GLACIER: Rethinking Mass Spectrum Prediction as an Object Detection Problem

Rui-Xi Wang **Runzhong Wang*** **Connor W. Coley***
Massachusetts Institute of Technology
Cambridge, MA 02139
{rxwangtw, runzhong, ccoley}@mit.edu

Abstract

Predicting tandem mass spectra (MS/MS) from molecular structures represents a central task in analytical chemistry with direct relevance to clinical metabolomics, systems biology, and adjacent disciplines. In this work, we revisit the problem through the lens of object detection on molecular graphs. Molecular fragmentation, a central step in MS/MS prediction, can be approximated as detecting a set of subgraphs (i.e., fragments) and their associated spectral contributions. Existing fragment-based models follow a two-stage paradigm—first generating candidate fragments and then scoring them—analogous to two-stage R-CNNs in computer vision. Towards higher accuracy and faster inference, we introduce GLACIER, a single-stage transformer-based fragment detection neural network for molecular graphs. This unified formulation eliminates the need for candidate enumeration, enabling scalable and globally consistent modeling of molecular fragmentation. GLACIER is faster and more accurate than existing state-of-the-art by a significant margin, achieving 70.0% and 69.7% Top-1 retrieval accuracy with and without contrastive finetuning on the MassSpecGym dataset (from the previous SOTA of 64.0%) and 52.5% and 38.5% respectively on the NIST'20 dataset (from 33.2%). Furthermore, GLACIER provides nearly 8-fold inference speedup over our prior two-stage model. Code is available at `https://github.com/coleygroup/ms-pred`.

1 Introduction

Tandem mass spectra (MS/MS) are measured at scale in chemistry and biology campaigns to help elucidate the structures of unknown molecules. MS/MS prediction aims to model the fragmentation of molecules, where a given molecular graph yields a set of fragment ions with associated intensities. An accurate MS/MS predictor can generate pseudo-standard spectra for candidate molecules, accelerating chemical and biomedical discoveries as demonstrated in case studies including Qiang et al. [19] and Wang et al. [31]. Existing approaches typically formulate this task as formula or fragment enumeration, relying on domain-specific heuristics [1, 22] or multi-stage model architectures [9, 10, 16, 36]. While effective, these formulations obscure a vital design perspective: MS/MS prediction can be viewed as an object detection problem by drawing analogy between visual objects and molecular graphs.

Specifically, MS/MS prediction can be formulated as taking a molecular graph as input, proposing a set of subgraphs (i.e., fragments) and predicting their properties (i.e., intensities). Since we can calculate the mass of each fragment and we have the predicted intensities, such a framework predicts mass spectrum as a set of (`mass`, `intensity`) pairs. This framework closely parallels object detection in computer vision—subgraph detection is the graph-based version of bounding-box detection, and the intensity regressor is analogous to the classifier head in computer vision. We also want to note

*Corresponding to Runzhong Wang and Connor W. Coley.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

that chemical rearrangement may occur in reality, and that a set of subgraphs does not fully reflect the physics of collision-induced dissociation [26], but it can be viewed as a reasonable approximation, empirically, in the context of spectral prediction.

Such a perspective suggests that the methodological evolution of object detection offers a useful lens for rethinking MS/MS prediction. Early object detection approaches were dominated by two-stage pipelines, such as R-CNNs [7, 8, 11, 21], which first predict region proposals and then classify and refine them by a separate neural network. Analogously, current state-of-the-art MS/MS predictors [9, 28, 31, 36] first propose candidate fragments using fragmentation simulators, and then use a separate model to score these candidates by predicting intensities. While effective, such approaches inherit key limitations of two-stage designs, including low inference throughput, complex training pipelines, and, specifically for MS/MS prediction, heuristic-based fragmentation.

In computer vision, these limitations motivated a shift towards one-stage architectures. One-stage detectors improved efficiency by predicting both bounding boxes and classification labels in a single pass, with further benefits brought by stabilized training [14, 20]. More recently, transformer-based models such as DETR [4] reformulated detection as a set prediction problem, eliminating the need for heuristic components like anchor boxes and non-maximum suppression. These models enable unified one-stage training while capturing global dependencies through attention mechanisms.

Inspired by this progression, we envision that MS/MS prediction should undergo a similar paradigm shift. Instead of decomposing the problem into separate stages of candidate generation and scoring, we propose to directly model molecular fragmentation as a set prediction problem. As shown in Figure 1, we present Graph Learning of Atomic Component Instances via End-to-end Recognition (GLACIER), a transformer-based architecture that jointly proposes plausible subgraphs and predicts their intensities in the mass spectrum without relying on hand-crafted intermediate steps.

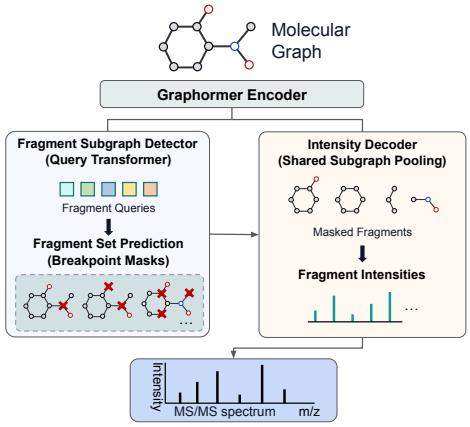


Figure 1: Overview of GLACIER as a one-stage MS/MS predicton neural network.

Developing such an one-stage graph detection model also presents unique challenges, including predicting multiple breakpoints to generate complex fragments, enabling efficient subgraph feature pooling to maximize parameter efficiency, and mitigating the limitation of heuristic-based fragmentation. To address these challenges, we propose a principled one-stage formulation of MS/MS prediction that unifies fragment generation and intensity prediction with a transformer. This design generates fragments with various depths in a single forward pass, improves computational efficiency, and enables globally consistent modeling of molecular fragmentation.

Our contributions are summarized as follows:

1) We incorporate a differentiable breakpoint predictor into GLACIER that supports multiple breakings. In contrast, existing single-stage MS/MS predictors only allow single-atom/bond fragmentation [18], resulting in limited coverage. Our approach matches the flexibility of two-stage models such as ICEBERG [31] while retaining a one-stage formulation. This is achieved with multi-head predictors coupled with a differentiable constraint projection layer [30] that enables control over the number of broken atoms, and a separate branch to predict the cardinality. Inspired by DETR [4], we utilize a query-based transformer to handle parallel prediction of multiple fragmentations; in training, the predicted breaking sites are mapped to the closest labeled patterns using the Hungarian algorithm.

2) In GLACIER, we utilize a shared Graphormer backbone [34] and introduce an efficient subgraph pooling strategy that aggregates features of selected atoms directly from the global representation. This approach, referred to as dynamic embedding in prior work [29], avoids redundant message passing over individual fragments, leading to improved inference speed and parameter efficiency. More broadly, while single-stage MS/MS prediction can benefit from a shared feature backbone, extracting subgraph-level features remains non-trivial. Naively applying message passing to each fragment in a one-stage model would undermine its theoretical efficiency, motivating our design.

3) In GLACIER, we propose a training strategy that uses fragment supervision with heuristic labels while progressively transitioning to an end-to-end objective. Specifically, we initialize the model with fragmentation labels generated by the MAGMa heuristic [22] and apply teacher forcing to learn spectral intensities, then gradually replace this signal with an end-to-end intensity objective. This design is motivated by a key challenge in MS/MS prediction: unlike visual object detection ground-truth fragment annotations are virtually always unavailable [26]. As a result, naive end-to-end training from scratch can be unstable. Our approach balances training stability with the goal of learning an accurate intensity predictor that remains robust to imperfect fragment assignments.

4) Extensive experiments on standard benchmarks demonstrate GLACIER's significant improvement in speed and accuracy. On the NIST'20 [17] dataset, compared to existing state-of-the-art, GLACIER improves top-1 retrieval accuracy from 33.5% to 52.5% on a random split and from 31.6% to 46.0% on a scaffold split. On the MassSpecGym dataset [3], GLACIER improves top-1 retrieval accuracy from 64.0% to 70.0% (mass challenge) and from 44.4% to 49.9% (formula challenge). The single-stage design also benefits efficiency, achieving a $\approx$8-fold speedup over a two-stage baseline [31].

2 Related Work

Tandem Mass Spectra Prediction. Machine learning-based MS/MS prediction has enabled *in silico* spectra generation directly from molecular structure. Several methods rely on kernel models and probabilistic fragmentation trees [1, 23, 27], while others leverage deep learning to capture complex fragmentation patterns [33, 35]. Current best-performing approaches are dominated by two-stage, fragment-centric methods, such as ICEBERG [10, 31], FraGNNet [36], and MARASON [28], as well as a flow-based model, MolSpecFlow [32], aiming to better capture the generative nature of spectra. However, key limitations remain: the dominant two-stage pipeline introduces substantial computational overhead. While FIORA [18] can perform one-step prediction with explicit fragmentation, it only supports single-step breakage and hence results in limited spectral peak coverage. These challenges motivate us to develop more efficient, single-stage models that better integrate fragmentation structure with downstream objectives.

Object Detection in Computer Vision. Object detection aims at predicting bounding boxes and classes for visual objects. Early object detection models were two-stage, such as R-CNNs [7, 8, 11, 21], which rely on region proposal networks followed by per-region classification and refinement, introducing architectural complexity and hand-designed components. One-stage detectors like YOLO [20] and SSD [14] improve efficiency, but still depend on heuristics such as anchor boxes and post-processing. More recently, DETR [4] reformulates detection as a set prediction problem using transformers, eliminating these components via bipartite matching and object queries. DETR-style models have been primarily developed for images and point clouds [15], while their extension to discrete graph-structured data remains underexplored. Unlike images, graphs lack a canonical spatial structure and exhibit irregular connectivity, making object definition ambiguous. In this work, we extend the DETR paradigm to molecular graphs by treating subgraph partitions as detection targets, enabling single-stage prediction of overlapping molecular fragments without predefined rules.

3 The GLACIER Model

This section discusses three building blocks of GLACIER, as shown in Figure 2, and the training loss.

3.1 Graphormer Feature Backbone

We adapt Graphormer [34] to encode molecules. Given a molecular graph, we represent each atom and bond as a feature fingerprint vector (see Appendix A for details). We then project these features, along with instrumental parameters such as adduct type (one-hot encoded) and collision energy (positional encoded), into node embeddings and edge-level attention biases via linear layers.

We process the resulting representations through multiple Graphormer layers together with a learned graph-level token. This produces node-level embeddings $\mathbf{N} \in \mathbb{R}^{n_{\text{atom}} \times D}$ for n_{atom} nodes and a graph-level embedding $\mathbf{G} \in \mathbb{R}^{1 \times D}$, where D is the feature dimension. We subsequently use these embeddings with downstream modules for fragment generation and intensity prediction.

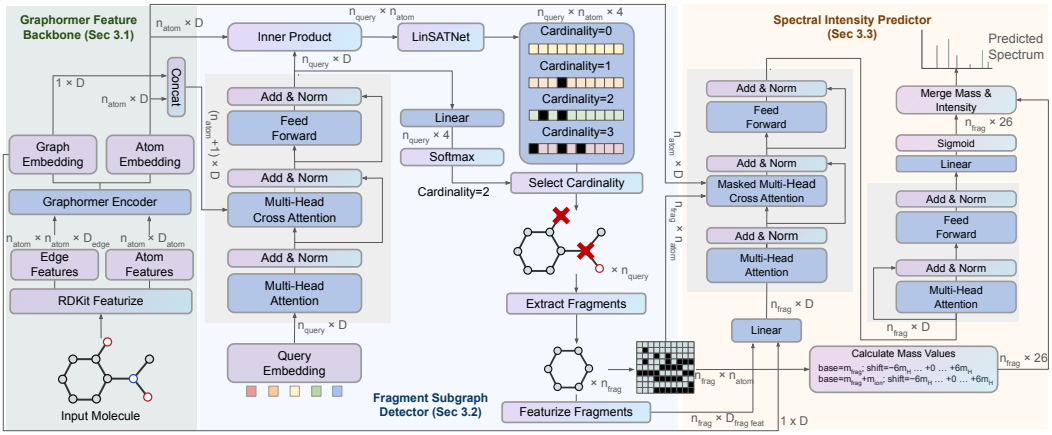


Figure 2: GLACIER architecture. Given a molecule as input, we first encode its structure using Graphormer to obtain contextualized graph and atom embeddings (Section 3.1). These embeddings are then used by the fragment subgraph detector to predict breakpoints and generate candidate fragment substructures (Section 3.2). Finally, the spectral intensity predictor extracts subgraphs using Graphormer embeddings and predicts peak intensities for each predicted fragment (Section 3.3).

3.2 Fragment Subgraph Detector

Graph Detection. The largest technical challenge of GLACIER is how to design a single-pass fragment detector with minimal additional enumerations to avoid significant inference burden on CPU. We formulate fragment prediction as an object detection problem on graphs, where each "object" corresponds to a fragment subgraph of the molecular graph. To achieve this, we adapt the transformer-based object detection framework, DETR [4], to the graph domain. Following Carion et al. [4], we initialize a set of learnable query tokens $\mathbf{T} \in \mathbb{R}^{n_{\text{query}} \times D}$. "Object queries" were proposed in DETR to provide a fixed number of anchors for predicted objects. In GLACIER, queries are used to predict atom-breaking patterns. n_{query} is the number of queries. $\mathbf{T}$ is initialized using the orthogonal initialization scheme [24]. These query tokens cross-attend the concatenated node embeddings $\mathbf{N}$ and the graph-level embedding $\mathbf{G}$, and $[\cdot \parallel \cdot]$ means tensor concatenation along the feature dimension:

$$\bar{\mathbf{T}} = \text{TransformerDecoder}(\mathbf{T}, [\mathbf{G} \parallel \mathbf{N}]). \tag{1}$$

$\bar{\mathbf{T}}$ is then used to compute a query-node affinity matrix $\mathbf{S} \in \mathbb{R}^{n_{\text{query}} \times n_{\text{atom}}}$, where n_{atom} is the number of heavy atoms and $S_{ij} = \bar{\mathbf{T}}_i \cdot \mathbf{N}_j$ measures the relevance between query i and node j.

Breakpoint Prediction. We convert the affinity matrix $\mathbf{S}$ into binary masks representing fragment subgraphs. However, directly predicting node-level masks is challenging in the mass spectrum setting, because even a single-atom difference in the mask will result in completely different mass values, and existing mass spectrum similarity metrics [12] will treat that as unmatched. Instead, GLACIER predicts fragments by predicting the boundaries of the subgraphs, i.e., the breaking sites that induce the subgraphs.

Given an adjacency matrix $\mathbf{A} \in \{0,1\}^{n_{\text{atom}} \times n_{\text{atom}}}$ and a binary subgraph mask $\mathbf{m} \in \{0,1\}^{n_{\text{atom}}}$, we define the ground-truth boundary (breakpoint) mask $\mathbf{b}^{gt} \in \{0,1\}^{n_{\text{atom}}}$ as:

$$\mathbf{b}^{gt} = (\mathbf{A}\mathbf{m}) \odot (\mathbf{1} - \mathbf{m}), \tag{2}$$

where $\odot$ denotes element-wise multiplication. This formulation identifies nodes that are adjacent to the subgraph but not contained within it, analogous to the bounding box of an object in images.

With the observation that fragmentation events involve only a small number of bond breaks and to constrain the model dimension, we assume that the number of breakpoints is bounded by a hyperparameter k (set to $k = 3$ in our experiments). Given fragment subgraphs generated from the MAGMa heuristic [22], we use Eq. (2) to derive corresponding breakpoint patterns for supervision.

To predict breakpoint patterns, we project each affinity vector $\mathbf{S}_i \in \mathbb{R}^{n_{\text{atom}}}$ into a set of cardinality-constrained vectors using the LinSATNet layer [30]:

$$\mathbf{C}_i = \left[\text{LinSATNet}(\mathbf{S}_i, 0, \tau), \text{LinSATNet}(\mathbf{S}_i, 1, \tau), \ldots, \text{LinSATNet}(\mathbf{S}_i, k, \tau)\right] \in \mathbb{R}^{(k+1) \times n_{\text{atom}}},$$

where $\mathbf{C}_{i,k'} = \mathrm{LinSATNet}(\mathbf{S}_i, k', \tau)$ is a differentiable top-k' projection layer that enforces the cardinality by optimal transport, so that $\mathbf{C}_{i,k'}$ satisfies $\mathbf{1} \cdot \mathbf{C}_{i,k'} = k'$, and $\mathbf{C}_{i,k'} \in [0,1]^{n_{\mathrm{atom}}}$. Its closeness to binary $\{0,1\}$ is controlled by the temperature τ. Therefore, each row of $\mathbf{C}_i$ is a soft top-k' activation of $\mathbf{S}_i$. The first row denotes the entire molecule and is always all 0s, corresponding to a cardinality of zero.

At the same time, for each i, we predict which cardinality to use, i.e., which row of $\mathbf{C}_i$ to select:

$$\mathbf{\Gamma} = \mathrm{Softmax}(\mathrm{MLP}(\bar{\mathbf{T}})) \in \mathbb{R}^{n_{\mathrm{query}} \times (k+1)}, \tag{3}$$

where each entry in $\mathbf{\Gamma}$ represents the probability that a particular query corresponds to a breakpoint pattern with a particular cardinality between 0 and k.

Set Prediction Loss for Breakpoint Detection. Following the DETR framework [4], our model produces a fixed-size set of n_{query} predictions, where n_{query} is chosen to be larger than the typical number of breakpoint patterns in a molecular graph. In training, it is treated as a permutation-invariant set prediction problem via bipartite matching.

We define the pairwise loss $\hat{\mathcal{L}}^{ij}_{\text{breakpoint}}$ between query i and ground-truth breakpoint pattern j as:

$$\hat{\mathcal{L}}^{ij}_{\text{breakpoint}} = \mathrm{CrossEntropy}\left(\frac{\mathbf{C}_{i,k_j^{gt}}}{\sum \mathbf{C}_{i,k_j^{gt}}},\ \frac{\mathbf{b}_j^{gt}}{k_j^{gt}}\right) + \mathrm{CrossEntropy}\left(\mathbf{\Gamma}_i,\ \mathrm{OHE}(k_j^{gt})\right), \tag{4}$$

where k_j^{gt} denotes the ground-truth cardinality of pattern j, $\mathrm{OHE}(k_j^{gt})$ is the one-hot encoding of k_j^{gt}, and $\mathbf{C}_{i,k_j^{gt}} \in \mathbb{R}^{n_{\mathrm{atom}}}$ is the predicted node distribution conditioned on cardinality k_j^{gt}. The first term measures the alignment between predicted and ground-truth breakpoints, while the second term supervises the predicted cardinality.

The remaining question is how to map each query i to the ground-truth pattern j while keeping the permutation invariance. We compute an optimal bipartite matching between the predicted and ground-truth patterns using the Hungarian algorithm, and optimize the model based on the resulting matched pairs. Specifically, we find permutation $\pi(\cdot)$ that minimizes the per-layer auxiliary loss,

$$\hat{\mathcal{L}}_{\mathrm{fragment}} = \min_{\pi} \sum_{i=1}^{n_{\mathrm{query}}} \hat{\mathcal{L}}^{i\pi(i)}_{\text{breakpoint}}, \tag{5}$$

where $\hat{\mathcal{L}}^{i\pi(i)}_{\text{breakpoint}}$ is the pairwise matching cost between query i and ground-truth pattern with index $\pi(i)$. Following DETR [4], we further incorporate auxiliary decoding losses at the output of each transformer decoder layer to promote faster convergence and improved prediction accuracy. The final loss objective, $\mathcal{L}_{\mathrm{fragment}}$, is defined as a weighted sum of these auxiliary losses across all decoder layers, with decaying weights by a factor of 0.9 with decreasing depth.

Breakpoint Inference. At inference time, each query produces a breakpoint pattern by first selecting its predicted cardinality, $\hat{k}_i = \arg\max \mathbf{\Gamma}_i$. Conditioned on this cardinality, we extract the corresponding node scores $\mathbf{C}_{i,\hat{k}_i} \in \mathbb{R}^{n_{\mathrm{atom}}}$ and construct the predicted breakpoint mask $\hat{\mathbf{b}}_i \in \{0,1\}^{n_{\mathrm{atom}}}$ by selecting the top-$\hat{k}_i$ nodes:

$$\hat{\mathbf{b}}_i = \mathrm{LinSATNet}(\mathbf{C}_{i,\hat{k}_i}, \hat{k}_i, 0). \tag{6}$$

Given breakpoints, fragments are further constructed as connected components after removing all predicted breakpoints. The details of GPU-friendly fragment construction during inference can be found in Appendix B.

3.3 Spectral Intensity Predictor

Existing two-stage models [31, 36] compute embeddings for each fragment subgraph independently, leading to significant computational overhead due to redundant message passing over overlapping regions of the molecular graph. To address this inefficiency and fully leverage the expressive power of the Graphormer encoder, we propose an intensity predictor with an efficient strategy for subgraph pooling. Prior work in graph edit distance learning [29] shows that one can simply pool node

embeddings to build subgraph features instead of rerunning message passing on every subgraph. This strategy, known as dynamic embedding, inspire the design of the intensity model.

Given a set of fragment masks $\mathcal{F}$ (where $|\mathcal{F}| = n_{\text{frag}}$) a molecular graph, and the learned graph-level token $\mathbf{G}$, we construct fragment-level embeddings $\mathbf{H}$ using chemically relevant features (see Appendix C for details), where each $\mathbf{H}_i$ corresponds to a fragment $F_i \in \mathcal{F}$.

We then refine these embeddings using a transformer decoder that performs masked attention over the full set of node embeddings $\mathbf{N}$:

$$\bar{\mathbf{H}} = \text{TransformerDecoder}(\mathbf{H}, \mathbf{N}, \mathcal{F}), \tag{7}$$

where $\mathbf{N}$ serves as the memory and $\mathcal{F}$ defines binary attention masks that restrict each fragment to attend only to its corresponding nodes. To further capture interactions between fragments, we apply a transformer encoder over the fragments:

$$\hat{\mathbf{H}} = \text{TransformerEncoder}(\bar{\mathbf{H}}). \tag{8}$$

Finally, fragment intensities are predicted by passing the refined embeddings $\hat{\mathbf{H}}$ through an MLP that maps each fragment embedding to a 26 dimensional vector that represents 13 hydrogen shift states (losing or obtaining up to 6 hydrogen atoms) and both charge retention fragmentation (CRF) and charge migration fragmentation (CMF). This setup is modified from [31] with details in Appendix D.

3.4 Training Objective

The overall training objective consists of three components: the breakpoint prediction loss (discussed in Section 3.2), a spectral distance loss, and an auxiliary ground-truth fragment spectral loss. We define the spectral distance loss as

$$\mathcal{L}_{\text{inten}} = \mathcal{D}(s^{gt}, \hat{s}), \tag{9}$$

where $\mathcal{D}$ denotes the spectral cosine distance, s^{gt} is the ground-truth spectrum, and $\hat{s}$ is the predicted spectrum. All spectral vectors are discretized into m/z bins with a width of 0.1 Dalton or 0.01 Dalton depending on the dataset specification.

To further stabilize training, we introduce an auxiliary loss $\hat{\mathcal{L}}_{\text{inten}}$ with the same formulation as $\mathcal{L}_{\text{inten}}$, but computed by feeding heuristically-labeled fragment patterns into the Intensity Predictor, similar to teacher forcing. This auxiliary objective helps stabilize the learning of the Intensity Predictor, particularly during the early stages of training. We validate its importance in Table 5.

The full training objective is defined as

$$\mathcal{L}_{\text{train}} = w_{\text{magma}} \cdot \left(w_{\text{inten}} \cdot \hat{\mathcal{L}}_{\text{inten}} + w_{\text{frag}} \cdot \mathcal{L}_{\text{fragment}} \right) + (1 - w_{\text{magma}}) \cdot w_{\text{inten}} \cdot \mathcal{L}_{\text{inten}}, \tag{10}$$

where w_{inten} and w_{frag} are fixed weighting hyperparameters, and w_{magma} starts from 1 for n_{magma} steps and and exponentially decays by a fixed rate λ_{magma} after each n_{decay} steps, where n_{magma}, n_{decay}, and λ_{decay} are the hyperparameters that control the number of warmup steps, the number of decay steps, and the decay rate, respectively. This design initializes fragment prediction using MAGMa heuristic [22] and stabilizes training in the early phase, while gradually shifting focus toward the real spectral prediction objective as training progresses.

Finally, we adopt a separate contrastive finetuning step following ICEBERG [31] to further distinguish isomers, improving the downstream retrieval accuracy. We provide the details of contrastive finetuning in Appendix E.

4 Experiments

We evaluate GLACIER on established benchmarks for computational mass spectrometry, including NIST'20 [17] and MassSpecGym [3]. We focus on two tasks: molecular retrieval and spectrum prediction (forward prediction). The experiments are designed to answer the following research questions: (1) *Does GLACIER improve the performance of mass spectrum prediction compared to two-stage fragment-prediction models and other state-of-the-art models?* (2) *How accurate is GLACIER on an authentic assessment of identifying unknown compounds to support its application in real-world*

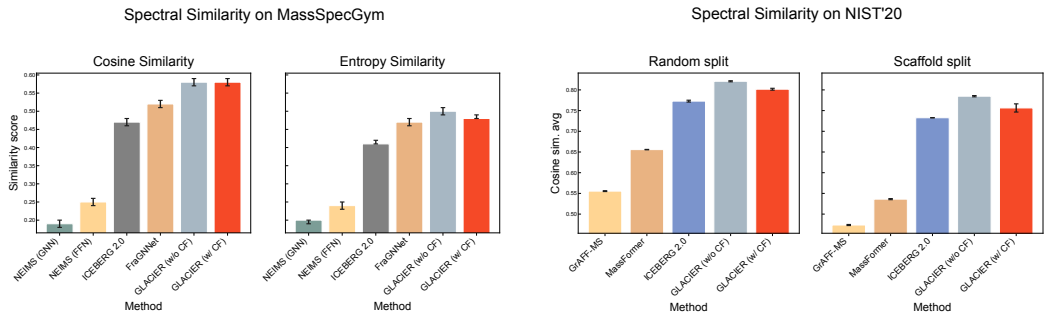


Figure 3: **Left**: Cosine and entropy similarity between experimental spectra and predictions on the MassSpecGym [3] dataset. **Right**: Cosine similarity between experimental spectra and predictions on the NIST'20 [17] dataset with all adduct types. This plot does not include MolSpecFlow because it uses a different bin width. CF: contrastive finetuning.

chemical and biological campaigns? (3) *Does GLACIER generate reasonable fragmentation patterns that align with the chemical intuition?* (4) *Does GLACIER accelerate the inference pipeline with its single-stage integration?* All experiments are conducted on a workstation with AMD 3995WX CPU, 4×NVIDIA A5000 GPU, and 512GB RAM. We conduct experiments and answer these research questions quantitatively and qualitatively.

4.1 Datasets and Evaluation Metrics

NIST'20. We train our models on the NIST'20 [17] dataset with 530,640 high-energy collision-induced dissociation (HCD) spectra and 25,541 unique molecular structures. We evaluate on two different splits: (1) a random split that partitions data by InChI key and (2) a Murcko scaffold [2] split that clusters different molecular scaffolds and requires more generalization to out-of-distribution scaffolds. We include all peer methods reported in Wang et al. [31] for benchmarking purposes.

MassSpecGym. We adopt the official MassSpecGym data split, which contains 231,104 tandem mass spectra (MS/MS) spanning 31,602 unique molecules. Importantly, this benchmark uses a Maximum Common Edge Subgraph (MCES)-based splitting strategy. The MCES protocol enforces substantial structural dissimilarity between training and testing molecules (edit distance ≥ 10). As a result, it provides a more stringent evaluation of a model's ability to generalize to previously unseen chemical space, rather than relying on memorization of known structural patterns.

Evaluation Metrics. We evaluate GLACIER's capability in spectral simulation using top-k retrieval accuracy, cosine similarity, and Jensen-Shannon similarity (interchangeable with spectral entropy similarity [12]). Binned spectral vectors are used with 0.1 resolution on the NIST'20 dataset and 0.01 resolution on the MassSpecGym dataset to align with prior work on both benchmarks.

4.2 Results and Discussions

4.2.1 Spectral Prediction Similarity and Visualization

We evaluate the prediction power of GLACIER by comparing the spectral similarity between the predicted spectrum and the experimental spectrum on the MassSpecGym and the NIST'20 datasets. As shown in Figure 3 (left), on MassSpecGym, GLACIER demonstrates better spectral similarity to ICEBERG. In Figure 3 (right), GLACIER outperforms all baselines on the NIST'20 dataset. Error bars are reported on three random seeds. Comparison with more baselines that only supports positive adduct types and the $[M+H]^+$ adduct type along with the peak coverage statistics is reported in Appendix G, where GLACIER still outperforms all prior methods. To evaluate MS/MS predictions qualitatively, we report a visualization of predicted spectra in Figure 4 in the Appendix.

4.2.2 Molecular Retrieval

We evaluate the real-world applicability of MS/MS simulators and summarize the evaluation results in Table 1 and Table 2. GLACIER outperforms all baselines on MassSpecGym retrieval tasks, including the mass challenge and the formula bonus challenge, and the NIST'20 retrieval benchmark

Table 1: Retrieval accuracy (mean with 99.9% confidence intervals upon bootstrapping, 20,000 resamples) on the MassSpecGym dataset [3]. GLACIER surpasses all baselines in retrieval accuracy. Note that MolSpecFlow [32] does not have an open-source implementation and only reports on the mass challenge. It also uses a different bin width (0.1 Dalton) instead of the bin width adopted by MassSpecGym (0.01 Dalton); the comparison has to be made with extra care.

Model	Recall @ 1	Recall @ 5	Recall @ 20
Mass Challenge: Sample up to 256 candidates with the same mass			
NEIMS (FFN) [33]	8.44 (7.56-9.34)	21.43 (20.10-22.79)	38.57 (36.99-40.23)
NEIMS (GNN) [37]	3.95 (3.37-4.62)	11.92 (10.87-13.00)	26.27 (24.83-27.82)
FraGNNet [36]	46.64 (44.98-48.26)	72.56 (71.18-74.00)	83.58 (82.34-84.75)
MolSpecFlow [32]	55.32 (52.97-56.87)	79.98 (79.01-81.32)	89.54 (88.32-90.76)
ICEBERG 2.0 [31]	63.95 (62.39-65.58)	83.83 (82.56-84.99)	92.56 (91.61-93.41)
GLACIER (ours, w/o CF)	69.69 (68.16-71.21)	86.58 (85.41-87.69)	93.34 (92.42-94.14)
GLACIER (ours, w/ CF)	**69.95 (68.36-71.50)**	**86.47 (85.30-87.56)**	**93.52 (92.62-94.31)**
Formula Challenge: Sample up to 256 candidates with the same chemical formula			
NEIMS (FFN) [33]	7.62 (6.77-8.54)	22.70 (21.32-24.12)	44.12 (42.51-45.75)
NEIMS (GNN) [37]	3.63 (3.05-4.29)	13.55 (12.46-14.68)	33.77 (32.26-35.37)
FraGNNet [36]	31.93 (30.40-33.50)	63.20 (61.64-64.76)	82.70 (81.45-83.93)
MARASON [28]	34.03 (32.86-35.20)	64.04 (62.77-65.19)	85.39 (84.48-86.24)
ICEBERG 2.0 [31]	44.35 (42.75-46.02)	73.57 (72.06-75.02)	89.45 (88.44-90.43)
GLACIER (ours, w/o CF)	48.49 (46.88-50.15)	76.56 (75.19-78.01)	90.60 (89.62-91.57)
GLACIER (ours, w/ CF)	**49.88 (48.25-51.49)**	**78.13 (76.78-79.47)**	**90.75 (89.75-91.70)**

Table 2: Retrieval accuracy (mean $\pm$ 95% confidence interval on 3 random restarts) on NIST'20 [17] dataset on all adduct types. GLACIER surpasses all baselines in terms of retrieval accuracy. Results of all baselines are taken from Wang et al. [31].

Accuracy @ Top-k	1	2	3	4	5	8	10
Random (InChI Key) Split							
GrAFF-MS [16]	$0.165_{\pm 0.003}$	$0.305_{\pm 0.003}$	$0.406_{\pm 0.004}$	$0.479_{\pm 0.005}$	$0.535_{\pm 0.001}$	$0.653_{\pm 0.004}$	$0.708_{\pm 0.007}$
MassFormer [35]	$0.212_{\pm 0.004}$	$0.360_{\pm 0.005}$	$0.465_{\pm 0.005}$	$0.538_{\pm 0.004}$	$0.595_{\pm 0.006}$	$0.715_{\pm 0.003}$	$0.765_{\pm 0.003}$
ICEBERG 2.0 [31]	$0.335_{\pm 0.000}$	$0.549_{\pm 0.009}$	$0.662_{\pm 0.004}$	$0.733_{\pm 0.005}$	$0.777_{\pm 0.004}$	$0.858_{\pm 0.008}$	$0.889_{\pm 0.005}$
GLACIER (ours, w/o CF)	$0.390_{\pm 0.007}$	$0.573_{\pm 0.003}$	$0.680_{\pm 0.007}$	$0.741_{\pm 0.007}$	$0.785_{\pm 0.003}$	$0.861_{\pm 0.003}$	$0.892_{\pm 0.004}$
GLACIER (ours, w/ CF)	$\mathbf{0.525_{\pm 0.008}}$	$\mathbf{0.686_{\pm 0.006}}$	$\mathbf{0.774_{\pm 0.002}}$	$\mathbf{0.822_{\pm 0.002}}$	$\mathbf{0.857_{\pm 0.003}}$	$\mathbf{0.909_{\pm 0.005}}$	$\mathbf{0.925_{\pm 0.004}}$
Murcko Scaffold Split							
GrAFF-MS [16]	$0.140_{\pm 0.004}$	$0.268_{\pm 0.005}$	$0.364_{\pm 0.007}$	$0.442_{\pm 0.005}$	$0.506_{\pm 0.003}$	$0.634_{\pm 0.003}$	$0.698_{\pm 0.001}$
MassFormer [35]	$0.179_{\pm 0.007}$	$0.319_{\pm 0.004}$	$0.420_{\pm 0.003}$	$0.497_{\pm 0.002}$	$0.559_{\pm 0.003}$	$0.694_{\pm 0.005}$	$0.757_{\pm 0.006}$
ICEBERG 2.0 [31]	$0.316_{\pm 0.007}$	$0.524_{\pm 0.003}$	$0.648_{\pm 0.005}$	$0.720_{\pm 0.004}$	$0.775_{\pm 0.005}$	$0.866_{\pm 0.004}$	$0.899_{\pm 0.008}$
GLACIER (ours, w/o CF)	$0.384_{\pm 0.002}$	$0.576_{\pm 0.007}$	$0.692_{\pm 0.007}$	$0.761_{\pm 0.003}$	$0.809_{\pm 0.007}$	$\mathbf{0.885_{\pm 0.003}}$	$\mathbf{0.912_{\pm 0.003}}$
GLACIER (ours, w/ CF)	$\mathbf{0.460_{\pm 0.004}}$	$\mathbf{0.636_{\pm 0.004}}$	$\mathbf{0.731_{\pm 0.009}}$	$\mathbf{0.788_{\pm 0.010}}$	$\mathbf{0.826_{\pm 0.009}}$	$\mathbf{0.885_{\pm 0.007}}$	$0.907_{\pm 0.007}$

on both scaffold split and random split. We want to note that on MassSpecGym, the mass challenge is practically easier compared to the formula challenge, despite the name suggesting the opposite, because the number of candidates is always capped at 256 and the mass challenge introduces more easily distinguishable candidates that have different formulae. Results of retrieval accuracy of positive adduct types and $[M+H]^+$ adducts on NIST'20 are included in Appendix H to compare with baselines that support fewer adducts, where GLACIER still outperforms prior methods.

4.2.3 Analysis of Breakpoint Prediction

One open question is whether a single-stage model can predict reasonable fragments as well as two-stage models. We visualize a predicted breakpoint pattern in Figure 5 in the Appendix. The model captures several chemically plausible breakpoint patterns, although some predicted breakpoints are not chemically reasonable. Importantly, these implausible fragments are subsequently suppressed by the spectral prediction module during intensity prediction. Table 18 also demonstrates a comprehensive coverage of peaks quantitatively. These results suggest that our architecture effectively mirrors the object detection paradigm, in which an initial set of coarse predictions is generated and then refined by downstream modules to produce the final output.

Table 4: Ablation study of contrastive finetuning (CF) and its impact on Top-1 retrieval accuracy and spectral similarity. All metrics are reported in the same way with CI as in Table 1 and Table 2.

	MassSpecGym (99.9% CI)				NIST'20 (±95% CI)			
Model	Mass Top-1	Formula Top-1	Cos. Sim.	J-S Sim.	Rand. Top-1	Scaff. Top-1	Rand. Cos.	Scaff. Cos.
ICEBERG 2.0 (w/ CF)	$63.95_{(62.39–65.58)}$	$44.35_{(42.75–46.02)}$	$0.47_{(0.46–0.48)}$	$0.41_{(0.41–0.42)}$	$0.335_{\pm 0.000}$	$0.316_{\pm 0.007}$	$0.773_{\pm 0.002}$	$0.733_{\pm 0.000}$
GLACIER (w/o CF)	$69.69_{(68.16–71.21)}$	$48.49_{(46.88–50.15)}$	$\mathbf{0.58_{(0.57–0.59)}}$	$\mathbf{0.50_{(0.49–0.51)}}$	$0.390_{\pm 0.007}$	$0.384_{\pm 0.002}$	$\mathbf{0.821_{\pm 0.001}}$	$\mathbf{0.785_{\pm 0.002}}$
GLACIER (w/ CF)	$\mathbf{69.95_{(68.36–71.50)}}$	$\mathbf{49.88_{(48.25–51.49)}}$	$\mathbf{0.58_{(0.57–0.59)}}$	$0.48_{(0.48–0.49)}$	$\mathbf{0.525_{\pm 0.008}}$	$\mathbf{0.460_{\pm 0.004}}$	$0.802_{\pm 0.002}$	$0.756_{\pm 0.010}$

4.2.4 Inference Speed

We evaluate the inference speed of GLACIER against the state-of-the-art two-stage model, ICEBERG 2.0 [31]. Specifically, we measure the average time required to complete spectral prediction for the molecular retrieval task on the scaffold split of the NIST'20 dataset (requiring prediction of 2,707,082 spectra in total, including all decoy structures), averaged over three random seeds for both models. We use 2×NVIDIA A5000 GPUs and the results are summarized in Table 3. GLACIER achieves a nearly 8-fold speedup compared to ICEBERG, highlighting its efficiency. This improvement is primarily attributed to our one-step, fully GPU-based fragment inference, which eliminates the need for CPU-intensive multi-step fragment enumeration. Speed comparison to MolSpecFlow [32] is not possible because it is not open-sourced, and its paper does not report any speed statistics.

Table 3: Inference speed comparison.

Model	ms/spec	spec/s/GPU
ICEBERG 2.0 [31]	10.8	46.4
GLACIER	**1.3**	**368.7**
Speedup	7.95×	

4.2.5 Ablation Studies

MAGMa supervision weight decaying. We study the impact of MAGMa heuristic and weight decaying in the loss function Eq. (10). Table 5 reports experiment results on the NIST'20 dataset under a random split with seed=1. If we rely on the intensity objective alone (first row), the model does not converge at all. If we always utilize MAGMa supervision and enforce teacher forcing (last row), the model converges but performs worse compared to the balanced weight-decaying strategy (second row). We let the model initialize for 2,500 steps before starting to decay w_{magma}. Our explanation is that MAGMa-derived signals are particularly important during the early stages of training. As training progresses, the intensity decoder becomes capable of predicting low intensities for false positive fragments, allowing the model to rely more heavily on the full spectral prediction objective, the second term of Eq. (10).

Table 5: MAGMa supervision ablation.

λ_{decay}	w_{magma}	Test Loss
0	always $= 0$	Not Converged
0.9	decays from 1	**0.177**
1	always $= 1$	0.243

Alternative breakpoint prediction and end-to-end training objective. We test an alternative fragment prediction strategy that directly predicts the atom-level subgraph mask, $\mathbf{m}$ by adapting GraphDETR [6] instead of predicting breakpoints as described in Section 3.2. We discover that the performance is suboptimal compared with our breakpoint prediction approach, suggesting that direct fragment prediction is not a trivial task. We also implemented an end-to-end model with spectral similarity loss as the only objective and the model fails to converge. We summarize the results in Table 6. Our boundary-based prediction approach enforces subgraph connectivity and relaxes the precision needed to generate meaningful fragment candidates, ultimately leading to more stable training and improved downstream performance.

Table 6: Fragment Prediction ablation.

Prediction strategy	Test Loss
end-to-end	not converge
GraphDETR	0.477
Breakpoint	0.418

Contrastive finetuning. Contrastive finetuning of spectral prediction to improve retrieval performance was introduced by ICEBERG 2.0 [31], which samples negative structures from PubChem and by random mutation, and separates them from the positive structure. The technical details are discussed in Appendix E. As shown in Table 4, we find contrastive finetuning is still an important technique for GLACIER, where it introduces a slight negative impact on spectral accuracy but significant improvement to retrieval accuracy. The base version of GLACIER without contrastive finetuning already outperforms ICEBERG. This technique seems to benefit more for in-distribution data, as the improvement observed is the most significant on NIST'20 random split and the least significant on MassSpecGym. There are several hyperparameters for contrastive finetuning. We further show that GLACIER works better with "harsher" contrastive conditions compared to that used by ICEBERG,

as shown in Table 11, 12, 13, and 14 in the Appendix. However, the harsher the contrastive condition, the lower the spectral similarity and peak coverage are, as summarized in Table 16. This can be reasoned as the tradeoff between spectral similarity and distinguishability. As the CF condition gets harsher, the model will mitigate the impact of non-essential noise peaks from the spectra, resulting in lower spectral similarity and higher distinguishability. We acknowledge that depending on the method of sampling decoys, CF could present a risk of pathological shortcut-learning [13], though that is not empirically the case in this work (Table 15; Appendix E).

Bin width. We study the effect of bin width resolution during training and evaluation on the MassSpecGym dataset. As shown in Appendix F, we find that having higher bin-width resolution improves retrieval performance of GLACIER on both the formula challenge and the mass challenge. This demonstrates the effectiveness of the fragment-based prediction approach in high-resolution spectral prediction task due to its capability of predicting exact m/z ratios using fragment mass.

Alternative graph encoder We study the influence of different graph encoder backbones on the prediction power of GLACIER on the MassSpecGym dataset. We tested 3 graph encoder variants: Graphormer, NeuralWalker [5] (which is used in GraphDETR), and vanilla GNN. As shown in Table 8, choosing Graphormer as the graph encoder backbone results in the best test loss, supporting our choice of adopting it as the default graph encoder backbone used in GLACIER. **Alternative fragment encoding initialization.** In order to make sure that GLACIER does not heavily rely on predefined heuristic and learn useful representation, we added ablations using two alternatives: (1) Transformer-decoder pooling without descriptors and (2) Transformer-decoder pooling without graph tokens, with results summarized in Table 8. The results show that removing external descriptors does not negatively affect performance. This demonstrates GLACIER 's ability to learn effective atom-level and graph-level representation without relying on external descriptors.

Table 7: Graph encoder ablation.

Graph Encoder	Test Loss
GNN	0.433
NeuralWalker	0.423
Graphormer	0.418

Table 8: Fragment token ablation.

Initialization method	Test Loss
graph token	0.411
graph token+descriptor	0.418
descriptor	0.421

5 Discussion and Outlook

In this paper, we establish the conceptual connection between MS/MS prediction in molecular machine learning and object detection in computer vision. We design GLACIER, a transformer-based single-stage MS/MS prediction model that supports multiple bond-breaking events and accurate intensity prediction. GLACIER establishes new state-of-the-art performance with better retrieval accuracy in downstream applications and a significant boost in inference throughput. We envision that the development of GLACIER will open up a new space of model design for single-stage MS/MS prediction networks and raise attention from the broader machine learning community, whereby faster and more accurate models will shed light on the long-lasting challenge of structural elucidation, enabling critical science areas including clinical metabolomics, precise medicine, and drug discovery.

Limitations and future work. As the first model for one-stage multi-breakpoint MS/MS predictions, GLACIER poses several limitations to be addressed in future work. First, though being one-stage, GLACIER is not a fully end-to-end model. The fragment generation step truncates the gradient, and we use MAGMa heuristic to supervise fragment generation. An end-to-end trainable one-stage model in future work might further improve the performance. Second, GLACIER does not handle complex rearrangements that involve changes in the heavy atom connectivity. Future work could focus on bond forming and breaking as the fragment generation step.

Broader impact. Tools developed in this paper for structural elucidation have broad potential for positive societal impact across chemical and biomedical sciences, with minimal foreseeable risk.

Acknowledgments and Disclosure of Funding

This work was supported by DSO National Laboratories in Singapore, the MIT Generative AI Impact Consortium (MGAIC), and the National Institute of General Medical Sciences of the National Institutes of Health (K99GM166738). The authors thank Shitong Luo and Mrunali Manjrekar for discussions and Hongxuan Liu for suggestions on parallel inference. NIST mass spectral libraries are copyright protected and do not allow for distribution of derivative products.

References

[1] Felicity Allen, Russ Greiner, and David Wishart. Competitive fragmentation modeling of esi-ms/ms spectra for putative metabolite identification. *Metabolomics*, 11:98–110, 2015.

[2] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[3] Roman Bushuiev, Anton Bushuiev, Niek F de Jonge, Adamo Young, Fleming Kretschmer, Raman Samusevich, Janne Heirman, Fei Wang, Luke Zhang, Kai Dührkop, et al. Massspecgym: A benchmark for the discovery and identification of molecules. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. URL `https://arxiv.org/abs/2005.12872`.

[5] Dexiong Chen, Till Hendrik Schulz, and Karsten Borgwardt. Learning long range dependencies on graphs via random walks, 2024. URL `https://arxiv.org/abs/2406.03386`.

[6] Dexiong Chen, Till Hendrik Schulz, and Karsten Borgwardt. End-to-end subgraph detection with graphdetr, 2026. URL `https://arxiv.org/abs/2606.06364`.

[7] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] Samuel Goldman, John Bradshaw, Jiayi Xin, and Connor Coley. Prefix-tree decoding for predicting mass spectra from molecules. *Advances in Neural Information Processing Systems*, 36:48548–48572, 2023.

[10] Samuel Goldman, Janet Li, and Connor W Coley. Generating molecular fragmentation graphs with autoregressive neural networks. *Analytical Chemistry*, 96(8):3419–3428, 2024.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

[13] Hongxuan Liu, Roman Bushuiev, Ivy Lightheart, Mrunali Manjrekar, Anton Bushuiev, Magdalena Lederbauer, Filip Jozefov, Yinkai Wang, Soha Hassoun, Josef Sivic, James Taylor, Runzhong Wang, David Healey, Tomáš Pluskal, and Connor W. Coley. Massspecgym in the wild: Uncovering and correcting evaluation pitfalls in ai-driven molecule discovery, 2026. URL `https://arxiv.org/abs/2606.19624`.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[15] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection, 2021. URL `https://arxiv.org/abs/2109.08141`.

[16] Michael Murphy, Stefanie Jegelka, Ernest Fraenkel, Tobias Kind, David Healey, and Thomas Butler. Efficiently predicting high resolution mass spectra with graph neural networks. In *International Conference on Machine Learning*, pages 25549–25562. PMLR, 2023.

[17] NIST. NIST standard reference database. National Institute of Standards and Technology, 2020. URL `https://www.nist.gov/srd`.

[18] Yannek Nowatzky, Francesco Friedrich Russo, Jan Lisec, Alexander Kister, Knut Reinert, Thilo Muth, and Philipp Benner. Fiora: Local neighborhood-based prediction of compound mass spectra from single fragmentation events. *Nature Communications*, 16(1):2298, 2025.

[19] Hantao Qiang, Fei Wang, Wenyun Lu, Xi Xing, Hahn Kim, Sandrine AM Mérette, Lucas B Ayres, Eponine Oler, Jenna E AbuSalim, Asael Roichman, et al. Language model-guided anticipation and discovery of mammalian metabolites. *Nature*, pages 1–10, 2026.

[20] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. URL `http://arxiv.org/abs/1506.02640`.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL `https://arxiv.org/abs/1506.01497`.

[22] Lars Ridder, Justin J. J. van der Hooft, and Stefan Verhoeven. Automatic compound annotation from mass spectrometry data using magma. *Mass Spectrometry*, 3(Special_Issue_2):S0033–S0033, 2014. doi: 10.5702/massspectrometry.S0033.

[23] Christoph Ruttkies, Emma L Schymanski, Sebastian Wolf, Juliane Hollender, and Steffen Neumann. Metfrag relaunched: incorporating strategies beyond in silico fragmentation. *Journal of cheminformatics*, 8:1–16, 2016.

[24] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014. URL `https://arxiv.org/abs/1312.6120`.

[25] Austin Tripp and José Miguel Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation. *arXiv preprint arXiv:2310.09267*, 2023.

[26] Lara van Tetering, Sylvia Spies, Quirine DK Wildeman, Kas J Houthuijs, Rianne E van Outersterp, Jonathan Martens, Ron A Wevers, David S Wishart, Giel Berden, and Jos Oomens. A spectroscopic test suggests that fragment ion structure annotations in ms/ms libraries are frequently incorrect. *Communications Chemistry*, 7(1):30, 2024.

[27] Fei Wang, Jaanus Liigand, Siyang Tian, David Arndt, Russell Greiner, and David S Wishart. Cfm-id 4.0: more accurate esi-ms/ms spectral prediction and compound identification. *Analytical chemistry*, 93(34):11692–11700, 2021.

[28] Rui-Xi Wang, Runzhong Wang, Mrunali Manjrekar, and Connor W. Coley. Neural graph matching improves retrieval augmented generation in molecular machine learning, 2025. URL `https://arxiv.org/abs/2502.17874`.

[29] Runzhong Wang, Tianqi Zhang, Tianshu Yu, Junchi Yan, and Xiaokang Yang. Combinatorial learning of graph edit distance via dynamic embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5241–5250, 2021.

[30] Runzhong Wang, Yunhao Zhang, Ziao Guo, Tianyi Chen, Xiaokang Yang, and Junchi Yan. LinSATNet: The positive linear satisfiability neural networks. In *International Conference on Machine Learning (ICML)*, 2023.

[31] Runzhong Wang, Mrunali Manjrekar, Babak Mahjour, Julian Avila-Pacheco, Joules Provenzano, Erin Reynolds, Magdalena Lederbauer, Eivgeni Mashin, Samuel Goldman, Mingxun Wang, Jing-Ke Weng, Desirée L. Plata, Clary B. Clish, and Connor W. Coley. Neural spectral prediction for structure elucidation with tandem mass spectrometry. *bioRxiv*, 2025. doi: 10.1101/2025.05.28.656653. URL `https://www.biorxiv.org/content/early/2025/06/01/2025.05.28.656653`.

[32] Yu Wang, Fan Yang, Kaikun Xu, Li Yuan, Jun Zhu, Jingjie Zhang, Zhenchao Tang, Yatao Bian, Cheng Chang, Yonghong Tian, and Jianhua Yao. Molspecflow: Mass-constrained hybrid flow matching for joint molecular-spectral analysis. *bioRxiv*, 2026. doi: 10.64898/2026.01.28.702438. URL `https://www.biorxiv.org/content/early/2026/02/01/2026.01.28.702438`.

[33] Jennifer N Wei, David Belanger, Ryan P Adams, and D Sculley. Rapid prediction of electron–ionization mass spectrometry using neural networks. *ACS central science*, 5(4):700–708, 2019.

[34] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform bad for graph representation?, 2021. URL `https://arxiv.org/abs/2106.05234`.

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

[36] Adamo Young, Fei Wang, David Wishart, Bo Wang, Hannes Röst, and Russ Greiner. Fragnnet: A deep probabilistic model for mass spectrum prediction. *arXiv preprint arXiv:2404.02360*, 2024.

[37] Hao Zhu, Liping Liu, and Soha Hassoun. Using graph neural networks for mass spectrometry prediction. *arXiv preprint arXiv:2010.04661*, 2020.

Technical Appendices and Supplementary Materials

A Graphormer Node and Edge Features

To construct more expressive node and edge representations for the molecular graph, we incorporate chemically informative features derived from RDKit and use heavy atom scaffold as the graph. Because our model shares embeddings across multiple subgraph fragments from the same molecule, we can compute these features once at the molecule level and reuse them across fragments. In contrast, prior work [31] generates separate embeddings for each fragment, which prevents efficient integration of such global, RDKit-derived features. We summarize the resulting atom and edge features in Table 9 and Table 10.

Table 9: Node feature construction for each atom. Here $C = \texttt{MAX_ABS_FORMAL_CHARGE}$.

Feature Name	Description	Dimension
Atom Type (One-hot)	One-hot over VALID_ATOM_NUM	|VALID_ATOM_NUM|
Atom Type (Other)	Indicator for atoms not in valid set	1
Degree (One-hot)	One-hot over $0 \ldots$ MAX_COMMON_DEGREE (excl. H)	MAX_COMMON_DEGREE $+ 1$
Degree (Overflow)	Indicator for degree $>$ max	1
Formal Charge (One-hot)	One-hot over $[-C, \ldots, C]$	$2C + 1$
Formal Charge (Overflow)	Indicator for $\|q\| > C$	1
Hybridization (One-hot)	One-hot over COMMON_HYBRIDIZATION	|COMMON_HYBRIDIZATION|
Hybridization (Other)	Indicator for unseen hybridization	1
Aromaticity	Atom is aromatic	1
Hydrogen Count (One-hot)	One-hot over $0 \ldots$ COMMON_MAX_HYDROGEN_COUNTS	COMMON_MAX_HYDROGEN_COUNTS $+ 1$
Hydrogen Count (Overflow)	Indicator for count $>$ max	1
Ring Membership	[not in ring, in ring]	2
Ring Size (Multi-hot)	For each size in COMMON_RING_SIZES: [in, not in]	$2 \times$ |COMMON_RING_SIZES|
Fused Ring	[not fused, fused]	2
Chirality (One-hot)	One-hot over COMMON_CHIRALITY	|COMMON_CHIRALITY|
Element Group Embedding	Element group representation (optional)	ELEMENT_GROUP_DIM
Gasteiger Charge	Continuous partial charge	1

Table 10: Edge (bond) feature construction. Fused rings are defined as bonds belonging to ≥ 2 rings.

Feature Name	Description	Dimension
Bond Type (One-hot)	One-hot over COMMON_BOND_TYPES	|COMMON_BOND_TYPES|
Bond Type (Other)	Indicator for unseen bond types	1
Conjugation	[not conjugated, conjugated]	2
Ring Membership	[not in ring, in ring]	2
Ring Size (Multi-hot)	For each size in COMMON_RING_SIZES: [in, not in]	$2 \times$ |COMMON_RING_SIZES|
Fused Ring	[not fused, fused]	2
Stereo (One-hot)	One-hot over VALID_BOND_STEREO	|VALID_BOND_STEREO|
Stereo (Other)	Indicator for unseen stereo types	1

B Fragment Pattern Construction in Inference

Once the breakpoints are predicted, the next step is to generate the fragments. Let $\mathbf{A} \in \{0,1\}^{n_{\text{atom}} \times n_{\text{atom}}}$ denote the adjacency matrix of a graph and let $\mathbf{m}^{(l)} \in \{0,1\}^{n_{\text{atom}}}$ denote the l-th breakpoint mask for a given molecular graph.

Reachability. For each mask $\mathbf{m}^{(l)}$, we construct a masked adjacency matrix:

$$\mathbf{A}^{(l)} = \mathbf{A} \odot \left(\mathbf{m}^{(l)}(\mathbf{m}^{(l)})^\top\right), \tag{11}$$

which restricts edges to those entirely within the subgraph.

We compute the reachability matrix $\mathbf{R}^{(l)} \in \{0,1\}^{n_{\text{atom}} \times n_{\text{atom}}}$ as the transitive closure of $\mathbf{A}^{(l)}$:

$$\mathbf{R}^{(l)}_{ij} = \mathbb{I}\big(\text{there exists a path from } i \text{ to } j \text{ within } \mathbf{A}^{(l)}\big). \tag{12}$$

Component extraction. Each row $\mathbf{R}_i^{(l)}$ encodes the set of nodes reachable from node i, corresponding to a connected component within the masked subgraph. We treat each row as a candidate fragment pattern: $\mathbf{p}^{(l,i)} = \mathbf{R}_i^{(l)} \in \{0,1\}^{n_{\text{atom}}}$. Since multiple nodes within the same connected component yield identical rows, we remove duplicate patterns to get the fragment set $\mathcal{F}$:

$$\mathcal{F} = \text{Unique}\big(\{\mathbf{p}^{(l,i)}\}_{l,i}\big). \tag{13}$$

We discard trivial patterns with no active nodes, and $\mathbf{p}$ is kept only if $\sum_j p_j > 0$. In the degenerate case where $|\mathcal{F}| = 0$, we insert a dummy pattern corresponding to valid nodes.

C Fragment Embedding for Intensity Prediction

We construct fragment-level embeddings by incorporating chemically meaningful structural signals. We first generate fragment formula vectors, encoding chemical formula as a multi-hot vector for each fragment. In parallel, we compute fragment-specific statistics, including the number of hydrogens within each fragment, the maximum allowable hydrogen addition and removal (i.e., hydrogen shift, bounded by the number of broken bonds), and the number of broken bonds, which we define as edges crossing the fragment boundary. We derive these quantities using adjacency matrices and fragment membership masks. We then compare each fragment form vector with the corresponding root (molecule-level) form vector to obtain difference features. We pass both the fragment representations and their differences through a shared embedding network. Additionally, we discretize the number of broken bonds using a clamped one-hot encoding. Finally, we concatenate the expanded root token embeddings, encoded fragment features, difference encodings, and broken-bond features, and map the result through a learned projection layer to produce the final token representations. In particular, we can define the fragment embedding token $\mathbf{t}_k$ as follows of fragment k:

$$\mathbf{t}_k = \text{MLP}\Big(\big[\mathbf{G} \parallel \psi(\mathbf{f}_k) \parallel \psi(\mathbf{r}^{\text{form}} - \mathbf{f}_k) \parallel \text{OHE}\big(n_k^{\text{broken}}\big)\big]\Big), \tag{14}$$

where $\mathbf{G}$ is the embedding of the root molecule, ψ is the absolute-sine embedder for formula embedding, $\mathbf{r}^{\text{form}}$ is the formula vector of the root molecule, $\mathbf{f}_k$ is the formula vector of fragment k, and n_k^{broken} is the number of broken bonds required when generating fragment k.

D Intensity Prediction Head

ICEBERG predicts unnormalized intensity weights at each possible hydrogen shift alongside an attention weight to determine how heavily to weight each prediction for its specified hydrogen shift for all fragment embeddings. The attention weight is calculated by taking a softmax over all prediction indices that fall into the same intensity bin. The final binned intensity is calculated as the weighted sum over all predictions that fall within this mass bin followed by a sigmoid activation function. The final intensity is calculated as the weighted sum over all predictions that fall within this mass bin followed by a sigmoid activation function. The unbinned intensity of is reassigned back to each fragment with the predicted attention score that and the total intensity falls into a given bin.

This setting is problematic because it mixes up the assignment score before and after activation and will lead to inaccurate results if we want to adapt the predicted fragment intensity into different bin-width setting. To address this, we remove the attention weight component and predict the fragment intensity as follows:

$$y_\delta^{(i)} = \sigma\left(\text{MLP}_{\text{inten}}(\hat{\mathbf{H}}_i)_\delta\right), \tag{15}$$

where σ is the Sigmoid activation function, δ is the hydrogen shift count, and $y_\delta^{(i)}$ is the predicted unbinned intensity of the fragment with hydrogen shift δ. The binned intensity $\hat{y}_m$ at m/z value m is calculated as the sum of all unbinned intensity predictions that fall within a given mass bin:

$$\hat{y}_m = \sum_i \sum_\delta y_\delta^{(i)} \, \mathbb{I}[M(i,\delta) = m], \tag{16}$$

where $M(i,\delta)$ is the m/z value of fragment i with hydrogen shift δ. This formulation enables the model to adapt to different mass bin widths at inference time, even when trained using a fixed binning resolution.

E Contrastive Finetuning Details

We adapt the formulation of the contrastive finetuning objective from Wang et al. [31].Of our sampled negative structures, 50% share the same chemical formula (from PubChem) and the remaining 50% are generated by single-step graph mutations [25]. During training, GLACIER predict spectra for one positive and k negative structures simultaneously, yielding the following pairwise distance matrix:

$$\mathbf{D} \in \mathbb{R}^{(k+1)\times(k+1)}, \quad \mathbf{D}_{ij} = d_i - d_j, \quad i, j \in \{1, \ldots, k+1\}, \tag{17}$$

where d_i is the entropy distance between the predicted spectrum for structure i and the ground-truth spectrum, and d_1 is the positive sample. We compute $\bar{\mathbf{D}} = \mathrm{Sinkhorn}(-\mathbf{D}, \tau)$ and $\bar{d_1} \in [0, 1]$ is the top-left entry of $\bar{\mathbf{D}}$. The temperature hyperparameter τ is set to 0.5. We define our contrastive loss term $\mathcal{L}_{\mathrm{contr}}$ as follows:

$$\mathcal{L}_{\mathrm{contr}} = \mathrm{ReLU}(-\log(\bar{d_1} + \delta)), \tag{18}$$

where the offset hyperparameter δ prevents overfitting by discouraging further maximization when $\bar{d_1}$ is already large. The overall contrastive finetuning objective is defined as follows:

$$\mathcal{L}^{\mathrm{train}}_{\mathrm{contr}} = w_{\mathrm{magma}} \cdot w_{\mathrm{frag}} \cdot \mathcal{L}_{\mathrm{fragment}} + w_{\mathrm{inten}} \cdot \mathcal{L}_{\mathrm{inten}} + w_{\mathrm{contr}} \cdot \mathcal{L}_{\mathrm{contr}}, \tag{19}$$

where $w_{\mathrm{magma}}, w_{\mathrm{frag}}, \mathcal{L}_{\mathrm{fragment}}, w_{\mathrm{inten}}$, and $\mathcal{L}_{\mathrm{inten}}$ are described in Section 3.4. Note that we remove the magma inten loss $\hat{\mathcal{L}}_{\mathrm{inten}}$ since the intensity predictor is pretrained. Similar to the pretraining stage, we also remove the fragmentation prediction objective $\mathcal{L}_{\mathrm{fragment}}$ and the overall contrastive finetuning objective becomes

$$\mathcal{L}^{\mathrm{val}}_{\mathrm{contr}} = w_{\mathrm{inten}} \cdot \mathcal{L}_{\mathrm{inten}} + w_{\mathrm{contr}} \cdot \mathcal{L}_{\mathrm{contr}}. \tag{20}$$

We study the impact of the performance of contrastive finetuning hyperparameters on NIST'20 random split and scaffold split and on the MassSpecGym dataset. Results are summarized in Table 11, Table 12, Table 13, and Table 14. We find out that using a "harsher" contrastive finetuning condition (lower δ and higher d) results in significant performance boost on random split while its impact on scaffold split is relatively limited. Our explanation is that contrastive finetuning is more effective for learning on in-distribution data where the model can learn to differentiate between similar molecular structures, while having distinct structures across training and validation sets makes it difficult for the model to separate structures it has not seen before.

Table 11: Top-k accuracy comparison between three contrastive finetuning (CF) conditions on NIST'20 random split.

	Top-1	Top-2	Top-3	Top-4	Top-5	Top-8	Top-10
GLACIER ($\delta = 0.4, d = 5$)	**0.525**$_{\pm 0.008}$	**0.686**$_{\pm 0.006}$	**0.774**$_{\pm 0.002}$	**0.822**$_{\pm 0.002}$	**0.857**$_{\pm 0.003}$	**0.909**$_{\pm 0.005}$	0.925$_{\pm 0.004}$
GLACIER ($\delta = 0.5, d = 3$)	0.498$_{\pm 0.008}$	0.667$_{\pm 0.010}$	0.754$_{\pm 0.009}$	0.808$_{\pm 0.006}$	0.845$_{\pm 0.005}$	0.907$_{\pm 0.008}$	**0.929**$_{\pm 0.007}$
GLACIER ($\delta = 0.6, d = 3$)	0.486$_{\pm 0.007}$	0.657$_{\pm 0.004}$	0.748$_{\pm 0.004}$	0.802$_{\pm 0.003}$	0.840$_{\pm 0.004}$	0.906$_{\pm 0.003}$	**0.929**$_{\pm 0.005}$

Table 12: Top-k accuracy comparison between three contrastive finetuning (CF) conditions on NIST'20 scaffold split.

	Top-1	Top-2	Top-3	Top-4	Top-5	Top-8	Top-10
GLACIER ($\delta = 0.4, d = 5$)	**0.460**$_{\pm 0.004}$	**0.636**$_{\pm 0.004}$	0.731$_{\pm 0.009}$	0.788$_{\pm 0.010}$	0.826$_{\pm 0.009}$	0.885$_{\pm 0.007}$	0.907$_{\pm 0.007}$
GLACIER ($\delta = 0.5, d = 3$)	0.448$_{\pm 0.004}$	0.633$_{\pm 0.010}$	**0.736**$_{\pm 0.008}$	**0.798**$_{\pm 0.010}$	**0.836**$_{\pm 0.006}$	0.899$_{\pm 0.004}$	0.923$_{\pm 0.001}$
GLACIER ($\delta = 0.6, d = 3$)	0.441$_{\pm 0.006}$	0.625$_{\pm 0.011}$	0.729$_{\pm 0.012}$	0.793$_{\pm 0.006}$	0.834$_{\pm 0.007}$	**0.901**$_{\pm 0.003}$	**0.926**$_{\pm 0.004}$

Further, we do not currently believe that contrastive finetuning is a source of "data leakage"— the information provided by contrastive finetuning is that PubChem structure A should not have a spectrum that looks similar to structure B, which is in the training set. It might or might not be true in reality. As a standard approach, it does not seem to leak any information because all synthetic structures are used as negative samples in contrastive finetuning. To verify this idea, we conduct another ablation study testing whether filtering out all retrieval candidate structures in retrieval during contrastive finetuning training affects the overall performance, with results summarized in Table 15. GLACIER performs similarly with or without candidate structure filtering, suggesting that the resulting performance boost is not due to data leakage but the enhancement of model's ability to differentiate structures. Nevertheless, we emphasize that even in the absence of any contrastive fine-tuning, GLACIER achieves state of the art retrieval accuracy for both MassSpecGym and NIST'20.

Table 13: Top-k accuracy comparison between three contrastive finetuning (CF) conditions on the MassSpecGym (mass challenge).

	Top-1	Top-2	Top-3	Top-4	Top-5	Top-8	Top-10
GLACIER ($\delta = 0.4, d = 5$)	0.681	0.784	0.823	0.847	0.862	0.890	0.901
GLACIER ($\delta = 0.5, d = 3$)	0.698	**0.792**	0.828	0.850	0.864	**0.895**	**0.905**
GLACIER ($\delta = 0.6, d = 3$)	**0.699**	**0.792**	**0.832**	**0.853**	**0.865**	**0.895**	**0.905**

Table 14: Top-k accuracy comparison between three contrastive finetuning (CF) conditions on the MassSpecGym (formula challenge).

	Top-1	Top-2	Top-3	Top-4	Top-5	Top-8	Top-10
GLACIER ($\delta = 0.4, d = 5$)	0.490	0.624	0.692	0.738	0.770	0.827	0.849
GLACIER ($\delta = 0.5, d = 3$)	0.490	0.627	**0.700**	0.745	0.775	0.835	**0.855**
GLACIER ($\delta = 0.6, d = 3$)	**0.499**	**0.633**	0.699	**0.749**	**0.781**	**0.836**	0.853

Table 15: Top-k accuracy comparison between models on random split with the same CF condition ($\delta = 0.5, d = 3$) and random seed (1) on the NIST'20 dataset. GLACIER performs similarly across the two conditions, suggesting that the performance boost of contrastive finetuning is not the results of data leakage arising from inclusion of exactly matching decoys.

	Top-1	Top-2	Top-3	Top-4	Top-5	Top-8	Top-10
GLACIER (w/o decoy filtering)	**0.510**	**0.677**	**0.761**	0.814	**0.849**	0.905	0.928
GLACIER (with decoy filtering)	0.505	0.674	0.759	**0.815**	0.848	**0.912**	**0.934**

Table 16: Ablation studies on the effect of different CF condition on spectral similarities and coverage on the NIST'20 dataset.

Method	**Positive+negative mode, Random split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
GLACIER ($\delta = 0.4, d = 5$)	0.802 ± 0.002	0.736 ± 0.004	0.843 ± 0.003
GLACIER ($\delta = 0.5, d = 3$)	0.820 ± 0.002	0.760 ± 0.002	0.862 ± 0.002
GLACIER ($\delta = 0.6, d = 3$)	$\mathbf{0.824 \pm 0.000}$	$\mathbf{0.767 \pm 0.001}$	$\mathbf{0.868 \pm 0.000}$
Method	**Positive+negative mode, Scaffold split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
GLACIER ($\delta = 0.4, d = 5$)	0.756 ± 0.010	0.687 ± 0.009	0.826 ± 0.008
GLACIER ($\delta = 0.5, d = 3$)	0.778 ± 0.002	0.717 ± 0.002	0.852 ± 0.003
GLACIER ($\delta = 0.6, d = 3$)	$\mathbf{0.787 \pm 0.000}$	$\mathbf{0.730 \pm 0.001}$	$\mathbf{0.862 \pm 0.001}$

F Ablation Studies on Bin Width Resolution

We conduct ablation studies on the effect of retrieval accuracy on models trained on bin width of 0.1 Dalton or 0.01 Dalton and evaluate using 0.1 Dalton and 0.01 Dalton on the MassSpecGym formula and mass challenges, and summarize the result in Table 17. GLACIER performs better when tested on bin width set to 0.01 Dalton, demonstrating the power of fragment-based approach on high-resolution spectral prediction.

Table 17: Top 1 accuracy of GLACIER trained/tested on the MassSpecGym dataset with bin width set to 0.1 Dalton or 0.01 Dalton. Rows correspond to the training bin width and columns correspond to the testing bin width. GLACIER performs better using 0.01 Dalton as the testing bin widths.

Train Bin Width (Dalton)	Mass Challenge		Formula Challenge	
	Test Bin Width (Dalton)		Test Bin Width (Dalton)	
	0.1	0.01	0.1	0.01
0.1	0.664	0.685	0.479	**0.490**
0.01	0.652	**0.697**	0.471	0.485

G Additional Experimental Results on NIST'20 Spectra Prediction

We provide cosine similarity, entropy similarity, and coverage of predicted spectra on NIST'20 clustered by all adduct types, positive adduct types, and $[M+H]^+$ adduct type in Table 18, Table 19, and Table 20. GLACIER outperforms all baseline methods, demonstrating its power on predicting accurate spectra across diverse datasets and instrumental parameters.

Table 18: Detailed evaluation results on spectral prediction accuracy for $[M+H]^+$ adduct on NIST'20 dataset. $\pm$: 95% CI. Baseline results are adapted from Wang et al. [31].

Method	**$[M+H]^+$, Random split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
CFM-ID [27]	0.470 ± 0.000	0.427 ± 0.000	0.261 ± 0.000
GrAFF-MS [16]	0.565 ± 0.001	0.529 ± 0.001	0.740 ± 0.001
MassFormer [35]	0.632 ± 0.001	0.605 ± 0.000	0.787 ± 0.001
FraGNNet [36]	0.717 ± 0.001	-	-
ICEBERG 1.0 [10]	0.754 ± 0.002	0.700 ± 0.001	0.809 ± 0.001
ICEBERG 2.0 [31]	0.794 ± 0.002	0.763 ± 0.001	0.856 ± 0.000
GLACIER, w/o CF	$\mathbf{0.838 \pm 0.001}$	$\mathbf{0.781 \pm 0.001}$	$\mathbf{0.887 \pm 0.001}$
GLACIER, w/ CF	0.815 ± 0.003	0.752 ± 0.003	0.858 ± 0.004
Method	**$[M+H]^+$, Scaffold split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
CFM-ID [27]	0.450 ± 0.000	0.401 ± 0.000	0.228 ± 0.000
GrAFF-MS [16]	0.470 ± 0.001	0.448 ± 0.001	0.706 ± 0.001
MassFormer [35]	0.526 ± 0.002	0.512 ± 0.002	0.742 ± 0.002
FraGNNet [36]	0.654 ± 0.003	-	-
ICEBERG [10]	0.701 ± 0.001	0.641 ± 0.004	0.790 ± 0.001
ICEBERG 2.0 [31]	0.735 ± 0.000	0.708 ± 0.000	0.847 ± 0.000
GLACIER, w/o CF	$\mathbf{0.788 \pm 0.002}$	$\mathbf{0.733 \pm 0.001}$	$\mathbf{0.876 \pm 0.001}$
GLACIER, w/ CF	0.763 ± 0.020	0.697 ± 0.025	0.840 ± 0.022

Table 19: Detailed evaluation results on spectral prediction accuracy for positive mode adducts on NIST'20 dataset. ±: 95% CI from three random seeds. Baseline results are adapted from Wang et al. [31].

Method	**Positive mode, Random split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
GrAFF-MS [16]	0.578 ± 0.001	0.533 ± 0.001	0.724 ± 0.000
MassFormer [35]	0.640 ± 0.001	0.605 ± 0.000	0.776 ± 0.001
ICEBERG 1.0 [10]	0.727 ± 0.002	0.664 ± 0.000	0.754 ± 0.002
ICEBERG 2.0 [31]	0.782 ± 0.002	0.749 ± 0.001	0.837 ± 0.000
GLACIER, w/o CF	**0.829 ± 0.001**	**0.769 ± 0.001**	**0.873 ± 0.001**
GLACIER, w/ CF	0.809 ± 0.002	0.743 ± 0.003	0.847 ± 0.004
Method	**Positive mode, Scaffold split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
GrAFF-MS [16]	0.477 ± 0.001	0.452 ± 0.001	0.701 ± 0.001
MassFormer [35]	0.532 ± 0.002	0.515 ± 0.002	0.740 ± 0.001
ICEBERG 1.0 [10]	0.698 ± 0.001	0.633 ± 0.004	0.770 ± 0.001
ICEBERG 2.0 [31]	0.734 ± 0.000	0.707 ± 0.000	0.842 ± 0.000
GLACIER, w/o CF	**0.787 ± 0.002**	**0.731 ± 0.001**	**0.872 ± 0.001**
GLACIER, w/ CF	0.758 ± 0.010	0.689 ± 0.009	0.832 ± 0.008

Table 20: Detailed evaluation results on spectral prediction accuracy for both positive and negative mode adducts on NIST'20 dataset. ±: 95% CI from three random seeds. Baseline results are adapted from Wang et al. [31].

Method	**Positive+negative mode, Random split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
GrAFF-MS [16]	0.556 ± 0.001	0.512 ± 0.001	0.710 ± 0.000
MassFormer [35]	0.656 ± 0.001	0.610 ± 0.000	0.782 ± 0.001
ICEBERG 2.0 [31]	0.773 ± 0.002	0.740 ± 0.002	0.834 ± 0.001
GLACIER, w/o CF	**0.821 ± 0.001**	**0.761 ± 0.001**	**0.869 ± 0.000**
GLACIER, w/ CF	0.802 ± 0.002	0.736 ± 0.004	0.843 ± 0.003
Method	**Positive+negative mode, Scaffold split**		
	Cosine sim. (↑)	Entropy sim. (↑)	Coverage (↑)
GrAFF-MS [16]	0.474 ± 0.001	0.448 ± 0.001	0.692 ± 0.001
MassFormer [35]	0.536 ± 0.001	0.516 ± 0.002	0.738 ± 0.002
ICEBERG 2.0 [31]	0.733 ± 0.000	0.706 ± 0.000	0.837 ± 0.000
GLACIER, w/o CF	**0.785 ± 0.002**	**0.729 ± 0.001**	**0.867 ± 0.001**
GLACIER, w/ CF	0.756 ± 0.010	0.687 ± 0.009	0.826 ± 0.008

H Additional Experimental Results on NIST'20 Retrieval

We provide the retrieval results on the NIST'20 with candidates of the $[M+H]^+$ adduct type in Table 21 and of positive adduct types in Table 22. GLACIER outperforms all baseline methods, improving top-1 retrieval accuracy on the $[M+H]^+$ adduct type from 40.0% to 53.3% and 34.0% to 52.0% on all positive adduct types. Results of all baseline methods are adapted from Wang et al. [31].

Table 21: Detailed retrieval accuracy for $[M+H]^+$ adduct on NIST'20 dataset. $\pm$: 95% CI. Baseline results are adapted from Wang et al. [31].

	$[M+H]^+$, Random split				
Top-k accuracy	Top-1	Top-2	Top-3	Top-4	Top-5
MetFrag [23]	0.107 ± 0.000	0.193 ± 0.000	0.267 ± 0.000	0.317 ± 0.000	0.375 ± 0.000
GrAFF-MS [16]	0.211 ± 0.004	0.365 ± 0.009	0.472 ± 0.015	0.551 ± 0.013	0.608 ± 0.005
MassFormer [35]	0.252 ± 0.001	0.422 ± 0.002	0.539 ± 0.005	0.617 ± 0.007	0.675 ± 0.004
FraGNNet [36]	0.238	-	0.504	-	0.652
ICEBERG 1.0 [10]	0.251 ± 0.016	0.454 ± 0.004	0.576 ± 0.006	0.654 ± 0.004	0.711 ± 0.007
ICEBERG 2.0 [31]	0.400 ± 0.008	0.611 ± 0.013	0.721 ± 0.010	0.790 ± 0.007	0.829 ± 0.006
GLACIER (ours, w/ CF)	**0.533 ± 0.009**	**0.713 ± 0.001**	**0.802 ± 0.001**	**0.846 ± 0.002**	**0.874 ± 0.005**
Top-k accuracy	Top-6	Top-7	Top-8	Top-9	Top-10
MetFrag [23]	0.414 ± 0.000	0.457 ± 0.000	0.497 ± 0.000	0.537 ± 0.000	0.567 ± 0.000
GrAFF-MS [16]	0.655 ± 0.006	0.690 ± 0.004	0.724 ± 0.008	0.754 ± 0.010	0.776 ± 0.009
MassFormer [35]	0.724 ± 0.005	0.761 ± 0.008	0.794 ± 0.010	0.822 ± 0.005	0.843 ± 0.006
FraGNNet [36]	-	-	-	-	0.831
ICEBERG 1.0 [10]	0.753 ± 0.007	0.783 ± 0.005	0.810 ± 0.001	0.833 ± 0.007	0.850 ± 0.009
ICEBERG 2.0 [31]	0.859 ± 0.005	0.881 ± 0.006	0.900 ± 0.009	0.913 ± 0.008	0.923 ± 0.007
GLACIER (ours w/ CF)	**0.892 ± 0.006**	**0.907 ± 0.005**	**0.917 ± 0.004**	**0.924 ± 0.005**	**0.930 ± 0.004**
	$[M+H]^+$, Scaffold split				
Top-k accuracy	Top-1	Top-2	Top-3	Top-4	Top-5
MetFrag [23]	0.127 ± 0.000	0.225 ± 0.000	0.302 ± 0.000	0.363 ± 0.000	0.420 ± 0.000
GrAFF-MS [16]	0.143 ± 0.004	0.270 ± 0.002	0.368 ± 0.008	0.450 ± 0.006	0.518 ± 0.002
MassFormer [35]	0.184 ± 0.008	0.328 ± 0.005	0.431 ± 0.001	0.512 ± 0.005	0.575 ± 0.004
ICEBERG 1.0 [10]	0.229 ± 0.004	0.415 ± 0.006	0.535 ± 0.008	0.617 ± 0.009	0.675 ± 0.004
ICEBERG 2.0 [31]	0.335 ± 0.005	0.543 ± 0.004	0.669 ± 0.006	0.745 ± 0.002	0.799 ± 0.004
GLACIER (ours, w/ CF)	**0.452 ± 0.005**	**0.637 ± 0.005**	**0.735 ± 0.010**	**0.794 ± 0.011**	**0.833 ± 0.010**
Top-k accuracy	Top-6	Top-7	Top-8	Top-9	Top-10
MetFrag [23]	0.471 ± 0.000	0.510 ± 0.000	0.548 ± 0.000	0.582 ± 0.000	0.612 ± 0.000
GrAFF-MS [16]	0.569 ± 0.002	0.614 ± 0.003	0.650 ± 0.000	0.687 ± 0.002	0.716 ± 0.002
MassFormer [35]	0.629 ± 0.004	0.674 ± 0.006	0.712 ± 0.004	0.749 ± 0.004	0.775 ± 0.004
ICEBERG 1.0 [10]	0.721 ± 0.001	0.756 ± 0.003	0.787 ± 0.001	0.812 ± 0.002	0.834 ± 0.002
ICEBERG 2.0 [31]	0.838 ± 0.006	0.867 ± 0.005	**0.890 ± 0.006**	**0.907 ± 0.009**	**0.920 ± 0.009**
GLACIER (ours, w/ CF)	**0.859 ± 0.011**	**0.875 ± 0.009**	0.889 ± 0.008	0.900 ± 0.008	0.909 ± 0.010

Table 22: Detailed retrieval accuracy for positive mode adducts on NIST'20 dataset. ±: 95% CI from three random seeds. Baseline results are adapted from Wang et al. [31].

	Positive mode, Random split				
Top-k accuracy	Top-1	Top-2	Top-3	Top-4	Top-5
GrAFF-MS [16]	0.173 ± 0.004	0.316 ± 0.003	0.417 ± 0.007	0.490 ± 0.006	0.545 ± 0.000
MassFormer [35]	0.219 ± 0.002	0.369 ± 0.004	0.476 ± 0.005	0.551 ± 0.006	0.607 ± 0.005
ICEBERG 1.0 [10]	0.189 ± 0.012	0.375 ± 0.005	0.489 ± 0.007	0.567 ± 0.005	0.623 ± 0.004
ICEBERG 2.0 [31]	0.340 ± 0.004	0.550 ± 0.011	0.661 ± 0.005	0.735 ± 0.006	0.777 ± 0.006
GLACIER (ours, w/ CF)	**0.520 ± 0.011**	**0.684 ± 0.002**	**0.774 ± 0.002**	**0.821 ± 0.002**	**0.854 ± 0.005**
	Positive mode, Random split				
Top-k accuracy	Top-6	Top-7	Top-8	Top-9	Top-10
GrAFF-MS [16]	0.591 ± 0.003	0.625 ± 0.002	0.660 ± 0.004	0.690 ± 0.005	0.714 ± 0.006
MassFormer [35]	0.654 ± 0.004	0.693 ± 0.004	0.725 ± 0.004	0.755 ± 0.001	0.775 ± 0.004
ICEBERG 1.0 [10]	0.666 ± 0.002	0.698 ± 0.002	0.725 ± 0.003	0.749 ± 0.003	0.770 ± 0.002
ICEBERG 2.0 [31]	0.810 ± 0.006	0.834 ± 0.008	0.855 ± 0.011	0.871 ± 0.008	0.885 ± 0.008
GLACIER (ours, w/ CF)	**0.876 ± 0.005**	**0.893 ± 0.004**	**0.905 ± 0.005**	**0.915 ± 0.005**	**0.922 ± 0.005**
	Positive mode, Scaffold split				
Top-k accuracy	Top-1	Top-2	Top-3	Top-4	Top-5
GrAFF-MS [16]	0.138 ± 0.003	0.262 ± 0.005	0.358 ± 0.008	0.437 ± 0.006	0.502 ± 0.003
MassFormer [35]	0.179 ± 0.007	0.319 ± 0.004	0.421 ± 0.003	0.499 ± 0.003	0.562 ± 0.003
ICEBERG 1.0 [10]	0.210 ± 0.004	0.397 ± 0.006	0.518 ± 0.006	0.599 ± 0.007	0.657 ± 0.002
ICEBERG 2.0 [31]	0.321 ± 0.006	0.525 ± 0.003	0.650 ± 0.007	0.724 ± 0.005	0.778 ± 0.005
GLACIER (ours, w/ CF)	**0.451 ± 0.004**	**0.631 ± 0.004**	**0.729 ± 0.009**	**0.787 ± 0.010**	**0.825 ± 0.009**
Top-k accuracy	Top-6	Top-7	Top-8	Top-9	Top-10
GrAFF-MS [16]	0.553 ± 0.002	0.598 ± 0.003	0.633 ± 0.001	0.670 ± 0.002	0.699 ± 0.001
MassFormer [35]	0.615 ± 0.004	0.660 ± 0.007	0.698 ± 0.006	0.734 ± 0.005	0.761 ± 0.006
ICEBERG 1.0 [10]	0.703 ± 0.001	0.738 ± 0.003	0.769 ± 0.001	0.794 ± 0.003	0.817 ± 0.003
ICEBERG 2.0 [31]	0.818 ± 0.005	0.848 ± 0.005	0.871 ± 0.004	0.889 ± 0.008	0.903 ± 0.008
GLACIER (ours, w/ CF)	**0.852 ± 0.010**	**0.869 ± 0.008**	**0.885 ± 0.007**	**0.896 ± 0.008**	**0.907 ± 0.008**

I Visualization of Predicted Spectra

We sample three different molecules from the NIST'20 dataset and plot the predicted spectra with the highest and lowest collision energy in Figure 4 with model trained on the NIST'20 random split with seed set to 1, d set to 3, and δ set to 0.5.

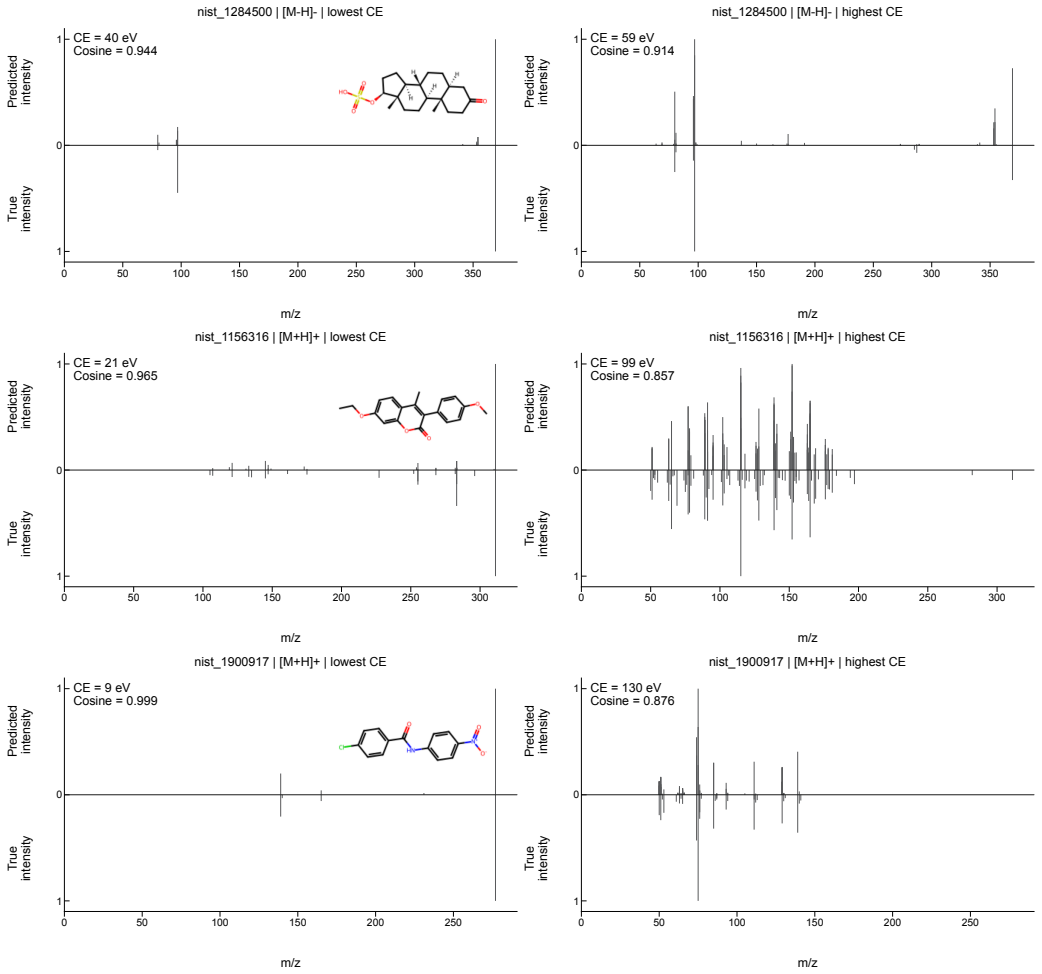


Figure 4: Visualization of predicted spectra from three different molecules with the highest and the lowest collision energy.

J Visualization of Predicted Break Points

We sample a molecule from the NIST'20 dataset and plot the predicted breakpoints with in Figure 4 with model trained on the NIST'20 random split with seed set to 1, d set to 3, and δ set to 0.5. Predicted breakpoints are labeled in red.



Figure 5: Visualization of predicted breakpoints.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe our methods in Section 3 and we run experiments that verify our claim in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations and future work explicitly in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The work in this paper is empirical and does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide implementation details and hyper-parameters regarding our models, and the datasets we use for evaluation are open-sourced.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is available at `https://github.com/coleygroup/ms-pred`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The code is available at `https://github.com/coleygroup/ms-pred`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experimental results in Section 4 include confidence intervals.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We document the compute resources we use in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow all guidelines described in the protocol.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential impacts of the paper in Section 5. We believe this work has a potentially positive societal impact with minimal risk.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in the paper are licensed and properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All assets introduced in this paper can be accessed in `https://github.com/coleygroup/ms-pred`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This Research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM is only used for paper writing assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.