BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 02/25 pp577–589
DOI: 10.26599/BDMA.2024.9020006
Volume 7, Number 3, September 2024

DMSS: An Attention-Based Deep Learning Model for High-Quality Mass Spectrometry Prediction

Yihui Ren†, Yu Wang†, Wenkai Han, Yikang Huang, Xiaoyang Hou, Chunming Zhang,
Dongbo Bu, Xin Gao*, and Shiwei Sun*

Abstract: Accurate prediction of peptide spectra is crucial for improving the efficiency and reliability of proteomic analysis, as well as for gaining insight into various biological processes. In this study, we introduce Deep MS Simulator (DMSS), a novel attention-based model tailored for forecasting theoretical spectra in mass spectrometry. DMSS has undergone rigorous validation through a series of experiments, consistently demonstrating superior performance compared to current methods in forecasting theoretical spectra. The superior ability of DMSS to distinguish extremely similar peptides highlights the potential application of incorporating our predicted intensity information into mass spectrometry search engines to enhance the accuracy of protein identification. These findings contribute to the advancement of proteomics analysis and highlight the potential of the DMSS as a valuable tool in the field.

Key words: mass spectrometry; proteomics; machine learning; deep learning

1 Introduction

Mass spectrometry is a powerful analytical tool used in the field of proteomics to identify and characterize proteins[1, 2]. Over recent years, various methods have emerged for peptide identification based on mass spectrometry data[3, 4]. Predicting peptide spectra is a crucial facet of mass spectrometric analysis, facilitating the identification and quantification of peptides in intricate biological samples. Accurate prediction of peptide spectra is crucial for enhancing the efficiency and reliability of protein identification and for gaining insights into various biological processes. The prediction of theoretical spectra involves generating the expected fragmentation pattern for a given peptide sequence based on its amino acid composition and known fragmentation rules[5]. By comparing the experimental spectrum with the theoretical spectrum, researchers can identify the peptide sequence that best matches the observed fragmentation pattern.

• Yihui Ren, Xiaoyang Hou, Dongbo Bu, and Shiwei Sun are with Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and with University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: renyihui17@mails.ucas.ac.cn; houxiaoyang22s@ict.ac.cn; dbu@ict.ac.cn; dwsun@ict.ac.cn.
• Yu Wang is with Syneron Technology, Guangzhou 510000, China. E-mail: yu.wang@synerontech.com.
• Yikang Huang is with College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China. E-mail: 2020307140312@cau.edu.cn.
• Wenkai Han and Xin Gao are with Computer Science Program, Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia. E-mail: wenkai.han@kaust.edu.sa; xin.gao@kaust.edu.sa.
• Chunming Zhang is with Insitute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and with Western Institute of Computing Technology, Chongqing 400000, China. E-mail: zhangchunming@ncic.ac.cn.
† Yihui Ren and Yu Wang contribute equally to this work.
∗ To whom correspondence should be addressed.

Manuscript received: 2023-11-07; revised: 2023-12-25; accepted: 2024-01-29

The prediction of theoretical spectra is an important technique in peptide identification, regardless of whether the data acquisition method used is Data-Dependent Acquisition (DDA)[6, 7] or Data-Independent Acquisition (DIA)[8].

In DDA, mass spectrometry is used to select and fragment precursor ions based on their intensity. The resulting tandem mass spectra are then compared with the theoretical spectra generated from a peptide sequence database[9], which uses search engines[10−12]. The prediction of theoretical spectra helps to match the observed experimental spectra with the expected fragmentation pattern of the peptides in the database to find the best Peptide Spectrum Match (PSM). This aids in the identification of the peptides present in the sample.

Similarly, in DIA, mass spectrometry is used to acquire fragment ion spectra for all ions within a specific m/z range in a single experiment. The acquired spectra are then analyzed by comparing them to theoretical spectra from a peptide sequence database. The prediction of theoretical spectra assists in matching the observed fragment ion spectra with the expected fragmentation patterns, enabling the identification of peptides in the sample.

Using theoretical spectra prediction, researchers can improve the accuracy and efficiency of peptide identification. It helps in reducing false-positive identifications by filtering out non-matching spectra and focusing on the most relevant matches. Additionally, it enables the identification of post-translational modifications and other sequence variations that may be present in the peptides.

Overall, theoretical spectra prediction plays a crucial role in both DDA and DIA methods, aiding in the accurate identification of peptides and facilitating proteomic analysis.

Over the years, several computational methods have been developed to predict peptide spectra, commonly known as theoretical spectra, including those based on physical as well as machine learning models.

The most commonly employed physical model is the proton transfer model, on which MassAnalyzer[13] and MS-Simulator[14] are based. MassAnalyzer relies on a molecular dynamics model, considering multiple competing fragmentation pathways and employs molecular dynamics simulations to calculate the theoretical intensity of peptide fragment ions. On the other hand, MS-Simulator simplifies the proton transfer model, taking into account factors, such as the protonation probability of amino acids, proton distribution, amino acid energy levels, as well as the influences of amino acids at the fragmentation site and the peptide sequence. It utilizes a model with numerous parameters to predict the intensity of theoretical spectrum y-ion fragments.

Dating back two decades, the earliest machine learning based methods for predicting theoretical spectra involved the use of decision trees[15] and single-layer neural networks[16]. The advancement in deep learning[17] has revolutionized the field of theoretical spectra prediction[18, 19]. Recurrent Neural Networks (RNN), such as Long Short-Term Memory (LSTM)[20] and Gate Recurrent Unit (GRU)[21], are well suited to modeling peptide sequences and predict their fragmentation spectra due to their inherent abiliby to handle variable-length sequences[22], which are highly appropriate for modeling sequential data. Among all these methods, the most representative ones are pDeep[23, 24] and Prosit[25]. pDeep employs a bidirectional LSTM architecture for theoretical spectrum prediction. The model’s structure takes one-hot vector representations of peptide sequences as inputs. It then processes these inputs through two layers of bidirectional LSTMs to predict fragment ion intensities. The median Pearson Correlation Coefficient (PCC) for theoretical spectra predicted by pDeep often exceeds 0.90. However, this model still has certain limitations, including that the accuracy of predictions is not sufficiently high, particularly when precursors are charged singly or quadruply.

Unlike pDeep, which is made up of BiLSTM models, Prosit, proposed by Gessulat et al.[25], utilizes an encoder-decoder architecture. The encoder component encodes sequence information using a bidirectional GRU model, while precursor ion charge state and Normalized Collision Energy (NCE) are encoded as metadata through a feedforward neural network. Both the sequence encoding and metadata encoding are combined to create the latent space for data prediction. Afterward, both encodings are inputted into the decoder, constructed using bidirectional GRU, to predict retention time index and fragment ion intensities.

Transformer model[26], underpinned by the fundamental concept of attention, has emerged as a ubiquitous deep learning architecture in recent years. It has found widespread applications in various domains such as computer vision[27, 28] and natural language process[29–32], and even extends its utility into the realm of theoretical spectrum prediction[33, 34].

In the MS-Simulator, based on the proton mobility model, certain assumptions have been made. Firstly, it is assumed that the intensity of y-ions is directly proportional to the probability of protonation at the terminal site of the ion fragment, and the distribution of protons along the peptide follows the Boltzmann distribution. Secondly, the energy levels of protonated microstates of residues are influenced by other amino acids, with the effects being dependent on the distance between the amino acids. These assumptions, combined with the attention mechanism of Transformer, have sparked the development of our model, an attention-based model named Deep MS Simulator (DMSS).

As illustrated in Fig. 1a, DMSS initiates its operations by encoding the amino acid sequence through a one-hot encoding scheme. In this scheme, each amino acid is represented as a binary vector, where every element is zero except for the element corresponding to the specific amino acid, which is set to one. This encoding method enables the model to effectively distinguish between different amino acids. Subsequently, the encoded amino acid sequences traverse multiple layers of attention. Each attention layer comprises self-attention mechanisms that empower the model to focus on the crucial relationships among amino acids. These relationships are characterized by attention weights, which determine the relative importance of each amino acid in the context of others. The computation of these attention weights involves the dot product of query, key, and value vectors derived from the encoded amino acid sequences. The ensuing attention scores are then used to weigh the values, which are essentially linear combinations of the encoded amino acid vectors. This sophisticated process enables the model to emphasize relevant amino acids while efficiently filtering out

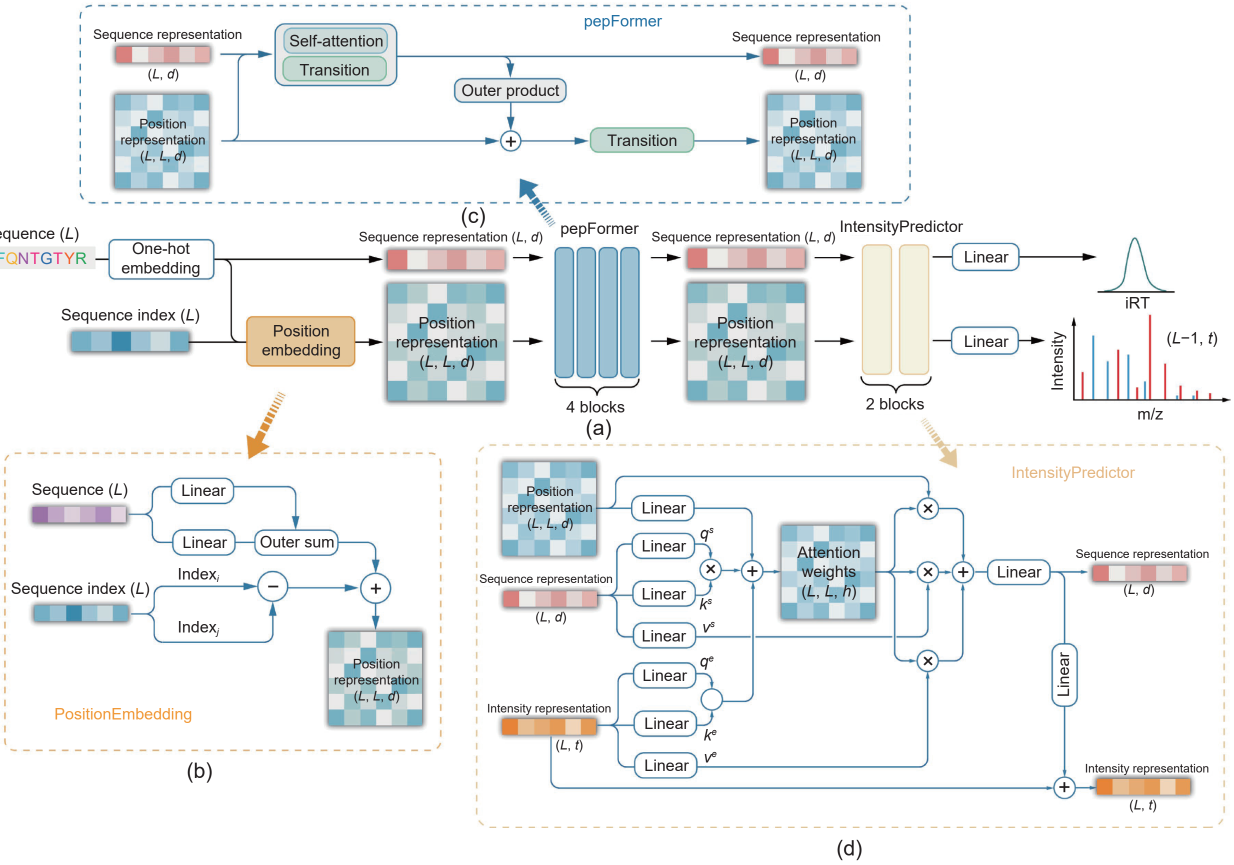


Fig. 1 Model architecture of DMSS. Arrows illustrate the information flow within the network. (a) Backbone network of DMSS, (b) PositionEmbedding, (c) pepFormer, and (d) IntensityPredictor (q^s, q^e, k^s, k^e, v^s, and v^e are intermediate variables).

irrelevant ones.

In experimental evaluations, DMSS has demonstrated superior performance in mass spectra prediction tasks, achieving state-of-the-art results on benchmark datasets. Its attention-based approach allows it to effectively capture important relationships and dependencies within amino acid sequences, leading to improved accuracy and predictive power.

2 Method

2.1 Model architecture

The DMSS network is illustrated in Fig. 1a and described in Algorithm 1. The DMSS model receives peptide sequence along with its position index as input, and provides predictions for the intensities of length $L-1$ for peptides of length L of each ion type (b+, y+, b++, y++, b+++, y+++). Taking into account the variety of ion types generated by precursor ions with different charges (1+, 2+, 3+, 4+), the numbers of predicted ion types denotes as t, are 2 (b+, y+), 2 (b+, y+), 4 (b+, y+, b++, y++), and 6 (b+, y+, b++, y++, b+++, y+++), respectively. Additionally, we consider the prediction of indexed Retention Time (iRT) as a special case where the number of predicted ions is 1.

The DMSS network is composed of two main stages. The first part is made up of four pepFormer blocks. The original inputs are processed through these blocks, generating $L\times d$ arrays for sequence representations and $L\times L\times d$ arrays for position representations. The network trunk is followed by the IntensityPredictor module, where the fully updated sequence representations are utilized to predict fragment ion intensity. In this module, we have designed an attention module, where the input consists of sequence representations and position representations, as well as current fragment ion intensities, while the output yields updated sequence representations. After the updates, the IntensityPredictor module utilizes the updated representations to predict the fragment ion intensities (b- and y- ion intensities) and iRT.

Algorithm 1 DMSS

```
Input: amino acid sequence $\{m_i\}$, amino acid index $\{\text{index}_i\}$
Output: intensity $\{I_{i,t}\}$
$\{s_i\}$ = Embedding ($\{m_i\}$); ▹ $s_i$ denotes single representation; Embedding ( ) is the function used to embed a tensor
$\{z_{i,j}\}$ = PositionEmbedding ($\{m_i\}$, $\{\text{index}_i\}$); ▹ $z_{i,j}$ denotes position representation
for $k$=1 to $N_{\text{update}}$ do ▹ $N_{\text{update}}$ denotes the number of updates
    $\{s_i\}$ = pepFormer ($\{s_i\}$, $\{z_{i,j}\}$);
end for
$e_i = 0$;
for $k$=1 to $N_{\text{predict}}$ do ▹ $N_{\text{predict}}$ denotes the number of iterations in the prediction process
    $\{\{s_i\}, \{e_i\}\}$ = IntensityPredictor ($\{s_i\}$, $\{z_{i,j}\}$, $\{e_i\}$);
    ▹ $e_i$ denotes intensity representation
end for
$I_{i,t}$ = exp (view ($e_i$)); ▹ exp ( ) is a function used to get the exponentiation of a tensor; view ( ) is a function used to flatten a tensor; $I_{i,t}$ denotes predicted intensities
return $\{I_{i,t}\}$
```

The incorporation of attention in our model arises from the objective of capturing the interplay between amino acids and their mutual influence. As attention mechanisms are intrinsically agnostic to positional information, we address this limitation by leveraging the PositionEmbedding module to encode the relative positions of amino acids. The resulting position representation is then introduced as a bias term during the sequence representation update process in pepFormer module. The PositionEmbedding module and the pepFormer module utilize attention models to depict the probability of protonation for each amino acid. Within the IntensityPredictor module, we employ the sequence representation to facilitate the prediction of tandem mass spectrometry (MS/MS) spectra using attention-based blocks. Our aim is for the sequence representation to acquire a universal representation that captures the essence of mass spectrometry prediction. Remarkably, our expectations are demonstrated in Section 3.3, where we conduct fine-tuning on the dataset with Q-Exactive HF instrument, while keeping the Embedding and pepFormer modules frozen, ultimately achieving impressive performance.

2.1.1 Position embedding

Position representations are generated using the PositionEmbedding module, as illustrated in Fig. 1b and described in Algorithm 2. In this module, the one-hot encoded sequence representations are processed through two independent linear layers, yielding a_i and b_i, which are then summed to form one part of the position representations. The second part of position representations involves relative positional encoding, where $r_{i,j}$ is calculated by the difference of sequence index of any two residues. $r_{i,j}$ is encoded as a one-hot vector, and then linearly transformed to obtain the

Algorithm 2 PositionEmbedding

```
Input: sequence representation $\{s_i\}$, sequence index $\{\text{index}_i\}$
Output: position representation $\{z_{i,j}\}$
$a_i$, $b_i$ = Linear $(s_i)$; ▷ Linear( ) is a function used for linear transformation on a tensor
$z_{i,j} = a_i + b_j$;
$r_{i,j}$ = $\text{index}_i - \text{index}_j$; ▷ $r_{i,j}$ denotes the distance between amino acids at positions $i$ and $j$
$z_{i,j}$ += Embedding $(r_{i,j})$;
return $\{z_{i,j}\}$
```

positional encoding $p_{i,j}$. $p_{i,j}$ is then added to $z_{i,j}$ to obtain the position representations.

2.1.2 pepFormer

The pepFormer block (see Algorithm 3) takes sequence representations and position representations as input, with size $L \times d$ and $L \times L \times d$, respectively. During the update, sequence representations are updated using a gated row-wise self-attention module with bias. The sequence representation s_i goes through linear layers to produce the query q_i, value v_i, and key k_i used in the attention module. The bias $b_{i,j}$ for position representation $z_{i,j}$ is generated by a linear layer, and used with q_i and k_i to build the attention weight $a_{i,j}$, incorporating positional information. In each block, the updated sequence representation is incorporated into the position representation through the operation of the outer product, effectively synchronizing the updated information of the sequence representation with the position representation.

Algorithm 3 pepFormer

```
Input: sequence representation $\{s_i\}$, position representation $\{z_{i,j}\}$
Output: sequence representation $\{s_i\}$
Constant: feature dimension c
$q_i$, $v_i$, $k_i$ = Linear $(s_i)$;
$b_{i,j}$ = Linear $(z_{i,j})$;
$g_i$ = sigmoid (Linear $(s_i)$); ▷ $g_i$ denotes the gate
$a_{i,j}$ = softmax $(\frac{1}{\sqrt{c}} q_i k_j^{\mathrm{T}} + b_{i,j})$;
$o_i = g_i \odot \Sigma_j a_{i,j} v_j$; ▷ $o_i$ denotes the output after gating
$s_i = s_i$ + Linear $(o_i)$;
$s_i = s_i$ + Linear $(s_i)$;
$a_i$, $b_i$ = Linear $(s_i)$;
$o_{i,j}$ = flatten $(a_i \otimes b_i)$ ▷ $o_{i,j}$ denotes the output of the outer product of $a_i$ and $b_i$
$z_{i,j}$ += Linear $(o_{i,j})$;
$z_{i,j} = z_{i,j}$ + Linear $(z_{i,j})$;
return $\{s_i\}$
```

2.1.3 IntensityPredictor

The input to the IntensityPredictor module (see Fig. 1d, Algorithm 4) consists of three components: a sequence representation s_i of shape $L \times d$, a position representation $z_{i,j}$ of shape $L \times L \times d$, and an initial intensity vector e_i of shape $L \times t$. The shape of intensity vectors should be $(L-1) \times t$, but for dimension alignment, a $1 \times t$ tensor is padded at the end and expanded to $L \times t$. The output consists of updated intensity vectors and sequence representations, with the shapes of both tensors remaining unchanged.

We construct a complex attention module, in which attention weights are contributed by the sequence representations, position representations, and intensity representations. The output of the attention calculation on the three parts is concatenated, and a linear layer is applied to align the dimensions with the sequence representations. The updated sequence representations are obtained by the residual connection. Using the updated sequence representations, a linear layer is applied to obtain the update of intensity representation of shape $L \times t$, denoted as Δe_i, which is directly added to e_i. To give practical meaning to the addition operation and provide interpretability even when the

Algorithm 4 IntensityPredictor

```
Input: sequence representation $\{s_i\}$, position representation $\{z_{i,j}\}$, intensity representation $\{e_i\}$
Output: sequence representation $\{s_i\}$, intensity representation $\{e_i\}$
Constant: feature dimension c
$q_i^s$, $v_i^s$, $k_i^s$ = Linear ( $s_i$); ▷ $q_i^s$, $v_i^s$, $k_i^s$ denote the linear encodings for $s_i$
$q_i^e$, $v_i^e$, $k_i^e$ = Linear ( $e_i$); ▷ $q_i^e$, $v_i^e$, $k_i^e$ denote the linear encodings for $e_i$
$b_{i,j}$ = Linear ( $e_{i,j}$);
$a_{i,j}$ = softmax ( $\frac{w_s}{\sqrt{c}} q_i^s (k_j^s)^{\mathrm{T}} + \frac{w_e}{\sqrt{c}} q_i^e (k_j^e)^{\mathrm{T}} + w_b b_{i,j}$ ); ▷ $w_s$, $w_e$, and $w_b$ denote weights assigned to each item
$o_i^s = \Sigma_j a_{i,j} v_j^s$; ▷ $o_i^s$ denotes output of single representation
$o_i^e = \Sigma_j a_{i,j} v_j^e$; ▷ $o_i^e$ denotes output of intensity representation
$o_i^p = \Sigma_j a_{i,j} z_{i,j}$; ▷ $o_i^p$ denotes output of position representation
$o_i$ = Linear (contact ( $o_i^s$, $o_i^e$, $o_i^p$)); ▷ $o_i$ denotes output after concatenating
$s_i$ += Linear ( $o_i$);
$e_i$ += Linear ( $s_i$);
return $\{s_i\}$ and $\{e_i\}$
```

output intensities are negative, we change the target of predicting intensities to predict the natural logarithm of fragment ion intensities, i.e., $e_i = \ln(I_{i,t})$ (where $I_{i,t}$ is the fragment ion intensity). In the first stacked intensity prediction module, the given current fragment ion intensities are all set to 0.

2.2 Statistics

The network is trained end-to-end, with gradients coming from the similarity loss of the theoretical spectrum and experimental spectrum. Here, we employ cosine similarity to measure the similarity between theoretical spectra and experimental spectra, which was calculated by torch.nn.functional.cosine_similarity. Since the cosine similarity values typically fall within the range of −1 to 1, we normalize this range to be between 0 and 1 through mathematical operations. Subsequently, we employ this scaled cosine similarity as our training loss, and its expression is as follows:

$$\mathcal{L} = \frac{1}{2}\left(1 - \frac{A \cdot B}{\|A\|_2 \cdot \|B\|_2}\right) \tag{1}$$

where A and B are the predicted and experimental spectras, respectively.

We utilize PCC and Spectral Angle (SA) to evaluate the similarity between A and B. The definition is as follows:

$$\text{PCC}(S_a, S_b) = \frac{\text{MD}(S_a) \times \text{MD}(S_b)}{\|\text{MD}(S_a)\|_2 \times \|\text{MD}(S_b)\|_2} \tag{2}$$

where S_a is predicted mass spectrum, S_b is the experimental mass spectrum, and $\text{MD}(S)$ is the mean deviation of S. The MD (S) of a tensor S with length n is defined as

$$\text{MD}(S) = S - \frac{1}{n}\sum_{i=1}^{n} S_i \tag{3}$$

$\| S \|_2$ is the 2-norm of S,

$$\|S\|_2 = \sqrt{\sum_{i=1}^{n} S_i^2} \tag{4}$$

Spectral contrast angle SA[35] is defined as

$$\text{SA}(S_a, S_b) = 1 - \frac{2\cos^{-1}\left(\frac{S_a}{\|S_a\|_2} \cdot \frac{S_b}{\|S_b\|_2}\right)}{\pi} \tag{5}$$

2.3 Model training

We employ the Adam optimizer[36] with a batch size of 32, and train for 64 epochs using the entire dataset. The initial learning rate is set to 0.001, and a cosine decay strategy is utilized to adjust the learning rate. The latent variables have a dimension of 256, and there are 8 attention heads. The training is performed on 4 Nvidia RTX 2080Ti/11G GPUs, with Pytorch 1.10.0 and Python 3.8 as dependencies.

2.4 Competing methods

For the pDeep2 method, we download its code from https://github.com/pFindStudio/pDeep/tree/master/pDeep2 and use the trained model mixed-180 322-multi_ce_Layer.ckpt to predict the intensities of fragment ions. For the Prosit method, we organize the sequence and collision energy information of the test set into a .hdf5 file and submit it to the server of Prosit (https://www.proteomicsdb.org/prosit), and then utilize Prosit_2020_intensity_hcd model to calculate the predicted results.

3 Result

3.1 Dataset

For our evaluation, we choose the ProteomeTools dataset[37], which consists of high-quality synthetic peptides. This dataset is created using a Lumos instrument and utilizes Higher-energy Collisional Dissociation (HCD) fragmentation[38]. The fragmentation process involves six commonly used NCEs: 20, 23, 25, 28, 30, and 35.

We acquire data from https://figshare.com/projects/prosit/35582, which are divided into three sets: a training set, a validation set to prevent overfitting, and a testing set to accurately estimate performance, in a 7: 2: 1 ratio, and convert the .hdf5 format to .pkl format for evaluation. The training, validation, and testing sets each consists of 8 230 360, 2 342 856, and 1 169 429 PSMs, respectively. The length of peptides falls within the range of 7 to 30. Refer to Table 1 for more information on the data distribution. Additionally, there is a pre-divided retention time dataset available, also obtained from the link above, consisting of distinct peptide sequences across the training, validation, and testing datasets, with sizes of 921 296, 263 226, and 109 133, respectively.

We download Bekker-Jensen et al.[39] dataset, which is obtained from Q-Exactive HF instrument, and the PRIDE repository with the identifier PXD004452. We expect to achieve optimal results through fine-tuning using a relatively small amount of data. Therefore, we randomly partition the dataset illustrated in Table 2, ensuring that there is no overlap between peptide

Table 1 Dataset information.

Dataset	Precursor charge				Total
	1+	2+	3+	4+	
Training	378 844	4 736 942	2 554 873	559 701	8 230 360
Validation	106 347	1 339 767	733 446	163 296	2 342 856
Testing	55 566	669 350	364 519	79 994	1 169 429

Table 2 Bekker-Jensen et al.[39] dataset information.

Dataset	Precursor charge				Total
	1+	2+	3+	4+	
Training	4867	19 787	21 858	22 820	69 332
Testing	30 369	1 389 091	1 093 665	431 764	2 944 889

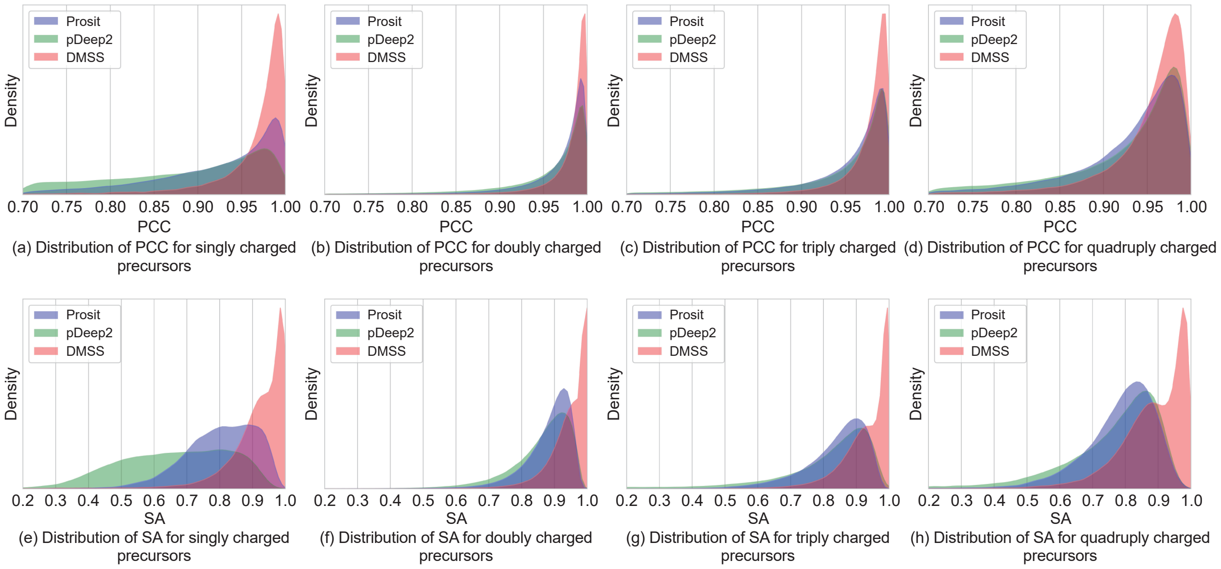
sequences in the training and testing sets.

3.2 Performance of DMSS in predicting b/y intensities

In our study, separate models are developed for precursor charges of 1, 2, 3, and 4. Each model predicts fragmentation spectra for six NCEs: 20, 23, 25, 28, 30, and 35. We evaluate the predictive capabilities of DMSS in determining b/y peak intensities, and compare it with Prosit and pDeep2. Our results, presented in Fig. 2, demonstrates that DMSS possesses exceptional abilities that set it apart from other methods. To establish a benchmark, we use the PCC and SA between experimental spectra of the same sequence in the test set. Figure 2 presents the distributions of PCC and SA between the predicted and experimental b/y intensities for DMSS, pDeep2, and Prosit with precursor ion charges of 1, 2, 3, and 4. From Fig. 2, it is evident that among the three methods, DMSS exhibits a higher value of PCC and SA values in the high similarity range compared to the other two methods, reflecting its superior accuracy.

The results of DMSS are presented in Tables 3−6, including the mean (SAmean) and median (SAmedian) of SA, the proportions of results with SA are greater than 0.70 (SA70), 0.90 (SA90), and 0.95 (SA95), as well as the same for PCC. The median SA values for precursor charges of 1, 2, 3, and 4 are 0.896, 0.925, 0.900, and 0.840, respectively. The median PCC values for the same precursor charges are 0.981, 0.990, 0.985, and 0.965, respectively. DMSS has demonstrated superior performance in predicting fragment ion intensities, especially for doubly charged precursors, where its predictions closely approach the upper limits for both PCC and SA compared to experimental b/y peak intensities. In addition, DMSS outperforms Prosit and pDeep2 significantly for singly charged precursors, with an improvement of around 8% in SA median compared to Prosit and 22% compared to pDeep2, as shown in Table 3.

In Tables 3−6, DMSS demonstrates significant improvements in metrics measuring the proportion of high-quality predicted spectra (SA90, SA95, PCC90, and PCC95) compared to Prosit and pDeep2 for each precursor charge. These results underscore the impressive capability of DMSS in accurately predicting high-quality mass spectra. Figure 3 displays the box plots of SA and PCC for each precursor charge at six



(a) Distribution of PCC for singly charged precursors

(b) Distribution of PCC for doubly charged precursors

(c) Distribution of PCC for triply charged precursors

(d) Distribution of PCC for quadruply charged precursors

(e) Distribution of SA for singly charged precursors

(f) Distribution of SA for doubly charged precursors

(g) Distribution of SA for triply charged precursors

(h) Distribution of SA for quadruply charged precursors

Fig. 2 Peptide MS/MS spectrum prediction.

Table 3 Performance metrics for SA and PCC of singly charged precursors.

Method	Index									
	SAmedian	SAmean	SA70	SA90	SA95	PCCmedian	PCCmean	PCC70	PCC90	PCC95
pDeep2	0.674	0.662	0.448	0.048	0.003	0.810	0.737	0.636	0.338	0.200
Prosit	0.814	0.801	0.832	0.206	0.044	0.940	0.905	0.949	0.658	0.450
DMSS	0.896	0.871	0.952	0.474	0.103	0.981	0.953	0.977	0.899	0.776

Table 4 Performance metrics for SA and PCC of doubly charged precursors.

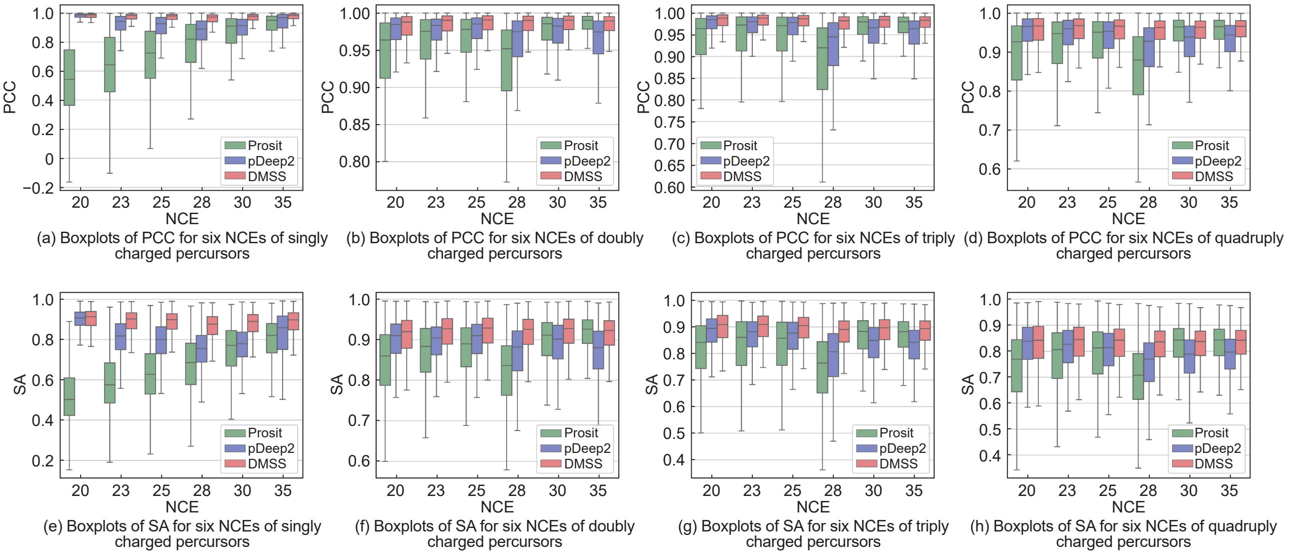
Method	Index									
	SAmedian	SAmean	SA70	SA90	SA95	PCCmedian	PCCmean	PCC70	PCC90	PCC95
pDeep2	0.883	0.859	0.933	0.409	0.112	0.975	0.946	0.977	0.852	0.697
Prosit	0.899	0.879	0.967	0.493	0.112	0.981	0.961	0.988	0.916	0.786
DMSS	0.925	0.907	0.980	0.687	0.251	0.990	0.974	0.992	0.957	0.891

Table 5 Performance metrics for SA and PCC of triply charged precursors.

Method	Index									
	SAmedian	SAmean	SA70	SA90	SA95	PCCmedian	PCCmean	PCC70	PCC90	PCC95
pDeep2	0.852	0.804	0.822	0.299	0.070	0.968	0.912	0.928	0.774	0.619
Prosit	0.860	0.835	0.901	0.293	0.051	0.972	0.945	0.983	0.850	0.674
DMSS	0.900	0.880	0.965	0.498	0.120	0.985	0.969	0.992	0.945	0.855

Table 6 Performance metrics for SA and PCC of quadruply charged precursors.

Method	Index									
	SAmedian	SAmean	SA70	SA90	SA95	PCCmedian	PCCmean	PCC70	PCC90	PCC95
pDeep2	0.800	0.761	0.740	0.117	0.010	0.944	0.893	0.921	0.677	0.463
Prosit	0.803	0.782	0.811	0.103	0.012	0.947	0.920	0.972	0.747	0.480
DMSS	0.840	0.819	0.904	0.161	0.015	0.965	0.943	0.983	0.864	0.650



(a) Boxplots of PCC for six NCEs of singly charged percursors
(b) Boxplots of PCC for six NCEs of doubly charged percursors
(c) Boxplots of PCC for six NCEs of triply charged percursors
(d) Boxplots of PCC for six NCEs of quadruply charged percursors
(e) Boxplots of SA for six NCEs of singly charged percursors
(f) Boxplots of SA for six NCEs of doubly charged percursors
(g) Boxplots of SA for six NCEs of triply charged percursors
(h) Boxplots of SA for six NCEs of quadruply charged percursors

Fig. 3 Boxplots of peptide MS/MS spectrum prediction.

NCEs. It is evident that DMSS surpasses the other two methods in all scenarios. It also highlights that DMSS exhibits a more concentrated distribution of PCC and SA for each precursor ion charge state and across the six NCEs. Upon calculation, DMSS shows lower variances in both PCC and SA, with values of 0.004 and 0.007, respectively, compared to pDeep2 (0.017, 0.019) and Prosit (0.006, 0.010). These findings

Table 7 Performance on Bekker-Jensen dataset[39].

Method	Index			
	SAmedian	SAmean	PCCmedian	PCCmean
pDeep2	0.738	0.639	0.887	0.731
Prosit	0.685	0.576	0.837	0.661
DMSS	0.857	0.806	0.968	0.910

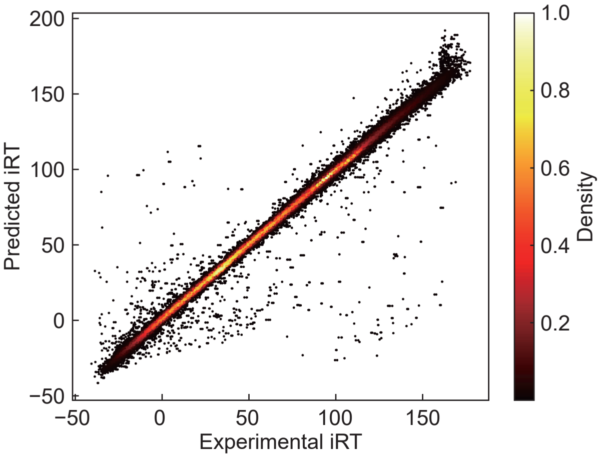


Fig. 4 Peptide iRT prediction.

demonstrate the stability of our method's prediction performance.

Despite the increase in model complexity, our execution time remains relatively fast. Employing our proposed method for inference on a single 2080Ti/11G GPU, the prediction time for 10 000 spectra is approximately 13.8 seconds. Considering the relatively affordable cost of the 2080 Ti GPU, it is a viable option for most researchers and practitioners.

3.3 Performance on Other Datasets

To assess the generalization capability of our model, we conduct experiments using the Bekker-Jensen et al.'s[39] dataset acquired from the Q-Exactive HF instrument. Specifically, we perform fine-tuning on the train set while keeping the Embedding and pepFormer modules unchanged, and solely focus on fine-tuning the IntensityPredictor module. The experimental results, as presented in Table 7, reveal that both the median and mean values of PCC are above 0.9 and of SA are above 0.8, indicating the effectiveness of our proposed approach. Furthermore, we conduct a comparative analysis with the QE mode of pDeep2 and the Prosit model. Notably, our approach exhibits superior performance compared to both of others.

3.4 Performance of DMSS in predicting iRT

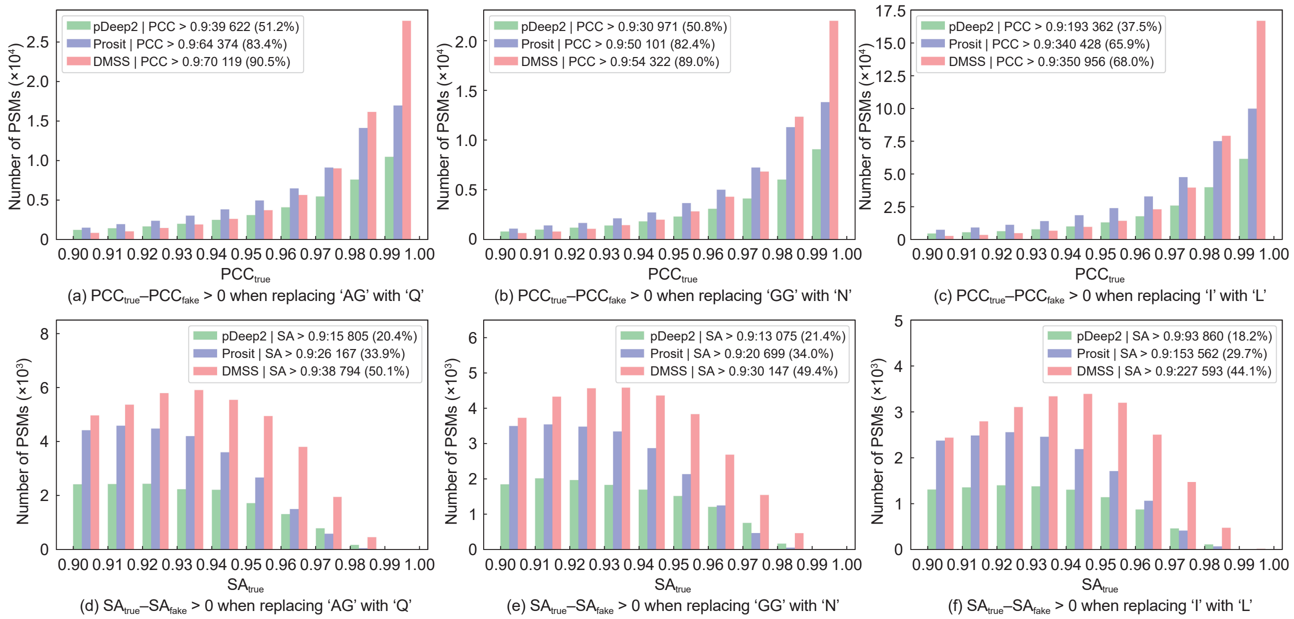
To assess the predictive capability of our model for iRT, we train it using a dataset of 921 296 peptides, and then evaluate it on an independent test set of 109 133 peptides. The results, as shown in Fig. 4, reveal a PCC of 0.993, indicating a strong agreement between the predicted and experimental iRT values. The $\Delta \text{iRT}_{95\%}$ is calculated to be 4.52, which is the difference between the predicted iRT values and the experimental iRT values within a 95% confidence interval. This small $\Delta \text{iRT}_{95\%}$ suggests that the majority of predicted iRT values are very close to the experimental values, demonstrating that the model's predictions are highly accurate and reliable. These findings demonstrate that the trained model has a strong predictive capability for iRT, being able to accurately predict the retention times of peptides based on their characteristics and properties. This information is beneficial in various applications, such as peptide identification in proteomics research or optimizing chromatography conditions in analytical chemistry.

3.5 Distinguish extremely similar peptides by DMSS

The identification of peptides in existing proteomics search engines heavily relies on mass information alone. However, accurately identifying peptides becomes problematic when amino acids with similar masses are present, such as 'I' and 'L', 'GG' and 'N', or 'AG' and 'Q'. In order to address this challenge, we propose incorporating intensity information to improve the matching outcomes. To evaluate the effectiveness of our proposed approach in distinguishing these ambiguous cases, we conduct a series of experiments.

In our experiments, we select peptides from test set that contains the aforementioned ambiguous amino acid pairs ('AG', 'GG', and 'I'), which are considered true peptides. We then alter these amino acid pairs to generate fake peptides, replacing 'AG' with 'Q', 'GG' with 'N', and 'I' with 'L'. We use Prosit, pDeep2, and DMSS to predict both the true peptides and the fake peptides. The differences between the true peptides (PCC_{true} and SA_{true}) and the fake peptides (PCC_{fake} and SA_{fake}) are analyzed and visualized in Fig. 5. The results obtained by swapping true peptides and fake peptides are illustrated in Fig. 6.

Figures 5 and 6 depict the count of cases in which the difference between the metrics of true peptides and fake peptides, for $\text{PCC}_{\text{true}} > 0.9$ and $\text{SA}_{\text{true}} > 0.9$, is greater than zero for the three methods. Figures 5 and 6 also illustrate the number of peptides for each method



(a) PCC_{true}–PCC_{fake} > 0 when replacing ‘AG’ with ‘Q’

(b) PCC_{true}–PCC_{fake} > 0 when replacing ‘GG’ with ‘N’

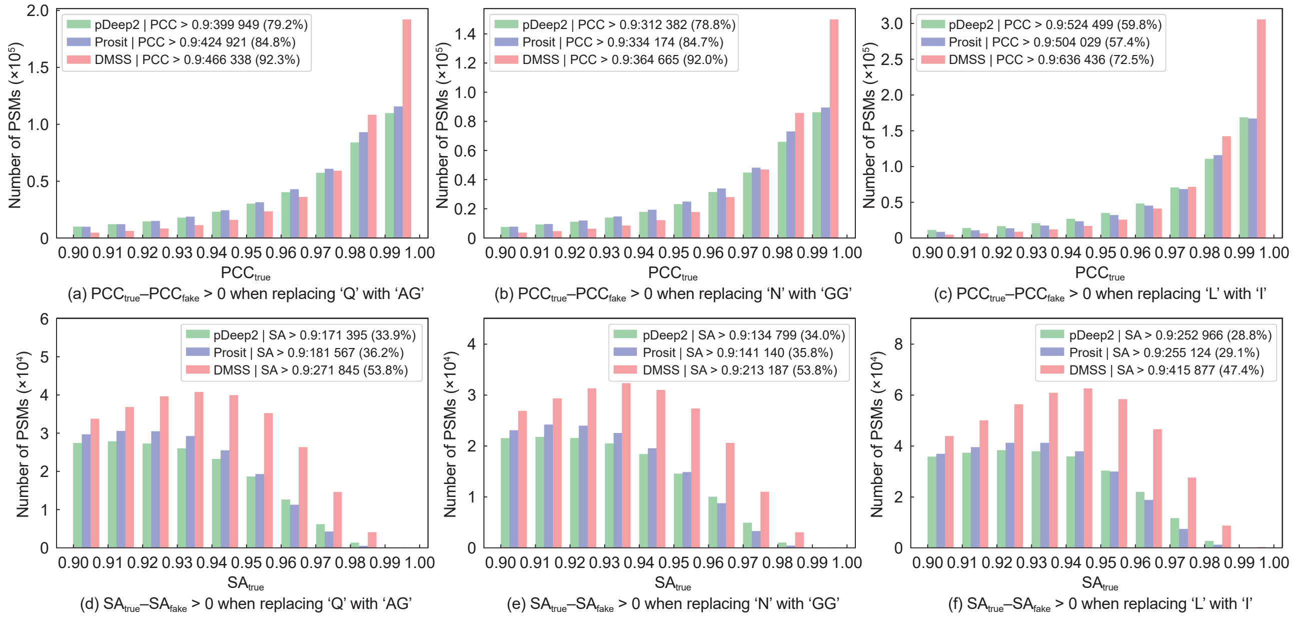
(c) PCC_{true}–PCC_{fake} > 0 when replacing ‘I’ with ‘L’

(d) SA_{true}–SA_{fake} > 0 when replacing ‘AG’ with ‘Q’

(e) SA_{true}–SA_{fake} > 0 when replacing ‘GG’ with ‘N’

(f) SA_{true}–SA_{fake} > 0 when replacing ‘I’ with ‘L’

Fig. 5 Distinguish similar peptides when replacing ‘AG’ with ‘Q’, ‘GG’ with ‘N’, and ‘I’ with ‘L’.



(a) PCC_{true}–PCC_{fake} > 0 when replacing ‘Q’ with ‘AG’

(b) PCC_{true}–PCC_{fake} > 0 when replacing ‘N’ with ‘GG’

(c) PCC_{true}–PCC_{fake} > 0 when replacing ‘L’ with ‘I’

(d) SA_{true}–SA_{fake} > 0 when replacing ‘Q’ with ‘AG’

(e) SA_{true}–SA_{fake} > 0 when replacing ‘N’ with ‘GG’

(f) SA_{true}–SA_{fake} > 0 when replacing ‘L’ with ‘I’

Fig. 6 Distinguish similar peptides when replacing ‘Q’ with ‘AG, ‘N’ with ‘GG’ and ‘L’ with ‘I’.

when PCC_{true} and SA_{true} are greater than 0.9, as well as the proportion of this number to the total number of test data. It is evident that under conditions where the metrics exceed 0.9, signifying high confidence in the predicted spectra, DMSS demonstrates superior capability in distinguishing similar peptides compared to the other two methods.

These findings demonstrate the robust discriminatory capabilities of DMSS compared to Prosit and pDeep2 when confronted with ambiguous amino acids. The superior ability of DMSS to distinguish between ‘AG’ and ‘Q’, ‘GG’ and ‘N’, and ‘I’ and ‘L’ highlights the potential application of incorporating our predicted intensity information into mass spectrometry search engines to enhance peptide identification accuracy. The outstanding discrimination of these ambiguous amino acids by DMSS suggests that leveraging the predicted intensity information can provide valuable insights for improving the accuracy and reliability of peptide identification in proteomic studies.

4 Conclusion

In this study, we introduce DMSS, a novel approach for accurately predicting iRT and fragment ion intensities in proteomics analysis. The results obtained from our experiments yield valuable insights into the effectiveness of DMSS, and the remarkable performance for predicting iRT and fragment ion intensities also shed light on the potential applications and prospects of DMSS in advancing peptide identification and spectral library generation.

Additionally, we investigate the capability of DMSS to distinguish extremely similar peptides. Specifically, we examine the discrimination between ‘I’ and ‘L’, ‘GG’ and ‘N’, and ‘AG’ and ‘Q’ cases. The results show that DMSS achieves superior performance in distinguishing these similar peptides compared to existing methods, which demonstrates potential application of incorporating our predicted intensity information into mass spectrometry search engines to enhance peptide identification accuracy.

While our study has yielded promising results, there are several avenues for future research. Firstly, further investigations could focus on expanding the scope of similar peptide discrimination to other ambiguous amino acid cases. Additionally, the performance of DMSS on larger and more diverse datasets could be explored to evaluate its scalability and generalization capability. Furthermore, the potential application of DMSS in other aspects of proteomics analysis, such as post-translational modification identification, warrants further exploration.

In conclusion, our study demonstrates the effectiveness of DMSS in predicting iRT and fragment ion intensities from peptide sequences. These findings contribute to the advancement of proteomics analysis and highlight the potential of DMSS as a valuable tool in the field. Further research in this direction has the potential to enhance the accuracy and reliability of proteomics data analysis, facilitating new insights into biological systems and disease mechanisms.

Acknowledgment

We would like to express our gratitude to the authors of the previous methods for generously making their web servers available. This work was supported by the National Natural Science Foundation of China (Nos. 62072435, 82130055, 32271297, and 32370657) and the National Key Research and Development Program of China (No. 2020YFA0907000).

References

[1] K. Biemann, Mass spectrometry of peptides and proteins, *Annu. Rev. Biochem.*, vol. 61, pp. 977–1010, 1992.

[2] R. Aebersold and M. Mann, Mass spectrometry-based proteomics, *Nature*, vol. 422, pp. 198–207, 2003.

[3] M. Wilhelm, D. P. Zolg, M. Graber, S. Gessulat, T. Schmidt, K. Schnatbaum, C. Schwencke-Westphal, P. Seifert, N. de Andrade Krätzig, J. Zerweck, et al., Deep learning boosts sensitivity of mass spectrometry-based immunopeptidomics, *Nat. Commun.*, vol. 12, no. 1, p. 3346, 2021.

[4] Z. Mao, R. Zhang, L. Xin, and M. Li, Mitigating the missing-fragmentation problem in de novo peptide sequencing with a two-stage graph-based deep learning model, *Nat. Mach. Intell.*, vol. 5, no. 11, pp. 1250–1260, 2023.

[5] J. Cox, Prediction of peptide mass spectral libraries with machine learning, *Nature Biotechnology*, vol. 41, no. 1, pp. 33–43, 2023.

[6] V. Lange, P. Picotti, B. Domon, and R. Aebersold, Selected reaction monitoring for quantitative proteomics: A tutorial, *Mol. Syst. Biol.*, vol. 4, no. 1, p. 222, 2008.

[7] L. C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner, and R. Aebersold, Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: A new concept for consistent and accurate proteome analysis, *Mol. Cell. Proteom.*, vol. 11, no. 6, p. O111.016717, 2012.

[8] A. Doerr, DIA mass spectrometry, *Nat. Meth.*, vol. 12, no. 1, p. 35, 2015.

[9] P. Sinitcyn, J. D. Rudolph, and J. Cox, Computational methods for understanding mass spectrometry–based shotgun proteomics data, *Annu. Rev. Biomed. Data Sci.*, vol. 1, pp. 207–234, 2018.

[10] J. Cox, N. Neuhauser, A. Michalski, R. A. Scheltema, J. V. Olsen, and M. Mann, Andromeda: A peptide search engine integrated into the MaxQuant environment, *J. Proteome Res.*, vol. 10, no. 4, pp. 1794–1805, 2011.

[11] D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell, Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999.

[12] J. K. Eng, A. L. McCormack, and J. R. Yates, An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database, *J. Am. Soc. Mass Spectrom.*, vol. 5, no. 11, pp. 976–989, 1994.

[13] M. Scigelova and A. Makarov, Orbitrap mass analyze—overview and applications in proteomics, *Proteomics*, vol. 6, no. S2, pp. 16–21, 2006.

[14] S. Sun, F. Yang, Q. Yang, H. Zhang, Y. Wang, D. Bu, and

B. Ma, MS-simulator: Predicting *Y*-ion intensities for peptides with two charges based on the intensity ratio of neighboring ions, *J. Proteome Res.*, vol. 11, no. 9, pp. 4509–4516, 2012.

[15] J. E. Elias, F. D. Gibbons, O. D. King, F. P. Roth, and S. P. Gygi, Intensity-based protein identification by machine learning from a library of tandem mass spectra, *Nat. Biotechnol.*, vol. 22, no. 2, pp. 214–219, 2004.

[16] R. J. Arnold, N. Jayasankar, D. Aggarwal, H. Tang, and P. Radivojac, A machine learning approach to predicting peptide fragmentation spectra, in *Proc. Pacific Symposium on Biocomputing*, Kohala Coast, HI, USA, pp. 219–230.

[17] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] S. Li, R. J. Arnold, H. Tang, and P. Radivojac, On the accuracy and limits of peptide fragmentation spectrum prediction, *Anal. Chem.*, vol. 83, no. 3, pp. 790–796, 2011.

[19] S. Degroeve, D. Maddelein, and L. Martens, MS2PIP prediction server: Compute and visualize MS2 peak intensity predictions for CID and HCD fragmentation, *Nucleic Acids Res.*, vol. 43, no. W1, pp. W326–W330, 2015.

[20] S. Hochreiter and J. J. Schmidhuber, Long short-term memory, *Neural Comput.*, vol. 9, pp. 1–32, 1997.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv: 1412.3555, 2014.

[22] Y. Yu, X. Si, C. Hu, and J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures, *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[23] X Zhou, W. Zeng, H. Chi, C. Luo, C. Liu, J. Zhan, S. He, and Z. Zhang, pDeep: Predicting MS/MS spectra of peptides with deep learning, *Anal. Chem.*, vol. 89, no. 23, pp. 12690–12697, 2017.

[24] W. Zeng, X. Zhou, W. Zhou, H. Chi, J. Zhan, and S. He, MS/MS spectrum prediction for modified peptides using pDeep2 trained by transfer learning, *Anal. Chem.*, vol. 91, no. 15, pp. 9724–9731, 2019.

[25] S. Gessulat, T. Schmidt, D. P. Zolg, P. Samaras, K. Schnatbaum, J. Zerweck, T. Knaute, J. Rechenberger, B. Delanghe, A. Huhmer, et al., Prosit: Proteome-wide prediction of peptide tandem mass spectra by deep learning, *Nat. Meth.*, vol. 16, no. 6, pp. 509–518, 2019.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. of the 31st Int. Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, pp. 6000–6010.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv: 2010.11929, 2020.

[28] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, Swin Transformer: Hierarchical vision Transformer using shifted windows, in *Proc. IEEE/CVF Int. Conf. Computer Vision* (*ICCV*), Montreal, Canada, 2021, pp. 9992–10002.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805, 2018.

[30] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, Improving language understanding by generative pretraining, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018.

[31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, Language models are unsupervised multitask learners, https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf, 2019.

[32] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165**.**

[33] R. Lou, W. Liu, R. Li, S. Li, X. He, and W. Shui, DeepPhospho accelerates DIA phosphoproteome profiling through in silico library generation, *Nat. Commun.*, vol. 12, no. 1, p. 6685, 2021.

[34] M. Ekvall, P. Truong, W. Gabriel, M. Wilhelm, and L. Käll, Prosit Transformer: A transformer for prediction of MS2 spectrum intensities, *J. Proteome Res.*, vol. 21, no. 5, pp. 1359–1364, 2022.

[35] U. H. Toprak, L. C. Gillet, A. Maiolica, P. Navarro, A. Leitner, and R. Aebersold, Conserved peptide fragmentation as a benchmarking tool for mass spectrometers and a discriminating feature for targeted proteomics, *Mol. Cell. Proteom.*, vol. 13, no. 8, pp. 2056–2071, 2014.

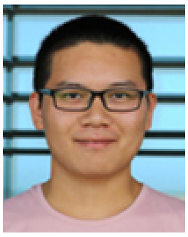
[36] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv: 1412.6980, 2014.

[37] D. P. Zolg, M. Wilhelm, K. Schnatbaum, J. Zerweck, T. Knaute, B. Delanghe, D. J. Bailey, S. Gessulat, H.-C. Ehrlich, M. Weininger, et al., Building proteome tools based on a complete synthetic human proteome, *Nature Methods*, vol. 14, no. 3, pp. 259–262, 2017.

[38] J. V. Olsen, B. Macek, O. Lange, A. Makarov, S. Horning, and M. Mann, Higher-energy C-trap dissociation for peptide modification analysis, *Nat. Meth.*, vol. 4, no. 9, pp. 709–712, 2007.

[39] D. B Bekker-Jensen, C. D Kelstrup, T. S Batth, S. C Larsen, C. Haldrup, J. B Bramsen, K. D Sorensen, S. Hoyer, T. F Orntoft, C. L Andersen, et al., An optimized shotgun strategy for the rapid generation of comprehensive human proteomes, *Cell Systems*, vol. 4, no. 6, pp. 587–599, 2017.

Yihui Ren received the BEng degree from University of Chinese Academy of Sciences, China in 2021. She is currently a PhD candidate at Institute of Computing Technology, Chinese Academy of Sciences (CAS) , China. Her main research interests include mass spectrometry, bioinformatics, and deep learning.



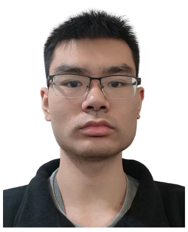
Wenkai Han received the BEng degree from University of Science and Technology of China in 2018. He is currently a PhD candidate under the supervision of professor Xin Gao at King Abdullah University of Science and Technology, Kingdom of Saudi Arabia. His research interests include computational biology, generative models, and deep learning.

Xiaoyang Hou received the BEng degree from Shandong University, China in 2022. She is currently a master student at Institute Computing Technology, Chinese Academy of Sciences, China. Her main research interest is molecule discovery.

Dongbo Bu received the PhD degree from Institute of Computing Technology, Chinese Academy of Sciences, China in 2001, where he is currently a professor. His research interests include algorithm design, especially algorithm design with the aid of AI, protein structure prediction, protein design, and glycan identification using mass spectrometry.

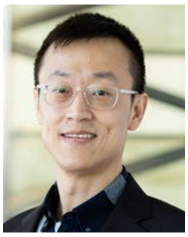
Shiwei Sun received the PhD degree from Institute of Computing Technology, Chinese Academy of Sciences, China in 2007, where he is currently an associate professor. His research focuses on bioinformatics, artificial intelligence, and various other related fields.

Yu Wang received the BEng degree in automation from Tsinghua University, China in 2019, and the MEng degree in computer science and technology from Institute of Computing Technology, CAS, China in 2022. He is currently employed as an algorithm engineer at Syneron Technology, China. His research interests include bioinformatics, artificial intelligence, and development of peptide-based pharmaceuticals.



Yikang Huang is currently a undergraduate student at College of Information and Electrical Engineering, China Agriculture University, China. His main research interest is bioinformatics.

Chunming Zhang received the BEng degree in electronics and communication engineering from Tsinghua University, China in 2008. He is currently working as a senior engineer at Institute of Computing Technology, Chinese Academy of Sciences, mainly engaged in research work in the fields of bioinformatics, high-performance computing, etc.



Xin Gao received the BEng degree in computer science from Tsinghua University, China in 2004, and the PhD degree in computer science from University of Waterloo, Canada in 2009. He is a professor of computer science, acting as an associate director of Computational Bioscience Research Center (CBRC), and the lead of the Structural and Functional Bioinformatics Group, King Abdullah University of Science and Technology (KAUST), Kingdom of Saudi Arabia. His research interests include computational biology, generative models, and deep learning.