MSAlign: Aligning Molecule and Mass Spectra Representations for Metabolite Identification

Paul Krzakala
LTCI, Télécom Paris & CMAP, Ecole Polytechnique,
Institut Polytechnique de Paris

Gabriel Melo
LTCI, Télécom Paris,
Institut Polytechnique de Paris

Camille Lançon
CEA, INRAE, MetaboHUB,
Université Paris-Saclay

Charlotte Laclau
LTCI, Télécom Paris,
Institut Polytechnique de Paris

Rémi Flamary
CMAP, Ecole Polytechnique,
Institut Polytechnique de Paris

Etienne Thévenot
CEA, INRAE, MetaboHUB,
Université Paris-Saclay

Florence d'Alché-Buc
LTCI, Télécom Paris,
Institut Polytechnique de Paris

Abstract

Accurately identifying metabolites i.e. small molecules from mass spectrometry data remains a core challenge in metabolomics, with broad applications in drug discovery, environmental analysis, and clinical research. We address the Molecule Retrieval task, which consists in recovering the chemical structure of a metabolite from its MS/MS spectrum given a set of candidate molecules. We make three contributions. First, we propose a unified framework encompassing recent approaches based on representation alignment and contrastive learning. Second, we introduce MSAlign, a lightweight model that achieves state-of-the art performances by aligning two frozen foundation models (DreaMS for mass spectra and MolDeBERTa for molecules) and demonstrate that a score fusion strategy further improves the performance for a very small computational cost. Third, we investigate a long-standing evaluation problem: data splitting strategies in molecule retrieval implicitly trade off data leakage against domain shift. We formalize this tension by introducing a quantitative measure of distribution shift, and use it to evaluate splitting strategies in existing benchmarks. All datasets, splits, candidate sets, and a unified implementation of MSAlign and baselines are publicly released to support reproducible research.

1 Introduction

Metabolomics, the study of small molecules in biological systems, is central to understanding disease mechanisms, discovering biomarkers, and developing therapeutics [45, 63]. To identify small molecules (metabolites) present in biological samples, biochemists rely mainly on Liquid Chromatography coupled with tandem Mass Spectrometry (LC-MS/MS) [2]. In a nutshell, the different types of ions generated from the molecule entering the mass spectrometer can be selected individually according to their mass-to-charge ratio (m/z) and fragmented at specific collision energies [18]. The resulting measured MS/MS spectrum consists of a list of tuples (m/z, intensity) corresponding to the fragment ions, which provides information about the chemical bonds and substructures of the compound [16]. Recovering the full 2D structure from this spectrum, i.e., *metabolite identification*, is critical yet fundamentally hard [21, 60, 65]. The dominant practical approach to this problem relies on spectral library matching, where an experimental spectrum is

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

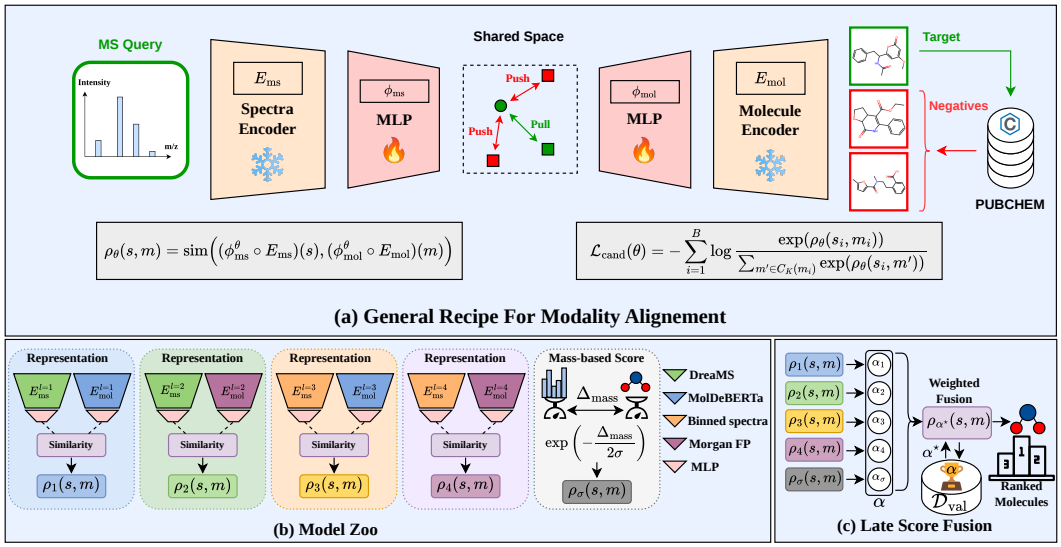


Figure 1: (a) Modality-alignment recipe; MSAlign instantiates it with frozen foundation encoders (DreaMS, MolDeBERTa). (b) The recipe yields a zoo of models, complemented by a target-mass score. (c) Late fusion aggregates them via an optimal convex combination of scores.

compared against a database of reference spectra from known compounds [35, 56]. However, these libraries cover less than 1% of the 110 million compounds in PubMed [4], which leaves most molecules unidentified.

Machine learning offers a way around this coverage bottleneck [29, 42], with rapid progress driven by large-scale datasets and the emergence of foundation models dedicated to mass spectrometry [11]. Because the space of metabolites is largely unknown, the most general setting, *de novo* structure elucidation, remains very difficult and its performance is still limited [5, 41, 53, 57, 62]. We focus instead on the more tractable *molecule retrieval* task. Given a spectrum and a set of candidate molecules (e.g., from PubChem), the goal is to retrieve the correct structure. This mirrors practical settings, where spectra are searched against large molecular datasets. Most of the proposed ML methods, even if not presented as such, fit the general energy-based framework [39], well suited to tackle structured output prediction as a retrieval task. The approaches to learning the scoring function are divided into two groups: some rely on fingerprint regression [20, 29] or more sophisticated variants [9, 17, 22], while others focus on learning spectra or molecule representations in a contrastive way [13, 26, 33]. In this work, we build on the latter approaches, exploiting the recent advances of multimodal alignment and chemical foundational models [1, 11].

Contributions. This paper addresses molecular retrieval from mass spectra through several complementary contributions detailed below:

- **A Unified framework** for molecule retrieval, clarifying how prior methods relate (Section 2).
- **MSAlign**, a model that aligns frozen foundation models into a shared spectrum and molecule space, achieving state-of-the-art performance despite lightspeed training. We show that its components are optimal among the alternatives identified in the unified framework (Section 3).
- **$\overline{\text{MSAlign}}$**, which further improves MSAlign with a score aggregation scheme, combining both alternative representations and one-dimensional chemical information.
- **Distribution shift analysis.** We show that splits that minimize data leakage tend to maximize domain shift, and vice versa. We then introduce a metric to navigate this trade-off and make benchmarks easier to interpret (Section 4).
- **Experiments.** We run extensive experiments and ablations that establish state-of-the-art results and validate our design choices (Sections 5.2, 5.3).

The open-source code for reproducible implementation of MSAlign baselines is provided at:

https://github.com/KrzakalaPaul/MSAlign-NeurIPS2026

Table 1: Comparison of baselines methods within the proposed unified framework.

Method	Spectra encoder		Molecule encoder		Training objective		Unsupervised data		Require Formula
	Fixed (E_{ms})	**Trained** (ϕ_{ms})	**Fixed** (E_{mol})	**Trained** (ϕ_{mol})	**Pretrain**	**Finetune**	**Spectra**	**Mol.**	
FFN	Binarize Spectra	—	Fingerprint	—	$\mathcal{L}_{\text{FP}}$ (3)	—	×	×	No
DeepSets	Fourier Features	DeepSets	Fingerprint	—	$\mathcal{L}_{\text{FP}}$ (3)	—	×	×	No
MIST	Peaks Subformulas	Transformer	Fingerprint	MLP	$\mathcal{L}_{\text{FP}}$ (3)	$\mathcal{L}_{\text{cand}}$ (6)	Simulated	Negatives	Yes
Emb-Cos	Binarize Spectra	MLP	Fingerprint	MLP	$\mathcal{L}_{\text{cand}}$ (6)	—	×	Negatives	No
JESTR	Binarize Spectra	MLP	Graph	GNN	$\mathcal{L}_{\text{batch}}$ (4)	$\mathcal{L}_{\text{cand}}$ (6)	×	Negatives	No
FLARE	Peaks Subformulas	Transformer	Graph	GNN	$\mathcal{L}_{\text{batch}}$ (4)	$\mathcal{L}_{\text{cand}}$ (6)	×	Negatives	Yes
MSAlign	DreaMS	MLP	ChemBERTa	MLP	$\mathcal{L}_{\text{cand}}$ (6)	—	DreaMS	ChemBERTa	No

2 A unified framework for molecule retrieval

In this section, we introduce a unified framework to analyze the different approaches proposed in the recent literature for molecule retrieval from MS/MS spectra.

Molecule retrieval problem. Let $\mathcal{S}$ denote the space of mass spectra and $\mathcal{M}$ the space of molecules. Given a dataset of paired observations $(s_i, m_i)_{i=1}^N \in \mathcal{S} \times \mathcal{M}$, the goal of Metabolite Identification is to learn a model $f : \mathcal{S} \to \mathcal{M}$ that predicts the 2D molecular structure associated with a given spectrum. This problem presents difficult challenges due to the discreteness and combinatorial nature of the output space $\mathcal{M}$ and fits the general framework of Structured Output Prediction [3, 43]. Molecule Identification is generally cast into a Molecule Retrieval problem, where the search is limited to a candidate set $C(s) \subset \mathcal{M}$. In that case, f_θ, the retrieval function, based on a scoring function $\rho_\theta : \mathcal{S} \times \mathcal{M} \to \mathbb{R}$ of parameter θ, writes as follows:

$$f_\theta(s) = \underset{m' \in C(s)}{\arg\max}\, \rho_\theta(s, m'). \tag{1}$$

Then, the goal is to learn a scoring function that assign higher scores to the correct pairs than to the incorrect ones which connects to the large body of works in energy-based models [39].

In our problem, the candidate sets are typically defined via a mass filter, i.e., $C(s) = \{m' \in \mathcal{M} : \text{abs}(|m'| - |m_s|) < \epsilon\}$ where $|m'|$ (resp. $|m_s|$) is the mass of a candidate (resp. target) molecule and ϵ a threshold. This is a reasonable assumption, as the determination of the mass of the neutral molecule given the MS/MS spectrum (i.e., the determination of the ion species that was fragmented) is a well-managed task both manually and automatically [60].

Scoring function and training loss. We focus here on the large class of scoring functions ρ_θ defined as a similarity *sim*: $\mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$ over a shared representation $\mathcal{Z}$ of spectra and molecules, i.e.

$$\rho_\theta(s, m) = \text{sim}(g_\theta(s), h_\theta(m)) \tag{2}$$

where $g_\theta : \mathcal{S} \to \mathcal{Z}$ and $h_\theta : \mathcal{M} \to \mathcal{Z}$ are encoders or *feature maps* and sim is typically the cosine similarity. The two following approaches have been proposed to learn the score:

Retrieval through regression. A first approach is to concentrate on learning the feature map g_θ while keeping $h_\theta(m) = h(m)$ fixed, either as a molecular fingerprint $FP(m)$ i.e. a binary vector representing the presence/absence of certain substructures [26, 29, 32] or as a feature vector in a Hilbert space [7, 9]. Learning the score function boils down to estimate the regression of the variable $h(m)$ by the variable s. Taking the example of fingerprint regression, the loss function can then be chosen as:

$$\mathcal{L}_{\text{FP}}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} \text{sim}(g_\theta(s_i), \text{FP}(m_i)). \tag{3}$$

Retrieval through constrastive learning. The alternative is to jointly learn the feature maps g_θ and h_θ. This formulation connects self-supervised multimodal alignment [46] with supervised retrieval, naturally leading to contrastive learning objectives [12, 61]. The most common choice is the InfoNCE loss [13, 17, 33], which encourages the similarity of the true pair (s, m) to be higher than that of negative pairs (s, m') and practical implementation differ in the definition of negatives pairs. As training is performed through mini-batch iteration, a common choice is to use the other molecules in

the batch as negatives, leading to the following *in-batch* InfoNCE loss:

$$\mathcal{L}_{\text{batch}}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\rho_\theta(s_i, m_i)/\epsilon)}{\sum_{j=1}^{B} \exp(\rho_\theta(s_i, m_j)/\epsilon)}. \tag{4}$$

However, this approach is limited by the fact that in-batch negatives are not necessarily hard to distinguish, making the training loss less informative. A large body of work attempt to address this limitation through hard negative mining or by scaling the number of negatives [49, 58, 66]. In the next section, we argue that these strategies are unnecessary for molecular retrieval, as the chemically defined candidate set $C(s_i)$ provides a natural source of hard negatives which yields the *candidate-based* InfoNCE loss (6).

Architecture and data representation. The second key design choice is the architecture of the encoders g_θ and h_θ and the underlying data representation for each modality [44].

For spectra, common approaches include discretizing the m/z axis into fixed bins [17, 33], applying convolutional neural networks [37], or modeling spectra as sequences of peaks (optionally enriched with formula annotations) and processed by transformer architectures [13, 14, 26].

For molecules, classical representations rely on handcrafted fingerprints, optionally refined with an MLP [26], while more recent methods use graph neural networks (GNNs) to directly encode molecular structures [13, 33]. An alternative paradigm represents molecules as SMILES or SELFIES strings and leverages transformer-based language models to learn molecular embeddings [23, 28].

Unified framework. We summarize the possible encoding choices in a compact formulation:

$$g_{\theta_1}(s) = (\phi_{ms}^{\theta_1} \circ E_{ms})(s), \quad h_{\theta_2}(m) = (\phi_{mol}^{\theta_2} \circ E_{mol})(m), \tag{5}$$

in which E_{ms} and E_{mol} are non-trainable preprocessing steps (e.g. binning for spectra, fingerprinting for molecules) and $\phi_{ms}^{\theta_1}$ and $\phi_{mol}^{\theta_2}$ are trainable encoders (e.g. MLP, GNN, transformer). Under this formulation, a large body of work can be reduced to the choice of E_{ms}, E_{mol}, $\phi_{ms}^{\theta_1}$ and $\phi_{mol}^{\theta_2}$, as well as the loss function among those presented in (3), (4) or (6). Table 1 summarizes the comparison between recent methods along with the key design features discussed above.

3 MSAlign: lightweight model with state-of-the-art performances

Foundation model alignment. We propose a lightweight implementation of the general framework introduced above where $E_{ms} : \mathcal{S} \mapsto \mathbb{R}^{d_{\text{ms}}}$, $E_{mol} : \mathcal{M} \mapsto \mathbb{R}^{d_{\text{mol}}}$ are fixed vector representation for each modality, and $\phi_{ms}^{\theta_1} : \mathbb{R}^{d_{\text{ms}}} \mapsto \mathbb{R}^d$, $\phi_{mol}^{\theta_2} : \mathbb{R}^{d_{\text{mol}}} \mapsto \mathbb{R}^d$ are lightweight MLP adapters. Inspired by recent works in vision–language alignment [51, 66] and the Platonic Representation Hypothesis [31] we propose to align frozen foundation-model representations and leverage DreaMS [11] as E_{ms} and MolDeBERTa [15] as E_{mol}. Critically, since the foundation-model are frozen, the representations $E_{ms}(s)$ and $E_{mol}(m)$ can be precomputed which enables lightspeed training and makes it possible to leverage the rich candidate-based InfoNCE loss described below. We refer to Appendix B for additional architecture details and Table 4 for comparison with alternative frozen encoders.

Candidate-based training. We argue that the *in-batch* InfoNCE loss (4) is not suited for molecular retrieval, as molecules in the same batch are typically easy to distinguish (by mass or formula for instance). We refer to Appendix A for further discussion on the topic. This is a classical limitation in contrastive learning, which typically calls for hard negative sampling strategies [49]. Fortunately, in our setting, the chemically defined candidate set $C(s_i)$ provides a natural source of hard negatives which can be exploited via the *candidate-based* InfoNCE loss:

$$\mathcal{L}_{\text{cand}}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\rho_\theta(s_i, m_i)/\epsilon)}{\sum_{m' \in C_K(s_i)} \exp(\rho_\theta(s_i, m')/\epsilon)}, \tag{6}$$

where $C_K(s_i) \subset C(s_i)$ is a subset of K negatives sampled from the candidate set. This loss is much more informative than its *in-batch* counterpart, and in particular it is a valid convex surrogate of the top-1 error as formalized in Proposition 2. The only limitation of (6) is its computational costs as it requires $K \times B$ similarity evaluations $\rho_\theta(s_i, m')$ with limited reuse of molecule embeddings across spectra. Critically, the pre-computation of embeddings, make the use of this loss tractable and even surprisingly fast, with less than 2minutes/epochs on MassSpecGym as reported in Table 10.

Late score fusion. The lightweight recipe described above makes it inexpensive to train several alignment models by varying the spectral and molecular frozen representations, yielding a model zoo (Figure 1b). We hypothesize that models based on different representations yields uncorrelated errors (this is verified experimentally, see Figure 3c), which naturally calls for a model aggregation strategy.

Following earlier work on molecular identification [8, 40, 54, 55], we combine L fixed scores through a convex combination:

$$\rho_{\boldsymbol{\alpha}}(s,m) = \sum_{\ell=1}^{L} \alpha_\ell \rho_\ell(s,m), \qquad \boldsymbol{\alpha}^\star \in \underset{\boldsymbol{\alpha}\in\Delta^{L-1}}{\arg\max}\ \mathrm{R@1}_{\mathrm{val}}(\boldsymbol{\alpha}). \tag{7}$$

where $\mathrm{R@1}_{\mathrm{val}}$ is the top-1 accuracy on the held-out validation split and Δ^{L-1} is the simplex. This is a non-convex problem in L weights which we approximate using a smooth surrogate and validation-based temperature selection. The procedure is detailed in Appendix B.2 and Algorithm 1.

Fusion also accommodates one-dimensional chemical information. Let $\widehat{M}(s)$ be the neutral mass estimated from the precursor m/z, accounting for charge and adduct, and $M(m)$ the candidate's monoisotopic mass. We define

$$e_{\mathrm{ppm}}(s,m) = 10^6 \frac{\widehat{M}(s) - M(m)}{M(m)}, \qquad \rho_\sigma^{\mathrm{mass}}(s,m) = \exp\left[-\frac{e_{\mathrm{ppm}}(s,m)^2}{2\sigma^2}\right]. \tag{8}$$

Ultimately, the proposed model, **MSAlign**, combines $L = 7$ scores: four alignment scores from pairing binned spectra or DreaMS with Morgan fingerprints or MolDeBERTa, and three mass scores with bandwidths $\sigma \in \{0.01, 0.1, 1\}$ ppm. Table 6 isolates the gains from alignment-score fusion and the additional mass scores, while Table 3 compares the resulting model with competing methods.

MSAlign refers to the raw model achieved by aligning DreaMS and MolDeBERTa representations (no score fusion), it is the strongest individual components of **MSAlign** (Table 6).

4 Assessing the challenges of data splitting in metabolite

Before moving to the experimental evaluation of MSAlign, we first discuss a critical aspect of the experimental protocol: the choice of data splitting strategy.

Challenge. Data splitting is a non-trivial aspect of MS/MS benchmarking. The same molecule can appear in multiple spectra acquired under different experimental conditions, making naive random splits prone to **data leakage**. To mitigate this issue, existing benchmarks typically enforce some degree of molecular dissimilarity across folds. However, increasingly stringent splitting criteria can induce substantial **distribution shifts** between training and test sets [47], which may not reflect practical deployment settings and may require dedicated domain adaptation techniques [24].

A wide range of splitting strategies have therefore been proposed, differing primarily in how strongly they separate molecules across folds. For instance, the MassSpecGym MCES split requires molecules from different folds to have an edit distance of at least 10 [10], whereas NPLIB1 only requires the corresponding structures to be distinct (InChIKey split). We refer to Appendix D for a detailed presentation of alternative criterions.

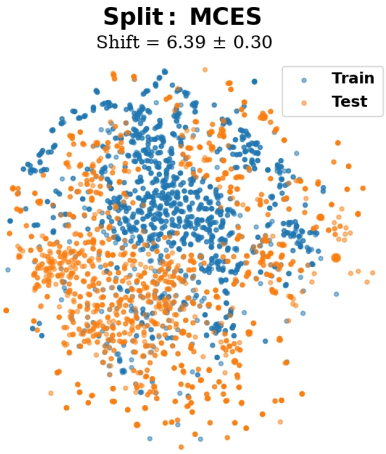


Figure 2: t-SNE visualization of the MassSpecGym MCES splits.

Yet, to the best of our knowledge, there is no principled way to compare these strategies or quantify the distribution shift they induce. This makes it difficult to assess the trade-off between preventing data leakage and preserving a realistic train–test distribution. To address this issue, we introduce a quantitative metric that measures the degree of distribution shift induced by a given splitting strategy.

Encoders	Data Splitting Criterion					
	Random	**InChIKey**	**Formula**	**Murcko Scaffold**	**MCES**	**Murcko Histogram**
Binned – Morgan	1.00 ± 0.03	4.71 ± 0.17	5.19 ± 0.35	9.24 ± 1.02	9.10 ± 0.33	15.22 ± 4.83
DreaMS – Morgan	1.00 ± 0.03	4.99 ± 0.20	5.71 ± 0.35	9.76 ± 1.26	11.95 ± 0.53	17.66 ± 5.64
Binned – MolDeBERTa	1.00 ± 0.02	4.00 ± 0.42	4.65 ± 0.27	7.01 ± 0.71	12.41 ± 0.86	15.12 ± 4.32
DreaMS – MolDeBERTa	1.00 ± 0.04	4.44 ± 0.28	4.98 ± 0.44	8.51 ± 0.57	15.89 ± 0.75	18.42 ± 5.62

Table 2: Normalized Sliced Wasserstein shift between the training and test sets for the different data-splitting strategies. We report average and standard deviation over five splits, except for MCES which has two splits.

Quantitative measure. We propose to compute the Sliced Wasserstein distance [6] between the training and test distributions. This requires encoding each sample $x = (s, m)$ as a vector

$$F(x) = \text{concat}\left(\frac{\text{encode}_s(s)}{\|\text{encode}_s(s)\|_2}, \frac{\text{encode}_m(m)}{\|\text{encode}_m(m)\|_2}\right) \in \mathbb{R}^{d_s+d_m}, \tag{9}$$

and we report the results both for foundation-model embeddings ($\text{encode}_s = \text{DreaMS}$ and $\text{encode}_m = \text{MolDeBERTa}$) and for simple vectorization baselines i.e. a fingerprint ($\text{encode}_m = \text{Morgan}$) for molecules and a binned spectrum ($\text{encode}_s = \text{Binned}$) for spectra.

Finally, to make the metric more interpretable, we normalize it by the value obtained with a random split. Formally, let $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ denote the training and test sets to evaluate, and let $(\mathcal{D}_1, \mathcal{D}_2)$ be a random split such that $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$. We define the normalized sliced Wasserstein distance by

$$\text{Shift}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) = \frac{\text{SWD}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}})}{\mathbb{E}_{(\mathcal{D}_1,\mathcal{D}_2)}[\text{SWD}(\mathcal{D}_1, \mathcal{D}_2)]}. \tag{10}$$

where SWD is the Sliced Wasserstein distance between the empirical distributions of the two sets. We use the implementation from the POT library [25] with 100 random projections.

Analysis on MassSpecGym. In Table 2, we report the proposed metric on MassSpecGym across the different splitting strategies and encoders encode_m, encode_s. We observe that the overall ranking is robust to the choice of encoders. MCES and Murcko splits are consistently the splits that induce the largest shifts. We illustrate this in Figure 2 by plotting a 2D visualization of the joint latent space (9) for the MCES split: we observe that the test set forms a distinct cluster far from the training data.

Hence, those splits are particularly challenging, and can be used to measure the out-of-distribution generalization capabilities of a model, this may be relevant for specific applications such as de novo drug discovery. However, such a strong shift may not reflect practical deployment in retrieval settings, where test molecules are likely to share some similarity with those in the training set. We encourage future works to address these challenging splits with dedicated domain adaptation techniques.

Overall, the Formula and 2D InChIKey splits provide a more balanced trade-off, with Shift values around 4-5. Of the two, we recommend Formula-split as it is strictly more challenging (two molecules with the same InChiKey necessarily have the same formula), and we adopt it for all subsequent experiments. For completeness, MassSpecGym's harder original MCES split is also evaluated.

5 Experimental Results

In this section, we first present the experimental setting 5.1, then we report the retrieval performances of MSAlign and MSAligncompared to a range of baselines 5.2. Finally, we validate step by step the key design choices of our method 5.3.

5.1 Experimental Setting

Supervised Datasets We consider two open-source benchmarks. **MassSpecGym** [10] provides 231,104 carefully curated and standardized pairs, and is rapidly becoming the community standard for benchmarking. **Spectraverse** [27] is the larger (488,797 pairs) and provides broader coverage of adducts and ionization modes, with more 50% of spectra corresponding to non-$[M+H]^+$ adducts. Note that pair counts overestimate chemical diversity as molecules frequently appear under multiple ionization conditions; Table 7 reports unique molecule counts along with pair counts for all datasets.

Table 3: Recall score across datasets and models. We report the standard deviation **over 3 random splitting** of the dataset. Methods reported on the right require an oracle to predict the target's formula.

		Without molecular formula					Requires target molecular formula			
Dataset	**Metric**	**DeepSet**	**JESTR**	**Emb-Cos**	**MSAlign**	**$\overline{\text{MSAlign}}$**	**MIST**	**MVP**	**FLARE**	**$\overline{\text{MSAlign}}$ (+Filter)**
SpectraVerse	R@1	2.0 ± 0.4	5.4 ± 0.2	26.6 ± 1.9	31.8 ± 2.6	**41.3** ± 2.7	22.7 ± 1.6	9.6 ± 0.6	14.2 ± 1.1	**44.4** ± 2.6
	R@5	8.4 ± 1.0	17.5 ± 1.3	49.5 ± 1.0	58.2 ± 1.9	**68.3** ± 1.7	41.5 ± 0.8	25.9 ± 1.8	34.7 ± 2.6	**71.4** ± 1.5
	R@20	22.9 ± 0.9	39.4 ± 2.0	71.0 ± 1.3	78.3 ± 1.7	**85.4** ± 0.9	64.1 ± 0.5	48.3 ± 0.7	60.3 ± 2.0	**87.5** ± 0.9
MassSpecGym (Formula Split)	R@1	4.7 ± 0.6	13.9 ± 0.5	42.7 ± 3.0	47.5 ± 3.4	**58.2** ± 3.4	41.4 ± 1.7	23.3 ± 3.4	32.8 ± 2.5	**60.4** ± 3.7
	R@5	13.0 ± 1.3	32.5 ± 0.8	65.2 ± 2.8	71.3 ± 2.6	**80.7** ± 3.0	60.8 ± 0.7	47.3 ± 2.0	61.8 ± 2.1	**82.1** ± 3.3
	R@20	27.5 ± 2.0	57.0 ± 0.7	80.4 ± 1.6	87.0 ± 1.1	**92.8** ± 1.1	76.2 ± 0.9	71.8 ± 0.6	82.3 ± 1.5	**93.6** ± 1.1
MassSpecGym (MCES Split)	R@1	1.9 ± 0.5	7.9 ± 1.1	12.6 ± 0.2	19.0 ± 3.2	**30.8** ± 2.5	16.4 ± 0.1	25.6 ± 0.8	28.4 ± 2.1	**37.7** ± 1.3
	R@5	8.4 ± 0.6	22.3 ± 1.4	29.9 ± 0.2	41.2 ± 2.5	**59.3** ± 0.7	35.8 ± 0.7	55.3 ± 0.1	60.2 ± 1.4	**67.2** ± 1.0
	R@20	21.8 ± 1.4	44.8 ± 1.9	51.6 ± 0.9	64.9 ± 0.9	**82.0** ± 0.8	60.2 ± 2.3	79.9 ± 2.3	82.8 ± 2.8	**87.1** ± 2.3

Candidates Sets. Following the standardized protocol of MassSpecGym [10], candidate molecules are retrieved from PubChem [34] (118M compounds) by selecting structures whose molecular mass matches the target molecule within a 10 ppm tolerance, prioritizing compounds sampled from databases of decreasing metabolite relevance (see Algorithm 1 of [10] for details). We keep $K = 256$ candidates per spectrum; as illustrated in Figure 3, these candidates are typically structurally close to the target, making retrieval challenging even at this moderate pool size.

Data Splitting and seeds. Following the discussion in Section 4, we split the dataset using the chemical formula as a criterion, ensuring that molecules in the train/validation/test sets have distinct formulas, and report results averaged over three random seeds of the splitting algorithm. For completeness, we also report results on the official MassSpecGym MCES split. Since generating alternative MCES splits is computationally expensive, we instead average over its two variants obtained by swapping the validation and test sets, an important source of variability since the validation set is used for both calibration and checkpoint selection by MSAlign and baselines.

Baselines. We propose a comprehensive list of recent baselines: **DeepSets** is a fingerprint prediction baseline introduced in MassSpecGym [10]. To maximize the performances, we use the Fourier Features variants and increase the number of epochs to 50 which significantly improves this baseline compared to default settings. **MIST** [26] is more complex model that pretrains with fingerprint prediction, finetunes with contrastive learning, and augments the data with a forward model for spectra simulation. **JESTR** [33] is a simple contrastive model that pretrains with batch InfoNCE (4) and finetunes with the more computationally expensive candidates InfoNCE (6). **MVP** [67] builds on JESTR by introducing a multi-view setting (fingerprint, graph, spectra and consensus spectra). **FLARE** [13] is a direct follow-up work that considers a stronger spectra encoder leveraging peak annotation with subformulas and a more interpretable, richer similarity. **Emb-Cos** is the strongest of the four models proposed in [17]; similarly to MSAlign, it uses candidates InfoNCE to learn a shared space for molecules and mass spectra, but it does not use pretrained encoders or score fusion.

5.2 Retrieval Performances and comparison to competitors

The performances of the **MSAlign** and the baselines presented above, are collected in Table 3.

Formula oracle. We note that some baselines require the molecular formula of the target to be provided. This is a strong assumption as in practice it is unknown and must be estimated, which remains an active research problem [19, 30, 64]. Consequently, we propose to explicitly distinguish the methods that make this assumption (right panel) from those that don't (left panel). For fair comparison, we propose a simple variant of our method (**$\overline{\text{MSAlign}}$ + Filter**) that leverage this oracle by filtering the candidates to keep only those matching the ground-truth formula.

Results. Overall, we observe that **MSAlign** and its variants achieves strikingly strong results across settings. Standalone **MSAlign** already provides a strong and computationally efficient baseline: it achieves the best mean recall on **Spectraverse** and **MassSpecGym** formula splits and outperforms all formula agnostic methods on **MassSpecGym** MCES split, while also being the fastest method to train (Table 10). Each proposed extension further improves retrieval performance. Score fusion increases R@1 by 9.5, 10.7, and 11.8 percentage points on SpectraVerse, MassSpecGym Formula,

Table 4: MassSpecGym performance (Formula split) across encoder pairs. The upper grid compares foundational and classical representations; the lower grid compares molecular foundation models.

Encoders		MassSpecGym		
Spectra (dim)	**Molecules** (dim)	**R@1**	**R@5**	**R@20**
Binned (10050)	Fingerprint (4096)	34.3 ± 1.3	56.2 ± 2.4	71.8 ± 1.3
DreaMS (1024)	Fingerprint (4096)	38.6 ± 2.8	60.4 ± 2.1	77.6 ± 0.8
Binned (10050)	MolDeBERTa (768)	44.5 ± 5.1	68.5 ± 3.9	84.9 ± 0.7
DreaMS (1024)	MolDeBERTa (768)	**47.5** ± 3.4	71.3 ± 2.6	87.0 ± 1.1
DreaMS (1024)	ChemBERTa-13M (768)	46.9 ± 3.0	**71.6** ± 1.7	**87.8** ± 1.6
DreaMS (1024)	IBM MoLFormer (768)	46.5 ± 5.6	69.1 ± 3.1	85.6 ± 0.9
DreaMS (1024)	GROVER (3200)	42.2 ± 5.0	67.4 ± 2.6	84.9 ± 0.7
DreaMS (1024)	MHG-GED (1024)	47.0 ± 5.2	70.3 ± 2.1	85.9 ± 1.8

and MassSpecGym MCES, respectively. When the ground-truth molecular formula is additionally provided, formula filtering yields another gain of 3.1, 2.2, and 6.9 percentage points. Consequently, the complete oracle variant, **MSAlign + Filter**, establishes a substantial performance gap over the strongest competing method in each setting, outperforming the second best method by 17.8, 17.7 and 9.3 percent, respectively. Finally, we note some discrepancies with previously reported results: JESTR performs below its original report, which echoes issues discussed in public implementations[1]. We observe a similar trend for FLARE, potentially due to shared implementation components.

5.3 Ablation studies and interpretation of the results

Encoders. In Section 3, we introduced the alignment of precomputed representations as a general recipe for molecular retrieval. MSAlign instantiates this approach by aligning DreaMS and MolDeBERTa representations. We now assess this default choice against a broad range of alternative encoders. On the molecular side, we consider representations ranging from classical Morgan fingerprints [59] to SMILES-based [1, 15, 52] and graph-based foundation models [36, 50]. For spectra, we compare DreaMS [11] with a classical histogram representation using bins of width 0.1Da [33].

Results on MassSpecGym are reported in Table 4. We observe that foundation models consistently outperform classical vectorization techniques: replacing spectral histograms with DreaMS improves performance by $+4.3\%$, replacing Morgan fingerprints with MolDeBERTa by $+10.2\%$, and combining both foundation models yields $+13.2\%$. In contrast, the choice among molecular foundation models has a comparatively limited impact: the different models perform similarly overall, with MolDeBERTa consistently achieving the best or second-best results in R@k scores.

Training loss. MSAlign is trained with the candidate-based InfoNCE loss (6), we compare this approach against batch InfoNCE (4) and regression losses, $\mathcal{L}_{\text{reg}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\rho_\theta(s_i, m_i)$. For regression, we consider both spectrum prediction, $\rho_\theta(s,m) = \text{sim}(E_{ms}(s), h_\theta(m))$, and molecule prediction, $\rho_\theta(s,m) = \text{sim}(g_\theta(s), E_{mol}(m))$. As shown in Table 5, candidate-based InfoNCE substantially outperforms all alternatives. We attribute this gap to three observations:

- **Regression** (mol $\rightarrow$ ms): under-performs because this mapping is inherently one-to-many: in MassSpecGym, each molecule is associated with 19 spectra on average.
- **Regression** (ms $\rightarrow$ mol): is also challenging as candidates are tightly clustered in molecular space (Figure 3a), such that even small regression errors can substantially degrade retrieval accuracy.
- **In-batch InfoNCE** trades informative candidate negatives for cheaper training. Figure 3b shows that this trade-off is unfavorable: candidate-based InfoNCE consistently outperforms its in-batch counterpart across batch sizes B and numbers of negatives K.

Late score fusion. In Section 3, we introduced late score fusion as a simple yet effective approach to model aggregation and defined MSAlign as the fusion of 7 complementary scorers: four alignment models (including MSAlign) and several one-dimensional heuristics based on the estimated mass of the target molecule. As shown in Table 3, score fusion consistently improves R@1 by approxi-

[1] https://github.com/HassounLab/JESTR1/issues/5

Table 5: Training loss comparison on MassSpecGym Formula split (Average over 3 seeds).

Training Loss	MassSpecGym		
	R@1	R@5	R@20
Regression (mol → ms)	22.8 ± 2.1	42.0 ± 1.3	60.3 ± 0.6
Regression (ms → mol)	16.7 ± 1.9	32.3 ± 2.4	49.6 ± 2.4
Contrastive (in-batch)	19.5 ± 2.1	42.6 ± 1.8	67.7 ± 2.8
Contrastive (candidate-based)	**47.5** ± 3.4	**71.3** ± 2.6	**87.0** ± 1.1

Table 6: Scorer ablation on MassSpecGym Formula split (Average over 3 seeds).

Model	MassSpecGym		
	R@1	R@5	R@20
MSAlign	47.5	71.3	87.0
+ 3 Modalities	54.4$^{+6.9}$	75.8$^{+4.5}$	89.9$^{+2.9}$
+ Mass $\sigma \in \{0.01, 0.1, 1\}$	51.3$^{+3.8}$	76.7$^{+5.4}$	91.0$^{+4.0}$
+ Both ($\overline{\text{MSAlign}}$)	**58.2**$^{+10.7}$	**80.7**$^{+9.4}$	**92.8**$^{+5.8}$

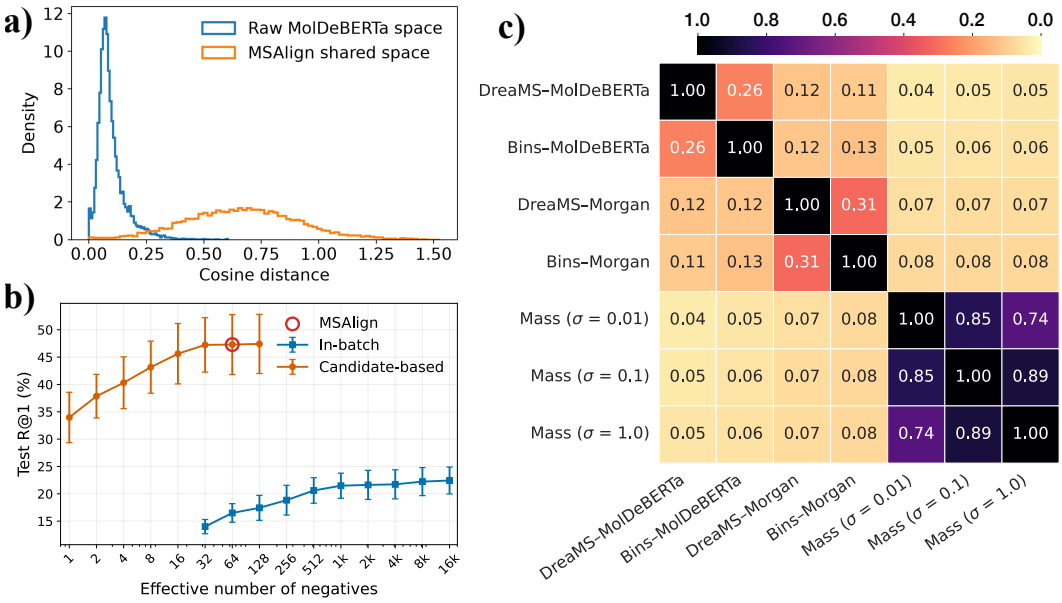


Figure 3: **a)** distribution of pairwise cosine distance between pairs of candidates $(m', m'') \in C(s)$. Candidates are very close in MolDeBERTa space ($E[d(m', m'')] = 0.09$) and thus hard to distinguish. On the contrary, MSAlign shared disentangle the candidates which facilitates retrieval ($E[d(m', m'')] = 0.67$). **b)** Performances of in-batch and candidate InfoNCE when scaling the number of negatives i.e. the batchsize B for in-batch and the hyperparameter K for candidate InfoNCE. In the second case batchsize is fixed to $B = 128$. **c)** Pairwise Jaccard index between the sets of candidates incorrectly ranked above the target by each scorer.

mately 10% over MSAlign, making $\overline{\text{MSAlign}}$ the best-performing method by a substantial margin. Importantly, this gain is not only due to combining several equally strong models: Table 6 shows that MSAlign is itself the strongest individual scorer while the mass scorer are weak on their own.

The ablation in Table 6 further reveals a striking near-additivity of the gains: the alternative alignment models improve R@1 by +6.9%, the mass-based scorers by +3.8%, and combining both yields +10.7%. To understand this complementarity, we analyze whether the different scorers make the same errors. For a scorer ρ, query spectrum $s \in \mathcal{S}$, and target molecule $m \in \mathcal{M}$, define the set of candidates incorrectly ranked above the target as $C_\rho(s, m) = \{m' \in C(s) \mid \rho(s, m') > \rho(s, m)\}$. For each pair of scorers (ρ, ρ'), we then compute the average Jaccard overlap between these sets,

$$J(\rho, \rho') = \frac{1}{N} \sum_{i=1}^{N} \frac{|C_\rho(s_i, m_i) \cap C_{\rho'}(s_i, m_i)|}{|C_\rho(s_i, m_i) \cup C_{\rho'}(s_i, m_i)|},$$

where the terms with zero denominator are discarded from the average.

Results are reported in Figure 3c. Overall, the pairwise overlaps are small, particularly between alignment- and mass-based scorers. For instance, the overlap between MSAlign (DreaMS–MolDeBERTa) and the mass scorer with $\sigma = 1$ is smaller than 5% . This confirms that the different scorers make largely complementary errors, providing a natural explanation for the effectiveness and near-additivity of late score fusion. Note that the Jaccard index can be somewhat misleading and we report a more detailed metric in Figure 7.

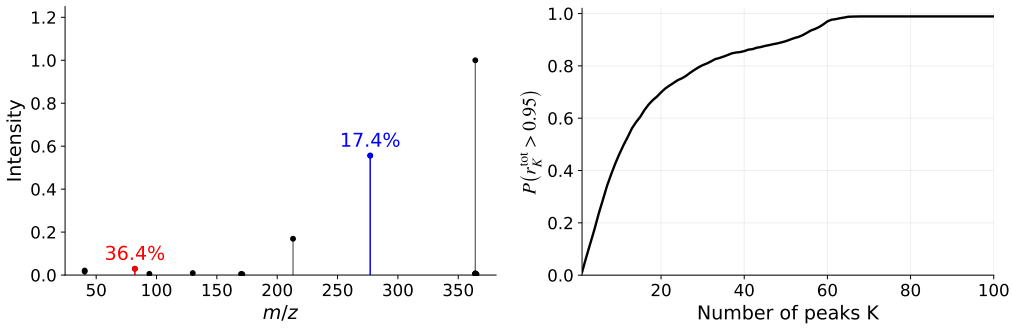


Figure 4: Using LIME to quantify peak importance. (Left) We illustrate the output of LIME by highlighting the two most important peaks of a spectra from MassSpecGym. (Right) We report the fraction samples (MassSpecGym test set) such that K peaks are sufficient to explain MSAlign output.

5.4 Interpretability of peak importance with LIME

MSAlign is a deep learning model with no built-in interpretability features which can be a limitation for practitioners. In this section, we demonstrate that this limitation can be mitigated by adapting **LIME**, a celebrated general purpose interpretability technique [48], to our setting.

Adapating LIME to molecular retrieval. LIME estimates individual feature contribution by fitting a linear regression under a user-defined distribution of feature perturbations. This approach can be smoothly adapted to molecular retrieval. Formally, consider a sample $(s, m) \in \mathcal{S} \times \mathcal{M}$ where the spectra is composed of n peaks: $s = (p_1, \dots, p_n)$. We propose to explain the contribution of each peak to the scalar prediction margin

$$f(s) = \rho(s, m) - \max_{c \in \mathcal{C} \setminus \{m\}} \rho(s, c). \tag{11}$$

To this end, we perturb the input spectra s by randomly dropping some peaks: for any $z \in \{0, 1\}^n$, we denote s_z the spectra such that $z_i = 1$ if and only if peak p_i is dropped. Then, following the general LIME strategy, we fit a linear regression to approximate $f(s)$ locally and the linear coefficient define the importance r_i of peak i:

$$f(s_z) \approx \beta_0 + \sum_{i=1}^{n} \beta_i z_i, \quad r_i = \frac{|\beta_i|}{\sum_{j=1}^{n} |\beta_j|}.$$

In practice, z is sampled from a Bernoulli distribution with parameter $q = 0.1$ for each peak and the linear regression is fit using $n = 1024$ samples.

Number of peaks contributing to MSAlign output. To illustrate the interest of this method, we propose to quantify how many peaks are needed to support MSAlign prediction. To this end, we define $r_K^{\text{tot}} = \sum_{i=1}^{K} r_i$, the total importance of the K most important peaks (assuming $r_1 > ... > r_n$). For different values of K, we compute the fraction on spectra in MassSpecGym test set such that $r_K^{\text{tot}} > 0.95\%$ i.e. the fraction of samples such that K peaks are sufficent to explain most of MSAlign prediction. The results are reported in Figure 4 (Right). Overall, we observe that MSAlign decision are based on at most $K = 60$ peaks and require less that $K \leq 10$ for about half of the samples.

6 Conclusion

We addressed molecule retrieval from mass spectra and unify recent approaches within a single scoring framework. Inspired by vision–language alignment, we introduced MSAlign, a model that aligns pretrained spectrum and molecule foundation models within a shared space using candidate-based InfoNCE loss. Additionally, we presented a simple late score fusion strategy that significantly boosts the performances of the model. Finally, we contributed to the interpretability of the existing benchmarks by defining a practical metric to navigate the tradeoff between data leakage and distribution shift. A natural direction for future work is to attach a (possibly pre-trained) decoder to MSAlign shared space to tackle the challenging task of *de novo* molecular identification. Pseudo-labeling with a forward model also appears as a key addition which should be explored in the future.

Acknowledgments and Disclosure of Funding

The study was funded by French National Research Agency (ANR) through the following projects: PEPR IA FOUNDRY (ANR-23-PEIA-0003), e-Lucid (ANR-25-TSIA-0002-01), MetaboHUB (ANR-11-INBS-0010) and through the France 2030 program (ANR-23-IACL-0005) under the MacLeOD project (ANR-25-PEIA-0005).

It also received funding from the European Union's Horizon Europe research and innovation programme under grant agreement 101120237 (ELIAS).

The first author received a PhD scholarship from Institut Polytechnique de Paris. We thanks NeurIPS reviewers and AC for the fruitful rebuttal. Anonymous reviewer vvVF questions on mass filtering motivated the addition of mass based scorers and score fusion in general.

References

[1] Ahmad, W., Simon, E., Chithrananda, S., Grand, G., and Ramsundar, B. (2022). ChemBERTa-2: Towards chemical foundation models. arXiv preprint arXiv:2209.01712.

[2] Alseekh, S., Aharoni, A., Brotman, Y., Contrepois, K., D'Auria, J., Ewald, J., C. Ewald, J., Fraser, P. D., Giavalisco, P., Hall, R. D., Heinemann, M., Link, H., Luo, J., Neumann, S., Nielsen, J., Perez de Souza, L., Saito, K., Sauer, U., Schroeder, F. C., Schuster, S., Siuzdak, G., Skirycz, A., Sumner, L. W., Snyder, M. P., Tang, H., Tohge, T., Wang, Y., Wen, W., Wu, S., Xu, G., Zamboni, N., and Fernie, A. R. (2021). Mass spectrometry-based metabolomics: A guide for annotation, quantification and best reporting practices. Nature Methods, 18(7):747–756.

[3] Bakır, G., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., and Vishwanathan, S. V. N., editors (2007). Predicting Structured Data. MIT Press, Cambridge, MA.

[4] Bittremieux, W., Wang, M., and Dorrestein, P. C. (2022). The critical role that spectral libraries play in capturing the metabolomics community knowledge. Metabolomics, 18(12):94.

[5] Bohde, M., Manjrekar, M., Wang, R., Ji, S., and Coley, C. W. (2025). DiffMS: diffusion generation of molecules conditioned on mass spectra. In Proceedings of the 42nd International Conference on Machine Learning, ICML'25. JMLR.org.

[6] Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and radon wasserstein barycenters of measures. Journal of Mathematical Imaging and Vision, 51(1):22–45.

[7] Brogat-Motte, L., Flamary, R., Brouard, C., Rousu, J., and d'Alché Buc, F. (2022). Learning to predict graphs with fused gromov-wasserstein barycenters. In International Conference on Machine Learning, pages 2321–2335. PMLR.

[8] Brouard, C., Bassé, A., d'Alché-Buc, F., and Rousu, J. (2019). Improved Small Molecule Identification through Learning Combinations of Kernel Regression Models. Metabolites, 9(8):160.

[9] Brouard, C., Shen, H., Dührkop, K., d'Alché Buc, F., Böcker, S., and Rousu, J. (2016). Fast metabolite identification with input output kernel regression. Bioinformatics, 32(12):i28–i36.

[10] Bushuiev, R., Bushuiev, A., de Jonge, N. F., Young, A., Kretschmer, F., Samusevich, R., Heirman, J., Wang, F., Zhang, L., Dührkop, K., et al. (2024). MassSpecGym: A benchmark for the discovery and identification of molecules. Advances in Neural Information Processing Systems, 37:110010–110027.

[11] Bushuiev, R., Bushuiev, A., Samusevich, R., Brungs, C., Sivic, J., and Pluskal, T. (2025). Self-supervised learning of molecular representations from millions of tandem mass spectra using DreaMS. Nature Biotechnology, pages 1–11.

[12] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR.

[13] Chen, Y. Z., Rushing, B., and Hassoun, S. (2026). FLARE: Fine-grained learning for alignment of spectra-molecule representation enhances metabolite annotation. bioRxiv, pages 2026–01.

[14] de Jonge, N., van der Hooft, J. J. J., and Probst, D. (2025). To Bin or not to Bin: Alternative Representations of Mass Spectra.

[15] de Oliveira, G. B. and Saeed, F. (2026). Moldeberta: Foundational model for physicochemical and structural-informed molecular representation learning. bioRxiv, pages 2026–02.

[16] De Vijlder, T., Valkenborg, D., Lemière, F., Romijn, E. P., Laukens, K., and Cuyckens, F. (2018). A tutorial in small molecule identification via electrospray ionization-mass spectrometry: The practical art of structural elucidation. Mass Spectrometry Reviews, 37(5):607–629.

[17] De Waele, G., Wydmuch, M., Waegeman, W., et al. (2026). Small molecule retrieval from tandem mass spectrometry: what are we optimizing for? arXiv preprint arXiv:2602.16507.

[18] Domingo-Almenara, X., Montenegro-Burke, J. R., Benton, H. P., and Siuzdak, G. (2018). Annotation: A Computational Solution for Streamlining Metabolomics Analysis. Analytical Chemistry, 90(1):480–489.

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

[20] Dührkop, K., Shen, H., Meusel, M., Rousu, J., and Böcker, S. (2015). Searching molecular structure databases with tandem mass spectra using CSI:FingerID. Proceedings of the National Academy of Sciences, 112(41):12580–12585.

[21] El Abiead, Y., Rutz, A., Zuffa, S., Amer, B., Xing, S., Brungs, C., Schmid, R., Correia, M. S. P., Caraballo-Rodriguez, A. M., Zarrinpar, A., Mannochio-Russo, H., Witting, M., Mohanty, I., Pluskal, T., Bittremieux, W., Knight, R., Patterson, A. D., van der Hooft, J. J. J., Böcker, S., Dunn, W. B., Linington, R. G., Wishart, D. S., Wolfender, J.-L., Fiehn, O., Zamboni, N., and Dorrestein, P. C. (2025). Discovery of metabolites prevails amid in-source fragmentation. Nature Metabolism, 7(3):435–437.

[22] El Ahmad, T., Brogat-Motte, L., Laforgue, P., and d'Alché Buc, F. (2024). Sketch in, sketch out: Accelerating both learning and inference for structured prediction with kernels. In International conference on artificial intelligence and statistics, pages 109–117. PMLR.

[23] Elser, D., Huber, F., and Gaquerel, E. (2023). Mass2SMILES: deep learning based fast prediction of structures and functional groups directly from high-resolution ms/ms spectra. bioRxiv, pages 2023–07.

[24] Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. (2021). A brief review of domain adaptation. Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020, pages 877–894.

[25] Flamary, R., Vincent-Cuaz, C., Courty, N., Gramfort, A., Kachaiev, O., Tran, H. Q., David, L., Bonet, C., Cassereau, N., Gnassounou, T., et al. (2024). Pot python optimal transport (version 0.9. 5), 2024. URL https://github. com/PythonOT/POT, 10.

[26] Goldman, S., Wohlwend, J., Stražar, M., Haroush, G., Xavier, R. J., and Coley, C. W. (2023). Annotating metabolite mass spectra with domain-inspired chemical formula transformers. Nature Machine Intelligence, 5(9):965–979.

[27] Gupta, V., Qiang, H., Chung, H.-H., Herbst, E., and Skinnider, M. A. (2026). Comprehensive curation and harmonization of small-molecule MS/MS libraries in Spectraverse. Analytical Chemistry, 98(5):3934–3943.

[28] Han, Y., Wang, P., Yu, K., Chen, X., and Chen, L. (2025). MS-BART: Unified modeling of mass spectra and molecules for structure elucidation. arXiv preprint arXiv:2510.20615.

[29] Heinonen, M., Shen, H., Zamboni, N., and Rousu, J. (2012). Metabolite identification and molecular fingerprint prediction through machine learning. Bioinformatics, 28(18):2333–2341.

[30] Hong, Y., Li, S., Ye, Y., and Tang, H. (2025). FIDDLE: a deep learning method for chemical formulas prediction from tandem mass spectra. Nature Communications, 16(1):11102.

[31] Huh, M., Cheung, B., Wang, T., and Isola, P. (2024). Position: The platonic representation hypothesis. In ICML, pages 20617–20642.

[32] Ji, H., Deng, H., Lu, H., and Zhang, Z. (2020). Predicting a molecular fingerprint from an electron ionization mass spectrum with deep neural networks. Analytical chemistry, 92(13):8649–8653.

[33] Kalia, A., Zhou Chen, Y., Krishnan, D., and Hassoun, S. (2025). JESTR: Joint embedding space technique for ranking candidate molecules for the annotation of untargeted metabolomics data. Bioinformatics, 41(7):btaf354.

[34] Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. (2023). PubChem 2023 update. Nucleic acids research, 51(D1):D1373–D1380.

[35] Kind, T., Tsugawa, H., Cajka, T., Ma, Y., Lai, Z., Mehta, S. S., Wohlgemuth, G., Barupal, D. K., Showalter, M. R., Arita, M., and Fiehn, O. (2018). Identification of small molecules using accurate mass MS/MS search. Mass Spectrometry Reviews, 37(4):513–532.

[36] Kishimoto, A., Kajino, H., Hirose, M., Fuchiwaki, J., Priyadarsini, I., Hamada, L., Shinohara, H., Nakano, D., and Takeda, S. (2023). Mhg-gnn: Combination of molecular hypergraph grammar with graph neural network. arXiv preprint arXiv:2309.16374.

[37] Kudriavtseva, P., Kashkinov, M., and Kertész-Farkas, A. (2021). Deep convolutional neural networks help scoring tandem mass spectrometry data in database-searching approaches. Journal of proteome research, 20(10):4708–4717.

[38] Landrum, G. et al. (2013). Rdkit documentation. Release, 1(1-79):4.

[39] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., et al. (2006). A tutorial on energy-based learning. Predicting structured data, 1(0).

[40] Li, X., Zhou Chen, Y., Kalia, A., Zhu, H., Liu, L.-p., and Hassoun, S. (2024). An ensemble spectral prediction (esp) model for metabolite annotation. Bioinformatics, 40(8):btae490.

[41] Litsa, E., Chenthamarakshan, V., Das, P., and Kavraki, L. (2021). Spec2Mol: An end-to-end deep learning framework for translating ms/ms spectra to de-novo molecules. ChemRxiv.

[42] Nguyen, D. H., Nguyen, C. H., and Mamitsuka, H. (2019). Recent advances and prospects of computational methods for metabolite identification: A review with emphasis on machine learning approaches. Briefings in Bioinformatics, 20(6):2028–2043.

[43] Nowozin, S. and Lampert, C. H. (2011). Structured prediction and learning in computer vision. Foundations and Trends in Computer Graphics and Vision, 6(3-4):3–4.

[44] Pollmann, J., Bushuiev, R., Bushuiev, A., Pluskal, T., and Huber, F. (2026). Bridging ms2 spectra and chemical space: Advances in spectral similarity, molecular retrieval, and de novo structure discovery. chemrxiv.15000536.

[45] Quinn, R. A., Melnik, A. V., Vrbanac, A., Fu, T., Patras, K. A., Christy, M. P., Bodai, Z., Belda-Ferre, P., Tripathi, A., Chung, L. K., Downes, M., Welch, R. D., Quinn, M., Humphrey, G., Panitchpakdi, M., Weldon, K. C., Aksenov, A., da Silva, R., Avila-Pacheco, J., Clish, C., Bae, S., Mallick, H., Franzosa, E. A., Lloyd-Price, J., Bussell, R., Thron, T., Nelson, A. T., Wang, M., Leszczynski, E., Vargas, F., Gauglitz, J. M., Meehan, M. J., Gentry, E., Arthur, T. D., Komor, A. C., Poulsen, O., Boland, B. S., Chang, J. T., Sandborn, W. J., Lim, M., Garg, N., Lumeng, J. C., Xavier, R. J., Kazmierczak, B. I., Jain, R., Egan, M., Rhee, K. E., Ferguson, D., Raffatellu, M., Vlamakis, H., Haddad, G. G., Siegel, D., Huttenhower, C., Mazmanian, S. K., Evans, R. M., Nizet, V., Knight, R., and Dorrestein, P. C. (2020). Global chemical effects of the microbiome include new bile-acid conjugations. Nature, 579(7797):123–129.

[46] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In ICML, pages 8748–8763.

[47] Rakhshaninejad, M., De Waele, G., Jürgens, M., and Waegeman, W. (2026). Reliable molecular retrieval from mass spectra using conformal prediction. bioRxiv, pages 2026–03.

[48] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144.

[49] Robinson, J., Chuang, C.-Y., Sra, S., and Jegelka, S. (2020). Contrastive learning with hard negative samples. arXiv preprint arXiv:2010.04592.

[50] Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. (2020). Self-supervised graph transformer on large-scale molecular data. Advances in neural information processing systems, 33:12559–12571.

[51] Roschmann, S., Krzakala, P., Mazelet, S., Bouniot, Q., and Akata, Z. (2026). SOTAlign: Semi-supervised alignment of unimodal vision and language models via optimal transport. arXiv preprint arXiv:2602.23353.

[52] Ross, J., Belgodere, B., Chenthamarakshan, V., Padhi, I., Mroueh, Y., and Das, P. (2022). Large-scale chemical language representations capture molecular structure and properties. Nature Machine Intelligence, 4(12):1256–1264.

[53] Russo, F. F., Nowatzky, Y., Jaeger, C., Parr, M. K., Benner, P., Muth, T., and Lisec, J. (2024). Machine learning methods for compound annotation in non-targeted mass spectrometry—A brief overview of fingerprinting, in silico fragmentation and de novo methods. Rapid Communications in Mass Spectrometry, 38(20):e9876.

[54] Ruttkies, C., Neumann, S., and Posch, S. (2019). Improving metfrag with statistical learning of fragment annotations. BMC bioinformatics, 20(1):376.

[55] Ruttkies, C., Schymanski, E. L., Wolf, S., Hollender, J., and Neumann, S. (2016). Metfrag relaunched: incorporating strategies beyond in silico fragmentation. Journal of cheminformatics, 8(1):3.

[56] Schymanski, E. L., Jeon, J., Gulde, R., Fenner, K., Ruff, M., Singer, H. P., and Hollender, J. (2014). Identifying Small Molecules via High Resolution Mass Spectrometry: Communicating Confidence. Environmental Science & Technology, 48(4):2097–2098.

[57] Stravs, M. A., Dührkop, K., Böcker, S., and Zamboni, N. (2022). MSNovelist: de novo structure generation from mass spectra. Nature Methods, 19(7):865–870.

[58] Thirukovalluru, R., Meng, R., Liu, Y., Su, M., Nie, P., Yavuz, S., Zhou, Y., Chen, W., Dhingra, B., et al. (2025). Breaking the batch barrier (B3) of contrastive learning via smart batch mining. arXiv preprint arXiv:2505.11293.

[59] Ucak, U. V., Ashyrmamatov, I., and Lee, J. (2023). Reconstruction of lossless molecular representations from fingerprints. Journal of cheminformatics, 15(1):26.

[60] Vaniya, A. and Fiehn, O. (2022). Revisiting CASMI: Compound ID for 500 new unknowns, using LC-MS/MS data.

[61] Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In International conference on machine learning, pages 9929–9939. PMLR.

[62] Wang, Y., Chen, X., Liu, L., and Hassoun, S. (2025). MADGEN: Mass-spec attends to de novo molecular generation. arXiv preprint arXiv:2501.01950.

[63] Wishart, D. S. (2019). Metabolomics for investigating physiological and pathophysiological processes. Physiological Reviews, 99(4):1819–1875.

[64] Xing, S., Shen, S., Xu, B., Li, X., and Huan, T. (2023). BUDDY: Molecular formula discovery via bottom-up MS/MS interrogation. Nature Methods, 20(6):881–890.

[65] Xu, R. and Zhu, J. (2025). Unveiling the dark matter of the metabolome: A narrative review of bioinformatics tools for LC-HRMS-based compound annotation. Talanta, 295:128327.

[66] Zhang, L., Yang, Q., and Agrawal, A. (2025). Assessing and learning alignment of unimodal vision and language models. In CVPR, pages 14604–14614.

[67] Zhou Chen, Y. and Hassoun, S. (2026). Learning from all views: A multiview contrastive framework for metabolite annotation. Analytical Chemistry, 98(9):6598–6606.

A Comparison of in-batch and candidate-based InfoNCE

A.1 In-batch InfoNCE is decorrelated from accuracy

In-batch InfoNCE (4) leverages the other samples in the batch as negatives. We argue that, in a molecular setting, this strategy yields weak negatives, i.e. negatives that are easy to distinguish from the target. Consequently, a weak scorer can achieve a low in-batch infoNCE while performing poorly on the retrieval accuracy.

Formally, for any score $\rho : \mathcal{S} \times \mathcal{M} \mapsto \mathbb{R}_+$ and batch of pairs $\mathcal{B} = (s_i, m_i)_{i=1}^B$, we define the *in-batch* InfoNCE as,

$$\mathcal{L}_{\text{batch}}(\rho) \equiv -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\rho(s_i, m_i))}{\sum_{j=1}^{B} \exp(\rho(s_i, m_j))}. \tag{12}$$

and the top-1 accuracy as,

$$\text{R@1}(\rho) \equiv \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}\left\{ \underset{m' \in C_K(s_i)}{\arg\max}\ \rho(s_i, m') = m_i \right\}, \tag{13}$$

and we claim that (12) and (13) can become surprisingly decorrelated.

This can be illustrated by introducing a handcrafted score, defined solely from the masses of the target and candidate:

$$\rho_{\lambda,\epsilon}(s, m) \equiv \lambda\ \mathbf{1}[\text{abs}(|m'| - |m_s|) < \epsilon] \tag{14}$$

where $|m_s|$ is the mass of target, which can be computed exactly if we assume access to the target's formula as metadata of the spectra s which is the case for FLARE [13] for instance.

In practice, all candidates sets feature at least one candidate with the exact same mass, and thus the accuracy of this score is always zero,

$$\text{R@1}(\rho_{\lambda,\epsilon}) = 0. \tag{15}$$

Despite this catastrophic performance (below a random guess), we can show both empirically and theoretically that the *in-batch* InfoNCE of the same scorer can go to zero.

Empirical result. Empirically, we set $\epsilon(\lambda) = \frac{1}{\lambda}$, and report the expected loss

$$\mathcal{L}(\lambda) = E_{\mathcal{B}}[\mathcal{L}_{\text{batch}}(\rho_{\lambda,\epsilon(\lambda)})], \tag{16}$$

on MassSpecGym and Spectraverse dataset at batchsize $B = 32$ (Figure 5). In both case, we see that simply increasing λ decrease the in-batch InfoNCE loss by several order of magnitude, while the accuracy remains null.

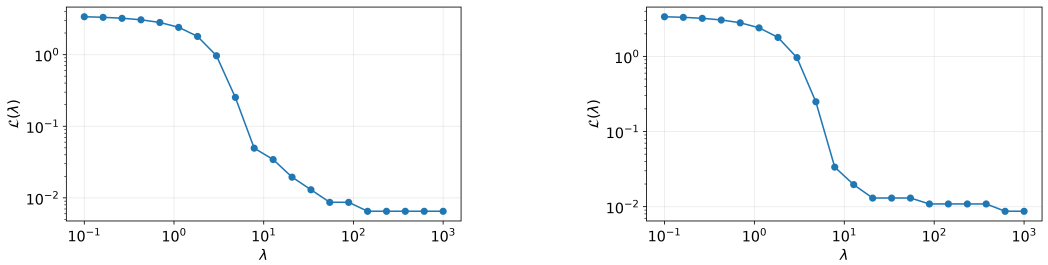


Figure 5: Expected in-batch InfoNCE loss (16) for MassSpecGym (Left) and SpectraVerse (Right). In both cases, the trivial scorer can achieve a loss below 10^{-2} while achieving 0% top-1 accuracy.

Theoretical observation. More generally, we can demonstrate that the trivial score $\rho_{\lambda,\epsilon}$ can always achieve low loss values and that this issue worsen when dataset size increase.

Proposition 1 (Upper bound of the in-batch loss)**.** *Introduce the expected loss,*

$$\mathcal{L}(\lambda, \epsilon) = E_{\mathcal{B}}[\mathcal{L}_{batch}(\rho_{\lambda,\epsilon})], \tag{17}$$

For any dataset we have,

$$\mathcal{L}(\lambda, \epsilon) \leq \log\left(1 + (B-1)(e^{-\lambda}(1-p_\epsilon) + p_\epsilon)\right) \tag{18}$$

where p_ϵ is the probability for two random samples to have the same mass up to ϵ. In particular, if the molecules in the dataset can take N_m different and equiprobable masses, we have

$$\mathcal{L}(+\infty, 0) \leq \log\left(1 + \frac{B-1}{N_m - 1}\right). \tag{19}$$

Thus, by construction, the loss goes to zero when N_m increases.

Proof. For a given batch we have,

$$\mathcal{L}_{\text{batch}}(\rho_{\lambda,\epsilon}) = \frac{1}{B}\sum_{i=1}^{N} \log\left(1 + \sum_{j\neq i} \exp(\rho_{\lambda,\epsilon}(s_i, m_j) - \rho_{\lambda,\epsilon}(s_i, m_i))\right)$$

Where $\rho_{\lambda,\epsilon}(s_i, m_j) = \lambda\delta_{i,j}$ for

$$\delta_{i,j} = \mathbf{1}[\text{abs}(|m_i| - |m_j|) < \epsilon].$$

This rewrites as,

$$\mathcal{L}_{\text{batch}}(\rho_{\lambda,\epsilon}) = \frac{1}{B}\sum_{i=1}^{N} \log\left(1 + \sum_{j\neq i} \exp(-\lambda)\exp(\lambda\delta_{i,j})\right)$$

Applying Jensen inequality we have,

$$\begin{aligned}
E_{\mathcal{B}}[\mathcal{L}_{\text{batch}}(\rho_{\lambda,\epsilon})] &\leq \frac{1}{B}\sum_{i=1}^{N} \log\left(1 + \sum_{j\neq i} \exp(-\lambda) E_{\mathcal{B}}[\exp(\lambda\delta_{i,j})]\right) \\
&\leq \frac{1}{B}\sum_{i=1}^{N} \log\left(1 + \sum_{j\neq i} (B-1)(p_\epsilon + e^{-\lambda}(1-p_\epsilon)]\right) \\
&\leq \log\left(1 + \sum_{j\neq i} (B-1)(p_\epsilon + e^{-\lambda}(1-p_\epsilon)]\right).
\end{aligned}$$

In particular $\mathcal{L}(+\infty, 0) = \log(1 + (B-1)p_\epsilon)$. Finally, in the hypothesis of a dataset with N_m equiprobable masses, we have $p_\epsilon = \frac{1}{N_m - 1}$ which concludes the proof. □

Thus, increasing the number of unique molecular masses in the dataset is sufficient to guarantee that the in-batch InfoNCE loss of the trivial scorer goes to zero, even though the accuracy is still null.

We envision three options to prevent this behavior:

- Option 1: increase the batchsize B. This is the simplest approach to make in-batch InfoNCE more informative, and in was successfully applied in language–vision alignment [51, 66]. We tested this approach (with limtied success) in the experimental section, see Figure 3**b**.
- Option 2: hard negative mining [49]. This is the approach we implement, using the candidate set as a cheap source of hard negatives.
- Option 3: batch sampling can be modified such that molecules that are similar are more likely to be sampled in the same batch. This approach is left for future works.

A.2 Candidate-based InfoNCE is a valid surrogate loss for accuracy

We first establish a scalar softmax bound, then apply it to the temperature-scaled candidate scores.

Lemma 1. *For $x^+, x_1^-, \dots, x_J^- \in \mathbb{R}$, with $J \geq 1$, define*

$$\ell\big(x^+, \{x_k^-\}_{k=1}^J\big) = -\log \frac{\exp(x^+)}{\exp(x^+) + \sum_{k=1}^J \exp(x_k^-)},$$
$$\mathrm{err}\left(x^+, \{x_k^-\}_{k=1}^J\right) = \mathbf{1}\left\{\max_{1\leq k\leq J} x_k^- > x^+\right\}.$$

Then

$$\mathrm{err}\left(x^+, \{x_k^-\}_{k=1}^J\right) \leq \frac{\ell\big(x^+, \{x_k^-\}_{k=1}^J\big)}{\log 2}. \tag{20}$$

Proof. Let $\Delta = \max_k x_k^- - x^+$. If $\Delta \leq 0$, then $\mathrm{err} = 0$ and the inequality follows from $\ell \geq 0$. Otherwise, $\mathrm{err} = 1$ and

$$\ell = \log\left(1 + \sum_{k=1}^J \exp(x_k^- - x^+)\right) \geq \log(1 + \exp(\Delta)) > \log 2.$$

This proves the strict-error bound. To obtain its non-strict version, observe that $\Delta = 0$ also gives $\ell \geq \log 2$. Therefore,

$$\mathbf{1}\left\{\max_k x_k^- \geq x^+\right\} \leq \frac{\ell\big(x^+, \{x_k^-\}_{k=1}^J\big)}{\log 2}. \tag{21}$$

□

Proposition 2. *Define the empirical top-1 retrieval accuracy as*

$$\mathrm{R@1}(\theta) = \frac{1}{B}\sum_{i=1}^B \mathbf{1}\left\{\operatorname*{arg\,max}_{m'\in C_K(s_i)} \rho_\theta(s_i, m') = m_i\right\}.$$

The candidate-based InfoNCE loss satisfies

$$\boxed{1 - \mathrm{R@1}(\theta) \leq \frac{\mathcal{L}_{\mathrm{cand}}(\theta)}{\log 2}.}$$

Proof. Apply Lemma 1 with $x^+ = \rho_\theta(s_i, m_i)/\epsilon$ and negative scores $x_k^- = \rho_\theta(s_i, m')/\epsilon$ for $m' \in C_K(s_i) \setminus \{m_i\}$. Since $\epsilon > 0$, the ranking is unchanged and the scalar loss equals $\ell_i(\theta)$. Let $\widehat{m}_i = \arg\max_{m'\in C_K(s_i)} \rho_\theta(s_i, m')$ denote the predicted molecule. An incorrect prediction implies that a negative scores at least as high as the target. Hence, by (21),

$$\mathbf{1}\{\widehat{m}_i \neq m_i\} \leq \mathbf{1}\left\{\max_{m'\in C_K(s_i)\setminus\{m_i\}} \rho_\theta(s_i, m') \geq \rho_\theta(s_i, m_i)\right\} \leq \frac{\ell_i(\theta)}{\log 2}.$$

Averaging over queries gives

$$1 - \mathrm{R@1}(\theta) \leq \frac{1}{B\log 2}\sum_{i=1}^B \ell_i(\theta) = \frac{\mathcal{L}_{\mathrm{cand}}(\theta)}{\log 2}.$$

□

B Implementation Details

B.1 Pre-processing and training.

Table 7: Dataset statistics.

	MassSpecGym	Spectraverse
# Mol/MS pairs	231,104	488,797
# Unique molecules	28,936	44,307
Non [M+H]+	15%	52%

Molecule pre-processing. All SMILES strings are canonicalized and sanitized using RDKit [38]. Chemical formulas and weights are computed by explicitly accounting for implicit hydrogen atoms. The molecular masses are computed with RDKit ExactMolWt function which uses the monoisotopic atomic mass (mass of the most abundant isotope). All baselines are trained using the hyperparameters reported in their respective original works.

Model training. MSAlign is trained on a single NVIDIA V100 GPU, using the AdamW optimizer with linear warmup followed by cosine decay. Following common practice, we augment the contrastive losses with a temperature parameter ϵ, for instance $\mathcal{L}_{\text{batch}}$ writes as:

$$\mathcal{L}_{\text{batch}}(\theta) = -\sum_{i=1}^{B} \log \frac{\exp(\rho_\theta(s_i, m_i)/\epsilon)}{\sum_{j=1}^{B} \exp(\rho_\theta(s_i, m_j)/\epsilon)}. \tag{22}$$

and ϵ is a trainable parameters parametrized in log scale and initialized to 0.07.

Table 8: Main MSAlign hyperparameters.

Hyperparameter	**MassSpecGym MCES**	**MassSpecGym Formula**	**SpectraVerse Formula**
MS representation	DreaMS	DreaMS	DreaMS
Molecular representation	MolDeBERTa	MolDeBERTa	MolDeBERTa
Shared dimension d_{model}	512	512	512
Metadata dimension d_{meta}	64	64	64
MS MLP hidden layers	1	2	2
Molecule MLP hidden layers	3	3	3
MLP hidden dimension	2048	2048	2048
Activation / normalization	GELU / LayerNorm	GELU / LayerNorm	GELU / LayerNorm
MLP dropout	0.3	0.3	0.2
Loss Score temperature	Learned (0.07 init.)	Learned (0.07 init.)	Learned (0.07 init.)
Candidates per query	128	128	128
Batch size	128	128	128
Learning rate	10^{-4}	10^{-4}	10^{-4}
Weight decay	10^{-6}	10^{-6}	10^{-6}
Warmup steps	2,000	2,000	2,000
Maximum training steps	20,000	30,000	50,000

B.2 Late score fusion.

Smooth surrogate. The validation top-1 accuracy is defined as

$$\text{R@1}_{\text{val}}(\boldsymbol{\alpha}) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(s_i, m_i) \in \mathcal{D}_{\text{val}}} \mathbf{1}\left\{ \arg\max_{m' \in C(s_i)} \rho_{\boldsymbol{\alpha}}(s_i, m') = m_i \right\}.$$

To approximate this objective, we maximize a concave log-softmax surrogate over the simplex for each fixed fusion temperature $T > 0$:

$$\boldsymbol{\alpha}_T \in \arg\max_{\boldsymbol{\alpha} \in \Delta^{L-1}} \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(s_i, m_i) \in \mathcal{D}_{\text{val}}} \log \frac{\exp(\rho_{\boldsymbol{\alpha}}(s_i, m_i)/T)}{\sum_{m' \in C(s_i)} \exp(\rho_{\boldsymbol{\alpha}}(s_i, m')/T)}. \tag{23}$$

This is a convex optimization problem. Here, T is distinct from the learned training temperature ϵ. The component scores remain fixed, and each full candidate set $C(s_i)$ includes the target m_i.

Temperature selection. The surrogate need not share the maximizer of R@1. We therefore fit weights for each temperature in a finite grid $\mathcal{T} \subset (0, \infty)$ and select them using the original validation metric:

$$T^\star \in \arg\max_{T \in \mathcal{T}} \mathrm{R@1}_{\mathrm{val}}(\boldsymbol{\alpha}_T), \qquad \widehat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_{T^\star}. \tag{24}$$

Both stages use validation data; the test split is reserved for evaluation. This procedure approximates the direct R@1 objective without guaranteeing its global optimum. Component scores are cached once and reused across temperatures, so fitting requires no further model evaluations. Algorithm 1 summarizes the procedure. At inference, rank candidates by $\rho_{\widehat{\boldsymbol{\alpha}}}$. The fusion temperature is used only to fit the weights.

Algorithm 1: Late fusion with temperature selection

Input: Fixed score functions $\rho_1, \ldots, \rho_L$; validation split $\mathcal{D}_{\mathrm{val}}$; full candidate sets $C(s_i)$ containing m_i; fusion-temperature grid $\mathcal{T} \subset (0, \infty)$.
Output: Fusion weights $\widehat{\boldsymbol{\alpha}} \in \Delta^{L-1}$.

1. Cache $S_{i\ell}(m') \leftarrow \rho_\ell(s_i, m')$ for every validation pair (s_i, m_i), candidate $m' \in C(s_i)$, and component ℓ.
2. For each $T \in \mathcal{T}$:
   (a) Solve (23) for $\boldsymbol{\alpha}_T$ using the cache and a convex solver over the simplex. Evaluate the objective with a numerically stable log-sum-exp.
   (b) Compute $r_T \leftarrow \mathrm{R@1}_{\mathrm{val}}(\boldsymbol{\alpha}_T)$ from the cached weighted scores, using the fixed tie-breaking rule.
3. Select $T^\star \in \arg\max_{T \in \mathcal{T}} r_T$ and return $\widehat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_{T^\star}$.

B.3 MSAlign architecture

MLPs. The main workhorse of the proposed alignement models are Multi Layers Perceptrons (MLPs). In this work, a MLP with L hidden layers applies an affine map, layer normalization, GELU, and dropout in each hidden block, followed by an affine output projection:

$$h_0 = x, \qquad h_\ell = \mathrm{Dropout}(\mathrm{GELU}(\mathrm{LayerNorm}(W_\ell h_{\ell-1} + b_\ell))), \qquad \mathrm{MLP}_L(x) = W_{\mathrm{out}} h_L + b_{\mathrm{out}}.$$

Adduct encoding. Query spectra often contains meta-data such as the ionization mode which is a presented as a discrete adduct class $a \in \{1, ..., N_a\}$. The adduct encoder concatenates a learned class embedding with a learned polarity embedding, allocating half of its 64-dimensional output to each:

$$\phi_a(a) = \left[E_{\mathrm{class}}(a)\,;\, E_{\mathrm{sign}}(\sigma(a))\right] \in \mathbb{R}^{64}.$$

Unknown or dropped adducts use a dedicated unknown class and unknown-sign embedding.

Collision energy. Similarly, the collision energy uses a fixed 64-dimensional Fourier encoding, with 32 sine–cosine pairs applied to the scalar normalized by $e_{\max} = 100$:

$$\phi_e(e)_{2j:2j+2} = \left[\sin\left(\frac{e/e_{\max}}{b^{j/32}}\right), \cos\left(\frac{e/e_{\max}}{b^{j/32}}\right)\right], \qquad j = 0, \ldots, 31, \quad b = 10^4.$$

A missing energy value is represented by a separate learned vector.

Full architecture. The spectrum representation passes through a linear input projection, is concatenated with both metadata encodings, and passes through an output MLP; the molecular representation has its own MLP. Both towers produce 512-dimensional vectors:

$$z_s = \mathrm{MLP}_{\mathrm{ms,out}}\left(\left[\,\mathrm{Linear}_{\mathrm{ms,in}}(s)\,;\,\phi_e(e)\,;\,\phi_a(a)\right]\right), \qquad z_m = \mathrm{MLP}_{\mathrm{mol}}(m).$$

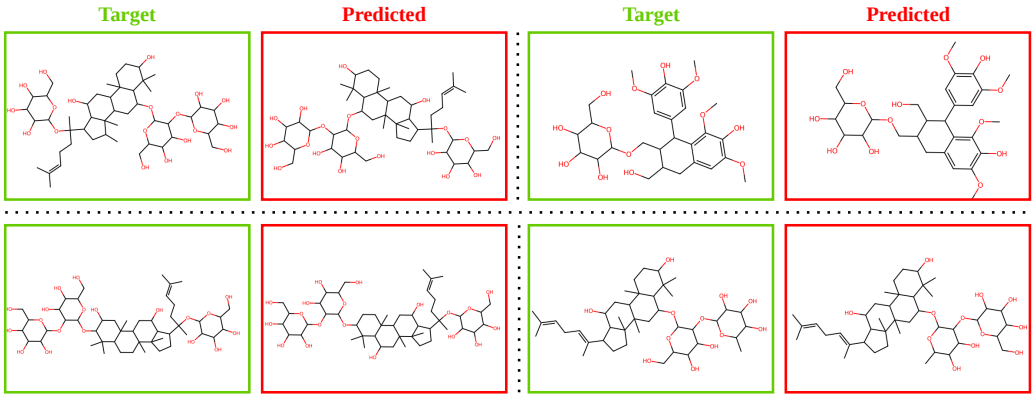


Figure 6: Four MSAlign failures cases, in each case we report both the prediction (in red) and the target (in green). The displayed failure cases are randomly sampled from the 5% hardest samples, in the sense that the Tanimoto distance between the target and closest candidate is the smallest.

C Additional Results

C.1 MSAlign failure cases

We observe that MSAlign failures are often due to a particularly challenging set of candidates i.e. when at least one of the candidate is particularly similar to the target.

To quantify this phenomenon, we propose to define the difficulty of a sample $(s, m) \in \mathcal{S} \times \mathcal{M}$ with candidate set $C(s) \subset \mathcal{M}$ as:

$$\text{Difficulty}(s) = \min_{c \in C(s)} d(m, c) \tag{25}$$

where d is distance over the set of molecules such as the (negative) MCES or Tanimoto similarity. Next we propose to restrict the test dataset the top-K% of most difficult samples and compute the accuracy on this subset. Results are reported in Table 9 and we observe a correlation between the difficulty of the sample as defined in (25) and MSAlign performances.

Table 9: MSAlign R@1 accuracy on increasingly difficult subsets of MassSpecGym Formula-split test set. The subsets are defined by selecting the top-K% of most difficult samples as defined by (25). We report the results both for MCES and Tanimoto metrics.

K	100%	80%	60%	40%	20%	10%	5%
MCES	47.52 ± 3.39	46.25 ± 4.10	45.83 ± 5.68	43.45 ± 7.57	35.26 ± 5.67	22.34 ± 6.76	21.96 ± 14.88
Tanimoto	47.52 ± 3.39	46.06 ± 4.18	42.93 ± 4.34	37.75 ± 8.30	29.31 ± 5.33	24.02 ± 9.90	25.17 ± 17.33

C.2 Methods training time comparison.

Table 10: Training time in hours.min for the different methods on a single NVIDIA V100 GPU.

Dataset	**DeepSet**	**JESTR**	**Emb-Cos**	**MIST**	**MVP**	**FLARE**	**MSAlign**	**$\overline{\text{MSAlign}}$**
MassSpecGym Formula	0.65	7.34	4.45	2.30	16.53	2.50	0.82	3.29
MassSpecGym MCES	0.76	9.18	4.58	1.05	15.00	2.22	0.75	3.02
SpectraVerse Formula	1.43	13.32	4.56	3.1.	25.00	4.17	1.31	5.71

We report the training time of the different methods in Table 10. Note that the training time reported for MSAlign* includes the training of the 4 independent alignment models (DreaMS/MolDeBERTa, DreaMS/Fingerprint...) and can be largely parallelized. The pre-computation of MolDeBERTa and DreaMS embedding takes about 5 hours but it needs to be run only once for the full project and is amortized across runs, seeds and models. Note that JESTR, FLARE and MVP are particularly slow despite using the cheaper candidate-based InfoNCE. This is due to the large number of epochs (1000 for JESTR) recommended in the original papers [13, 33, 67].

C.3 Additionnal results on score fusion.

Scorers ablation. In Table 11 we report the performances of each individual scorer used in MSAlign and different combinations of scorers aggregations. In particular, we observe that MSAlign (DreaMS/MolDeBERTa model) is the best performing of all the individual scorers while on the contrary the 1-dimensional mass scorers achieve very poor performances on their own.

Table 11: Individual scorers and calibrated scorer combinations. B: binned spectra; D: DreaMS; FP: Morgan fingerprint; MD: MolDeBERTa. Mass columns give the Gaussian scorer scale in ppm. R@1 values are mean ± sample standard deviation over the three MassSpecGym formula test splits.

Configuration	B/FP	B/MD	D/FP	D/MD	Mass .01	Mass .1	Mass 1	R@1
Alternative Modality Pair	✓	×	×	×	×	×	×	34.3 ± 1.3
	×	✓	×	×	×	×	×	44.5 ± 5.1
	×	×	✓	×	×	×	×	38.6 ± 2.8
MSAlign	×	×	×	✓	×	×	×	47.5 ± 3.4
MSAlign + Other modalities	✓	✓	✓	✓	×	×	×	<u>54.4</u> ± 3.0
Gaussian Mass Scorer	×	×	×	×	✓	×	×	0.6 ± 0.1
	×	×	×	×	×	✓	×	0.8 ± 0.2
	×	×	×	×	×	×	✓	0.7 ± 0.2
MSAlign + Mass	×	×	×	✓	✓	✓	✓	51.3 ± 3.4
MSAlign^{4M}	✓	✓	✓	✓	✓	✓	✓	**58.2** ± 3.4

Scorer failure-case overlaps. In Figure 3c, we reported the pairwise Jaccard index between the failure sets of the individual scorers, defined for a scorer ρ, query spectrum s, and target molecule m as

$$C_\rho(s, m) = \{m' \in C(s) \mid \rho(s, m') > \rho(s, m)\}. \tag{26}$$

In other words, $C_\rho(s, m)$ contains all candidate molecules incorrectly ranked above the target by ρ.

While the Jaccard index provides a symmetric measure of overlap, it can be difficult to interpret when the failure sets have substantially different sizes. A small Jaccard index may reflect genuinely complementary errors, but can also result from one failure set being much larger than the other. To obtain a more fine-grained picture, we therefore consider the following asymmetric overlap:

$$J'(\rho, \rho') = \frac{1}{N} \sum_{i=1}^{N} \frac{|C_\rho(s_i, m_i) \cap C_{\rho'}(s_i, m_i)|}{|C_\rho(s_i, m_i)|}.$$

$J'(\rho, \rho')$ can be interpreted as the average fraction of failure cases of ρ that also fool ρ'. Unlike the Jaccard index, it is asymmetric: $J'(\rho, \rho')$ and $J'(\rho', \rho)$ need not coincide. All pairwise values are reported in Table 7.

For completeness, we also report the average size of $C_\rho(s_i, m_i)$ for each scorer. Interestingly, these failure-set sizes exhibit a relatively heavy-tailed distribution: a scorer may correctly rank the target first for a large fraction of queries while ranking it below many candidates on a smaller number of difficult queries. For instance, for MSAlign (DreaMS–MolDeBERTa), we obtain

$$\text{R@1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[|C_\rho(s_i, m_i)|) = 0] = 47.5\% \tag{27}$$

$$\text{Average} = \frac{1}{N} \sum_{i=1}^{N} |C_\rho(s_i, m_i)| = 9.7 \tag{28}$$

Thus, despite ranking the correct molecule first for nearly half of the queries, MSAlign has an average of 9.7 candidates ranked above the target, illustrating the strong heterogeneity of failure severity across queries which also contributes to the efficiency of model aggregation.

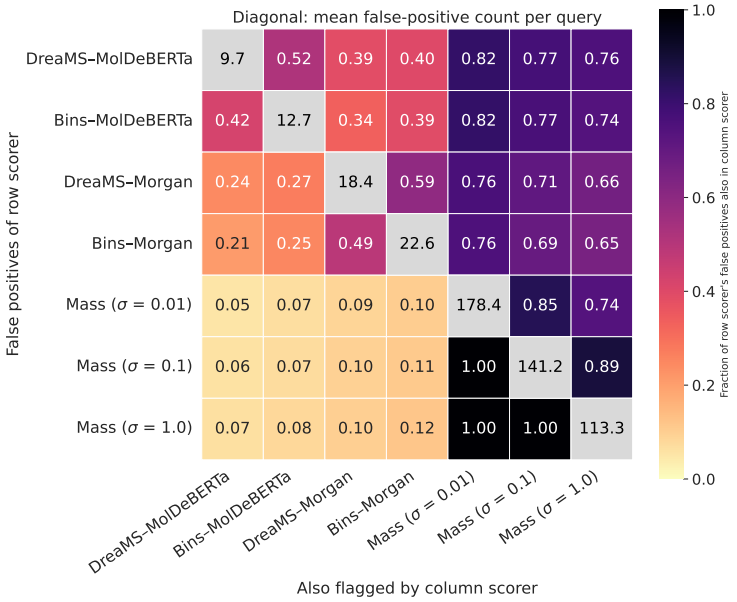


Figure 7: Average fraction of candidates incorrectly ranked by ρ (rows) that also fool scorer ρ' (column). The diagonal reports the average number of incorrectly ranked candidates for each individual scorer. MSAlign is the first scorer (DreaMS-MolDeBERTa) and $\overline{\text{MSAlign}}$ is the aggregate of all scorers via late score fusion.

D Existing criterion for ms/ms dataset data splitting.

Data splitting requires particular care in MS/MS benchmarks, as the same molecule may be associated with multiple spectra acquired under different experimental conditions. A naive random split can therefore place spectra of the same, or highly similar, molecules in both training and test sets, leading to data leakage and overly optimistic performance estimates. To mitigate this issue, several splitting strategies have been proposed, differing in the notion and degree of molecular similarity that they enforce across folds:

- **InChIKey split.** The InChIKey provides a unique identifier for a molecule's 2D structure. Splitting by InChIKey ensures that the same molecular structure does not appear in different folds [26].
- **Formula split.** Similarly, molecules are grouped according to their molecular formula, ensuring that molecules with the same formula do not appear in different folds. This criterion is strictly more restrictive than the InChIKey split: molecules with the same InChIKey necessarily have the same molecular formula, whereas distinct molecular structures may share the same formula.
- **Murcko scaffold split.** Molecules are grouped according to their Murcko scaffold [38], ensuring that molecules sharing the same underlying scaffold do not appear in different folds.
- **Murcko histogram split.** DreaMS [11] introduces a stronger variant of the Murcko scaffold split, in which even a coarse histogram representation of the Murcko scaffold must differ across folds.
- **MCES split.** MassSpecGym [10] uses the Maximum Common Edge Subgraph (MCES) distance to explicitly constrain structural similarity across folds, requiring an edit distance of at least 10 between molecules assigned to different folds.

These strategies do not form a single hierarchy of increasingly restrictive splits. The InChIKey split is strictly less restrictive than all the alternatives above, while the Murcko histogram split is strictly more restrictive than the standard Murcko scaffold split. Beyond these relations, the criteria are generally incomparable: for instance, separating molecules by molecular formula, scaffold, or MCES distance captures different notions of structural similarity. Consequently, there is no obvious way to rank these strategies by their strength, making it difficult to make a principled choice between them.

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: Yes all claims are justified and in particular we provide detailed ablations studies.

   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: Yes we explcitly state that our works relies on the quality of the underlying foundation models and that the proposed approach do not apply to de novo structure elucidation.

   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: [Yes].

Yes, the proof are provided and correct.

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: Yes, we provided a dedicated "implementation details" section in the appendix, which includes all the necessary information to reproduce our results. Moreover the code, dataset splits and pretrained checkpoints will be released upon publication.

   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: Yes, we will release the code, dataset splits and pretrained checkpoints upon publication.

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: Yes. In particular section 4 carefully explore the question of data splitting.

   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]

   Yes, we applicable, we report the standard deviation over different random splitting of the datasets.

   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: As explained in Appendix B all experiment are conducted on a single V100 GPU and the execution time are also reported.

   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: Yes, the research conducted in the paper conforms to the Code of Ethics.

   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: [N/A]

    Justification: Metabolite Elucidation is a fundamental problem in metabolomics, which is a key technology for understanding biological systems and has applications in health, agriculture, and environmental science. The proposed method, MSAlign, has the potential to significantly improve the accuracy of metabolite identification from mass spectrometry data, which could accelerate research in these fields and lead to new discoveries. We do not identify any direct negative societal impacts of this work.

    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: There is no risk for misuse of the data or models released in this paper, so we do not have any safeguards to put in place.

    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: The authors have properly credited the creators of both models, benchmarks and datasets mentioned in the paper.

    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: Upon publication, we will release the code, dataset splits and pretrained checkpoints. The code will be accompanied by documentation to facilitate its use by other researchers.

    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: We did not conduct any crowdsourcing experiments nor research with 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: We did not conduct any research with 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: The core method development in this research does not involve LLMs beyond formating the paper and standard code auto-completion, which do not impact the core methodology, scientific rigor, or originality of the research.

    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.