa's training is sensitive to SMILES token-level syntax, and the ground-truth SMILES consistently originates from non-RDKit sources (retaining PubChem-style conventions), the adapter learns to produce $\hat{\mathbf{z}}_m$ that matches the *syntactic formatting* of the positive pair, rather than encoding mass spectral information.

Method 2: ChemFormer Two-Stage Retrieval. This two-stage pipeline builds upon a frozen ChemFormer BART encoder and a jointly trained spectral encoder.

Stage 1 (Pre-Retrieval): The molecular encoder is a frozen ChemFormer encoder (BART architecture: $d = 512$, 6 layers, 8 heads, vocabulary size 523 SMILES tokens), producing a CLS embedding $\mathbf{e}_m = f_{\text{mol}}(\mathbf{x}_{\text{SMILES}}) \in \mathbb{R}^{512}$. The spectral encoder is an FC($2 \to 512$) projection followed by a 6-layer Transformer encoder with mean pooling: $\mathbf{e}_s = f_{\text{spec}}(\mathbf{x}_{\text{peaks}}) \in \mathbb{R}^{512}$. Both are L2-normalized before computing similarities. Training optimizes a dual-path InfoNCE loss at $\tau = 0.1$:

$$\mathcal{L}_{\text{mol2ms}} = \text{CE}\big([\mathbf{e}_{m,i}\cdot\mathbf{e}_{s,i},\ \mathbf{e}_{m,i}\cdot\mathbf{e}_{s,i}^{-}]/\tau,\ 0\big), \quad \mathcal{L}_{\text{ms2mol}} = \text{CE}(\mathbf{E}_s\mathbf{E}_m^T/\tau,\ \mathbf{I}) \tag{2}$$

where $\mathbf{e}_{s,i}^{-}$ is an augmented negative spectrum for molecule i, and $\mathcal{L}_{\text{ms2mol}}$ is a full-batch CLIP-style contrastive loss. The total loss is $\mathcal{L} = 0.5(\mathcal{L}_{\text{mol2ms}} + \mathcal{L}_{\text{ms2mol}})$.

Stage 2 (Generative Retrieval): After Stage 1, the top-$K = 40$ candidate molecules are pre-computed per spectrum. A cross-fusion module performs cross-attention where the spectral encoder sequence (queries) attends to the concatenated full-sequence hidden states of the K candidates (keys/values), producing a context-enriched spectral representation. A ChemFormer decoder (6-layer BART, initialized from pretrained weights) then autoregressively generates a SMILES string

conditioned on this cross-fused context via teacher-forcing cross-entropy loss. At inference, beam search (beam size 5) generates a refined candidate, which is then used to re-rank the initial top-K pool by cosine similarity between the generated molecule embedding and each candidate embedding.

Failure propagation across stages: Stage 1 inflates the pre-retrieved candidate pool by exploiting the canonicalization artifact in the InfoNCE training signal. The Stage 2 decoder, conditioned on these biased top-K candidates, further consolidates the shortcut, since the cross-fusion module learns to match the formatting signature of the positive candidate that consistently appears near the top of the Stage 1 ranking.

Method 3: ChemBERTa SMILES Classifier. A spectrum-blind binary classifier built on top of a pretrained SMILES transformer (Ahmad et al., 2022) (`DeepChem/ChemBERTa-77M-MLM`, loaded via the HuggingFace `transformers` library with a randomly initialized 2-class classification head) that maps a single SMILES string to the probability that it is a ground-truth query molecule rather than a decoy candidate (i.e., a randomly sampled SMILES from the corresponding candidate list, as per `MassSpecGym_retrieval_candidates_formula.json`). The model is fine-tuned for two epochs with a binary cross-entropy objective on balanced positive/negative pairs. At test time, the classifier scores each candidate independently and ranks the candidate set in descending order of predicted probability. No spectral input and no per-query ground-truth information are used at inference time.

Method 5: PubChem Default Ranking. This method exploits the popularity bias described in Section 5. For a given query, the candidate set is reordered in descending order of each molecule's PubChem Compound Identifier (CID) deposition count—a proxy for compound prevalence and commercial/research interest. No spectral information is used. Ground-truth test molecules, being predominantly well-characterized natural products and common metabolites, systematically rank higher under this criterion, achieving $> 49\%$ Hit rate@1. This baseline is aimed to expose a critical limitation that any benchmark whose ground-truth molecules are non-uniformly sampled from chemical space will be susceptible to this frequency-prior shortcut.

Method 6: Ranking by Num. of Chiral Atoms. A spectrum-blind ranker that exploits a systematic offset between ground-truth molecules and their candidates in the count of annotated stereocenters. For each candidate molecule c, the ranker computes

$$n(c) = \sum_{a \in c} \mathbb{1}\{\texttt{a.GetChiralTag()} \neq \texttt{Chem.ChiralType.CHI_UNSPECIFIED}\}, \tag{3}$$

the number of atoms a in c carrying an explicit stereochemistry annotation in the SMILES (i.e., @ or @@), evaluated via RDKit's `Chem.MolFromSmiles` followed by `GetAtoms` and `GetChiralTag`. Candidates are then ranked by $n(c)$. The ranking direction is decided based on the training set: descending if ground-truth molecules carry on average more annotated stereocenters than their candidates, ascending otherwise. The ranker uses no spectral input and no per-query ground-truth information at test time.

A.5. Pipeline of Annotating Missing Collision Energies in the MassSpecGym Test Set

Because collision energy annotations are incomplete in MassSpecGym, but forward models such as ICEBERG require collision energy labels, to compare head-to-head on the main benchmark, we used a staged ICEBERG workflow rather than training directly on the full dataset. We first constructed a subset containing spectra with observed collision energies and trained the ICEBERG fragmentation and intensity models on this curated subset. These models were then used to impute missing collision energies in the original full labels: for each spectrum lacking a collision energy, ICEBERG generated predictions over a grid of candidate normalized collision energies (from 0% to 150% NCE at steps of 5%), and the candidate whose predicted spectrum best matched the experimental spectrum was selected as a pseudo-label. The imputed labels were written back into a derived dataset, with retrieval candidate tables updated to use the same collision-energy values. Finally, ICEBERG was trained and evaluated on this completed full dataset, including the fragmentation model, intensity model, contrastive intensity fine-tuning, and retrieval benchmarks. This two-pass design preserves the calibration of collision-energy-dependent predictions from experimentally annotated spectra while still allowing the final model to be evaluated on the full MassSpecGym testing set. These ICEBERG-related results in Table 3 should not be mistaken with the "simulation_challenge" setting, where only a subset of training and testing data is involved.

A.6. Formal Metric Definitions for Reproducible Evaluation

We provide precise, implementation-pinned definitions for each evaluation metric referenced in Section 6.

InChIKey Hit Rate. An InChIKey is a 27-character hash encoding three layers: a 14-character connectivity hash (atoms and bonds, ignoring stereochemistry), an 8-character stereochemistry and charge hash, and a 1-character isotope flag. Since MS/MS fragmentation is largely insensitive to stereocenters, two structures sharing the same connectivity layer should be considered equivalent. Formally, a prediction $\hat{m}$ is a hit for ground truth m^* if $\text{InChIKey}_{14}(\hat{m}) = \text{InChIKey}_{14}(m^*)$, where $\text{InChIKey}_{14}(\cdot)$ denotes the first 14 characters of the InChIKey. Using the full 27-character InChIKey incorrectly rejects stereoisomers as non-hits and—in data splitting—fails to exclude structurally similar training molecules that differ only in stereochemistry.

MCES Distance. Given two molecular graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the Maximum Common Edge Subgraph (MCES) is the largest subgraph (by edge count) isomorphic to a subgraph of both. The MCES distance is:

$$d_{\text{MCES}}(G_1, G_2) = |E_1| + |E_2| - 2\,|\text{MCES}(G_1, G_2)| \tag{4}$$

Since computing MCES exactly is NP-hard, practical implementations (e.g., `myopic-mces` with PuLP solver) apply a timeout-based heuristic. Stereochemistry is stripped before comparison. All comparisons should use the same solver, timeout (default: 5 seconds), and atom-matching constraints to be comparable. MassSpecGym uses the `myopic-mces` implementation[9] with `threshold=15` and `always_stronger_bound=True`.

Tanimoto Fingerprint Similarity. Let A and B be binary fingerprint vectors of a predicted and ground-truth molecule. Tanimoto similarity is:

$$T(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}| + |\mathbf{B}| - \mathbf{A} \cdot \mathbf{B}} \tag{5}$$

Fingerprints must be ECFP4 Morgan fingerprints computed with radius $r = 2$ and $n = 2048$ bits (via `rdkit.Chem.AllChem.GetMorganFingerprintAsBitVect` or `rdkit.Chem.rdFingerprintGenerator.FPType.MorganFP`).

Close and Meaningful Match Rates. The Close Match and Meaningful match metrics are complementary structural similarity metrics introduced outside the original MassSpecGym benchmark introduced in (Butler et al., 2023). A *Close Match* requires $T(A, B) \geq 0.675$, and a *Meaningful Match* requires $T(A, B) \geq 0.40$, where T is Tanimoto similarity computed with the default **RDKit** fingerprint instead of ECFP4 (i.e., via `rdkit.Chem.rdFingerprintGenerator.FPType.RDKitFP`. Any deviation from the RDKit fingerprint should be explicitly reported.

Confidence Intervals. Per-example metric values (e.g., per-molecule InChIKey hits or per-prediction Tanimoto similarities) are aggregated by their mean across the test set. To quantify uncertainty around this mean, bootstrapped confidence intervals are computed: the test-set values are resampled with replacement 20,000 times, the mean is recomputed on each resample, and the 0.05th and 99.95th percentiles of the resulting distribution are taken as the lower and upper bounds of the 99.9% confidence interval. We use `scipy.stats.bootstrap` with `confidence_level=0.999`, `n_resamples=20,000`, and a fixed random seed for reproducibility.

A.7. Metric-Implementation Sensitivity Tables

Table 10 shows the influence of evaluation metric parameters on DiffMS performance. High deviation in the results highlights the importance of keeping metric implementations consistent with the original reference implementations (e.g., the ones implemented in the MassSpecGym codebase).

A.8. Agentic Infrastructure for Automated Leaderboard Submission and Code Review

To operationalize the recommendations of this audit and reduce the burden of manual review, MassSpecGym v1.5 introduces a structured submission workflow for new leaderboard entries. Authors submit a pull request to the main repository containing the relevant results, code, and a model card that specifies method metadata including paper and code URLs, pretraining data and filtering criteria, oracle component versions, and evaluation setup. On every pull request, a GitHub

[9] https://github.com/AlBi-HHU/myopic-mces

Table 10. **Comparison of metric variants on identical DiffMS predictions for the MassSpecGym *de novo* generation challenge w/ bonus.** "r" denotes the Morgan fingerprint radius, "thld" denotes the minimum exact distance (`threshold` parameter) below which MCES is computed exactly, and "bound" denotes the `always_stronger_bound` parameter in the MCES implementation. Bold rows show the reference implementation; gray rows show alternative implementations.

Metric	Variant	Top-1	Top-10
Accuracy	InChIKey-14	0.0246 (0.0209, 0.0286)	0.0443 (0.0394, 0.0495)
Accuracy	Full InChIKey	0.0246 (0.0209, 0.0286)	0.0443 (0.0394, 0.0495)
Tanimoto	Morgan r=2, 1024b	0.2968 (0.2925, 0.3012)	0.4034 (0.3990, 0.4084)
Tanimoto	Morgan r=2, 2048b	0.2857 (0.2814, 0.2902)	0.3925 (0.3881, 0.3976)
Tanimoto	Morgan r=2, 4096b	0.2818 (0.2774, 0.2863)	0.3892 (0.3848, 0.3944)
Tanimoto	Morgan r=3, 1024b	0.2375 (0.2335, 0.2415)	0.3266 (0.3224, 0.3314)
Tanimoto	Morgan r=3, 2048b	0.2209 (0.2169, 0.2251)	0.3102 (0.3059, 0.3151)
Tanimoto	Morgan r=3, 4096b	0.2139 (0.2098, 0.2181)	0.3036 (0.2992, 0.3086)
MCES	thld=100, bound=False	21.21 (20.77, 21.66)	16.51 (16.09, 16.94)
MCES	thld=15, bound=False	15.87 (15.49, 16.27)	14.24 (13.86, 14.65)
MCES	thld=15, bound=True	16.02 (15.64, 16.43)	14.39 (14.00, 14.81)

Actions workflow automatically runs a review script, which performs an analysis and heuristic checks covering all failure modes identified in this paper and generates a narrative review of the method. Hard failures and warnings require explicit sign-off by maintainers. A second workflow runs on every merge to `main` and validates the full set of results for schema consistency and confidence interval coverage. A maintainer review guide documents residual judgment calls beyond a static analysis, which includes reading the paper methods section for implicit formula use, verifying pretraining data safety against the InChIKey exclusion list, and spot-checking reproducibility. The system prompt and skill to guide the code review is shown below. Examples of identifying issues in a submission are shown in Figures 3 and 4. This resource is intended to support both the maintainers and users of MassSpecGym with updating the leaderboard and contributors with additional guardrails and feedback in the development of new models.

SKILL.md

```
---
name: review
description: Maintainer guide for reviewing MassSpecGym leaderboard submissions.
↪  Covers interpreting the automated review report, performing judgment calls the
↪  automation cannot make, and approving or rejecting PRs. The automated review
↪  (scripts/review_submission.py, triggered by the review_submission GH Action)
↪  handles deterministic checks; this guide covers the residual human review.
---

# MassSpecGym Submission Review -- Maintainer Guide

## Role of this guide

Every leaderboard PR triggers `scripts/review_submission.py` automatically and posts a
↪  structured report as a PR comment. That report handles all deterministic checks
↪  (schema, CIs, tier integrity, MIST bug, pretraining filter, metric overrides).
↪  **Your job as maintainer is to:**

1. Read the automated report and triage any WARNINGs that require judgment
2. Fetch and read the paper if the automated LLM review flagged concerns or couldn't
↪  access the paper
3. Check items the automation explicitly cannot verify (listed below)
4. Approve or request changes

Hard failures in the automated report must be resolved by the author before you even
↪  look at the PR. Do not override hard failures without a documented reason.

---

## Submission requirements (what a valid PR must contain)
```

```
1. A new row in the correct `results/*.csv` with all required metrics and 95% bootstrap
↪  CIs
2. A `submissions/<method_name>/model_card.yaml` filled from the template

The method name in the CSV `Method` column must exactly match `method_name` in
↪  `model_card.yaml` (underscored folder name, spaced method name). See
↪  `submissions/SUBMISSION_GUIDE.md`.

---

## Step 1: Read the automated report

The PR comment from the review bot contains:

- **Hard failures** -- must be fixed; CI blocks merge
- **Warnings** -- require your sign-off; use a PR review comment to document your
↪  decision
- **LLM narrative review** -- treat as a second opinion; verify any specific issues it
↪  flags

For warnings, document your reasoning inline on the PR before approving.

---

## Step 2: Things the automation cannot check -- do these manually

### 2a. Paper methods section vs. model card

Read the paper's methods/data section and cross-check against `model_card.yaml`:

- Does the paper describe using MIST-CF at inference to pre-filter candidates in the
↪  **mass-based** (standard) challenge? If yes and the model is submitted to
↪  `results/retrieval.csv` (not bonus), reject.
- Does the paper describe pretraining on data sources *not listed* in the model card?
↪  Flag the discrepancy.
- Does the paper use ICEBERG-generated spectra for pretraining? Check which ICEBERG
↪  version (data-safe v1.5 or upstream). If unclear, ask the author.
- Does the paper's reported number match the CSV entry? Values that differ from the
↪  paper by >0.5 pp on the primary metric need explanation.

### 2b. Spectrum-blind shortcut check (S2)

If the model is a retrieval model, mentally check: could a model that ignores the
↪  spectrum entirely (ranking purely by PubChem deposition frequency or SMILES
↪  format) achieve similar results?

The PubChem frequency-prior baseline achieves >90% Recall@1 on non-corrected datasets.
↪  If the submission's Recall@1 is at or below this level, it may not be learning
↪  from spectra at all. Ask the author to compare against the frequency-prior
↪  baseline if not already included.

**Spectrum-blind classifier test (run if suspicious):** A rule-based RDKit format
↪  check (`smiles != Chem.MolToSmiles(Chem.MolFromSmiles(smiles))`) achieves >99%
↪  Recall@1 on non-canonicalized datasets. If the candidate set for this submission
↪  is non-standard, ask the author to confirm the v1.5 pre-canonicalized candidate
↪  set is used.

If you have access to compute: train or run a version of the model with the spectral
↪  input zeroed out or permuted. If performance stays substantially above random
↪  (~1/pool_size), the model likely exploits a spurious correlation and should be
↪  investigated further before acceptance.

### 2c. Pretraining data -- if no parquet provided
```

```
If `pretraining.used=true` but no parquet URL is given, ask the author to either:
- Provide a public parquet for the InChIKey overlap check, or
- Run `python -m massspecgym.data.sanity_check --input their_data.parquet
↪  --inchikey-col inchikey_14` themselves and share the output

Do not accept claims of data safety without this check if pretraining is declared.

### 2d. Reproducibility spot-check

If the submission includes a checkpoint or inference script, spot-check one metric:

```bash
# In the massspecgym conda environment (Python 3.11)
git clone <author-fork-url>
cd MassSpecGym
pip install -e ".[dev]"
python scripts/run.py \
    --job_key review_run \
    --run_name <model_name>_review \
    --test_only \
    --no_wandb \
    --seed 42
```

A discrepancy >0.5 pp on the primary metric requires explanation. If the author cannot
↪ provide a reproducible checkpoint, note this in your review but it is not grounds
↪ for automatic rejection -- reproducibility is aspirational for submissions without
↪ public checkpoints.

Step 3: Metric specifications (reference)

All submitted metrics must match these pinned implementations:

Metric	Specification
InChIKey hit rate	First 14 characters (connectivity layer) only -- **not** full
↪ 27-char
Tanimoto similarity	Morgan ECFP4, radius=2, 2048 bits
MCES distance	`threshold=15`, `always_stronger_bound=True` -- see
↪ `massspecgym/utils.py:MyopicMCES`
Cosine similarity	Standard MS/MS cosine as in `massspecgym`
Jensen-Shannon similarity	As in `massspecgym`

Do not accept metric implementations reimported from third-party codebases unless
↪ independently verified against the specifications above.

Step 4: Required metrics per task

De novo (standard and bonus)
`Top-1 Accuracy`, `Top-1 MCES`, `Top-1 Tanimoto`, `Top-10 Accuracy`, `Top-10 MCES`,
↪ `Top-10 Tanimoto` -- all required, all with CIs.

Retrieval (standard)
`Hit rate @ 1`, `Hit rate @ 5`, `Hit rate @ 20`, `MCES @ 1` -- all required, all with
↪ CIs.

Retrieval (bonus)
`Hit rate @ 1`, `Hit rate @ 5`, `Hit rate @ 20`, `MCES @ 1` -- all required, all with
↪ CIs.
```

```
### Simulation (standard and bonus)
`Cosine Similarity`, `Jensen-Shannon Similarity`, `Hit rate @ 1`, `Hit rate @ 5`, `Hit
↪  rate @ 20` -- all required, all with CIs.

Standard and bonus variants must not be compared in the same table row. A model
↪  submitted to both must have two separate CSV rows.

---

## Step 5: MIST batching bug (I1) -- background

If a submission uses the MIST encoder from an external codebase (not
↪  `massspecgym/models/encoders/mist/`), verify the attention masking in batched
↪  inference. The original MIST encoder added `attn += attn_mask` without first
↪  converting the boolean mask to `-inf`, which allows padding tokens to contribute to
↪  the softmax. The v1.5 MassSpecGym MIST encoder has this fixed.

**Impact if not fixed:** Tanimoto similarity inflates from 0.37 to 0.52; de novo Top-1
↪  inflates by ~17 pp; retrieval Top-1 inflates from 10.73% to 28.50%. This
↪  completely reorders the de novo leaderboard.

**What to look for:**
```python
# BUG (wrong):
attn += attn_mask

# SUBTLE BUG -- fix code present but result discarded:
new_attn_mask = torch.zeros_like(attn_mask, dtype=q.dtype)
new_attn_mask.masked_fill_(attn_mask, float("-inf"))
attn += attn_mask   # still uses raw bool mask, new_attn_mask silently ignored

# FIX (correct):
new_attn_mask = torch.zeros_like(attn_mask, dtype=q.dtype)
new_attn_mask.masked_fill_(attn_mask, float("-inf"))
attn += new_attn_mask
```

The v1.5 MassSpecGym MIST encoder
↪ (`massspecgym/models/encoders/mist/transformer_layer.py`) already applies the fix.
↪ If the submission imports from there, this is satisfied.

Step 6: Data leakage -- gradient of stringency

Data leakage is not binary. The steepest performance gradient appears at the
↪ transition from exact-match exclusion to Tanimoto greater than or equal to
↪ 0.70-0.80 filtering -- not at the exact-match boundary itself. This means a model
↪ may claim "clean" data (exact-match excluded) while still benefiting substantially
↪ from near-neighbor contamination.

What to report in your review:
- What filtering criterion is declared in the model card?
- Is this consistent with what the paper states?
- Does the paper compare against baselines at the same filtering level?

The leaderboard accepts exact-match exclusion as the minimum. If the criterion is
↪ weaker or unstated, request clarification before approval.

Final approval checklist

Before approving the PR, confirm:
```

```
- [ ] Automated report has no hard failures (or failures have been resolved and
↪  re-reviewed)
- [ ] All warnings have been signed off with a documented reason
- [ ] Paper methods section consistent with model card
- [ ] No implicit formula use in standard-tier submission (from reading the paper)
- [ ] Pretraining data safety confirmed (parquet sanity check run or author confirmed)
- [ ] If MIST encoder used: v1.5 fix confirmed or external repo verified
- [ ] If MIST-CF or ICEBERG used: v1.5 data-safe version confirmed
- [ ] All required metrics present with CIs
- [ ] Method name in CSV matches model card exactly
- [ ] `df_test_path` configured so per-sample predictions are saved (recommended)
- [ ] Seed documented
```

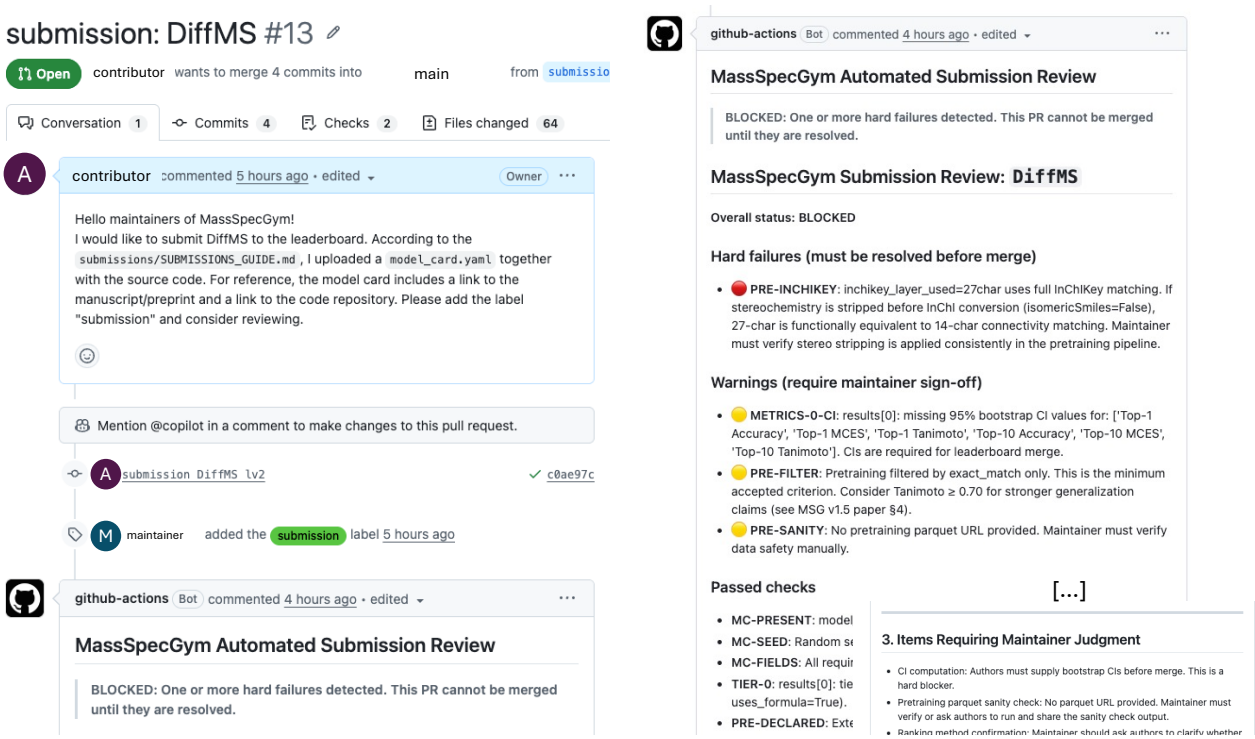

Figure 3. **Example 1 of an LLM-enabled workflow for reviewing submissions to the MassSpecGym leaderboard.** When a contributor submits a pull request for their model to be featured in the leaderboard, an automated GitHub Action workflow is evoked that parses through the provided model card and source code to review according to the guidelines outlined in the provided skill (*vide supra*) based on this publication. The output of the workflow is a set of errors, warnings and recommendations for a human maintainer. Once approved, the source code files are deleted, the model card is kept and the leaderboard updated.

submission: MIST+MolForge #11

Open contributor wants to merge 6 commits into main from submission/mistmolforge-wro

Conversation 1 Commits 6 Checks 2 Files changed 115

A contributor commented 6 hours ago Owner

Requesting to show MIST+MolForge on the leaderboard. According to the submissions/SUBMISSIONS_GUIDE.md, I uploaded a model_card.yaml together with the source code. It follows the provided template:

```
method_name: ""
paper_url: ""
code_url: ""
random_seed: null

results:
  - task: ""              # One of: de_novo | retrieval | simulation
    challenge: ""         # One of: standard | bonus
    uses_formula_at_inference: false  # true if formula predictor used to pre-filter candidate

    metrics:
      # De novo example (remove the block that doesn't apply):
      # Top-1 Accuracy: 0.00
      # Top-1 Accuracy CI Low: 0.00
      # Top-1 Accuracy CI High: 0.00
      # Top-1 MCES: 0.00
...
      # Top-10 Tanimoto CI High: 0.00

uses_mist_encoder:     # true if MIST spectrum encoder used
uses_mist_cf:          # true if MIST-CF formula predictor used at any stage
uses_iceberg:          # true if ICEBERG spectral simulator used at any stage

mist_cf_version: ""          # "v1.5_data_safe" or describe the version used
iceberg_version: ""          # "v1.5_data_safe" or describe the version used

# Pretraining data
pretraining:
  used: false              # Set true if ANY external data was used for pretraining
                           # (decoder pretraining, encoder pretraining, fine-tuning, etc.)

  # Fill in the fields below only if used=true:
  description: ""          # Free text: what datasets, how many molecules, which model compone
  filtering_criterion: ""  # How test/val molecules were excluded from pretraining data.
  inchikey_layer_used: ""  # "14char" (connectivity, correct) or "27char" (full, insufficient)
  parquet_url: ""
```

github-actions Bot commented 6 hours ago • edited

MassSpecGym Automated Submission Review

> BLOCKED: One or more hard failures detected. This PR cannot be merged until they are resolved.

MassSpecGym Submission Review: `MIST + MolForge`

Overall status: BLOCKED

Hard failures (must be resolved before merge)

- 🔴 **PRE-INCHIKEY**: inchikey_layer_used=27char uses full InChIKey matching. If stereochemistry is stripped before InChI conversion (isomericSmiles=False), 27-char is functionally equivalent to 14-char connectivity matching. Maintainer must verify stereo stripping is applied consistently in the pretraining pipeline.
- 🔴 **I1-MIST-BUG**: MIST batching bug detected: found 'attn += attn_mask' without -inf mask fill. This inflates Tanimoto similarity from 0.37 to 0.52 and de novo Top-1 by ~17 pp.

  `src/mist/models/transformer_layer.py`

Warnings (require maintainer sign-off)

- 🟡 **MC-SEED**: random_seed not specified in model_card.yaml. Document the seed used for reported results.
- 🟡 **METRICS-0-CI**: results[0]: missing 95% bootstrap CI values for: ['Top-1 Accuracy', 'Top-1 MCES', 'Top-1 Tanimoto', 'Top-10 Accuracy', 'Top-10 MCES', 'Top-10 Tanimoto']. CIs are required for leaderboard merge.
- 🟡 **PRE-FILTER**: Pretraining filtered by exact_match only. This is the minimum accepted criterion. Consider Tanimoto ≥ 0.70 for stronger generalization claims (see MSG v1.5 paper §4).
- 🟡 **PRE-SANITY**: Pretraining parquet declared at: https://zenodo.org/records/15122968. Automated InChIKey overlap check requires local access; maintainer should run: python -m massspecgym.data.sanity_check --input <downloaded.parquet> --inchikey-col inchikey_14

Passed checks

- **MC-PRESENT**: model_card.yaml found and parseable.

Figure 4. **Example 2 of an LLM-enabled workflow for reviewing submissions to the MassSpecGym leaderboard.** Similarly to Figure 3, the output of the workflow is a set of errors, warnings and recommendations for a human maintainer. The workflow correctly identifies data leakage due to stereoisomers and the MIST inference bug.