AdaNovo: Towards Robust *De Novo* Peptide Sequencing in Proteomics against Data Biases

Jun Xia[1*], **Shaorong Chen**[1*], **Jingbo Zhou**[1*], **Xiaojun Shan**[2], **Wenjie Du**[3], **Zhangyang Gao** [1], **Cheng Tan**[1], **Bozhen Hu**[1], **Jiangbin Zheng**[1], **Stan Z. Li**[1†]
[1]School of Engineering, Westlake University
[2]University of California San Diego [3]University of Science and Technology of China
{xiajun, stan.zq.li}@westlake.edu.cn

Abstract

Tandem mass spectrometry has played a pivotal role in advancing proteomics, enabling the high-throughput analysis of protein composition in biological tissues. Despite the development of several deep learning methods for predicting amino acid sequences (peptides) responsible for generating the observed mass spectra, training data biases hinder further advancements of *de novo* peptide sequencing. Firstly, prior methods struggle to identify amino acids with Post-Translational Modifications (PTMs) due to their lower frequency in training data compared to canonical amino acids, further resulting in unsatisfactory peptide sequencing performance. Secondly, various noise and missing peaks in mass spectra reduce the reliability of training data (Peptide-Spectrum Matches, PSMs). To address these challenges, we propose AdaNovo, a novel and domain knowledge-inspired framework that calculates Conditional Mutual Information (CMI) between the mass spectra and amino acids or peptides, using CMI for robust training against above biases. Extensive experiments indicate that AdaNovo outperforms previous competitors on the widely-used 9-species benchmark, meanwhile yielding 3.6% - 9.4% improvements in PTMs identification. The code for reproducing the results is available at: `https://github.com/Westlake-OmicsAI/adanovo_v1`.

1 Introduction

Proteomics research focuses on large-scale studies to characterize the proteome, the entire set of proteins in a living organism. Tandem mass spectrometry serves as the only high-throughput method to analyze the protein composition in complex biological samples, playing an essential role in drug target discovery [12], PTMs discovery [16] and precision medicine [28]. Peptide sequencing, i.e., predicting the peptide sequence for each observed mass spectrum, is at core of proteomics [1].

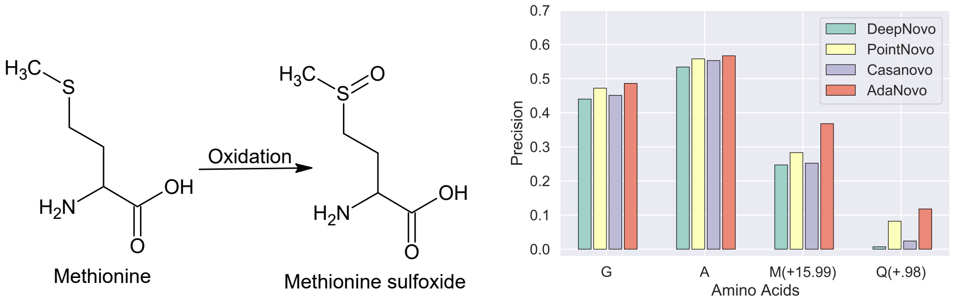
Currently, two mainstream methods are employed in peptide sequencing: database search and *de novo* peptide sequencing. The database search approaches [18] compare the observed spectrum against the spectra in a pre-constructed PSMs database and pick the peptide sequence of the most similar spectrum as the identification result. Obviously, database search cannot sequence the peptides out of the database. In contrast, *de novo* peptide sequencing deduces peptide sequence without prior knowledge of the database and thus it is essential in applications where the database is not available, such as antibody sequencing[26], human leukocyte antigen neoantigen sequencing[25], and identification of new proteins and peptides which are missing from the database[30].

[*] Equal contribution.

[†] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0057



(a) (b)

Figure 1: **(a)**: An example of PTMs (oxidation of methionine). **(b)**: Comparisons of previous *de novo* sequencing methods in terms of amino acid-level precision. 'G' and 'A' denote Glycine and Alanine, respectively. Both of them are canonical amino acids. 'M(+15.99)' and 'Q(+.98)' represent oxidation of methionine and deamidation of glutamin, both of which are modified amino acids (the amino acids with PTMs). The results are reported using the human dataset in 9-species benchmark as test set.

Since the early 1990s, *de novo* sequencing methods based on the graph theory [2, 8], Hidden Markov Model [6], or dynamic programming [3, 15, 7] were developed to score peptide sequences against observed spectra. With the prosperity of deep learning, some researchers train the deep neural networks with mass spectrum as the input and peptide sequence as the label [27, 19, 35]. Although these methods have achieved notable progress, as shown in Figure 1, we observe that they struggle to identify the amino acids with PTMs (such as the oxidation of methionine shown in Figure 1(a)), further leading to low peptide sequencing performance. On the other hand, the identification of amino acids with PTMs holds significant biological importance because PTMs plays a pivotal role in elucidating protein function and studying disease mechanisms [4]. Additionally, some expected peaks in mass spectra may be missing due to instrument malfunction or multiple cleavage events occurring on the peptides, and some additional peaks may undesirably appear in the spectrum, created by instrument noise or non-peptide molecules in the biological samples[17]. All of these make the spectra and peptides labels for training being poorly matched.

To address above issues, we propose a novel framework, AdaNovo, to calculate the conditional mutual information (CMI) between the spectrum and each amino acid in its peptide label. This is inspired by the domain knowledge that the mass shifts of PTMs over canonical amino acids are only manifested in the mass spectrum. The CMI can measure the importance of different target amino acids by their dependence on the source spectrum. Based on the amino acid-level CMI, we can obtain the Mutual Information (MI) between the spectrum and the entire peptide to measure the matching level of each spectrum-peptide pair in the training PSM data. Subsequently, we design a robust training approach based on both the amino acid- and PSM-level CMI or MI, which re-weights the training losses of the corresponding amino acids adaptively.

The extensive experiments on the 9-species benchmark [27] indicate that AdaNovo generally outperforms state-of-the-art *de novo* peptide sequencing methods in amino acid-level or peptide-level precision and demonstrates significantly higher precision in identifying the amino acids with PTMs.

2 Background and Related Work

As shown in Figure 2, in a standard protein identification workflow of shotgun proteomics [32], proteins undergo initial digestion by enzymes, yielding a mixture of peptides. The peptides are then separated using liquid chromatography. Each charged peptide is analysed by mass spectrometer, which produces the first scan (MS1) spectra, displaying the mass-to-charge (*m/z*) ratio of the intact peptide. And then, each peptide will be fragmented in the mass spectrometer, and each generated second scan (MS2) spectrum comprises a collection of peaks. Each peak is tuple constitutes *m/z* value and an associated intensity value. The core of the above pipeline is the **peptide sequencing**, where we aim to predict the peptide sequence using the observed MS2 spectrum and the corresponding precursor information (mass and charge of the intact peptide). Finally, we can infer the entire protein sequence using assembly methods [14]. There exist two lines of works for peptide sequencing.

https://doi.org/10.52202/079017-0057

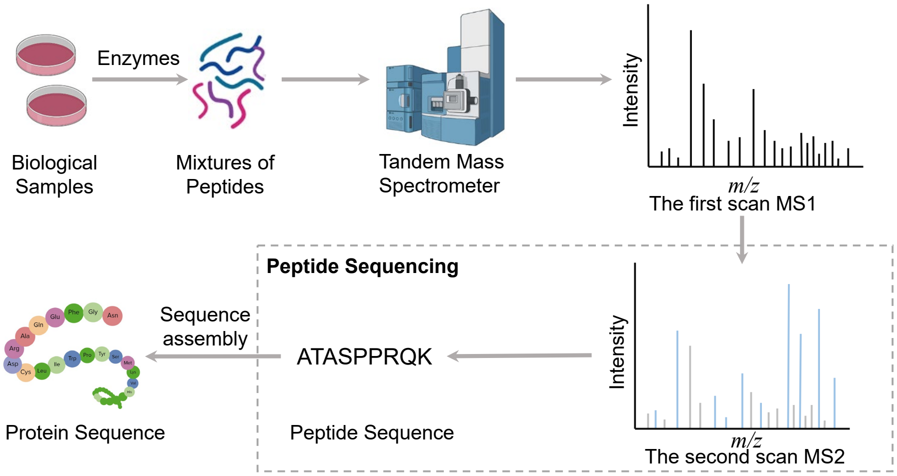


Figure 2: The identification workflow of shotgun proteomics [32]. The peptide sequencing task is to predict the peptide sequence (e.g., ATASPPRQK) for the observed MS2 spectrum, where peaks in blue are signal peaks (real ions) and grey peaks denote noisy ones. The spectrum annotation are obtained with ProteomeXchange [31].

The first line is **database search**, where we compare the observed mass spectra against the theoretical fragmentation mass spectra of peptide sequences in the database and pick the peptide sequence with the highest matching score as the result. Typical methods and tools include SEQUEST [5], pFind [11], MSFragger [10] and Open-pFind [23]. However, these methods cannot sequence the peptides out of the database.

The second line of works is ***de novo* peptide sequencing**, where we predict the peptide sequences for observed spectra without relying on pre-constructed databases. Initially, researchers cast the *de novo* peptide sequencing task as finding the largest path in the spectrum graph [3, 24] or compute the best sequences whose fragment ions can best interpret the peaks in the observed MS2 spectrum using Hidden Markov Model [6] or dynamic programming algorithm [15].

With the prosperity of deep learning, DeepNovo [27] is the first method applying deep neural networks to the task of *de novo* peptide sequencing. It regards the task as the image caption [22] in computer vision and incorporates the encoder-decoder architecture to predict the peptide sequence. To annotate the high-resolution mass spectrometry data, PointNovo [19] adopts an order invariant network structure for peptide sequencing. More recently, Casanovo [35] first employs a transformer encoder-decoder architecture [29] to predict the peptide sequence for the observed spectra. SearchNovo [34] integrates the strengths of database search and *de novo* sequencing to enhance peptide sequencing.

Although *de novo* peptide sequencing methods have achieved notable progress, we observe that they have difficulty in identifying the amino acids with PTMs because these amino acids occur much less frequently in datasets compared to other canonical ones. Additionally, mass spectrometry data contains a significant amount of noise. All of these make the peptides labels being less reliable. The AdaNovo model proposed in this paper effectively alleviates both of them.

3 Methods

3.1 Task Formulation

Formally, we denote mass spectrum peaks in a MS2 spectrum as $\mathbf{x} = \{(m_i, t_i)\}_{i=1}^{M}$, where each peak (m_i, t_i) forms a 2-tuple representing the *m/z* and intensity value, and M is the number of peaks that can be varied across different mass spectra. Also, we denote the precursor as $\mathbf{z} = \{(m_{prec}, c_{prec})\}$, consisting of the total mass $m_{prec} \in \mathbb{R}$ and charge state $c_{prec} \in \{1, 2, \ldots, 10\}$ of the spectrum. Additionally, we represent the peptide sequence as $\mathbf{y} = \{(y_1, y_2, \ldots, y_N)\}$, where y_i is the type of the i-th amino acid, N is the peptide length and can be varied across different peptides. $\mathbf{y}_{<j}$ means the previous amino acids sequence appearing before the index j in the peptide $\mathbf{y}$. The *de novo* peptide

https://doi.org/10.52202/079017-0057

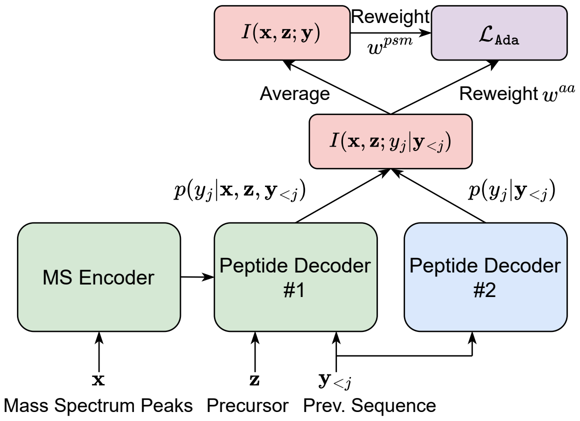


Figure 3: Schematic diagram of AdaNovo framework.

sequencing models are designed to predict the probability of each amino acid y_i given $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{y}_{<j}$:

$$P(\mathbf{y} \mid \mathbf{x}, \mathbf{z}; \theta) = \prod_{j=1}^{N} p\left(y_j \mid \mathbf{y}_{<j}, \mathbf{x}, \mathbf{z}; \theta\right), \tag{1}$$

where j is the index of each amino acid position in the peptide sequence and θ is the model parameter. In general, previous models [27, 35, 19] are optimized using the cross-entropy (CE) loss:

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\sum_{j=1}^{N} \log p\left(y_j \mid \mathbf{y}_{<j}, \mathbf{x}, \mathbf{z}; \theta\right). \tag{2}$$

During inference, these models typically predict the probabilities of target amino acids in an autoregressive manner and generate hypotheses using heuristic search algorithms like beam search [21].

3.2 Model Architectures

As shown in Figure 3, AdaNovo consists of a mass spectrum encoder (MS Encoder) and two peptide decoders (Peptide Decoder #1 and Peptide Decoder #2). All of these models are built on Transformer [29]. In order to feed the mass spectrum peaks to MS Encoder, following Casanovo[35], we regard each mass spectrum peak (m_i, t_i) as a 'word' in natural language processing and obtain the peak embedding by individually encoding its *m/z* value (m_i) and intensity value (t_i) before combining them through summation. We employ the similar embedding approach for the precursor $\mathbf{z} = \{(m_{prec}, c_{prec})\}$. As the embedding method is not our original contribution, we introduce the details in Appendix A. As for the peptide sequence, the amino acid vocabulary encompasses the 20 canonical amino acids, 3 PTMs (oxidation of methionine, deamidation of asparagine or glutamine) and a special `[stop]` token indicates the end of decoding. Peptide Decoder #1 and Peptide Decoder #2 undergo autoregressive training, wherein they receive the previous sequence $\mathbf{y}_{<j}$ prior to amino acid y_j during the prediction process. However, different from Peptide Decoder #1, Peptide Decoder #2 exclusively employs previous sequence $\mathbf{y}_{<j}$ as input because we want to calculate the conditional probability $p(y_j \mid \mathbf{y}_{<j})$, which is the prerequisite for calculating the conditional mutual information between the mass spectrum ($\mathbf{x}$ and $\mathbf{z}$) and amino acids.

3.3 Training Strategies

The training strategies consist of amino acid-level (§ 3.3.1) and PSM-level training methods (§ 3.3.2).

3.3.1 Amino Acid-level Training Objective

As mentioned above, previous *de novo* peptide sequencing models struggle to identify amino acids with PTMs because they occur much less frequently in datasets compared to other canonical amino

https://doi.org/10.52202/079017-0057

acids. Therefore, we expect to emphasize the amino acids with PTMs to improve the models' ability in identifying them during inference. This resembles the up-sampling methods in long-tailed classification where researchers emphasize samples from the tail class during training [36, 20]. We explain the reasons why these methods are unsuitable to *de novo* peptide sequencing and compare AdaNovo with them in Section 4.6. On the other hand, when predicting the amino acid with PTMs y_j, we should rely more on mass spectrometry data (peaks x and precursor z) and less on the historical predictions of previous amino acids sequence $y_{<j}$ because the mass shifts resulting from PTMs are only manifested in the mass spectrometry data. This unique attribute of PSMs data motivates us to measure the mutual information (MI) between each target amino acid (y_j) and the mass spectrum conditioned on previous amino acids, i.e., conditional mutual information (CMI) [33] between target amino acid and mass spectrum. Given that $\mathbf{x}, y_j, \mathbf{z}, \mathbf{y}_{<j}$ are drawn from the underlying random variable $X, Y_j, Z, Y_{<j}$, respectively, we can calculate the conditional mutual information (CMI) as,

$$\begin{aligned} I\left(X, Z ; Y_j \mid Y_{<j}\right) &= \mathbb{E}_{(X, Y_j, Z)}\left\{\log \left(\frac{p\left(y_j, \mathbf{x}, \mathbf{z} \mid \mathbf{y}_{<j}\right)}{p\left(y_j \mid \mathbf{y}_{<j}\right) \cdot p\left(\mathbf{x}, \mathbf{z} \mid \mathbf{y}_{<j}\right)}\right)\right\} \\ &= \mathbb{E}_{(X, Y_j, Z)}\left\{\log \left(\frac{p\left(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j}\right) \cdot p\left(\mathbf{x}, \mathbf{z} \mid \mathbf{y}_{<j}\right)}{p\left(y_j \mid \mathbf{y}_{<j}\right) \cdot p\left(\mathbf{x}, \mathbf{z} \mid \mathbf{y}_{<j}\right)}\right)\right\} \\ &= \mathbb{E}_{(X, Y_j, Z)}\left\{\log \left(\frac{p\left(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j}\right)}{p\left(y_j \mid \mathbf{y}_{<j}\right)}\right)\right\}. \end{aligned} \tag{3}$$

In this way, the CMI can be obtained with $p\left(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j}\right)$ and $p\left(y_j \mid \mathbf{y}_{<j}\right)$, which are the output of the Peptide Decoder #1 and Peptide Decoder #2, respectively. Each data point $(\mathbf{x}, \mathbf{z}, y_j)$ is independently sampled from the joint distribution uniformly. Therefore, we can measure the dependence between mass spectrometry $(\mathbf{x}, \mathbf{z})$ and y_j conditioned on $\mathbf{y}_{<j}$ using $I_j = \log\left(\frac{p(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j})}{p(y_j \mid \mathbf{y}_{<j})}\right)$. Moreover, to reduce the variances and stabilize the distribution of the amino acid-level CMI in each peptide, we normalize the CMI values in the peptide using Z-score normalization and then scale the normalized values to obtain the amino acid-level training weight w_j^{aa} for y_j,

$$w_j^{aa} = \max(s_1 \cdot \frac{I_j - \mu^{aa} + \sigma^{aa}}{\sigma^{aa}}, 0), \tag{4}$$

where μ^{aa} and σ^{aa} are the mean values and the standard deviations of all the CMI values in each peptide, and s_1 is a hyperparameter that controls the effect of amino acid-level adaptive training.

3.3.2 PSM-level Training Objective

As we introduced before, the training PSMs samples are of different matching levels because of the unexpected signal noise and missing peaks. To alleviate the negative effect of poorly matched mass spectrometry and peptide pairs and encourage the well-matched ones, we adopt the mutual information between them as a measure of matching levels. Formally,

$$\begin{aligned} I(X, Z ; Y) = \mathbb{E}_{(X, Y, Z)}\left\{\log\left(\frac{p(\mathbf{y} \mid \mathbf{x}, \mathbf{z})}{p(\mathbf{y})}\right)\right\} &= \mathbb{E}_{(X, Y, Z)}\left\{\log\left(\frac{\prod_{j=1}^{N} p\left(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j}\right)}{\prod_{j=1}^{N} p\left(y_j \mid \mathbf{y}_{<j}\right)}\right)\right\} \\ = \mathbb{E}_{(X, Y, Z)}\left\{\sum_{j=1}^{N} \log\left(\frac{p\left(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j}\right)}{p\left(y_j \mid \mathbf{y}_{<j}\right)}\right)\right\} &= \mathbb{E}_{(X, Y, Z)}\left\{\sum_{j=1}^{N} I_j\right\}. \end{aligned} \tag{5}$$

In other words, the mutual information can be derived by summarizing all the amino acid-level CMI over the peptide. Similarly, we can measure the matching level between mass spectrometrometry $(\mathbf{x}, \mathbf{z})$ and the peptide $(\mathbf{y})$ using $\mathrm{MI} = \sum_{j=1}^{N} I_j$. And then, we normalize all the MI values across all the PSMs in each mini-batch and scale the normalized values to obtain the PSM-level training weight w^{psm}.

$$w^{psm} = \max(s_2 \cdot \frac{\mathrm{MI} - \mu^{psm} + \sigma^{psm}}{\sigma^{psm}}, 0), \tag{6}$$

where μ^{psm} and σ^{psm} are the mean values and the standard deviations of the MI values of all the PSMs in each minibatch, and s_2 is a hyperparameter that controls the effect of PSM-level adaptive training. We studied the influence of s_1 and s_2 in Section 4.7.

https://doi.org/10.52202/079017-0057

3.3.3 Overall Training Objective

In the proposed method, we re-weight each target amino acid y_j with the following loss,

$$\mathcal{L}_1(\theta_1) = -\sum_{j=1}^{N} w_j \log p\left(y_j \mid \mathbf{y}_{<j}, \mathbf{x}, \mathbf{z}; \theta_1\right), \tag{7}$$

where θ_1 are the parameters of MS Encoder and Peptide Decoder #1, and

$$w_j = w_j^{aa} \cdot w^{psm}. \tag{8}$$

Additionally, Peptide Decoder #2 is trained with the following loss,

$$\mathcal{L}_2(\theta_2) = -\sum_{j=1}^{N} \log p\left(y_j \mid \mathbf{y}_{<j}; \theta_2\right), \tag{9}$$

where θ_2 are the parameters of Peptide Decoder #2. The overall training loss is,

$$\mathcal{L}_{\texttt{Ada}}(\theta_1, \theta_2) = \mathcal{L}_1(\theta_1) + \mathcal{L}_2(\theta_2). \tag{10}$$

3.4 Inference Strategies

In the inference phase, we feed the mass spectrometry to the encoder MS Encoder and the decoder Peptide Decoder #1 predicts the highest-scoring amino acid for each peptide sequence position. The decoder is then fed its preceding amino acid predictions at each decoding step. The decoding process concludes upon predicting the `[stop]` token or reaching the predefined maximum peptide length of $\ell = 100$ amino acids. We discuss the computational overhead of AdaNovo in Section 4.8.

4 Experiments

4.1 Datasets

We employ the nine-species benchmark initially introduced by DeepNovo [27]. This dataset amalgamates approximately 1.5 million mass spectra from nine distinct species, all employing the same instrument but analyzing peptides from different species. Each spectra is associated with a ground-truth peptide sequence, which comes from database search identification with a standard false discovery rate (FDR) set at 1%. Following previous works [27, 19, 37], we adopt a leave-one-out cross-validation framework. This entails training a model on eight species and testing it on the species held out for each of the nine species. We also split the eight species into training set and validation set with the ratio 9:1. This framework facilitates the testing of the model on peptide samples that have never been encountered before, which is precisely the advantage of *de novo* peptide sequencing methods over database search methods.

4.2 Evaluation Metrics

In our assessment of model predictions, we employ precision calculated at both the amino acid and peptide levels, following methodologies presented by previous works [15, 27]. We first calculates the number of matched amino acid predictions, N_{match}^{aa}, which are defined as predicted amino acids that exhibit a mass difference of $< 0.1\text{Da}$ from the real amino acids and have either a prefix or suffix with a mass difference of $\leq 0.5\text{Da}$ from the corresponding real amino acid sequence in the ground truth peptide. Amino acid-level precision is then defined as $N_{\text{match}}^{aa} / N_{\text{pred}}^{aa}$, where N_{pred}^{aa} represents the number of predicted amino acids in predicted peptide sequences. Similarly, PTMs identification precision can be formulated as $N_{\text{match}}^{ptm} / N_{\text{pred}}^{ptm}$, where N_{match}^{ptm} and N_{pred}^{ptm} denote the number of matched PTMs and predicted amino acids with PTMs, respectively. For peptide prediction, a predicted peptide is deemed a correct match only if all of its amino acids are matched. In a collection of N_{all}^{p} spectra, if a model accurately predicts N_{match}^{p} peptides, the peptide-level precision are defined as $N_{\text{match}}^{p} / N_{\text{all}}^{p}$. Kindly note that peptide-level performance measures are the primary quantifier of the model's practical utility because the goal is to assign a complete peptide sequence to each spectrum.

https://doi.org/10.52202/079017-0057

4.3 Baselines and Experimental Settings

We compare AdaNovo with protein database search tool Peaks [15] and previous *de novo* peptide sequencing methods including DeepNovo [27], Casanovo [35] and PointNovo [19]. The MS Encoder, Peptide Decoder # 1 and Peptide Decoder # 2 in AdaNovo are 9-layer Transformers, all of which come with 512 feed forward dimensions. During the training process, we used one Nvidia A100 GPU with the batchsize as 32. We set the learning rate at 0.0004 and applied a linear warm-up. For gradient updates, we used the AdamW optimizer [9]. The hyperparameters s_1 and s_2 are tuned within the range $\{0.05, 0.1, 0.3\}$ using the validation set.

4.4 Main Results

Table 1: Empirical comparison of previous models on 9-species benchmark. The best and the second best results are highlighted with **bold** and underlined, respectively. The reproduced performance of Casanovo is denoted as Casanovo (*rep.*). Following Casanovo [35], we train 5 models with different random initializations and report the standard deviation when the test dataset is Mouse or Human. The other standard deviations are not reported because it would be too computationally expensive.

Species	Peptide-level precision					Amino acid-level precision				
	DeepNovo	PointNovo	Casanovo	Casanovo (*rep.*)	AdaNovo	DeepNovo	PointNovo	Casanovo	Casanovo (*rep.*)	AdaNovo
Mouse	0.286	0.355	0.443 ± 0.019	0.449 ± 0.010	**0.493** ± 0.015	0.623	0.626	0.562 ± 0.021	0.612 ± 0.015	**0.667** ± 0.018
Human	0.293	0.351	0.367 ± 0.017	0.343 ± 0.016	**0.373** ± 0.012	0.610	0.606	0.424 ± 0.019	0.585 ± 0.010	**0.618** ± 0.013
Yeast	0.462	0.534	0.561	0.568	**0.612**	0.750	0.779	0.591	0.753	**0.825**
M. mazei	0.422	0.478	0.486	0.474	**0.523**	0.694	0.712	0.518	0.686	**0.757**
Honeybee	0.330	0.396	0.408	0.422	**0.431**	0.630	0.644	0.461	0.640	**0.650**
Tomato	0.454	0.513	0.460	0.463	**0.552**	0.731	0.733	0.471	0.720	**0.767**
Rice bean	0.436	0.511	0.437	**0.549**	0.546	0.679	**0.730**	0.442	0.727	0.719
Bacillus	0.449	0.518	0.540	0.513	**0.561**	0.742	**0.768**	0.573	0.711	**0.788**
Clam bacteria	0.253	0.298	0.371	0.347	**0.397**	0.602	0.589	0.405	0.617	**0.656**
Average	0.376	0.439	0.453	0.459	**0.499**	0.673	0.687	0.494	0.672	**0.716**

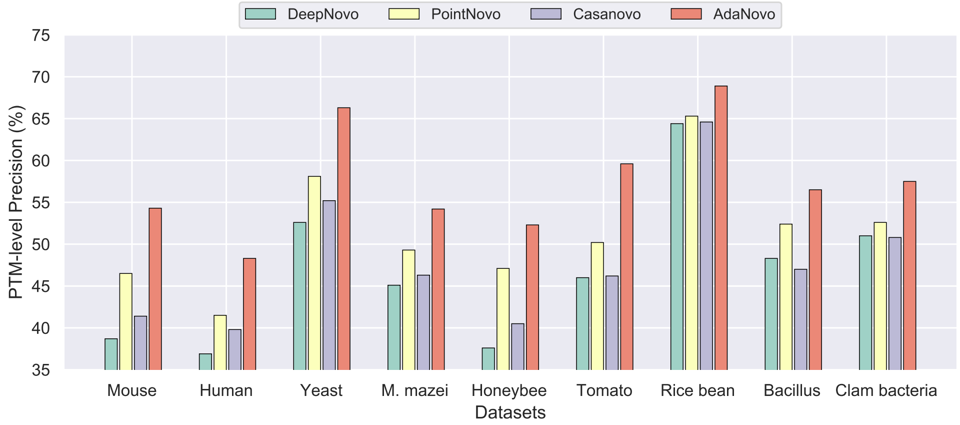


Figure 4: Empirical comparison of *de novo* sequencing models in terms of PTMs identification.

AdaNovo frequently outperforms previous methods on 9-species benchmark and notably excels in PTMs identification. As can be observed in Table 1, AdaNovo outperforms competitive models on most (8 out of 9) species in peptide-level precision compared to DeepNovo, PointNovo and CasaNovo. At amino acid-level, AdaNovo also outperforms baselines on most datasets. Also, as demonstrated in Figure 4, we compare AdaNovo with other methods in terms of identifying amino acids with PTMs because AdaNovo is designed to accurately identify the amino acids with PTMs. The results in the table indicate that AdaNovo consistently exceeds other competitors by significant margins (3.6% - 9.4%) in identifying amino acids with PTMs, verifying the effectiveness of the amino acid-level adaptive training objective. We further evaluate the stability of AdaNovo in Appendix B.

4.5 Ablation Study

Ablations on amino acid-level and peptide-level adaptive training strategies. To investigate the influence of the amino acid-level and peptide-level adaptive training strategies, we remove each of them from AdaNovo and evaluate the models' performance using the experimental settings in

https://doi.org/10.52202/079017-0057

Table 2: Ablations on amino acid-level (AA-level) and peptide-level training strategies. The results are reported using the Human datasets as test set.

Model	AA. Prec.	Peptide Prec.	PTM Prec.
Casanovo	0.585	0.343	0.300
AdaNovo (w/o PSM-level objective)	0.607	0.360	0.478
AdaNovo (w/o AA-level objective)	0.594	0.349	0.314
AdaNovo	0.618	0.373	0.483

Section 4.3. The results shown in Table 2 indicate that both modules are necessary and effective for the AdaNovo model. More specifically, when we remove the AA-level training strategy in AdaNovo, the precision of the amino acids with PTMs identification drops significantly because the amino acid-level training strategy is designed for identifying amino acids with PTMs.

Table 3: Models' Performance on mass spectrum dataset with synthetic noise. The results are reported using the Clam bacteria as test set.

Model	AA. Prec.	Peptide Prec.
CasaNovo	0.582	0.297
AdaNovo (w/o PSM-level objective)	0.617	0.336
AdaNovo (w/o AA-level objective)	0.644	0.372
AdaNovo	0.656	0.397

Performance on mass spectra with synthetic noise. To verify the effectiveness of the PSM-level adaptive training strategy, we randomly choose 20% spectrum in the training datasets, and add synthetic noise peaks or remove original peaks with higher intensity values. We report the results in Table 3, from which we can observe that the performance would degrade sharply when we remove the PSM-level training strategy. This indicates that PSM-level adaptive training strategy can enhance models' robustness against data noise in mass spectrum.

4.6 Comparisons with Alternative Methods for identifying amino acids with PTMs

Table 4: Comparisons with alternative methods in terms of identifying amino acids with PTMs. All results are reported using the yeast dataset as test set.

Model	AA. Prec.	Peptide Prec.	PTM prec.
Casanovo	0.753	0.568	0.552
+ Re-weight (Upsampling)	0.762	0.576	0.564
+ Focal loss	0.745	0.543	0.550
AdaNovo (w/o PSM-level objective)	0.793	0.594	0.616
AdaNovo	0.825	0.612	0.665

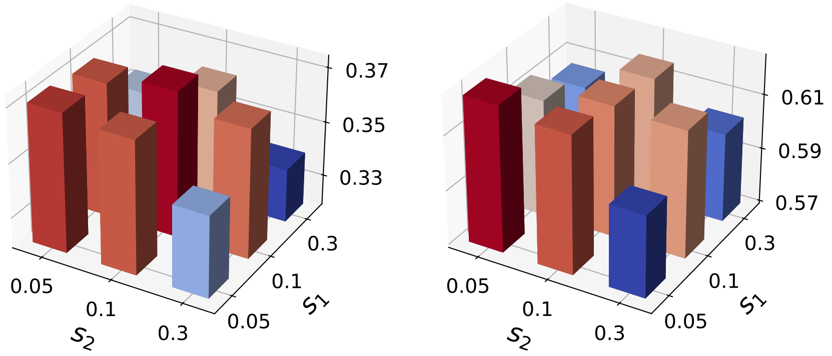
In this section, we show the performance of AdaNovo only with amino acid-level loss (denoted as 'AdaNovo w/o PSM-level objective') and compare to some alternative methods in terms of identifying amino acids with PTMs. The first alternative is to re-weight each amino acid y_j with $w_j = N_{total}/N_{y_j}$, where N_{total} and N_{y_j} represent the total number of amino acids and the number of amino acids in the y_j category in the dataset, respectively. The second alternative is the focal loss [13], we replace the cross entropy loss of Casanovo [35] with the focal loss,

$$\mathcal{L} = -(1 - \alpha p(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j}))^\gamma \log p(y_j \mid \mathbf{x}, \mathbf{z}, \mathbf{y}_{<j}),$$

where α and γ are hyperparameters to adjust the loss weight. The results shown in Table 4 indicate that both AdaNovo and the first alternative can help improve Casanovo's ability. Moreover, AdaNovo outperforms the first alternative by a notable margin probably because the training and testing datasets are derived from different species, there exists a significant difference in the distribution of PTMs quantities. Therefore, the above w_j obtained with the train set is not suitable for test set. Also, AdaNovo is inspired by the domain knowledge that the mass shift of PTMs only be manifested in the mass spectra, thus shows superiority over the re-weighting methods in long-tailed classification.

https://doi.org/10.52202/079017-0057

4.7 Sensitivity Analysis



(a) Human (Peptide-level)

(b) Human (AA-level)

Figure 5: The effects of the two hyperparameters s_1 and s_2 for AdaNovo.

In this section, we investigate the effects of the two hyperparameters s_1 and s_2, which determines the influence of amino acid-level and PSM-level training strategy. As shown in Figure 5, we tune both s_1 and s_2 within the range $[0.05, 0.1, 0.3]$ and observe that the values of these two hyperparameters significantly affect the final performance of the model. Additionally, the optimal hyperparameters for peptide-level metrics may be sub-optimal to amino acids-level metrics. It is necessary to finely adjust the values of s_1 and s_2 based on the dataset, striking the balance between amino acid-level and PSM-level training strategies.

4.8 Costs of Computing and Storage

Table 5: Comparisons with competitive methods in terms of computational overhead. The training and inference time are evaluated on Honeybee dataset with the same Nvidia A100 GPU.

Model	**#Params (M)**	**Training time (h)**	**Inference time (h)**
Casanovo	47.35	56.52	7.14
AdaNovo	66.31	60.17	7.09

In this part, we compare AdaNovo with Casanovo in terms of the number of model parameters, training time and inference time. The results shown in Table 5. Although AdaNovo outperforms Casanovo in peptide sequencing and PTMs identification, it inevitably introduced extra parameters (Peptide Decoder #2), resulting in a 40.04% increase in parameter count (from 47.35M to 66.31M). Also, under the same hardware settings (1 Nvidia A100-SXM4-80GB GPU), the training time of AdaNovo increased by 6.3% (from 56.52h to 60.17h) over Casanovo. However, AdaNovo and Casanovo share the similar inference speed, which is more important in real-world applications.

5 Conclusion and Future Work

In this paper, we discern that data biases limit progress in *de novo* peptide sequencing by hindering the accurate identification of PTMs and reducing the reliability of PSMs due to low PTM frequency and spectral noise. To address these issues, we introduce a novel approach involving the calculation of conditional mutual information between the spectrum and each amino acid, followed by re-weighting the training loss of corresponding amino acids. Extensive experiments on 9-species datasets affirm that AdaNovo surpasses previous *de novo* sequencing methods, showcasing superior performance in both amino acid- and peptide-level precision. Notably, AdaNovo exhibits a distinct advantage in identifying amino acids with PTMs. Despite the significant progress made by AdaNovo, identifying previously unseen PTMs remains challenging, prompting the need for further research in the future.

6 Acknowledgement

This work was supported by National Natural Science Foundation of China Project No. 623B2086 and No. U21A20427, the Science & Technology Innovation 2030 Major Program Project No. 2021ZD0150100, Project No. WU2022A009 from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University, and Project No. WU2023C019 from the Westlake University Industries of the Future Research. Finally, we thank the Westlake University HPC Center for providing part of the computational resources. Xiaojun Shan was not supported by any funds from China.

References

[1] Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, 2003.

[2] Christian Bartels. Fast algorithm for peptide sequencing by mass spectroscopy. *Biomedical & environmental mass spectrometry*, 19(6):363–368, 1990.

[3] 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.

[4] Yonathan Lissanu Deribe, Tony Pawson, and Ivan Dikic. Post-translational modifications in signal integration. *Nature structural & molecular biology*, 17(6):666–672, 2010.

[5] Jimmy K Eng, Ashley L McCormack, and John R Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the american society for mass spectrometry*, 5(11):976–989, 1994.

[6] Bernd Fischer, Volker Roth, Franz Roos, Jonas Grossmann, Sacha Baginsky, Peter Widmayer, Wilhelm Gruissem, and Joachim M Buhmann. Novohmm: a hidden markov model for de novo peptide sequencing. *Analytical chemistry*, 77(22):7265–7273, 2005.

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

[8] Ari M Frank. Predicting intensity ranks of peptide fragment ions. *Journal of proteome research*, 8(5):2226–2240, 2009.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Andy T Kong, Felipe V Leprevost, Dmitry M Avtonomov, Dattatreya Mellacheruvu, and Alexey I Nesvizhskii. Msfragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature methods*, 14(5):513–520, 2017.

[11] Dequan Li, Yan Fu, Ruixiang Sun, Charles X Ling, Yonggang Wei, Hu Zhou, Rong Zeng, Qiang Yang, Simin He, and Wen Gao. pfind: a novel database-searching software system for automated peptide and protein identification via tandem mass spectrometry. *Bioinformatics*, 21(13):3049–3050, 2005.

[12] Eugene Lin, Chieh-Hsin Lin, and Hsien-Yuan Lane. Relevant applications of generative adversarial networks in drug design and discovery: molecular de novo design, dimensionality reduction, and de novo peptide and protein design. *Molecules*, 25(14):3250, 2020.

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[14] Fan Liu, Dirk TS Rijkers, Harm Post, and Albert JR Heck. Proteome-wide profiling of protein assemblies by cross-linking mass spectrometry. *Nature methods*, 12(12):1179–1184, 2015.

[15] 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.

https://doi.org/10.52202/079017-0057

[16] Matthias Mann and Ole N Jensen. Proteomic analysis of post-translational modifications. *Nature biotechnology*, 21(3):255–261, 2003.

[17] Kevin McDonnell, Enda Howley, and Florence Abram. The impact of noise and missing fragmentation cleavages on de novo peptide identification algorithms. *Computational and Structural Biotechnology Journal*, 20:1402–1412, 2022.

[18] Alexey I Nesvizhskii. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of proteomics*, 73(11):2092–2123, 2010.

[19] 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.

[20] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[21] CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE. *Speech Understanding Systems. Summary of Results of the Five-Year Research Effort at Carnegie-Mellon University*. 1977.

[22] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559, 2022.

[23] Jinshuai Sun, Jiahui Shi, Yihao Wang, Shujia Wu, Liping Zhao, Yanchang Li, Hong Wang, Lei Chang, Zhitang Lyu, Junzhu Wu, et al. Open-pfind enhances the identification of missing proteins from human testis tissue. *Journal of proteome research*, 18(12):4189–4196, 2019.

[24] J Alex Taylor and Richard S Johnson. Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Analytical chemistry*, 73(11):2594–2604, 2001.

[25] Ngoc Hieu Tran, Rui Qiao, Lei Xin, Xin Chen, Baozhen Shan, and Ming Li. Personalized deep learning of individual immunopeptidomes to identify neoantigens for cancer vaccines. *Nature Machine Intelligence*, 2(12):764–771, 2020.

[26] Ngoc Hieu Tran, M Ziaur Rahman, Lin He, Lei Xin, Baozhen Shan, and Ming Li. Complete de novo assembly of monoclonal antibody sequences. *Scientific reports*, 6(1):31730, 2016.

[27] 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.

[28] Anuli Christiana Uzozie and Ruedi Aebersold. Advancing translational research and precision medicine with targeted proteomics. *Journal of proteomics*, 189:1–10, 2018.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] Rui Vitorino, Sofia Guedes, Fabio Trindade, Inês Correia, Gabriela Moura, Paulo Carvalho, Manuel AS Santos, and Francisco Amado. De novo sequencing of proteins by mass spectrometry. *Expert Review of Proteomics*, 17(7-8):595–607, 2020.

[31] Juan A Vizcaíno, Eric W Deutsch, Rui Wang, Attila Csordas, Florian Reisinger, Daniel Ríos, José A Dianes, Zhi Sun, Terry Farrah, Nuno Bandeira, et al. Proteomexchange provides globally coordinated proteomics data submission and dissemination. *Nature biotechnology*, 32(3):223–226, 2014.

[32] Dirk A Wolters, Michael P Washburn, and John R Yates. An automated multidimensional protein identification technology for shotgun proteomics. *Analytical chemistry*, 73(23):5683–5690, 2001.

[33] Aaron D Wyner. A definition of conditional mutual information for arbitrary ensembles. *Information and Control*, 38(1):51–59, 1978.

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

[35] Melih Yilmaz, William Fondrie, Wout Bittremieux, Sewoong Oh, and William S Noble. De novo mass spectrometry peptide sequencing with a transformer model. In *International Conference on Machine Learning*, pages 25514–25522. PMLR, 2022.

[36] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[37] Jingbo Zhou, Shaorong Chen, Jun Xia, Sizhe Liu, Tianze Ling, Wenjie Du, Yue Liu, Jianwei Yin, and Stan Z Li. Novobench: Benchmarking deep learning-based de novo peptide sequencing methods in proteomics. *arXiv preprint arXiv:2406.11906*, 2024.

A Peak Embedding Methods

In order to feed the MS2 peaks to MS Encoder, we regard each mass spectrum peak (m_i, t_i) as a 'word' in natural language processing and obtain its embedding by individually encoding its *m/z* value and intensity value before combining them through summation. For a given peak, we consider its *m/z* as its position and employ positional encoding with reference to [29],

$$f_{ij} = \sin\left(\frac{m_i}{n_1 n_2^{2j/d}}\right), \text{ for } j \leq \frac{d}{2}, \tag{11}$$

$$f_{ij} = \cos\left(\frac{m_i}{n_1 n_2^{2j/d}}\right), \text{ for } j > \frac{d}{2}, \tag{12}$$

where f_{ij} is the value of f_i in the j-th dimension, d is the embedding size of f_i, and n_1 and n_2 is an user-defined scalar and can be set to any value. Specifically, we set $n_1 = \frac{m_{\max}}{m_{\min}}$ and $n_2 = \frac{m_{\min}}{2\pi}$ where $m_{\max}$ = 10,000 and $m_{\min}$ = 0.001 in our work. The input embeddings furnish a detailed portrayal of high-precision *m/z* information. Analogous to the consideration of relative positions in the initial transformer model [29], these embeddings potentially facilitate the model's attention to *m/z* variations between peaks. Such attention to detail is crucial for the accurate identification of amino acids within the peptide sequence. The intensity information t_i is directly encoded through a linear layer W_g and mapped to the d-dimensional space $\mathbb{R}^d$,

$$g_i = W_g t_i, \tag{13}$$

where $W_g \in \mathbb{R}^d$ denote the parameters of the linear layer. Subsequently, the input peak (m_i, t_i) embedding h_i generated through summation,

$$h_i = g_i + f_i. \tag{14}$$

The input to adanovo's MS Encoder is the embedding of each peak, $\mathbf{h} = \{h_i\}_{i=1}^{M}$, where M is the number of peaks in mass spectra. Similarly, for the precursor $\mathbf{z} = \{(m_{prec}, c_{prec})\}$ to be fed into the Peptide Decoder #1, we embed m_{prec} using the same sinusoidal position embedding as the peaks in the spectrum. Additionally, the charge state c_{prec} is embedded using the embedding layer in PyTorch.

https://doi.org/10.52202/079017-0057

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: Please see the abstract and the introduction section 1.

   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: The proposed method cannot identify never-before-seen PTMs (see section 5).

   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]

https://doi.org/10.52202/079017-0057

Justification: The paper does not include 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: Please see section 4.3.

   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: We have released the code in the supplementary materials.

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: Please see section 4.3.

   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: [No]

   Justification: The error bars are not reported because it would be too computationally expensive.

   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).

https://doi.org/10.52202/079017-0057

- 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: Please see section 4.8.

   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 research 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: We have discussed the potential positive societal impacts in section 5. It has no negative societal impact.

    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.

https://doi.org/10.52202/079017-0057

- 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: **[TODO]**

    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 papers.

    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.

https://doi.org/10.52202/079017-0057

- 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 release the code in the supplement.

    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: **[TODO]**

    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: **[TODO]**

    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.

https://doi.org/10.52202/079017-0057