Towards Open-Search De Novo Peptide Sequencing via Mass-Based Zero-Shot Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Proteins are the main drivers of biochemical processes and play a pivotal role in almost all cellular functions. Through post-translational modifications (PTMs), residues within a protein can be chemically modified to fine-tune the protein's function in the cellular context. Despite the importance of PTMs, the plethora of deep learning-based de novo peptide sequencing (DNPS) models, which, in contrast to database searching approaches, predict peptide sequence solely from tandem mass spectra without any reference organism database, can only predict peptide sequences with a limited set of PTMs. This is because they rely on fixed vocabularies that map residue tokens to non-generalizable learned embeddings. To overcome this limitation, we propose a novel approach that leverages the fact that amino acids and their derivatives are characterized by their mass, a generalizable feature that enables zero-shot learning. Specifically, we reformulate DNPS as a mass prediction problem instead of a multiclass classification problem, where the model predicts the mass of the next residue instead of its token representation. To facilitate generalization to unseen PTMs, we leverage an adversarial multi-task learning scheme by supplementing the training data of experimental spectra with simulated spectra that mimic spectra containing unseen residues. We show that our approach allows the prediction of previously unseen PTMs, providing a promising proof of concept for mass-based representations as a path towards true open-search DNPS.

1 Introduction

Proteins are essential to nearly all biological processes, functioning as catalysts, structural components, signaling molecules, transporters, and immune effectors [1, 2]. Their functional diversity is further enhanced by post-translational modifications (PTMs)—chemical changes to amino acid side chains that influence protein structure and activity and are often linked to disease [3]. Proteomics, the study of proteins, depends on accurate protein identification for downstream analyses such as quantification and interaction mapping [4]. Bottom-up proteomics via liquid chromatography tandem mass spectrometry (LC-MS/MS) is the most common method for high-throughput protein identification [4, 5]. Here, proteins are enzymatically digested into peptides, which are analyzed in a first mass spectrometry scan to determine their mass-to-charge (m/z) ratios. Selected precursor ions are then fragmented, and the resulting fragment ions are measured in a second scan called a tandem mass spectrum [4]. Peptide sequences can, in principle, be inferred from m/z differences between consecutive peaks [6, 7], but this remains challenging due to missing or noisy peaks, co-isolated contaminants, and uncertainty about ion series assignment.

Peptide sequences are typically identified using database search methods, which compare experimental spectra to theoretical spectra derived from a reference protein database [8]. Including PTMs further

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

expands the space of candidate peptides, making exhaustive searches computationally challenging. De novo peptide sequencing (DNPS) bypasses the need for a reference database by inferring peptide sequences directly from tandem mass spectra. This makes it, in contrast to database search methods, well-suited for identifying novel peptides, rare or unknown PTMs, and proteins from unsequenced organisms. Several de novo peptide sequencing (DNPS) tools have been introduced in recent years, demonstrating promising performance across benchmark datasets [9–25]. However, accurately sequencing post-translationally modified (PTM) peptides remains a significant challenge. Despite the existence of over 400 known PTM types [26, 27], Casanovo—the first transformer-based de novo peptide sequencing (DNPS) model—can identify only seven [11]. Improving PTM prediction has attracted considerable research interest, with recent advances such as AdaNovo [9]. Another approach is to expand model coverage by enlarging the PTM vocabulary; for example, π-PrimeNovo is fine-tuned to recognize 21 PTMs [28]. However, to the best of our knowledge, no current DNPS model can detect PTMs not seen during training in a zero-shot manner, restricting their ability to discover novel or rare PTMs.

In this work, we propose a transformer-based mass prediction approach to peptide sequencing, enabling zero-shot inference for unseen residues, including novel PTMs. To improve generalization, we explore multi-task learning (MTL) with a training strategy combining experimental and simulated spectra. In the experimental spectra, the model encounters high spectral complexity with a limited set of PTMs, while in the simulated spectra, it learns to generalize to unseen PTMs with arbitrary peak differences. We introduce a generative adversarial network (GAN)-inspired MTL model, which demonstrates strong performance on simulated data and limited but encouraging generalization to unseen residues on experimental spectra. While not solving zero-shot residue inference on experimental data, we provide a framework for future research and highlight key challenges.

2 Background and Related Work

In bottom-up proteomics, proteins are extracted from a biological sample and enzymatically digested—typically with trypsin—into smaller peptides [4]. These peptides are separated by liquid chromatography (LC) and introduced into a tandem mass spectrometer (LC-MS/MS), typically operated in data-dependent acquisition (DDA) mode. In the first MS stage (MS1), peptide ions are detected to determine their precursor masses and intensities. The most intense precursors are selected for fragmentation in the second MS stage (MS2), generating spectra composed of fragment ion peaks (tandem mass spectra).[5] The mass differences between these peaks correspond to the masses of individual (modified) amino acids, enabling peptide sequence inference. [6, 7] Peptides are then identified either by matching spectra to theoretical peptides from a reference proteome with database search methods [29] or by interpreting spectra directly with de novo peptide sequencing (DNPS) methods. Identified peptides are subsequently used to infer protein presence and abundance in the original sample. [30]

In the early days of DNPS-based proteomic studies, spectra were manually annotated by experts—a process that was both time-consuming and expensive [30]. As LC-MS/MS throughput increased, early computational tools such as PEAKS [31], NovoHMM [32], and PepNovo [33] were developed to automate de novo peptide sequencing. However, despite their theoretical potential, these methods were often limited in accuracy and struggled with noisy or complex spectra, making database search methods the preferred choice for many applications.

The field has gained renewed momentum with the rise of deep learning. DeepNovo [34] first combined CNNs and LSTMs to model the peptide sequencing process directly. Following this, PointNovo [25] employed a pointer network approach to enhance the decoding of peptide sequences. Transformer-based architectures then marked a significant leap in performance, with Casanovo [11] introducing a streamlined, attention-based model that boosted both accuracy and inference speed. Several other transformer-based tools have emerged since [9, 10, 12, 17, 18, 24, 35], notably AdaNovo, which incorporated adaptive learning strategies to improve performance on peptides containing post-translational modifications (PTMs). Additionally, GraphNovo [16] introduced a graph-based approach to capture the relationships between peptide fragments, further improving sequencing accuracy and handling complex modification patterns. ContraNovo [12] enhanced the transformer decoder of Casanovo by incorporating mass information directly into the model, allowing it to better utilize the mass differences between peptide fragments and improve the accuracy of

peptide sequence prediction, particularly for spectra with ambiguous or overlapping peaks. Recently, π-PrimeNovo was proposed, a model fine-tuned on 21 PTMs, to expand PTM coverage [24].

These advancements have greatly enhanced the accuracy, generalization, and speed of DNPS tools, making them more viable for applications with incomplete reference databases. However, DNPS models capable of generalizing beyond the amino acids and PTMs present in the training data are still lacking, limiting their ability to accurately discover novel peptides or uncharacterized PTMs.

3 Methods

3.1 Datasets

3.1.1 Experimental Datasets

We use the MassIVE Knowledge Base spectral library v1 (MassIVE-KB v1), originally introduced by Casanovo for DNPS, to train and evaluate our method [11, 36]. This extensive dataset consists of high-resolution HCD mass spectrometry data from diverse experimental conditions, totaling 30,504,897 peptide-spectrum matches (PSMs) with extremely stringent false discovery rate (FDR) control [11]. To facilitate direct comparison with Casanovo [11], we employ the same train, validation, and test splits used in their study. The dataset includes peptide sequences composed of the 20 canonical amino acids, along with eight post-translationally modified amino acids.

3.1.2 Simulated Datasets

To establish a model with zero-short learning capacity, we simulated synthetic spectra with a wide variety of possible masses. We reasoned that the simulated masses did not have to match the masses of existing PTM-amino acids. This incentivized models to capture the mechanisms of mass spectrometry rather than memorizing predefined sets of masses.

We first generated peptide sequences as lists of residue masses, each sampled uniformly from the range [60, 300] Da. Peptide lengths were drawn from a uniform distribution between 5 and 20 residues, and precursor charges z were sampled from a discrete distribution: $z = 1$: (0.5), $z = 2$: (0.25), $z = 3$: (0.125), and $z = 4$: (0.125).

We implemented two strategies for generating the corresponding tandem mass spectra. In the first, simplified approach, we generated an ideal spectrum consisting of one peak per residue, corresponding directly to its mass over charge ratio (m/z). The second, more realistic strategy simulates peptide fragmentation by generating b- and y-ions. More specifically, let z be the precursor charge and $\mathbf{M}_{\text{Pep}} = (m_1, m_2, \ldots, m_l)$ the vector of masses for a given peptide of length l, where m_i is the mass of the i-th residue. The m/z ratios for b-ions are defined as $b_i = \frac{1}{z}\sum_{j=1}^{i} m_j + \mathrm{H}$. Analogously, the m/z values for y-ions are defined as $y_i = \frac{1}{z}\sum_{j=l-i+1}^{l} m_j + \mathrm{H_2O} + \mathrm{H}$. H and $\mathrm{H_2O}$ represent the mass of Hydrogen and Water molecules, respectively. To introduce variability, a random number (uniformly drawn from 0 to 5) of peaks were removed from each ion series, while ensuring that at least one peak (b- or y-ion) per residue was retained to preserve full sequence information.

To simulate more realistic spectra, we added between 0 and 10 noise peaks per spectrum, with m/z values sampled uniformly from the range [50, m/z_{max} + 300], where m/z_{max} is the maximal m/z in the simulated spectrum. Fragment ion intensities were sampled from a Gaussian distribution $\mathcal{N}(1.0, 0.1)$, while noise peak intensities were drawn from $\mathcal{N}(0.4, 0.1)$.

3.2 Model Architecture

3.2.1 Transformer-Based Mass Prediction

Our model builds upon the transformer architecture introduced in Casanovo [11], but reformulates the peptide sequencing task from next-token classification over (modified) amino acid tokens to a continuous mass prediction task. For this, we introduce the mass regression decoder, which outputs a scalar value representing the mass of each peptide residue in Dalton. We implement this by extending Casanovo's peptide decoder with a four-layer feed-forward network with PReLU activation and a single output dimension.

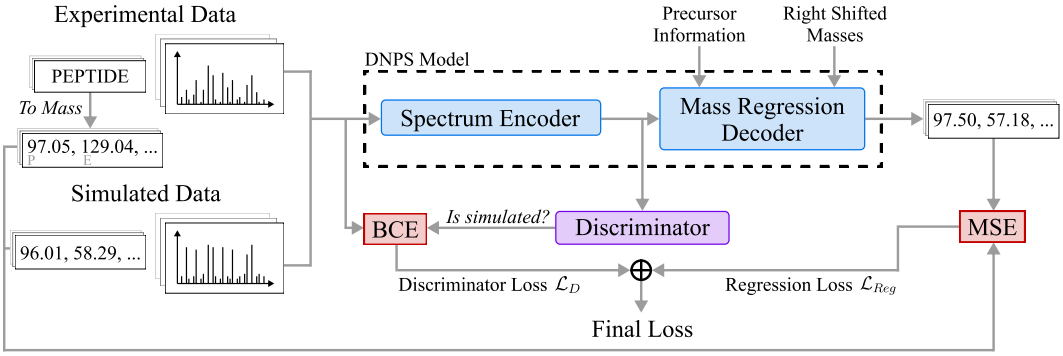

Figure 1: **Adversarial multi-task Learning model architecture for mass-based peptide identification.** Our model uses an encoder-decoder transformer to predict peptide sequences from mass spectra. The decoder predicts the next residue mass based on the spectrum encoding, precursor mass, and previous predictions. During training, the model alternates between experimental and simulated spectra, and is optimized using MSE loss. An adversarial discriminator, trained with BCE loss, encourages domain-invariant encodings by distinguishing simulated from experimental spectra.

For each prediction step, the decoder is conditioned not only on the encoded spectrum and precursor information (mass and charge) but also on the masses of previously predicted residues and the remaining precursor mass, similar to [12]. This setup enables autoregressive decoding of arbitrary residue masses consistent with the spectrum.

To map predicted masses back to residue tokens, we use an extendable mass lookup table, similar to the approach in Contranovo [12]. Specifically, we map each predicted mass to the PTM–amino acid combination that most closely matches its value. Token probabilities are computed via a softmax over the negative absolute differences between the predicted mass and each entry in the table. By extending this set post-training (e.g., by incorporating additional PTMs), the model can predict tokens beyond those encountered during training.

3.2.2 GAN-Inspired Latent Alignment

To align representations of experimental and simulated spectra in a shared latent space, we adopt a generative adversarial network (GAN)-inspired framework. The generator in this setup is the transformer encoder, which encodes input spectra into latent vectors. These representations are mean-pooled and passed to a discriminator—a three-layer feed-forward neural network (FNN) with Leaky ReLU activations—adapted from the discriminator used by Wu et al. [37]. The discriminator is trained to distinguish between embeddings of experimental and simulated spectra, while the DNPS model is adversarially regularized by the discriminator's loss, thereby encouraging the encoder to learn modality-invariant representations.

3.3 Mulit-Task Training Strategy

We train the model on a mixture of simulated and experimental spectra, sampling balanced batches from both sources. The mass prediction model is optimized using a mean squared error (MSE) loss $\mathcal{L}_{Reg}$ between predicted and ground truth residue masses. To train the adversarial extension of the multi-task learning (MTL) model, we incorporate an additional regularization term based on the discriminator's binary cross-entropy (BCE) loss $\mathcal{L}_D$. The MTL model is trained on a composite loss $\mathcal{L}_{Adv}$, defined as a linear combination of the mass regression loss $\mathcal{L}_{Reg}$ (weight 1) and the discriminator loss $\mathcal{L}_D$ (weight -50).

The MTL model weights are initialized with weights from a model pre-trained solely on simulated spectra from our simplified simulation approach for 80,000 steps. We train for 600,000 steps with a batch size of 64 (approximately 1.5 epochs) on a single A40 GPU with 8 CPU cores and 60 GB of CPU RAM for approximately 2 days. The learning rate is set to 4×10^{-4} for the mass prediction DNPS model and 1×10^{-6} for the discriminator. The validation set was evaluated every 50,000 training steps, and the final model corresponds to the checkpoint with the lowest validation loss.

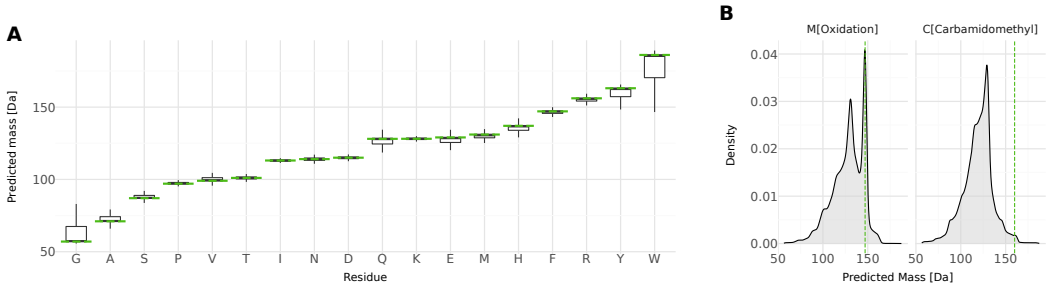

Figure 2: **Mass regression transformer evaluation on seen and unseen PTM-amino acid combinations during training. A**, Predicted masses on test set for all amino acids present in the training split of the MassIVE-KB V1 dataset. The horizontal dashed lines indicate the target mass, i.e., the true mass of the residue. Outliers are not displayed. **B**, Distribution of the predicted mass for the two PTM-amino acid combinations, methionine oxidation and cysteine carbamidomethylation, withheld from the training split of the MassIVE-KB V1 dataset. The vertical dashed lines indicate the target masses.

4 Experiments

To enable de novo peptide sequencing (DNPS) models to generalize beyond the set of amino acids and post-translational modifications (PTMs) observed during training, we reformulated the task as a continuous mass prediction problem. Instead of selecting residues from a fixed vocabulary, our model predicts a scalar mass for each peptide position, allowing for the potential inference of arbitrary or novel modifications. These predicted masses are then mapped to residue identities via a flexible, extendable lookup table. To encourage generalization, we adopt a multi-task training strategy that combines complex experimental spectra, limited to a known set of PTMs, with simulated spectra engineered to include a broad range of synthetic modifications. Additionally, we introduce a GAN-inspired architecture, which aims to align latent representations of real and simulated spectra and improve the model's ability to bridge between the two domains.

4.1 Evaluation Metrics

To assess the model's ability to generalize to modified amino acids not seen during training, we excluded all spectra containing methionine oxidation and cysteine carbamidomethylation from the training and validation sets—effectively treating these modifications as unseen during evaluation.

We evaluated performance at two levels: mass accuracy and residue identification. Mass accuracy was measured as the absolute difference between the predicted and ground truth masses for each residue. Residue identification was assessed by mapping each predicted mass to the nearest entry in a lookup table containing all amino acid and PTM combinations present in the MassIVE-KB v1 dataset. The predicted residue was then compared to the ground truth to compute recall at the amino acid level.

For evaluation, we ran the model in inference mode using teacher forcing. In this setup, the model is supplied with the ground truth masses of all previously decoded residues at each prediction step. This prevents error accumulation from incorrect predictions and isolates the model's ability to predict each residue mass independently. Additionally, by enforcing the correct number of decoding steps, teacher forcing ensures that the predicted and ground truth residue sequences are aligned, facilitating direct comparison of their respective masses.

4.2 Main Results

4.2.1 Mass Regression Decoder Confidently Predicts Masses Seen During Training

Our results indicated that recasting peptide sequencing as a mass prediction task preserved the model's ability to infer residue identities. The model reliably predicted the masses of unmodified residues seen during training on the defined test set (Fig. 2A). Across all unmodified residues seen during training, our model achieved a median absolute error of approximately 0.56 Da on the test

set (Pearson correlation coefficient: 0.89, P-value<0.05). Errors ranged from 1.10 Da for tryptophan (W) to 0.39 Da for glycine (G). This precision allowed the model to distinguish residues effectively, resulting in an overall amino acid-level recall of 62.37%. Because we assigned predicted masses to the closest matching residue in a lookup table, the model performed better for amino acids that were well-separated in mass. For example, arginine (R) was correctly recalled in approximately 79.91% of cases. In contrast, the model struggled with residues whose masses are close to others, such as lysine (K), which had a recall of only 26.97%—despite having a lower median error (0.48 Da) than arginine (0.61 Da). While these recall values were somewhat lower than those reported by classification-based transformer models such as AdaNovo [9], this was expected given the nature of our approach. Predicting scalar masses imposes a stricter requirement for precision: small deviations could lead to mismatches when mapping back to discrete residue tokens. In contrast, models like Casanovo [11] benefited from the flexibility of learned embedding spaces, where similar residues could be placed further apart to ease classification. Despite this inherent challenge, our results demonstrated that the mass-regression decoder effectively learned accurate mass representations for residues seen during training, providing a viable foundation for generalization to modified residues beyond the training set.

While the model successfully predicted the masses of residues seen during training, capturing a generalizable and interpretable biochemical feature, it struggled to generalize to truly novel modifications (Fig. 2B). For example, predictions for cysteine carbamidomethylation showed no enrichment near the correct mass, with the distribution shifted toward lower values and a recall of only 0.6%. A contributing factor to the failed extrapolation might be that all cysteines in the data were modified, meaning the model never encountered an unmodified cysteine during training. In contrast, predictions for methionine oxidation were modestly enriched around its correct mass, achieving a recall of 14.18%. However, this perhaps reflected a memorization artifact: the mass of methionine oxidation ($\sim$147.04 Da) was only $\sim$0.03 Da less than phenylalanine (F), a residue included in training, indicating the model might simply be reproducing familiar masses. The observed bias toward masses seen during training was consistent with trends in generalized zero-shot learning [38]. Since the training objective does not explicitly encourage extrapolation to unseen masses, the model was not incentivized to learn a truly continuous mass space. Instead, it might implicitly treat the task as a form of multi-class classification, where the "classes" were the residue masses encountered during training. As a result, the model could memorize these masses and reproduce them at inference time, without being penalized by the loss function for failing to predict novel ones.

4.2.2 Mass Regression Model Can Solve Simulated Open-DNPS Problem

To mitigate the model's bias toward training-set residue masses—an issue arising from the limited diversity of modifications in experimental spectra—we explored the use of simulated spectra spanning a broader amino acid–PTM search space. In these simulations, peptides were constructed by sampling random residue masses to mimic modified amino acids, thereby preventing memorization. We designed three simulation levels of increasing complexity and trained a separate model on each. Across all settings, the model successfully learned to predict the correct masses (Fig. 3A, B), though performance declined as simulation complexity increased (Pearson correlation coefficients of 1.0, 0.99 and 0.97 with two-sided P-values<0.05). Introducing a variable number of peaks led to higher mass prediction errors compared to simulations with a fixed number of peaks (Fig. 3B), as expected from the increased difficulty. Interestingly, the model remained robust when noise peaks and realistic fragmentation processes were introduced. However, these settings caused the MSE loss to more than double, largely due to occasional large prediction errors (108.96 vs. 263.98). This is likely because the simulation emulated missing peaks (although never both of the complementary b- and y-ion), causing the model to infer residue masses from unrelated fragments, leading to strong deviations and quadratic penalties from the MSE loss.

Although the model performed well on simulated spectra, it failed to generalize to experimental spectra and accurately predict real residue masses (Pearson correlation coefficients of -0.11, 0.19 and 0.04 with two-sided P-values<0.05). We reasoned that the domain shift between simulated and real data was too substantial, leading models trained on simplified simulations to break down entirely when applied to real spectra (Fig. 3C). These models produced nearly identical mass distributions across all residues, indicating a lack of residue-specific signal. The model trained on the most realistic simulation—incorporating both noise peaks and fragmentation mechanics—showed some limited signs of generalization to experimental spectra (Fig. 3C). It produced slightly differentiated mass

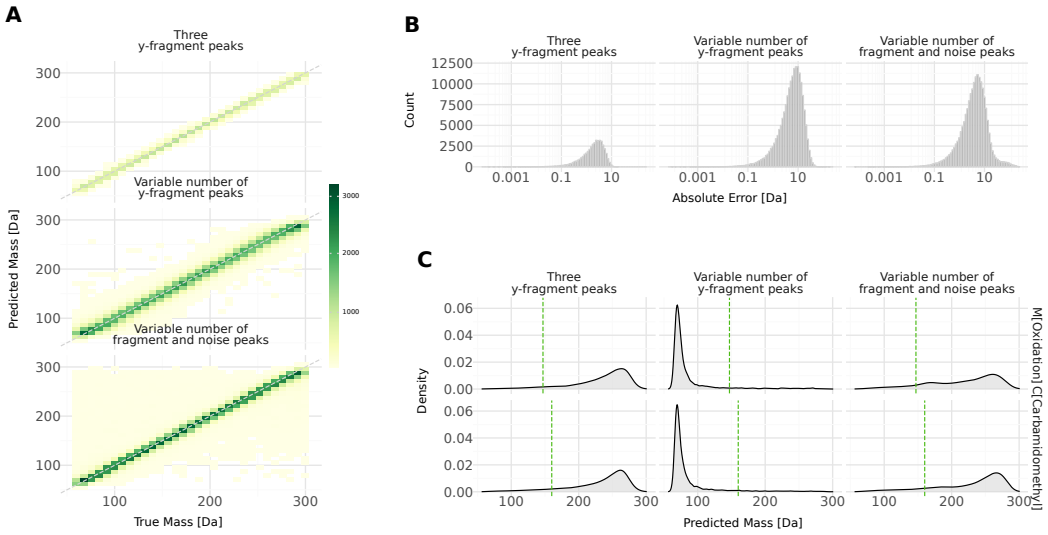

Figure 3: **Mass regression transformer on simulated data**. **A**, Predicted against true underlying simulated residue masses for three data simulation strategies (facets): (i) spectra with three peaks with arbitrary mass distance and precursor charge=1. (ii) a variable number of peaks with arbitrary mass distances and precursor charge=1 (iii) spectra with a variable number of peaks corresponding to b- and y-fragments, precursor charges>1, and noise peaks. **B**, Absolute errors between the predicted masses and true masses for the different data simulation strategies (facets). **C**, Distribution of predicted masses for experimental data from MassIVE-KB V1 containing methionine oxidation and cysteine carbamidomethylation residues. The vertical dashed lines indicate the target masses.

distributions for individual residues, with subtle enrichments around masses offset by approximately 18 Da from the target values. This offset aligned with the mass of a water molecule ($\sim$18 Da), which distinguishes y-ions from b-ions, and may reflect confusion between ion series in experimental spectra. While this suggested that the model was extracting some transferable features from the realistic simulations, the extent of generalization remained minimal. Performance on experimental data remained inadequate, with recall for cysteine carbamidomethylation and methionine oxidation residues reaching only 0.84% and 1.19%, respectively. These results highlight that, although realistic simulation improved alignment with experimental characteristics, simulated data alone were insufficient for teaching the model to handle the complexity of real-world spectra and unseen modifications.

4.2.3 Multi-Task Learning Improves Generalization

We explored Multi-Task Learning (MTL) as a strategy to improve the model's ability to generalize, particularly to unseen post-translational modifications. MTL has proven effective in various domain adaptation contexts [39–41], and we adapted it by training the model on a balanced combination of real and simulated spectra. The rationale was twofold: training on real experimental spectra encourages the model to learn the inherent complexity and noise characteristics of true mass spectrometry data, while training on simulated spectra—which include a diverse set of unrestricted residue masses—pushes the model to generalize beyond the limited set of modifications seen in the real data. By combining both sources, the model was encouraged to learn features that were robust across domains while being flexible enough to infer arbitrary modifications.

The MTL model improved performance on simulated spectra (Pearson correlation coefficient: 0.98 with two-sided P-value<0.05, Fig. 4A) but sacrificed precision on seen residues in experimental spectra compared to the model trained exclusively on experimental data (Pearson correlation coefficient: 0.83 with two-sided P-value<0.05, Fig. 4B, median absolute errors of 3.17 Da vs. 0.56 Da). On the other hand, the MTL model outperformed the model trained solely on the simulated data with a lower median absolute error of 2.95 Da compared to 4.25 Da. The imbalance between the MTL model's performance on simulated and experimental spectra might stem from the loss formulation:

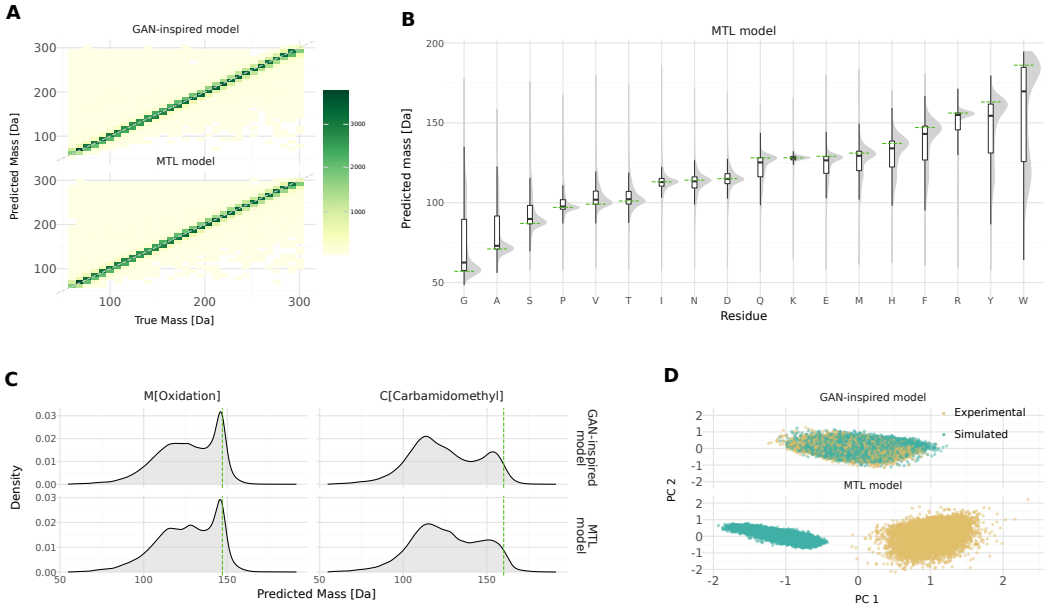

Figure 4: **Multi-task learning and adversarial-inspired model for PTM-amino acid mass prediction**. **A**, Predicted against the true masses on simulation data for the two different models (facets): (i) model using multi-task learning scheme with a mixture of experimental and simulated spectra and (ii) extended with an adversarial loss term that is obtained by training a discrimination module that predicts whether a spectrum is simulated or not. **B**, Distribution of predicted masses for all amino acids seen during training. The horizontal lines indicate the true target masses. **C**, Distribution of the predicted masses of the two unseen residues, cysteine carbamidomethylation and methionine oxidation, which were withheld from the training data. The vertical lines indicate the target masses. **D**, Two-dimensional PCA projection of pooled spectrum encoder embeddings for the two different models (facets).

since simulated spectra were easier to learn from, the model might prioritize reducing their loss, which offered a more efficient path to minimizing the overall objective, potentially at the expense of generalizing to real spectra. This behavior contrasted with the ideal MTL outcome, where both tasks mutually benefit from shared representation learning [42].

While the MTL model showed improved performance on simulated spectra, this came at the cost of reduced accuracy on known residues in experimental data. However, this trade-off coincided with emerging signs of generalization to unseen residues excluded from training (Fig. 4C). Notably, the peak near the target mass for methionine oxidation was more distinct and sharper than in the model trained without MTL (Fig. 2B and Fig. 4C), indicating increased confidence in these predictions. For the unseen cystein carbamidomethylation, the MTL model's predicted masses shifted closer to the target mass, with a pronounced peak just below it—contrasting sharply with the model trained only on experimental data. This suggests that MTL enables the model to better generalize and recover evidence for unseen modifications that were previously overlooked.

Despite improved generalization, the MTL model still showed some bias toward masses seen during training, though to a lesser extent. For instance, the secondary peak in predicted masses for cystein carbamidomethylation aligned with the amino acids (iso)leucine (I, L $\sim$113 Da), asparagine (N $\sim$114 Da), and aspartic acid (D $\sim$115 Da). Consequently, absolute prediction quality for unseen residues remained limited. While recall for cystein carbamidomethylation improved fivefold compared to the model trained only on experimental data, it remained low at about 3.18%. Recall for methionine oxidation increased only marginally, from 14.18% to 14.41%.

4.2.4 Adversarial Loss For Common Latent-Space For Spectrum Embedding

Although the MTL model performed well on simulated data, it was still biased toward seen masses. To investigate whether simulated data was effectively leveraged during training, we analyzed the embeddings with a 2D PCA projection (Fig. 4D). The embeddings for experimental and simulated spectra formed distinct clusters, suggesting that the transformer-based mass regression decoder could easily differentiate between the two data types. This separation might have caused the model to treat simulated spectra differently, reducing their effectiveness in helping the model generalize to realistic, unseen masses.

To reduce the separation between simulated and experimental spectra, we incorporated adversarial learning by adding a lightweight binary classifier to our MTL model that predicted whether a spectrum was simulated or experimental based on its encoding. The classification loss was subtracted from the MTL model's MSE loss to encourage a shared latent space for both types of spectra. Following the addition of this adversarial loss, the embeddings of simulated and experimental spectra no longer separated clearly in PCA space (Fig. 4D). Moreover, the discriminator's binary cross entropy loss during training plateaued at around 0.68, meaning its predicted probabilities lay consistently around 0.5 (-log(0.5)$\approx$0.69). However, the loss and PCA alone did not provide definitive evidence that the model could not still distinguish between the two spectrum types. This was further supported by the discriminator achieving an area under the ROC curve of around 0.78.

Unfortunately, the addition of the adversarial loss term did not lead to a noticeable improvement in the MTL model's ability to generalize to unseen residues. The distributions of predicted masses for unseen residues remained very similar between the MTL models with and without adversarial loss (Pearson correlation coefficient: 0.82 with two-sided P-value<0.05, Fig. 4C). Additionally, both models showed comparable recall rates for methionine oxidation (3.18% vs. 3.10%) and cystein carbamidomethylation (14.41% vs. 15.11%), along with similar MSE losses.

5 Conclusion and Future Work

We present a novel framework for de novo peptide sequencing that reformulates the task as mass regression rather than discrete classification. This shift enables the model to move beyond a fixed vocabulary of amino acids and modifications, allowing it to predict residues not encountered during training. Our approach combines transformer-based mass prediction with a multi-task learning setup trained on both experimental and simulated spectra. To encourage domain-invariant representations, we incorporate adversarial learning that aligns the spectrum encodings across data sources. Results demonstrate promising generalization to previously unseen modifications, marking a step toward more flexible and open-ended peptide sequencing. By enabling the discovery of novel or rare peptide modifications, this work may support future advancements in biomedical research.

Limitations. This work serves as a proof of concept rather than a production-ready system. Although we show that mass-based modeling can overcome vocabulary constraints, the current framework has only been evaluated under teacher-forced decoding. Nonetheless, it provides a foundation for developing fully open de novo sequencing models that support more reliable and practical applications.

Future directions. Several avenues merit further investigation: exploring the tradeoff between scalar mass prediction and vector-based encodings, improving robustness to noise in simulated spectra, and developing loss functions that better prioritize precision. The role of adversarial learning also warrants deeper analysis, particularly whether the model implicitly treats simulated data differently. Incorporating high-fidelity simulated spectra (e.g., from models like Prosit [43, 44]) and more advanced decoding strategies could further improve performance. Together, these advances could enable DNPS models that generalize across biological contexts, capturing both known and novel peptide modifications with greater reliability.

References

[1] David Eisenberg, Edward M Marcotte, Ioannis Xenarios, and Todd O Yeates. Protein function in the post-genomic era. *Nature*, 405(6788):823–826, 2000. Publisher: Nature Publishing Group UK London.

[2] Engelbert Buxbaum and others. *Fundamentals of protein structure and function*. Springer, 2nd edition, 2007.

[3] Dagmar Klostermeier and Markus G. Rudolph. *Biophysical Chemistry*. CRC Press, 1st edition, 2017.

[4] Yaoyang Zhang, Bryan R Fonslow, Bing Shan, Moon-Chang Baek, and John R Yates III. Protein analysis by shotgun/bottom-up proteomics. *Chemical reviews*, 113(4):2343–2394, 2013. Publisher: ACS Publications.

[5] Ruedi Aebersold and Matthias Mann. Mass-spectrometric exploration of proteome structure and function. *Nature*, 537(7620):347–355, 2016. Publisher: Nature Publishing Group UK London.

[6] Vlado Dančík, Theresa A Addona, Karl R Clauser, James E Vath, and Pavel A Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of computational biology*, 6 (3-4):327–342, 1999. Publisher: Mary Ann Liebert, Inc.

[7] J Alex Taylor and Richard S Johnson. Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry*, 11(9): 1067–1075, 1997. Publisher: Wiley Online Library.

[8] Jimmy K Eng, Brian C Searle, Karl R Clauser, and David L Tabb. A face in the crowd: recognizing peptides through database search. *Molecular & Cellular Proteomics*, 10(11), 2011. Publisher: ASBMB.

[9] Jun Xia, Shaorong Chen, Jingbo Zhou, Shan Xiaojun, Wenjie Du, Zhangyang Gao, Cheng Tan, Bozhen Hu, Jiangbin Zheng, and Stan Z Li. Adanovo: Towards robust\emph {De Novo} peptide sequencing in proteomics against data biases. *Advances in Neural Information Processing Systems*, 37:1811–1828, 2024.

[10] Siyu Wu, Zhongzhi Luan, Zhenxin Fu, Qunying Wang, and Tiannan Guo. Biatnovo: A self-attention based bidirectional peptide sequencing method. *bioRxiv*, pages 2023–05, 2023.

[11] Melih Yilmaz, William E Fondrie, Wout Bittremieux, Carlo F Melendez, Rowan Nelson, Varun Ananth, Sewoong Oh, and William Stafford Noble. Sequence-to-sequence translation from mass spectra to peptides with a transformer model. *Nature communications*, 15(1):6427, 2024. Publisher: Nature Publishing Group UK London.

[12] Zhi Jin, Sheng Xu, Xiang Zhang, Tianze Ling, Nanqing Dong, Wanli Ouyang, Zhiqiang Gao, Cheng Chang, and Siqi Sun. Contranovo: A contrastive learning approach to enhance de novo peptide sequencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 144–152, 2024.

[13] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31): 8247–8252, 2017.

[14] Cheng Ge, Yi Lu, Jia Qu, Liangxu Xie, Feng Wang, Hong Zhang, Ren Kong, and Shan Chang. Deps: an improved deep learning model for de novo peptide sequencing. *arXiv preprint arXiv:2203.08820*, 2022.

[15] Ruitao Wu, Xiang Zhang, Runtao Wang, and Haipeng Wang. Denovo-gcn: De novo peptide sequencing by graph convolutional neural networks. *Applied Sciences*, 13(7):4604, 2023.

[16] Zeping Mao, Ruixue Zhang, Lei Xin, and Ming Li. Mitigating the missing-fragmentation problem in de novo peptide sequencing with a two-stage graph-based deep learning model. *Nature Machine Intelligence*, 5(11):1250–1260, 2023.

[17] Kevin Eloff, Konstantinos Kalogeropoulos, Amandla Mabona, Oliver Morell, Rachel Catzel, Esperanza Rivera-de Torre, Jakob Berg Jespersen, Wesley Williams, Sam PB van Beljouw, Marcin J Skwark, et al. Instanovo enables diffusion-powered de novo peptide sequencing in large-scale proteomics experiments. *Nature Machine Intelligence*, pages 1–15, 2025.

[18] Joel Lapin, Alfred Nilsson, Mathias Wilhelm, and Lukas Käll. Pairwise attention: Leveraging mass differences to enhance de novo sequencing of mass spectra. *bioRxiv*, pages 2025–03, 2025.

[19] Kaiyuan Liu, Yuzhen Ye, Sujun Li, and Haixu Tang. Accurate de novo peptide sequencing using fully convolutional neural networks. *Nature Communications*, 14(1):7974, 2023.

[20] Jun Xia, Sizhe Liu, Jingbo Zhou, Shaorong Chen, Hongxin Xiang, Zicheng Liu, Yue Liu, and Stan Z Li. Bridging the gap between database search and de novo peptide sequencing with searchnovo. *bioRxiv*, pages 2024–10, 2024.

[21] Korrawe Karunratanakul, Hsin-Yao Tang, David W Speicher, Ekapol Chuangsuwanich, and Sira Sriswasdi. Uncovering thousands of new peptides with sequence-mask-search hybrid de novo peptide sequencing framework. *Molecular & Cellular Proteomics*, 18(12):2478–2491, 2019.

[22] Daniela Klaproth-Andrade, Johannes Hingerl, Yanik Bruns, Nicholas H Smith, Jakob Träuble, Mathias Wilhelm, and Julien Gagneur. Deep learning-driven fragment ion series classification enables highly precise and sensitive de novo peptide sequencing. *Nature Communications*, 15 (1):151, 2024.

[23] Tingpeng Yang, Tianze Ling, Boyan Sun, Zhendong Liang, Fan Xu, Xiansong Huang, Linhai Xie, Yonghong He, Leyuan Li, Fuchu He, et al. Introducing π-helixnovo for practical large-scale de novo peptide sequencing. *Briefings in Bioinformatics*, 25(2):bbae021, 2024.

[24] Xiang Zhang, Tianze Ling, Zhi Jin, Sheng Xu, Zhiqiang Gao, Boyan Sun, Zijie Qiu, Nanqing Dong, Guangshuai Wang, Guibin Wang, et al. π-primenovo: An accurate and efficient non-autoregressive deep learning model for de novo peptide sequencing. *bioRxiv*, pages 2024–05, 2024.

[25] Rui Qiao, Ngoc Hieu Tran, Lei Xin, Xin Chen, Ming Li, Baozhen Shan, and Ali Ghodsi. Computationally instrument-resolution-independent de novo peptide sequencing for high-resolution devices. *Nature Machine Intelligence*, 3(5):420–425, 2021.

[26] Joerg Seidler, Nico Zinn, Martin E Boehm, and Wolf D Lehmann. De novo sequencing of peptides by MS/MS. *Proteomics*, 10(4):634–649, 2010. Publisher: Wiley Online Library.

[27] Shahin Ramazi and Javad Zahiri. Post-translational modifications in proteins: resources, tools and prediction methods. *Database*, 2021:baab012, 2021. Publisher: Oxford University Press UK.

[28] Xiang Zhang, Tianze Ling, Zhi Jin, Sheng Xu, Zhiqiang Gao, Boyan Sun, Zijie Qiu, Nanqing Dong, Guangshuai Wang, Guibin Wang, and others. -PrimeNovo: An Accurate and Efficient Non-Autoregressive Deep Learning Model for De Novo Peptide Sequencing. *bioRxiv*, pages 2024–05, 2024. Publisher: Cold Spring Harbor Laboratory.

[29] Rovshan G Sadygov, Daniel Cociorva, and John R Yates III. Large-scale database searching using tandem mass spectra: looking up the answer in the back of the book. *Nature methods*, 1 (3):195–202, 2004.

[30] Katalin F Medzihradszky and Robert J Chalkley. Lessons in de novo peptide sequencing by tandem mass spectrometry. *Mass spectrometry reviews*, 34(1):43–63, 2015. Publisher: Wiley Online Library.

[31] Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. Peaks: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry*, 17(20):2337–2342, 2003.

[32] Matthias Mann and Matthias Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical chemistry*, 66(24):4390–4399, 1994. Publisher: ACS Publications.

[33] Ari Frank and Pavel Pevzner. Pepnovo: de novo peptide sequencing via probabilistic network modeling. *Analytical chemistry*, 77(4):964–973, 2005.

[34] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31): 8247–8252, 2017. Publisher: National Acad Sciences.

[35] Tingpeng Yang, Tianze Ling, Boyan Sun, Zhendong Liang, Fan Xu, Xiansong Huang, Linhai Xie, Yonghong He, Leyuan Li, Fuchu He, and others. Introducing -HelixNovo for practical large-scale de novo peptide sequencing. *Briefings in Bioinformatics*, 25(2):bbae021, 2024. Publisher: Oxford University Press.

[36] Mingxun Wang, Jian Wang, Jeremy Carver, Benjamin S Pullman, Seong Won Cha, and Nuno Bandeira. Assembling the community-scale discoverable human proteome. *Cell systems*, 7(4): 412–421, 2018. Publisher: Elsevier.

[37] Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. Adversarial sparse transformer for time series forecasting. *Advances in neural information processing systems*, 33:17105–17115, 2020.

[38] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. Publisher: IEEE.

[39] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1446–1455, 2019.

[40] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 762–771, 2018.

[41] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8160–8171, 2019.

[42] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021. Publisher: IEEE.

[43] Mathias Wilhelm, Daniel P Zolg, Michael Graber, Siegfried Gessulat, Tobias Schmidt, Karsten Schnatbaum, Celina Schwencke-Westphal, Philipp Seifert, Niklas de Andrade Krätzig, Johannes Zerweck, and others. Deep learning boosts sensitivity of mass spectrometry-based immunopeptidomics. *Nature communications*, 12(1):3346, 2021. Publisher: Nature Publishing Group UK London.

[44] Siegfried Gessulat, Tobias Schmidt, Daniel Paul Zolg, Patroklos Samaras, Karsten Schnatbaum, Johannes Zerweck, Tobias Knaute, Julia Rechenberger, Bernard Delanghe, Andreas Huhmer, et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature methods*, 16(6):509–518, 2019.

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: The claims made in the abstract and introduction are accurately reflected in both the Methods and Experiments sections of our paper.

   Guidelines:

   - The answer NA 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 NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have clearly stated the limitations of our work in the Experiments sections and summarized them in the Conclusion section of our paper.

   Guidelines:

   - The answer NA 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: [NA]

Justification: The paper does not include any theoretical results.

Guidelines:

- The answer NA 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: All information needed to reproduce the proposed approach is described in the Methods section of the paper.

Guidelines:

- The answer NA 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: The data (MassIVE-KB v1) is publicly available and not created by us. The code is publicly available and stored in a (currently private) GitHub repository, that will be made public upon release of the camera-ready version. The anonymized code is included with the supplemental material of the submission as a .zip file.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All training and test details have been described in the Methods section "Training strategy". Furthermore the code is provided.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: We statistically assess the correlation between predicted and ground truth continuous values with a two-sided Pearson test and report the obtained correlation coefficients along with P-values. Other statistical measures (e.g., error bars) were omitted due to computational constraints.

Guidelines:

- The answer NA 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: All computer resources are disclosed in the Methods section of the paper.

   Guidelines:

   - The answer NA 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: This work conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The positive societal impact has been addressed in the Conclusion section. There is no negative societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA 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: We have cited related publications.

Guidelines:

- The answer NA 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: We will make our GitHub repository available which includes documentation in our paper. Model trained weights will also become available.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects. We only use publicly available mass spectrometry data.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs are used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.